An Automated Allele-calling System for High-throughput Microsatellite Genotyping

Pratyaksha Jagad Wirapati

Submitted in total fulfillment of the requirements
of the degree of Doctor of Philosophy

January 2003

Department of Medical Biology
University of Melbourne

The Walter and Eliza Hall Institute of Medical Research

Abstract

Microsatellite markers are widely used for genetic analysis in biomedical research, agriculture, population and evolutionary biology, as well as for forensics and diagnostics. Advances in laboratory automation and data collection have increased the throughput and have reduced the cost of large-scale genotyping. One step in the measurement process that still needs improvement is "allele calling", where raw electrophoresis signals are converted into discrete genotypes. This is still largely a laborious manual process that constitutes more than a quarter of the genotyping cost.

Automating allele calling is hampered by, among others, the presence of "stutter patterns" (artefact peaks introduced during PCR amplification) and variation in electrophoresis migration behavior. Both effects are marker specific, making it difficult to devise an algorithm that works for all markers without marker-specific calibration. This thesis proposes an allele calling method that consists of two main computer programs: (1) STRAL: a trace alignment algorithm that normalizes variation in the "time domain" of the observed chromatograms, and (2) FA: a pattern recognition algorithm that performs allele calling on the aligned chromatograms. Both are adaptive and do not require marker-specific calibration. For a given observation, each possible genotype is associated with a quality score related to the probability of calling error. This quality score can be used to rank and select the most likely genotype(s).

Benchmark tests were performed on ∼33,000 genotypes taken arbitrarily from the daily output of a genotyping service laboratory. The performance is characterized by a trade-off between true calls and miscalls at a given cutoff of the quality value. At a level corresponding to less than 1% error (acceptable for most purposes), 55% of the data can be called correctly (or 70% of the data that can be called by human analysts). This performance is still far from manual calling (at 80% correct call of the total with $< 0.2\%$ error). However, it is useful for a hybrid system where up to 70% of the data is scored automatically, 15% of which is automatically rejected, and the remaining 30% needs to be manually examined, but with only 5% of them requiring corrections.

We conclude that this prototype is worth implementing for actual appli-

cations. The noteworthy features are the ability to adapt to marker-specific effects and to predict the error rate. More importantly, a framework has been established where automatic, training-set-driven optimization of the algorithms might yield better performance in the near future.

Declaration

This is to certify that

(i) the thesis comprises only my original work towards the PhD except where indicated in the Preface,

(ii) due acknowledgement has been made in the text to all other material used,

(iii) the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Preface

All genotyping data used in this thesis were produced at the Australian Genome Research Facility (AGRF), courtesy of Dr. Kelly Ewen. The data were taken from the archive of the AGRF's service operation. The genetic studies for which the genotyping were carried out were irrelevant to the topic of this thesis, and not mentioned to ensure confidentiality. No information about the patients was disclosed to the author.

Some work on the modeling of microsatellite stutter patterns and the development computer programs for reading ABIF trace data files were done prior to the PhD candidature enrollment, while the author was working at the Eijkman Institute for Molecular Biology in Jakarta, Indonesia.

Acknowledgements

It has been a most enlightening experience, and firstly I would like to thank Professors Suzanne Cory and Terence P. Speed for the opportunity to undertake my PhD studies at the WEHI. I owe a great debt to Professor Speed for his suggestion to upgrade from Masters to a PhD candidature, and for his recommendation for the necessary financial support. I should also thank Professor Sangkot Marzuki of the Eijkman Institute in Jakarta, for the opportunity to obtain a Masters degree scholarship, despite the difficult circumstances in Indonesia at that time.

I wish to thank Professor Speed again, as my PhD supervisor, for his trust, freedom and flexibility that allowed me to approach the thesis problem by exploring, discovering and evolving. I give my utmost gratitude for his patience and wisdom in the light of my occasional haphazardness and "persistence".

My project thrived on streams of data, problems and ideas in genomics and genetics generously supplied by many people at the WEHI and the AGRF. My sincerest thanks for them all, and particularly to Dr. Simon Foote, who stood at the center of these activities and posed the genotyping problem at our first encounter. He has subsequently provided easy access to the AGRF and his lab, which always have challenging problems, good data and enthusiastic people. Dr. Andrew Symons ensured that I understood the reality of genotyping and was the first adopter of my method. Ms. Farah Coutrier supplied data sets for the early studies, and was the one who initially suggested to study at the WEHI and introduced me to Dr. Foote and Prof. Speed.

I wish to thank Dr. Kelly Ewen, Mr. Wayne Ward, Dr. John Barlow and the whole team at the AGRF, for sharing their high quality genotyping data, without which this thesis would have been much less significant. I also thank Dr. Tony Kyne and his team for their first-class information technology support and accommodative computing arrangement.

Big thanks to the Speed Lab, which has been an excellent environment for research in bioinformatics, rich in ideas from people with many backgrounds. Thanks to our visitors from around the world, particularly Dr. Lei Li who laid the foundations for my work, and to the regular audiences of the "Tuesday

Morning Seminar", whose critical questioning has certainly shaped many aspects of this project. Dr. George Rudy helped me settle in my early days, and has been a great friend ever since. Henrik Bengtsson, Tim Beißbarth and Alex Gout ensured that I completed my thesis with their camaraderie and generous help at the eleventh hour.

Last but not least, my most heartfelt thanks to my faraway companion, Meta W. Djojosubroto, for her patience, encouragement and the "wake-up calls".

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ABI	Applied Biosystems
AGRF	Australian Genome Research Facility
ASCII	American Standard Code for Information Interchange
bp	base pairs
c.d.f.	cumulative distribution function
DTW	Dynamic time warping
GLSA	Genotypic Least-Squares Approximation
FA	Find allele (or fit allele)
FAL1	Find allele using L-score version 1; the name of a calling algorithm, specifically with the weights tuned by the training set used here.
MGS	Mammalian genotyping service (at Marshfield, Wisconsin)
PCR	Polymerase chain reaction
p.d.f.	probability density function
PHRED	a base-calling software by Phil Green *et al.* (not an abbreviation)
RSS	Residual sum of squares
SRSS	Standardized residual sum of squares
SSF	Size-standard fragments
STR	Short tandem repeat
STRAL	Short tandem repeat alignment algorithm (also "Made in Australia")
TA	TrueAllele^{TM} (a genotyping program by Mark Perlin *et al.*)

List of Symbols

n	the number of traces per marker data set
m	the number of data points in a marker interval
k	the number of possible fragments (one every bp) in a marker interval; therefore also the marker interval length
T	the number of data points per nucleotide (aligned trace resolution)
j	index to a trace
s	scan numbers of raw trace data
u	SSF-adjusted size
t	fragment length
$y_j^{(1)}$, $y_j^{(2)}$, $y_j^{(3)}$	resampled trace data j at various stages of alignment
$\phi(t)$	marker-wide systematic "warp"
$\psi_j(t)$	lane-specific "jitters"
a, b	allele indices
t_a, t_b	the length of allele a, b, etc.
$(t_0, t_k]$	marker interval
α, β	allelic coefficients
$\boldsymbol{\mu}_a$	pattern of allele a, $\boldsymbol{\mu}_a \in \mathbb{R}^m$
$\boldsymbol{y}_j$	the observed (and aligned) vector of trace j
$\boldsymbol{\theta}$	allelic pattern model parameters
ρ	heterozygote ratio slope parameter
$\|.\|$	2-norm
$\boldsymbol{D}$	diffusion linear operator
$\boldsymbol{S}$	polymerase slippage linear operator
$\boldsymbol{A}$	plusA addition linear operator
z^2	the 2-norm of the residual in GLSA fit divided by $\|\boldsymbol{y}_j\|$
h^2	deviation-squared from the heterozygote ratio model
$Q_{a,b,j}$	the Q-score for genotype (a, b) and trace j
$L_{a,b,j}$	the L-score for genotype (a, b) and trace j
$\boldsymbol{w}$	the weights of the Q-score

Chapter 1

Introduction

1.1 Motivation

Microsatellite or short tandem repeat (STR) markers are widely used for genetic analysis in biomedical research [Weber and Broman 2001], agriculture [Beuzen *et al* 2000, Dekkers and Hospital 2002], population and evolutionary biology [Kim *et al* 2002, Webster *et al* 2002], as well as for forensics [Carey and Mitnik 2002] and diagnostics [Sidransky 1994]. Advances in genotyping technology have driven the costs down dramatically (see figure 1.1). This is achieved through laboratory automation, miniaturization in chemical reactions and automated fluorescence electrophoresis machines [Weber and Broman 2001]. Economy of scale is achieved through high-throughput systems at specialized laboratories, or "core facilities", that provide generic genotyping services to a variety of genetic analysis projects. These "genotyping centers" have been established in many countries. One of the first of these facilities, and the pioneer in many aspects of the technology, was the Mammalian Genotyping Service in Marshfield[1]. Other prominent laboratories are the Center for Inherited Disease Research (CIDR)[2] and DeCode Genetics in Iceland[3]. In Australia, the high-throughput genotyping center is located at the Melbourne branch of the Australian Genome Research Facility (AGRF)[4], which supplied all the data used in this thesis project.

Figure 1.1 also indicates that the cost drop started to plateau in 1999. To see whether further cost reduction is possible, we need to examine the cost components. The breakdown is shown in figure 1.2. According to Weber and Broman [2001], about half of the costs are devoted to salaries (covering administration, scoring and a good portion of electrophoresis in figure 1.2a). Similar cost breakdown is found at the AGRF (figure 1.2b), although the system is

[1] `www.research.marshfieldclinic.org/genetics`
[2] `www.cidr.jhmi.edu`
[3] `www.decodegenetics.com`
[4] `www.agrf.org.au`

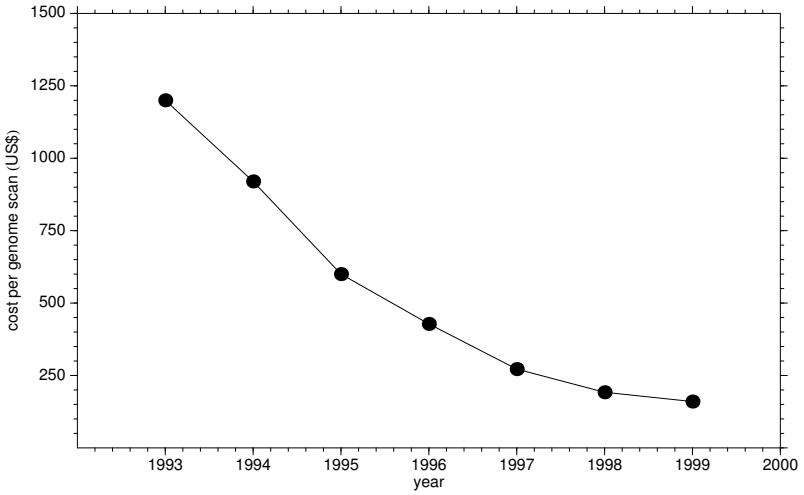
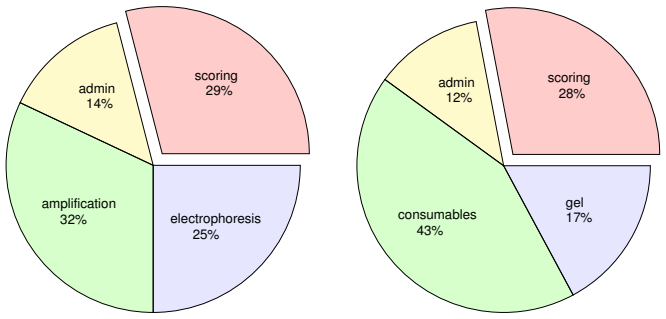


Figure 1.1: A rapid decrease in the cost of microsatellite genotyping in the last decade. The data is from the Mammalian Genotyping Service in Marshfield [Weber and Broman 2001, table 7.1]. The cost is for a genome scan with 400 markers.



a) MGS b) AGRF

Figure 1.2: The breakdown of microsatellite genotyping costs at two genotyping centers: a) Mammalian Genotyping Service (MGS) at Marshfield [Weber and Broman 2001] and b) the Australian Genome Research Facility (John Barlow, personal communication). Both are similar. The 'consumables' in the AGRF might include reagents for electrophoresis, while 'gel' is mostly labor cost for manual gel handling. Manual scoring (or allele calling) constitutes nearly 30% of the genotyping costs.

based on commercial genotyping technology (unlike the custom-built one used in Marshfield MGS). For the AGRF data, all components except 'consumables' are labor costs, which is slightly more than half of the total cost. In both cases, 'scoring' constitutes 28%-29% of the total cost. Also known as *allele calling*, this is a process where the patterns of continuous electrophoretic signals (known as 'traces') are converted into discrete genotypes, or a pair of alleles corresponding to the genetic variations at the marker loci. This process only involves analyzing information and thus can be potentially replaced by software. Substantial cost reduction is therefore expected in this area [Weber and Broman 2001, page 84], provided that allele calling software with acceptable performance can be devised.

Consequently, efforts have been made to develop software for automated allele calling. Some are published [Mansfield *et al* 1994, Perlin *et al* 1994, 1995, Stoughton *et al* 1997, Pálsson *et al* 1999], while others are proprietary or for in-house use only. However, there is not yet a satisfactory solution [Weber and Broman 2001, Li *et al* 2001]. The sources of difficulties will be detailed later in this chapter. In brief, there are various effects and noise introduced throughout a complex measurement process involving various technologies: polymerase chain reaction (PCR), electrophoresis and sample multiplexing. Although the underlying information of interest is simple (a pair of discrete labels corresponding to the length variants of the microsatellite loci), the observed signal is complex: multi-component, high-resolution time series data exhibiting complex patterns of peaks. The patterns are, however, highly consistent (at least from the point of view of a trained human analyst). Using intuitive knowledge about how the patterns are generated and the basic principle of genetics that there are at most two alleles in each individual, and aided by a graphical user-interface software for examining and editing the trace and genotypes, manual allele calling can be performed with high accuracy. The error rate is typically less than 0.2% [Ewen *et al* 2000, Weeks *et al* 2002][5].

The best published result for a fully automated algorithm is 1.34% error rate[6] reported by a team from DeCode Genetics [Pálsson *et al* 1999]. Depending on the specific requirements of the downstream statistical genetic analysis, this rate might not be acceptable [Weeks *et al* 2002]. Further manual examination and corrections is needed if an error rate below 1% is required. Furthermore, the DeCode Genetics algorithm is proprietary and might include a patented algorithm [Pálsson *et al* 1999, Perlin *et al* 1994, 1995, Perlin 2000]. There is clearly room for a new effort to develop an automated allele caller with,

[5]Note that this is the rate of calling error only, and does not include 'intrinsic' errors such as recent mutations or null alleles.

[6]78 miscalls out of 5806 observations as 'good' in the test data set.

hopefully, improved performance. There is also a need for a publicly and openly distributed implementation, which might evolve into a good solution through widespread use and community feedback and participation.

One feature that would be useful but is not yet found in the existing solutions is the ability to predict the error rate of the called genotypes. Such 'quality values' have been found to be extremely useful in another problem involving electrophoresis traces: base calling in DNA sequencing. The software suite PHRED/PHRAP/CONSED [Ewing *et al* 1998, Ewing and Green 1998, Gordon *et al* 1998, Richterich 1998] is a widely used package proven to be instrumental in many whole-genome sequencing projects. The 'PHRED quality score' has become a standard quality control for DNA sequence information. The score corresponds closely to the probability of calling error. A cutoff level can be flexibly chosen depending on the appropriate trade-off between error rate and yields. Additionally, the scores themselves can be incorporated as weighting factors in downstream analysis algorithms, allowing more optimal use of the available data. We would like to develop an analogous quality indicator for microsatellite genotyping.

The remainder of this chapter contains a brief overview of microsatellite markers, the way they are measured (genotyped) and the various effects that complicate the raw signals. Existing solutions are reviewed and finally we outline our general approach and the proposed method.

1.2 Genetic Markers and Microsatellite Loci

Genetic analysis attempts to correlate genetic states of individuals in the population being studied with other biological properties ranging from disease status, to various phenotypic traits in agricultural species, to geographical or ethnic origins. It is neither necessary nor feasible to measure the exact genetic state of an individual (which is the DNA sequence of all chromosomes). Knowing the states of a handful of specific sites along the genome is often sufficient. In order to be useful, these sites, also known as *markers* or *loci* (singular *locus*), have to exhibit *polymorphism*. That is, a number of variant states, or *alleles*, need to be present in the population. A good marker has many alleles, and each allele occurs with high frequency. There are many different types of markers based on the kind of sequence polymorphisms that are investigated, and various technologies can be used to genotype each type of markers [Edwards and Caskey 1991, Ahmadian and Lundeberg 2002].

A microsatellite or short tandem repeat (STR) locus is a stretch of sequence, whose location in the genome is defined by a pair of unique sequences at its ends, with a region in the middle containing a repeat sequence (see figure 1.3a).

The repeat sequence is a string of duplicated and tandemly arranged short sequences or *repeat units*. Each repeat unit may consist of six or less nucleotides. Microsatellites are often classified according to the length of their repeat units. The terms *mono-*, *di-*, *tri-*, *tetranucleotide repeats* and so on are often used to indicate the length of the repeat unit of a microsatellite locus. Di-, tri- and tetranucleotide repeats are the ones that are most useful for genetic analysis. The utility of microsatellites for genetic analysis comes from the highly variable nature of the number of repeats in the population. Each length variant can be genetically considered an allele (figure 1.3b).

Microsatellite loci are found throughout the genomes of eukaryotes [Hamada *et al* 1982, Toth *et al* 2000]. Their usefulness as markers for genetic analysis was first demonstrated in the late 1980's [Litt and Luty 1989, Tautz 1989, Weber and May 1989]. Isolation and characterization of new loci followed [Weber 1990, Beckmann and Weber 1992, Hudson *et al* 1992, Weissenbach *et al* 1992, Levitt *et al* 1994], culminating in a comprehensive map of 5,264 microsatellite markers [Dib *et al* 1996]. Not all of the markers are needed to perform a genetic analysis. For practical reasons (and cost), a set of well-chosen markers are used for typical 'genome scan' applications. They are chosen for high heterozygosity (the probability of being a heterozygote in the population, which confers higher 'dissecting' power), ease of genotyping, equal spacing across the genome, and optimal arrangement in a multiplexed genotyping system (described below). Such sets are called 'linkage mapping sets', and are available for various marker densities.

1.3 Microsatellite Genotyping

1.3.1 Overview of the genotyping procedure

The most common way to genotype a microsatellite locus is by a combination of polymerase chain reaction (PCR) and electrophoresis. This method does not give the full sequence information about the alleles. Only the information about (relative) length of the alleles can be obtained. This is sufficient for most genetic analysis applications, which only require arbitrary (but consistent) labels of the alleles. There might be some information loss when the length of two two alleles are the same but the underlying sequences are different due to mutations other than a change in the number of repeats (a phenomenon called *homoplasy*, Weber and Broman [2001]). However, more often the length variations of microsatellite markers are due to variation in the number of repeats. The loss of information due to homoplasy should be very small. Therefore, the main objective in microsatellite genotyping is to determine the relative length

Figure 1.3: Panel a) shows examples of the typical DNA sequence of microsatellite loci. In each locus, the repeat sequence is flanked by unique sequences that identify the marker. Locus 1 is a dinucleotide $(CA)_n$ repeat locus and locus 2 is a tetranucleotide $(TATC)_n$ repeat locus. Only one chromosome is shown is each case. Panel b) illustrates the length variations of a locus in a population. Each individual has two chromosomes (maternal and paternal). The length variations are due to difference in the number of repeat units. Each variant is called an allele. If the two alleles of an individual are identical (such as 'individual 2' shown above), the genotype is called a homozygote; otherwise it is called a heterozygote.

of the alleles. Next, we outline briefly how this is done at the biochemical level and introduce some of the important terminology used throughout this thesis.

PCR After the DNA of the whole genome is extracted from a tissue sample of an individual, the DNA sequence of a marker locus of interest needs to be "highlighted". This is done using polymerase chain reaction, which selectively replicates DNA fragments defined by two short flanking sequences. A pair of primers (oligonucleotides) complementary to the flanking sequences is used to start the replication. One of the primers is also labeled with easily detectable chemicals, such as fluorescent dyes or radioactive isotopes. PCR amplification multiplies the number of molecules by several orders of magnitude. The product of the PCR amplification is a solution containing the fragments of the two alleles. The source DNA is often called the 'PCR template', or simply the 'template'.

Electrophoresis The length of the DNA fragments corresponding to the microsatellite alleles can be determined using electrophoresis. A high electric field is used to separate charged molecules by forcing them to move to the electrode with the opposing polarity. The different molecules in the mixture will migrate with different velocities depending on their charge and physical interaction with the medium. For denatured DNA molecules, the main factor that determines the migration rate is their length (the number of nucleotides). In fact, the relationship between the migration time and the length is proportional [Southern 1979a,b]. The term *mobility* is often used to describe the migration rate of a molecule.

The migration rate of a DNA fragment can be determined by measuring either the time of travel over a fixed distance, or the distance traveled after a fixed time. In the former, the detector is set up at a fixed location from where the sample is loaded, and measurement is done continuously throughout the electrophoresis run. In the latter, detection is done only at the end of the run, usually by taking the photographic image of the whole medium (almost always a slab gel). For high-throughput genotyping, the fixed-distance, continuous-time detection is preferable because the data can be digitized directly. The linear relationship between time and size offers more uniform separation throughout the size range, unlike the reciprocal relationship between mobility and size in the fixed-time electrophoresis. In this report, we limit our scope to the data obtained from fixed-distance electrophoretic apparatus.

The result of electrophoresis is a spectrum of intensity (concentration of the chemically labeled substances) as a function of time. This spectrum is also called a *trace*, a *chromatogram* or an *electropherogram*. The measurement is done at regular time intervals, say 2400 times per hour. The physical time has

no significance in the analysis, so we will use the number of data points acquired since the start of the electrophoresis run, often called *scan numbers* or simply *scans*, as the unit of the domain of a trace. Subsequent analysis may resample the raw signal, with possible interpolation of the intensity value in between data points.

In high-resolution systems used for microsatellite genotyping, two DNA fragments differing by one nucleotide can be distinguished as two separate peaks. The measurement's sampling rate is usually much higher than the electrophoresis resolution (that is, the meaningful fluctuations in the signal are relatively smooth). The number of data points per nucleotide typically ranges from 8 to 20. It varies throughout the run as the migration rate changes slightly. It also depends on the instrument and the running conditions.

The time domain of the trace can be calibrated using a set of known fragments electrophoresed together with the unknown fragments. We will call these calibration fragments *size-standard fragments* or SSF. The SSF allows comparison between different traces, and identification of the same alleles in different individuals. However, the relationship between the *length* (the number of nucleotides) and the *size* (the location in the spectrum relative to the SSF) is not straightforward. Note that in the literature, the terms *size* and *length* might be used interchangeably. As we will see later, it is important to explicitly distinguish the two concepts.

Multiplexing The throughput of an electrophoresis run can be multiplied by simultaneously separating a large number of independent samples. A single *run* can have multiple *lanes* (typically up to 96 lanes, although recently a 384-lane device was introduced). Each lane can be considered a separate electrophoresis process. In capillary electrophoresis (CE), a 'lane' is a capillary tube, which is a physically separate entity. In the older slab gel systems, a 'lane' is a path through a two-dimensional gel. The separation between lanes relies on the physical distance between the wells where the multiple samples are loaded. Photo-detection in a slab gel apparatus is done across the width of the gel (if we consider the length to be the direction of the migration). The raw data is thus an image instead of a set of traces. The number of scans is usually larger than the number of lanes. Because there is no physical barrier between lanes, the relationship between the lane and the location along the width can be distorted. The samples may travel with some sideways movement, instead of in perfect straight lines. A data analysis procedure called 'lane tracking' needs to be performed to determine a path through the image, based on the patterns of the DNA fragments. We will not deal with this procedure since satisfactory solutions exist, both from the instrument vendor as well as other sources. Fur-

thermore, capillary devices will be more prominent in the future (they do not require lane tracking).

The second level of multiplexing is achieved by labeling DNA fragments from different markers with different fluorescent dyes. Each dye has its own characteristic electromagnetic spectrum (in the range of visible light), and therefore fragments of different markers can be distinguished even if they are loaded into the same lane and have the same length. Detection is done at several different wavelengths (typically four) that coincide with the peaks in the dyes' spectra. We will refer to this measurement point in the light spectrum as a *dye channel*. Thus, each time point in a trace can have, say, four measurements from four dye channels.

The last level of multiplexing relies on the range of possible alleles in a typical microsatellite marker. The range is quite narrow (typically less than 40 bp), while the length of the fragments that can be separated with predictable migration behavior and good resolution ranges from 75 to 400 bp. This means fragments from many markers can be run together if we can be sure that their windows do not overlap.

An arrangement of a set of markers that can be multiplexed (through the use of different dyes and non-overlapping size ranges) is called a *panel*. A panel typically consists of 10 to 20 markers. Only three dye channels are usually used by the markers, because one channel is dedicated for the SSF. Organizing markers into a standard set of panels greatly simplifies the management of high-throughput genotyping. Figure 1.4 illustrates the multiplexing scheme in an electrophoresis run.

In a typical genotyping project, the same set of markers needs to be genotyped for a large number of individuals. All PCR products in the same lane usually have the same DNA template, i.e. they are from one individual. The number of individuals in a project can be larger than the number of lanes in a run, therefore the individuals are organized into *boxes*; each box is always run together in the same gel. The whole genotyping data in a project is thus a Cartesian product of individuals and markers (or panels and boxes).

1.3.2 Sources of measurement artefacts

Allele calling is essentially a process of assigning allele labels corresponding to the (relative) length of the allelic fragments in the template DNA, based on the observed trace data. We have mentioned earlier that it is not easy to automate this task. This is due to artefacts and distortions accumulated throughout the measurement steps, in addition to occasional measurement failures and background noise. Some of the systematic effects and variations are illustrated in

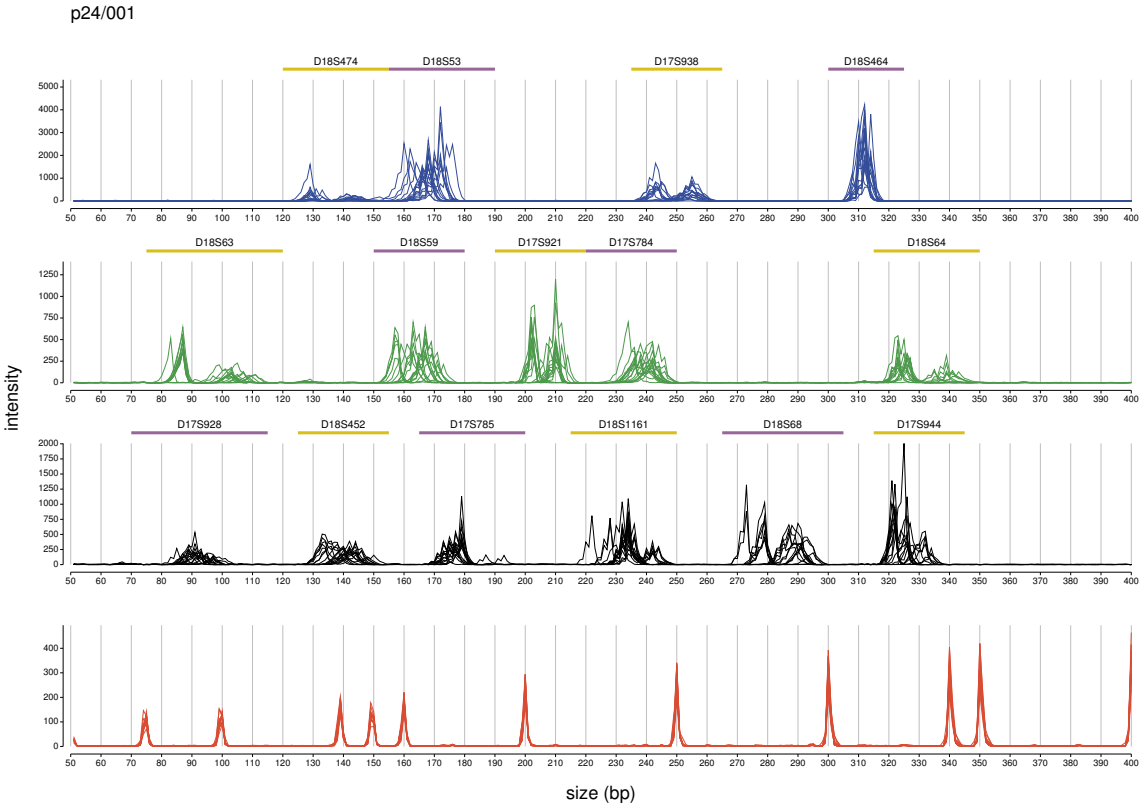


Figure 1.4: An example of a microsatellite genotyping run. The label, p24/001, indicates panel 24 of ABI linkage mapping set version 2 (10 cM density) and box number 001. The four plots show the traces (chromatograms) from four different fluorescence channels. In this figure, traces of different lanes are shown overlaid (only 12 out of 96 are shown to avoid clutter). The top three plots show, respectively, the blue, green and yellow (black in this figure) channels, which are used for the PCR products of each individual in the box. The name of the markers and the size ranges are indicated by the alternating yellow and purple bars. The red channel (the bottom plot) is dedicated for size-standard fragments, which are DNA fragments of known length, as indicated by the peak positions on the scale. The trace data shown here has been re-scaled (on the horizontal axis), such that size-standard fragments from different lanes, with the same length, coincide on the scale.

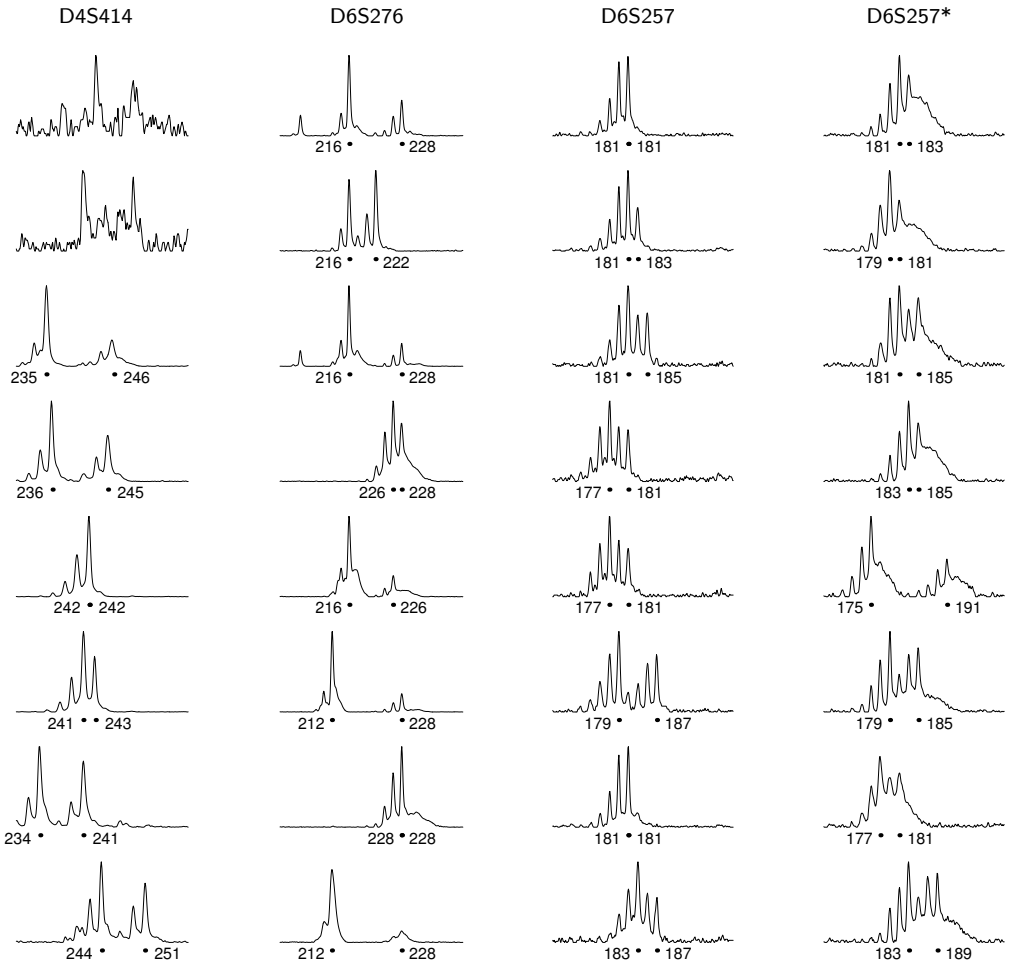


Figure 1.5: Some examples of trace data and their corresponding genotypes. Each column shows traces from several individuals genotyped at the same marker (the marker names are shown above). The integers underneath the traces are allele labels (called manually by human analysts). The dots correspond to the peaks that are likely to be the original template fragments. Instead of showing a single sharp peak, each allele manifests itself as a characteristic "stutter pattern". When the length difference between the two alleles is too small, the patterns overlap; roughly according to the principle of linear superposition. D4S414 shows some failed measurements (those that are not called). D6S276 shows that the intensity ratio between the two alleles can be significantly different, although both alleles are present in equal proportion in the template DNA. D6S257 and D6S257* are the same marker but different runs, showing the there is a run-specific effect (rounded 'bumps' always trail the sharp cluster of peaks in D6S257*).

figure 1.5. The origins of some of these effects have been studied and will be summarized shortly.

We view the measurement process as a sequence of transformations, where very simple information (a pair of numbers corresponding to a genotype) is converted into a complex signal represented by high-resolution time series data. This idea is illustrated by figure 1.6. The various effects that contribute to the complex signal are seen to occur in stepwise manner (at least conceptually; in reality they may occur simultaneously). Some of these effects are introduced by PCR (unequal amplification ratio, 'plusA' peaks, and polymerase slippage), and some by electrophoresis (diffusion and 'time warping').

PCR Artefacts An ideal PCR reaction selectively replicates the sequence defined by the flanking primers. When applied to a microsatellite locus, two fragments of different length (assuming a heterozygote genotype) should be produced (see figure 1.6a). The quantity of the product should be the same because the proportion of the allelic fragment in the template DNA is the same (equal number of maternal and paternal chromosomes). In real PCR reactions, more complex behavior is observed.

First, the assumption of equal amplification efficiency does not hold. When a mixture of template with different lengths is co-amplified, there is a tendency that the shorter allele is amplified more strongly. In the extreme cases, one of the alleles might not show at all [Ewen *et al* 2000], resulting in ambiguity between a homozygote and a heterozygote with a very weak allele. Furthermore, the relative efficiency seems to correspond to the length difference between the two alleles (for real data examples, see figure 1.5, especially the marker D6S276). There are (rare) exceptions to this rule, where the larger allele amplifies more efficiently. Note that the ratio is highly reproducible and in general can be considered the function of a specific pair of alleles. We can conceptualize the result of this phenomenon as shown in figure 1.6b.

The second PCR effect is known as the 'plusA' effect, or 3' untemplated addition [Smith *et al* 1995b, Brownstein *et al* 1996]. The polymerase enzyme might add a nucleotide at the 3' end of the newly synthesized strand with certain frequency. This effectively splits the product into two peaks (figure 1.6c). The intensity ratio of the peak with the original length to the 'plusA peak' (1 bp longer) depends on the frequency of the 3' addition. This is determined by the PCR reaction conditions. It is somewhat consistent within a batch of reactions, and there might be considerable variation between batches. Within a reaction tube, this effect is indifferent: all peaks are split with the same ratio between the original and the plusA peak. The proportion of the plusA peak might be larger or smaller than the main peak. Note that this effect is not observed in figure 1.5,

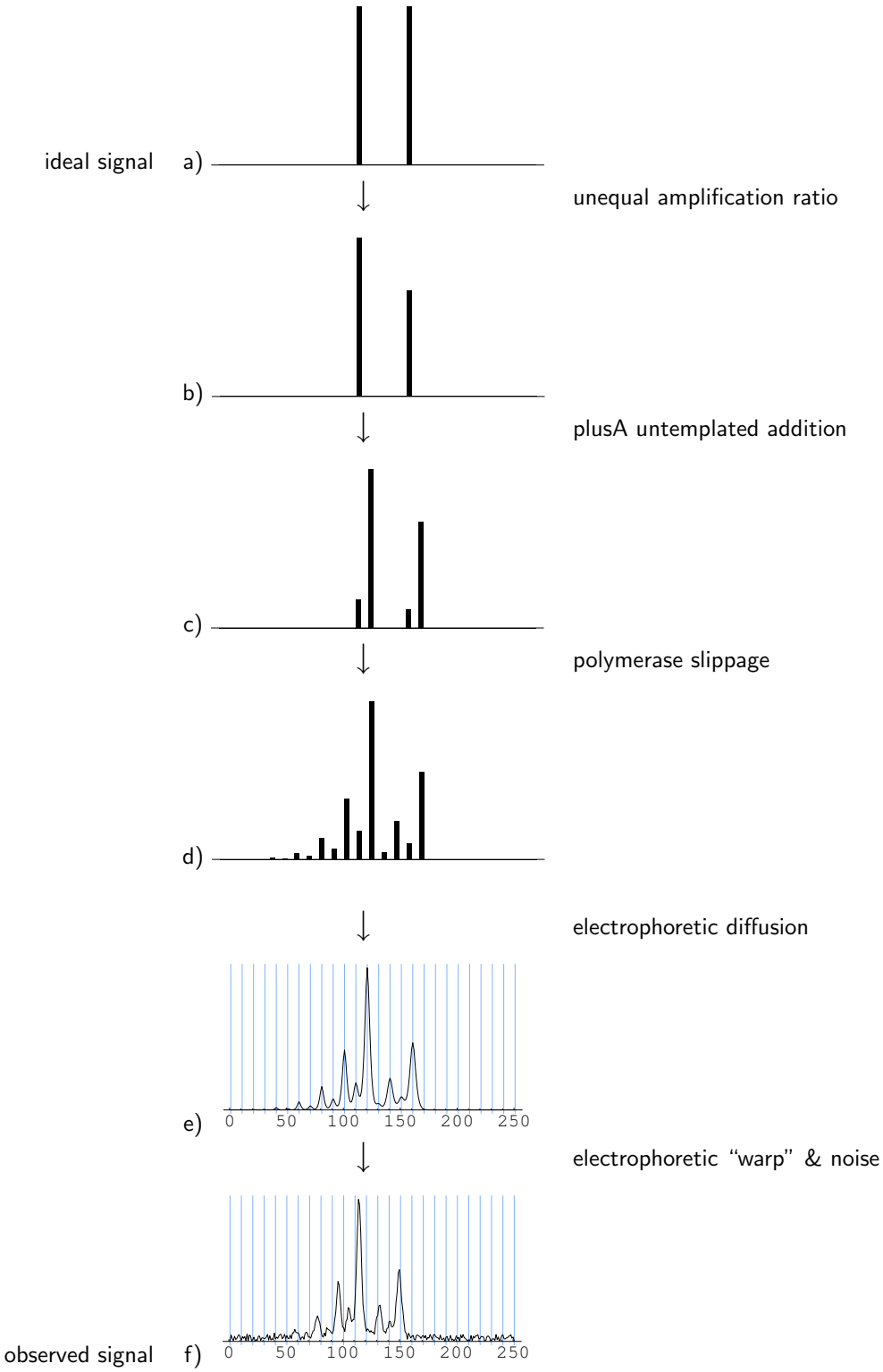


Figure 1.6: Microsatellite genotyping as a sequence of transformations.

because of a biochemical trick called 'PIG-tailing' [Brownstein *et al* 1996] that enforces plusA addition, resulting in complete shifts of the allele peaks. This procedure reduces the complexity of the pattern and simplifies allele calling. However, it is does not always work perfectly for all markers.

The most prominent PCR effect is 'stuttering' or 'polymerase slippage'. Unlike the previous two effects that are observed in PCR of any DNA fragments, polymerase slippage is caused by the repeat sequence itself. This phenomenon has been noted since the early attempts to amplify microsatellite using PCR [Litt and Luty 1989, Tautz 1989, Weber and May 1989]. Several studies were conducted specifically to investigate the cause and factors affecting this effect [Hauge and Litt 1993, Hite *et al* 1996, Walsh *et al* 1996]. The stutter peaks were found to differ from the main (template) peak by deletions (or less frequently, insertions) of some repeat units. This leads to the formulation of a (qualitative) model of stutter pattern generation [Hite *et al* 1996]. When the polymerase enzyme attempts to replicate a repeat sequence, 'slipped mispairing' might occur. Temporary denaturation of the double-stranded DNA followed by erroneous renaturation might result in loop formation either on the template strand (resulting in deletions) or on the nascent strand (resulting in insertions). Although these occur rarely, the recursive nature of PCR amplification exacerbates this effect. The products of one PCR cycle are added to the template for the next cycle. This spreads the stutter peaks further away from the main peak.

The extent of the stutter patterns depends on the alleles. Larger alleles tend to have more spread out patterns [Perlin *et al* 1995]. This is consistent with the fact that the larger the number of repeats, the higher the probability of a slippage to occur in one replication cycle (although the probability of slippage per repeat unit might be the same in all alleles). There are marker-specific effects, possibly related to the type of repeats (di-, tri-, tetra-nucleotides; CA or CT repeats, etc.) and the typical number of repeats for a marker.

Lastly, non-specific fragments might be observed in some markers. These are fragments from other regions in the genome, which are flanked by sequences that are co-incidentally similar to the primer sequences of the markers. These peaks might result in ambiguous genotypes, although in many cases they can be identified from the lack of stutter patterns.

Electrophoresis artefacts The ideal electrophoresis device will produce a signal similar to that in figure 1.6d. Each peak is very sharp and evenly spaced (because they differ by an integral number of nucleotides). However, in addition to migrating along the direction of the electric field, charged molecules in electrophoresis also diffuses in all directions, resulting in broadening of the

peaks. Because the extent of this broadening depends on the time the charged molecules spends in the electrophoresis medium, the resolution decreases with the fragment length. In addition to diffusion, electrophoresis also introduces baseline trend and noise, which are typically encountered in any spectral analysis instrumentation.

Electrophoresis separates DNA fragments according to their *mobility*, which means the domain (or 'horizontal axis') of the trace is measured in migration time, instead of the fragment length (indicated by the scan numbers along the axes of figure 1.6e and f). Although the relationship between fragment length and migration time is roughly linear, there are systematic and random components that distort the traces in the time domain. This results in uneven spacing between the peaks in the DNA fragment ladder, and shifts in location of the peaks from the expected integer positions (as shown by deviations from the grids in figure 1.6f). A mapping between electrophoretic migration time and fragment length needs to be found to correct this effect. This time warping effect varies between lanes, markers, and batches of measurements. Typically, size-standard fragments are used to approximately interpolate fragment sizes, which need to be rounded to integer allele labels by a process called *binning*. Different binning schemes, in addition to systematic differences in the electrophoretic behavior of the SSF and the unknown fragments, may produce incompatible allele labels when genotypes from multiple runs need to be combined [Ghosh *et al* 1997, Weeks *et al* 2002]. This issue will be discussed in details in chapter 2.

De-multiplexing artefacts All fluorescent dyes emit across a broad spectrum, and there is always interference between the channels. Assuming linearity, this cross-talk effect can be resolved by solving a system of four linear equations. The 'dye matrix' (the characteristic spectrum of each dye) can be determined from calibration samples [ABI 1996], or estimated from the data [Li and Speed 1999, Domnisoru 2000, Berno 1996]. These 'color-separation' methods may fail when the molecules in some peaks are so concentrated that the fluorescence intensity falls into the non-linear range of the photo-detector (i.e. the detector is nearly saturated). The violation of the linearity assumption results in false peaks (known as 'color bleed'), when linear inversion is applied to separate the signals. The contaminating false peaks sometimes cause genotyping ambiguity and errors. So far, there has not been satisfactory solution to this problem, other than making sure that not too much samples are loaded.

Another type of de-multiplexing errors are inter-lane leakage in slab gel electrophoresis. This might be due to mistracking of the gel image or inherently aberrant migration behavior of some lanes. Diffusion might also contribute to this. Lastly, 'stray alleles' might be encountered. This is a case where the size

of an allele falls outside the expected range of a marker, and the allele ventures into the window of an adjacent marker.

All of the artefacts above complicate allele calling, particularly because the extent of the artefacts vary between marker data sets. The transformations have to be undone to get to the information of interest. Whether this is to be done through stepwise 'inverse functions' (where deterministic procedures are applied to the raw data to produce successively 'cleaner' data) or through a 'generative approach' (where a model is used to reconstruct the complex patterns given a genotype, followed by finding the best fitting genotype), the method needs to take into account marker- and run-specific behaviors. We should also keep in mind that some of the transformations are inherently 'lossy'. The observed data may not distinguish alternative genotypes. In such cases, errors can be avoided only by discarding the data.

1.3.3 A review of existing solutions

Biochemical solutions

Because the sources of measurement effects are biochemical and physical, modifications of the laboratory procedures and experimental designs are obvious ways to solve the genotyping problem. Not all markers are equally difficult to score. Tri- and tetranucleotide repeats are known to have less stuttering, and mapping sets comprising mainly this type of markers are available [Weber and Broman 2001]. The drawback is that they may not be as abundant as dinucleotide repeats, in addition to having lower heterozygosity. The most successful biochemical modification to reduce the artefact peaks is the use of 'PIG-tailing' [Smith *et al* 1995b, Brownstein *et al* 1996]. By modifying the PCR primers, plusA addition can be enforced and thus eliminating the appearance of split peaks. This method, however, does not remove the stutter peaks due to polymerase slippage. Although modifications such as linear PCR [Odelberg and White 1993] and using different types of polymerase [Hite *et al* 1996] have been shown to reduce slippage in prototype genotyping, these techniques are too expensive for industrial-scale genotyping.

Data analysis methods

For practical genotyping operations, there are various information processing requirements: data acquisition and preprocessing, laboratory information management systems (LIMS), and graphical user interfaces for manually browsing and editing the genotypes. Software packages supplied by instrument vendors understandably focus on those more essential aspects. They do provide tools to

facilitate allele calling, with varying degrees of automation, which are parts of a fairly complex multi-step procedure.

The typical flow of data analysis, e.g. the one used by ABI PRISM GeneScan and Genotyper software for the Applied Biosystems instrument [ABI 1996, 2001a], consists of:

Lane tracking The migration tracks of the samples on the raw two-dimensional gel image are identified and traces (called 'sample files') are produced. This step is required for slab gel electrophoresis only.

Color separation A dye matrix is applied to remove fluorescence cross-talk.

Baselining The baseline intensity trend is zeroed. Smoothing may be performed to reduce noisy background fluctuations.

Peak identification The trace data are reduced to a list of peak locations and their corresponding intensities (either the peak heights or peak areas).

SSF identification The known lengths of the SSF are assigned to their respective peak locations.

Sizing of all peaks The SSF peaks are used to construct a sizing curve, and the sizes of all other peaks are interpolated using this curve.

Identification of allelic peaks Artefact peaks are removed, leaving only at most two peaks per sample, corresponding to the allelic fragments.

Binning The non-integer allele sizes are rounded by allocating them to 'bins'. This is essentially a classification problem for a mixture of clusters, where the clusters are periodically spaced in a one-dimensional space.

Merging Genotypes from different runs are combined.

We can see that the various steps are concerned with removing measurement effects introduced either by PCR, electrophoresis or multiplexing. The order of the steps above is only one of many possible ways to sequence the artefact removals. An example of slightly different strategy is to perform binning before allele calling [Mansfield *et al* 1994], which takes advantage of the fact that the stutter peaks can assist estimation of the bins. Another example is identification of the fragments directly on the two dimensional slab gel image [Perlin *et al* 1995]. Some steps might also be combined. Stoughton *et al* [1997] proposed an algorithm where the allelic patterns are matched directly using raw trace data (without first reducing them to peaks).

In the ABI GeneScan/Genotyper system, identification of allelic peaks is the task that requires the heaviest human intervention[7]. The software allows filtering unwanted peaks through a user-defined set of *ad hoc* rules [ABI 2001a]. This system is not intended for fully automated allele calling[8].

As we have seen in figure 1.5(page 11), the compound effect of plusA, polymerase slippage and electrophoretic diffusion is a characteristic pattern of peaks surrounding the allelic peak. The main peak is almost always the highest[9], suggesting that this property can be used to select the alleles. However, when the two alleles are not too far apart, the patterns overlap and seem to follow the principle of linear superposition. A procedure called "genotyping by deconvolution" was suggested by Perlin *et al* [1994, 1995], Perlin [2000]. Several algorithms were proposed [Perlin *et al* 1995]. The simplest ones use deconvolution techniques, assuming shift-invariant allelic patterns, which is not realistic. Assuming that the allele-specific patterns are known, they suggested that the allelic peaks can be recovered by performing linear inversion. It is not clear whether these can handle the potentially ill-conditioned nature of the stutter pattern matrices. Inverting a matrix with bad condition number usually results in unreliable solutions (and in this case, the allelic peak intensities might become negative, which is unrealistic). They also suggested what might be a better method. Enumeration of all possible genotypes is used to minimize the least-squares error between the predicted and observed patterns. This is essentially a linear inversion with model selection, which constrains the number of basis vectors used to at most two. They did not indicate whether a non-negativity constraint was used. This constraint could improve the reliability of the inversion.

Although linear models can be used to "deconvolve" the complex traces, the solutions are not necessarily the genotypes. Least-square fitting will always attempt to explain the observations as much as possible using all available degree of freedom. This means that two distinct "alleles" will always be produced to explain an observation, even if the genotype is a homozygote, due to the noisy nature of the data. Additional procedures are needed to eliminate one of the two coefficients that is "too small" to be an allele. This is not trivial because true alleles can have small intensity due to unequal amplification efficiency. For example, see the marker D6S276 in figure 1.5. In some traces the allele 226 and 228 are fairly weak, while there are contaminants, e.g. on the third trace from the top, that may be equally strong. The second trace from the bottom (a homozygous 228-228) is particularly difficult. Linear inversion will produce

[7]Note that manual scoring mentioned in figure 1.2b is performed using this software.

[8] The company has recently released better allele calling system called GeneMapper; but we are not yet familiar with the design and performance of this software.

[9] In some alleles, especially with extensive stutter patterns, the highest peak might shift to the next stutter peak [Perlin *et al* 1995, Miller and Yuan 1997].

a second "allele" around the "right shoulder" of the highest peak.

The "genotyping-by-deconvolution" approach requires the matrix containing the stutter patterns of each allele in a marker. This means that an extensive "stutter pattern library" needs to be constructed from hundreds of markers used in large genotyping projects. The training set needs to be carefully chosen so that all alleles are represented. Perlin *et al* [1995] also indicated that the library might not be valid if the reaction conditions are changed. An alternative approach is to make the algorithm "data-adaptive" [Stoughton *et al* 1997]. The basic idea is that it should be possible to construct the library on-the-fly from the observations. Traces with well-separated allelic patterns are searched. Each portion is then considered a characteristic allelic pattern (a basis vector of the stutter pattern matrix). Deconvolution is then performed afterward, using a least-squares procedure similar to Perlin's enumeration method. The procedure for constructing the allelic pattern library depends on the way the alleles are distributed in the data. Each allele has to be found either well separated from other alleles or as a homozygote.

Another interesting feature of Stoughton's solution is that the patterns are vectors with the same dimensionality as the trace data, unlike Perlin's approach where dimensionality reduction needs to be performed first (this also implies a binning procedure). The consequence of working on the trace data is that the peaks might not be aligned because there are "jitters" or small random fluctuations in the time domain. Their solution to this is extending the enumeration algorithm to search all possible small shifts of the allelic patterns.

Performance of the methods

Genotyping errors can cause significant loss of power in the downstream analysis. The acceptable error rate depends on the type of application [Weeks *et al* 2002]. For genome centers that provide generic service, an error rate of <1% seems to be the acceptable standard [Ewen *et al* 2000, Weber and Broman 2001, Weeks *et al* 2002]. Weber and Broman [2001] mentioned than an accuracy of 94% can be achieved for their automated allele caller[10]. Tedious manual editing is required to bring down the error rate to 1%. Stoughton *et al* [1997] claimed that their system was comparable to trained human reader. However, only two loci were tested.

For the "genotyping-by-deconvolution" method, the original paper [Perlin *et al* 1995] reported a test using real data from ABI-373 sequencer. 100% correct calls were reported, but the test set was small (only 5 markers and 32 individual

[10]It is not clear if this is their true positive rate or 100% minus 6% error rate. Note also that they used mostly tetranucleotide repeats that have less stutter artefacts.

on each marker). The training set was derived from traces in the same gel. A more extensive study was done later by a team at DeCode Genetics [Pálsson *et al* 1999]. The "genotyping-by-deconvolution" algorithm was implemented as a commercial product called TrueAllele (TA). When TA was used to call 7595 genotypes, 719 discrepancies was found when compared to that of manual calls, corresponding to an error rate of 9.4%. To improve the performance they created a post-processing software called Decode-GT, which implements a set of rules for throwing away alleles based on criteria such as the peak heights, TA quality value (possibly a measure of similarity in the linear fitting of the stutter pattern model), and the peak ratio. DeCode-GT flags the observations into 'good', 'ambiguous' and 'discarded' category.

In a test involving 6912 genotypes (from 72 markers and 96 individuals), 5806 (84.0%) were in the 'good' category, 78 of which were miscalled (1.34% of the 'good' ones). The corresponding yield (the true positive rate) is not clear; it is not mentioned explicitly if the true genotypes are available for all observations. Assuming that they are, the yield is 82.9%. [It is important to always consider the performance as a trade off between the error rate and the yield at a given cutoff, because it is always possible to reduce the error rate by choosing more conservative criteria.] To reduce the error rate further, they suggested re-examining the calls in the 'good' category if they belong to a marker that is found to contain any miscall. The miscall is detected through various means: control genotypes, miscalls in the 'ambiguous' category, inheritance checking, and visual examination of the allelic ladder plot (superimposed traces). This allowed them to bring down the error rate to 0.4% without manually examining everything, although it is not clear how many of those in the 'good' category need to be manually re-examined.

The three main instrument makers, Applied Biosystems, MegaBACE and Li-Cor, have systems for automatic allele calling. However, there is a lack of published report on rigorous and independent benchmark tests on their performance. MegaBACE reported in their web site:[11]

> A whole genome scan with 380 markers was performed by the Finnish Genome Center in Helsinki. Correct calls were made for 96.21% of the genotypes. Of the 4.79% incorrect calls, 99.75% had a quality score less than 2 and 100% had a quality score less than 4. A researcher can save valuable time by looking at only the genotypes with low quality scores.

They implied that a call made by the software was either correct or incorrect. This means that either the data set was particularly clean (all genotypes could be called), or that uncallable traces had been removed. Furthermore, the asso-

[11] `http://www.apbiotech.com/application/megabace/`, on 27 September 2002.

ciation between their quality score and the error rate cannot be used to fully judge the performance, because the corresponding true positive rates are not reported. This is not to say that their system is bad. We just point out the difficulty in assessing the performance of alternative solutions due to the lack of standard testing method and reporting. When comparing alternative methods, it is also important to use the same data set, because the quality of genotyping data can vary greatly, depending on the choice of the marker set, instrument technology and the quality of the DNA samples. It is important to establish a good testing method as well as a system to share trace data, if we want to converge to a reliable, fully automated microsatellite genotyping system.

1.4 Overview of the Proposed Method

High-throughput microsatellite genotyping is a complex process with many aspects and steps [Hall *et al* 1996, Ghosh *et al* 1997, Li *et al* 2001]. The information processing requirements covers many diverse areas such as raw data acquisition and preprocessing, laboratory information management systems (LIMS), allele calling, user-interface for manual examination and editing, error correction based on pedigree information, and exporting data for downstream statistical genetic analysis. As mentioned before, the allele calling aspect alone involves many analysis steps.

All of these areas contribute to some extent to the ease and accuracy of allele calling, but it is certainly impossible to develop a completely new integrated system within the time frame and resources available for this thesis project. Although not trivial, solutions in areas such as LIMS and graphical user interfaces can be considered straightforward applications of software engineering. We therefore decided to concentrate on the problems whose lack of automation consumed the largest portion of human analysts' time and whose development requires research in statistical and numerical algorithms. These are: 1) correcting electrophoretic distortions and 2) recognizing underlying genotypes that give rise to the stutter patterns.

These two topics cover the sizing, binning and merging problems, as well as identification of allelic peaks (see page 17). Explicit peak identification is not needed in our approach (to be discussed below). For the other steps, lane tracking, color separation, baselining and SSF identification, we still rely on the output of the existing software, e.g. ABI GeneScan. We did find that the output of vendor-supplied color separation and baselining algorithm may contain 'algorithmic artefacts' that may contribute to allele calling errors, particularly 'color bleed' artefacts due to inadequate color separation. However, anomalies due to these effects are reasonably rare (or can be identified when occurring

systematically), that we consider the existing preprocessed signals sufficient for our purpose of developing an allele calling method. Although we have started researches in this area (continuing Li and Speed [1999]), the results for microsatellite traces are not conclusive enough to be included in this thesis.

1.4.1 The unit of analysis

Our method processes data from one marker at a time, but simultaneously examining traces from different individuals in the same electrophoresis run. In figure 1.4, a marker data set corresponds to the trace intensity values from one dye channel, in a window defined by the yellow or purple bar that covers all possible alleles of the marker. This *marker interval* should be specified to include the whole allelic pattern, including the trailing stutter peaks. Some safety margins at both ends might be included, but care must be taken not to include the peaks from adjacent markers or noisy contaminants. If commercial mapping sets are used, the marker definitions can be automatically specified from existing databases (such as the ABI panel guide [ABI 2001b]), adding ± 5 bp on either side of the range specified by the database (which is intended to cover the main allelic peaks only). Manual examination of plots such as in figure 1.4 is recommended to ensure the intervals include the appropriate peaks[12]. Usually after fine-tuning the boundaries on data from a few runs, the intervals can be applied confidently to new data without looking. Note that different running conditions or different instruments might need different interval definitions.

In well-optimized mapping sets (such as the ABI Linkage Mapping Sets), the markers have been arranged to include a sufficient safety margin between them. In a few rare cases, the intervals of two adjacent markers may overlap (usually this is caused by only a few alleles). The boundary has to been chosen such that the inclusion of 'stray alleles' into the wrong marker interval is minimized. This will necessarily throw away a few alleles and cause calling errors. Currently we have no solution for this problem except to take notes and manually fix the genotypes of the truncated traces. An 'engineering' solution to this problem might be devised by blanking out the trace signals of stray alleles (manually identified). Automatic detection of such occurrences is complicated because it requires co-analysis of adjacent markers, assigning the joint genotype of adjacent markers to the combined trace intervals. Although not impossible, it will be computationally expensive.

We do not combine traces from different electrophoresis runs. This is because there might be run-specific systematic effects, even if the running conditions are

[12] Graphical software to do this is being developed.

assumed to be the same (controlling all subtle variations of physical conditions is not possible). One example is shown in figure 1.5. The traces in the columns D6S257 and D6S257* are from the same marker, but different runs. Slightly different electrophoresis conditions might be responsible for the 'trailing blurs' in D6S257*. Optimal matching of the allelic patterns clearly requires different templates for the two sets. Other effects that might vary between runs are the mobility relative to the SSF and plusA addition[13].

Our unit of data analysis is therefore a set of traces from all lanes in the same run, sliced from the whole electrophoresis range according to the marker interval definition. This set will be identified by the panel[14], box, and marker name (see figure 1.4), such as `p24/001/D18S474`. We will refer to this set of traces simply as a 'marker data' or the data set from one marker. This is admittedly a slight abuse of terminology, and implies that, for example, `p24/001/D18S474` is a different 'marker' from `p24/103/D18S474`. Most steps in our method do treat them separately (except when the final genotypes from the two sets are merged at the end of the process).

1.4.2 The general approach

We view allele calling as a process of modeling the sequence of transformations in figure 1.6 (page 13). The unknown variables are the genotypes (which are of interest) and the parameters governing the processes that generate the complex patterns. Two properties of the processes that we should take advantage of are reproducibility and regularity. Reproducibility means that all alleles and electrophoresis lanes behave uniformly, at least in the same run. This allows us to assume the model parameters to be the same for all traces in the same marker data[15]. Regularity means that there are relatively simple 'laws' that apply to all markers equally, differing only in a handful of parameters (such as the extent of slippage effect). A human analyst that has been trained on data from just relatively few markers can easily call the alleles of new markers, although the allelic patterns might be different from those he or she has encountered before. This means 'training' a computer program to perform the same task should be training in general principles, not allele- and marker-specific information. The advantage of this approach is that the same algorithm may work on all data, including those measured under different conditions. The implementation is simplified because there is no need for marker-specific calibration and a large

[13] This is actually a systematic effect affecting PCR reaction batches, but they often coincide with electrophoresis runs because of the way the samples are organized.

[14] The panel name is a redundant identifier since each marker appears only in one panel, but this is convenient for data management.

[15] In fact, this is the main rationale for defining our unit of analysis.

database to store the parameters (such as the stutter pattern library in Perlin *et al* [1995]).

The adaptive approach is not new. We have mentioned before the method of Stoughton *et al* [1997], where the allelic patterns are searched for in the data, relying on the knowledge that the extent of the stutter peaks is localized around the main allelic peak. This approach, however, relies too much on the distribution of alleles in the data. If an allele is never observed in isolation, either in a homozygote or in a heterozygote with an allele that differs significantly in length, it may be missing from the empirically constructed patterns. In general data analysis, there are methods such as independent component analysis (ICA) [Roberts and Everson 2001], where assumption of linearity and sparseness can be used to simultaneously estimate the basis vectors and the 'hidden variables'. Archetypal analysis [Cutler and Breiman 1994] and 'non-negative matrix factorization' [Lee and Seung 1999] perform similar tasks, with a constraint that both the patterns and the hidden variables are non-negative (which is appropriate for data such as electrophoresis traces). Cluster analysis can be seen as a special form of ICA, where only one of the hidden variable components may have non-zero value (and the value is one). Initially, this led us to consider an adaptive allele calling approach based on similar principles, with non-negativity constraints and a requirement that at most two of the hidden variables have non-zero values. Although we have managed to construct a prototype algorithm that works along these lines (and incorporating the constraint that a heterozygote pattern is a linear combination of the constituent allelic patterns), we soon encountered a inherently difficult problem: the number of genotypes (pair of alleles) in a marker can be quite large. For a marker of, say, 12 alleles, there are $\frac{1}{2}12(12+1) = 78$ genotypes. The typical sample size of a marker data is at most 96 traces, leaving most clusters (corresponding to genotypes) underpopulated. Estimation of the allelic patterns became unreliable, especially when the alleles are present only a few times in the data set (which is typical for microsatellite markers with many alleles).

The approaches above can be considered as 'non-parametric', because the allelic patterns are derived from the data, with minimal assumptions about how they are generated. If we are willing to make stronger assumptions about the patterns, such as by specifying a parameterized mathematical function that produces the stutter patterns, the number of parameter that needs to be estimated can be significantly reduced, leading to more robust estimates. Miller and Yuan [1997] proposed a rudimentary model for stutter pattern generation, based on a simple convolution where the main allelic peak is 'spread' to the adjacent fragment length positions, modeling a deletion and an insertion slippage by the polymerase enzyme, each with a certain probability. This convolution is then

applied repeatedly to simulate cycles of polymerase chain reactions. The authors noted, however, that the model was only a 'first-order' approximation, and did not take into account the fact that real allelic patterns vary with the allelic length[16]. Furthermore, it did not incorporate the non-templated plusA effects and unequal amplification ratio. Nevertheless, we believed that this approach was promising and we took it further by elaborating the model to include length-specific probability of slippage, plusA effect, and electrophoretic diffusion, covering almost all processes in figure 1.6. This model is the main engine of our allelic pattern recognition method. The details of the model and a method for optimizing the parameters directly from the data will be presented in chapter 3.

Including electrophoresis "time warps" should complete the model, allowing us to fit the theoretical patterns of a pair of alleles to an observed trace. However, it is difficult to specify a simple parametric model for the time warps. We cannot just slightly shift the patterns around like in the algorithm proposed by Stoughton *et al* [1997] because our allelic patterns are generated based on the allelic lengths, which do not correspond to the allelic sizes (migration rate relative to the SSF) in a simple way. In addition to that, repeatedly generating the warped versions of the patterns will be computationally expensive (we need to this while simultaneously estimating the allelic pattern parameters, which requires an iterative method). Therefore, the warp correction is performed first, using a method called 'trace alignment', which produces a trace data matrix where all peaks corresponding to the same DNA fragment are aligned across different lanes, and all the peaks are evenly spaced and located right at the integer points of a new scale which corresponds more closely to the ladder of DNA fragments seen in the data. The trace alignment method is the subject of chapter 2.

An allele calling method, which is essentially a classification method, is not complete if it cannot give estimates of the error probability of the calls, or at least a sort of quality scale which can be used to rank the genotypes according to the confidence of the calls. It is inevitable that some DNA samples will fail to amplify, or will be amplified weakly. Some markers may also contain systematic contaminants, due to high background signals, non-specifically amplified fragments, and color bleed. Some alternative genotypes might be inherently non-identifiable from the trace data (for example the allelic intensity ratio might be on the 'border-line' between a homozygote and a heterozygote). Those cases certainly need to be flagged, preferably by a quality indicator with a continuous value. Thus, assessing the performance of the allele caller can be done using a receiver operating characteristic (ROC) curve, or similar curves

[16]The convolution model is clearly time-invariant.

that indicate the trade-off between error rate and yields, as a function of the quality score cutoff. More importantly, the users of the called genotypes can flexibly choose a trade-off that is appropriate for their specific genetic analysis applications.

Quality indicators have a central role in our allele calling method. Instead of an attribute assigned to the genotype picked by the caller, it is computed to all possible genotypes, and the genotype with the best quality value is the call (or alternatively, when the trace data is ambiguous, a few genotypes with similarly good quality values may be reported). The quality value is derived from various feature variables based on the trace and the genotype. This will be detailed in chapter 4.

1.4.3 Implementation

A fully integrated implementation in a laboratory system is outside the scope of this thesis. Such implementation will require a lot of software engineering work related to data management and graphical user interface software. Our focus here is only on 'algorithmic engines'; small programs that perform the computationally intensive core algorithms of our allele calling system. It is still wise to consider some issues related to data exchange and operating systems, because we do want the programs to be portable and independent of the setup in a particular laboratory.

We need to be able to process data from a variety of instruments. Although all the data that we used in this project come from Applied Biosystems (ABI) instruments, nothing specific to the ABI system is assumed. The core algorithms only require a set of trace data, along with SSF information. A simple common data format for these types of information has been defined. It uses the structure of the filesystem (directory hierarchy and a strict file-naming convention), and a handful of file with simple formats (ASCII texts and a trivial binary format for the trace intensity arrays). We have written a `perl`[17] script for extracting the relevant data from ABI sample files into the common data format. It should not be too difficult to adapt this script for the trace files produced by other instruments.

There are many procedures in our system, and instead of writing a single complex program, the procedures are divided into many small programs (written either in C[18] or `perl`). This makes it easy to debug each component or test alternative approaches. The common data format serve as a 'hub' for communication between these components. The intermediate result of various algorithms

[17] `www.perl.org`

[18] ANSI/ISO/IEC 9899 standard or other compilers that support `inline` keyword and variable length arrays such as GCC (`www.gnu.org`).

can be dumped to the common data format to allow examination by programs such as generic data analysis packages (like Mathematica[19] or R[20]). The core algorithms themselves were written in C for maximum efficiency.

The common data format also facilitates development of graphical user interface software. The hierarchical file system is highly portable and the same directory structure can be mounted on UNIX or Linux machines (where the computation engine might be run) and on Apple Macintosh or Windows where the analysts can use a graphical user interface (written in Java) to browse and edit the trace data and genotypes. The interface is currently under development as a separate project (Keith Satterley, unpublished), and we will not discuss this any further. Although the common data format is only a collection of simple text files, they are designed with the principles of relational database in mind. Exporting the data, including the aligned trace data matrix, to other databases should only involve writing simple scripts.

[19] `www.wolfram.com`
[20] `www.r-project.org`

Chapter 2

Trace Alignment

2.1 Background

As outlined in chapter 1, the proposed allele calling system has two main steps: 1) trace alignment that compensates for electrophoretic time distortions, and 2) recognition of complex allelic patterns. The pattern recognition step treats the trace data as a multivariate data matrix, where the different electrophoresis lanes are the observation vectors, and the time point measurements along each trace are the variables. This treatment requires that we can identify the same variables across all lanes. The problem is that we cannot equate the time points of the raw data with the variables, because the lanes behave differently. Therefore, we need to find the mapping between electrophoretic migration time and the DNA fragment length. The concept of trace alignment is illustrated in figure 2.1. This chapter describes a method for estimating the mapping and for normalizing the time variations by resampling the trace data. We will examine the nature of the time variations first.

2.1.1 Electrophoresis of DNA fragments

The distortions in the time domain are introduced during electrophoresis. Electrophoresis is an analytical technique for separating soluble ions according to their charge and other physical properties, by placing them in a semi-solid, porous medium under an electric field created by two electrodes placed just outside the "starting" and "finishing" lines. Charged molecules will migrate toward the finishing line, driven by the electric field. The larger the charge on a molecule, the stronger the force of attraction. However, the medium also restricts the movement of the charged molecules selectively, depending on their physical properties and how they interact with the medium. The net result is that each type of molecule has its own migration rate, or velocity (measured in the distance traveled per unit time). In the case of microsatellite genotyp-

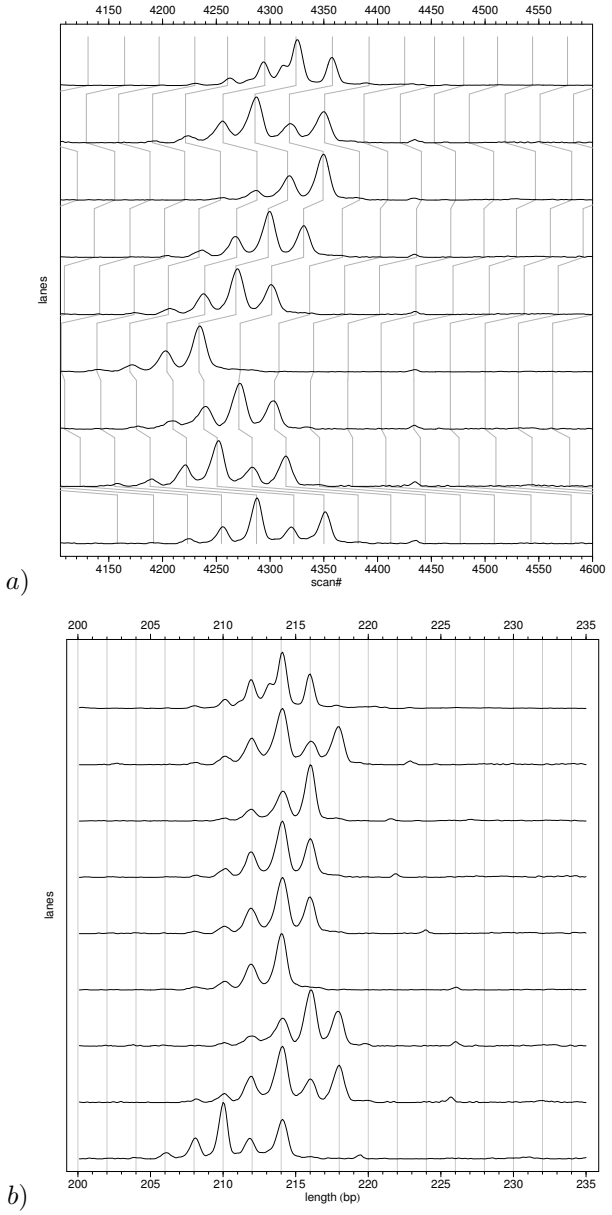


Figure 2.1: Trace alignment. Panel *a* shows preprocessed (color-separated and baselined) trace data from several lanes, from marker D1S2800 run on a ABI-3700 capillary electrophoresis machine. The horizontal axis is the detector sampling time, while the vertical axis of each lane is the fluorescence intensity. Each vertical gray line connects time points across different lanes that correspond to the same DNA fragment length. Panel *b* shows aligned representations of the same traces as in panel *a*. The trace data have been resampled and interpolated. Not only do the horizontal scales of all lanes become identical, but also the spacing between the DNA fragment peaks become uniform and the center of the peaks coincide with the integer points on the scale.

ing, fragments with different lengths will migrate with different velocities. To neutralize the effect of base composition and secondary structure (or folding) of the DNA molecules, denaturing conditions are used such that the velocity of a fragment is largely a function of the number of bases. The term 'mobility' is often used for the characteristic migration behavior of a type of molecules, under certain running conditions. Technically, mobility is the velocity per unit of electric field strength [Riekkola and Jonsson 2001]. The field strength can be assumed to be constant for each run and therefore, in practice, 'mobility' is often used interchangeably with 'velocity' (or even other related measures such as migration distance).

The relationship between the electrophoretic mobility and the physical property of the ions (such as chemical structure and size) is complicated, and depends greatly to the environmental conditions such as temperature and ionic strength of the buffer solution. For denatured DNA molecules, it is well-established that the migration time is roughly proportional to the fragment length [Southern 1979a, Carrano *et al* 1989]. This is illustrated in figure 2.2. Subtle changes in running conditions as well as idiosyncratic behavior of individual fragments might contribute to small systematic deviations from the linear relationship.

2.1.2 Size-standard fragments

The migration time cannot be used to identify the same fragment across different lanes, as illustrated in figure 2.1, and more clearly, in figure 2.3. This is because different lanes in the same gel might be loaded at different times, so that there are unknown lags. Additionally, in a large slab gel, conditions such as temperature and electric field strength might not be uniform throughout the gel. In capillary electrophoresis, each capillary tube might behave as if it is a separate "gel" [Gill *et al* 2001]. Thus, each capillary has its own characteristic running environment.

To normalize the variations and allow fragment identification, a set of fragments with known lengths are included in every lane [Mayrand *et al* 1992, Ziegle *et al* 1992]. One color channel is dedicated to the size-standard fragments (SSF) to avoid confusion with the unknown samples (see the red traces in figure 1.4, page 10). The fragments are chosen such that the pattern of the peaks can be used to assign the known lengths to the appropriate peaks. For example, the widely used ABI GS500 size-standard fragments consists of fragments with the length of 35, 50, 75, 100, 139, 150, 160, 200, 250, 300, 340, 350, 400, 450, 490, 500 bp. Note that they are mostly fragments spaced at 50 bp intervals, with a number of extra fragments irregularly placed to allow unambiguous identification.

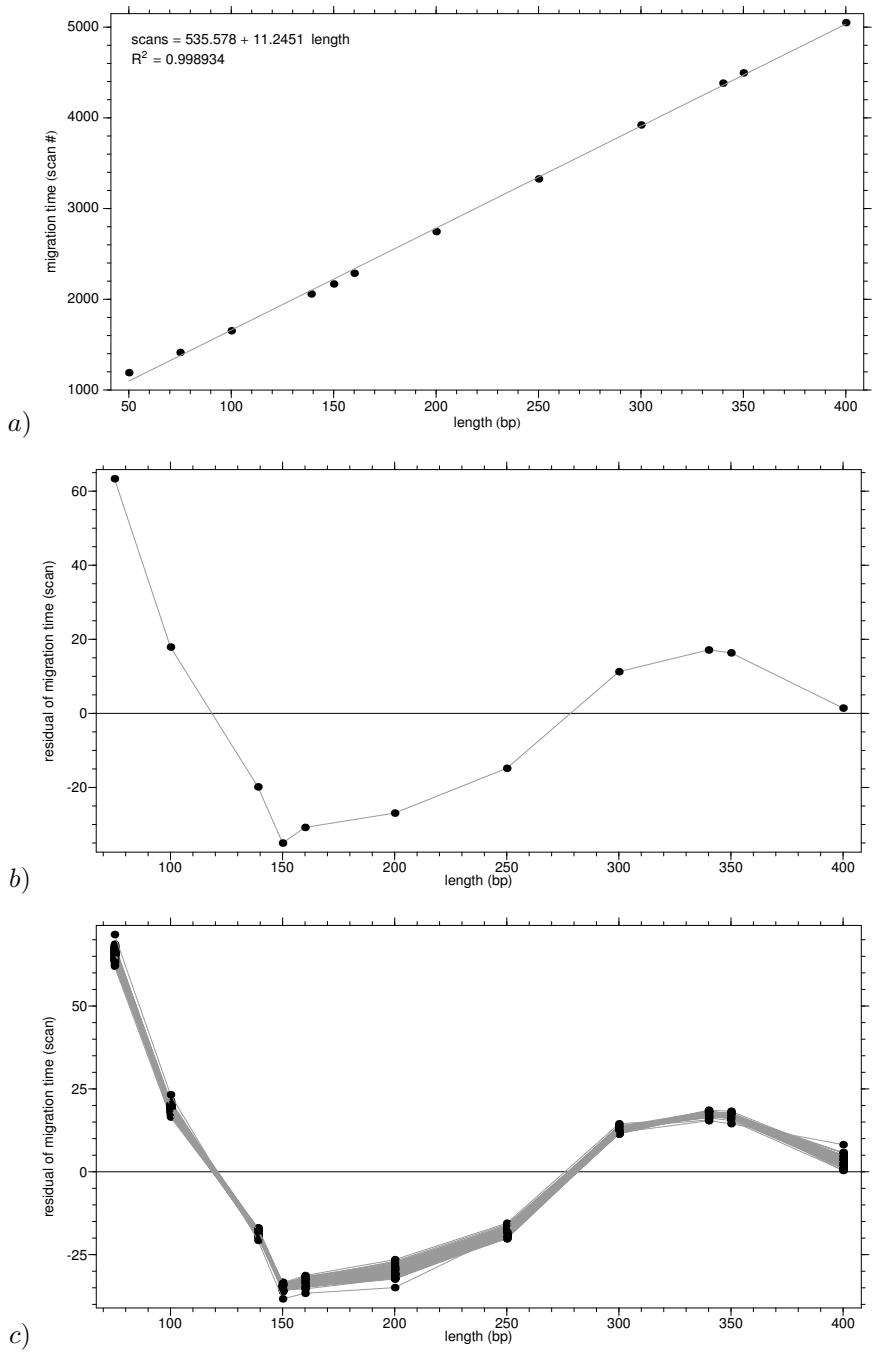


Figure 2.2: The relationship between fragment length (the number of nucleotides) and electrophoretic migration time. The data points correspond to fragments of known length (from ABI GS500 size standards, run on an ABI 377 machine). Panel *a* shows a straight line approximation using linear regression, with a very good fit. However, the residual plot (panel *b*) shows deviations around ±30 scans. Panel *c* shows residual plots of the same set of fragments, from 93 lanes in the same gel (linear fit is performed separately for each lane). This plot shows that the deviations are systematic (reproducible across different lanes).

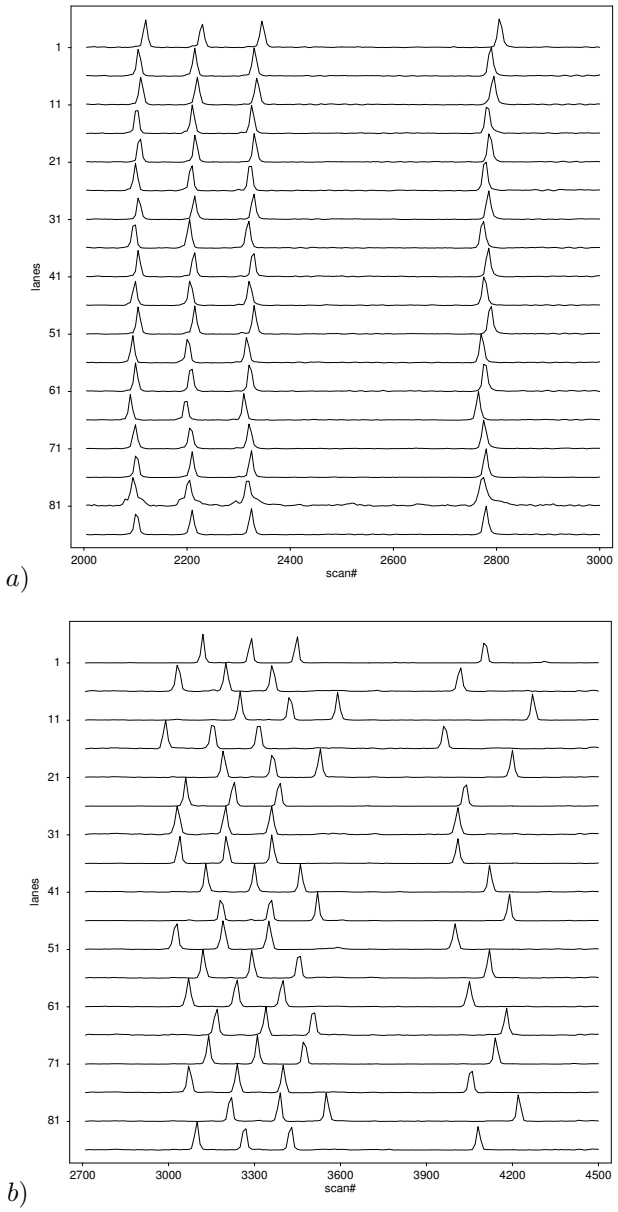


Figure 2.3: Lane-specific variations in migration time. The traces of ABI GS500 standard fragments were run on a slab gel electrophoresis (ABI-377), panel a, and capillaries (ABI-3700), panel b. Lane number $1, 6, 11, 16, \ldots$ are chosen to illustrate variations across the gel in panel a. Usually, even- and odd-numbered lanes are loaded at slightly different times, and the alternating lags can be seen on every other traces, in addition to the trend associated with the lane positions. In capillary electrophoresis, all lanes were started simultaneously, but each tube behaves differently.

Figure 2.4 shows how the SSF can be used to normalize migration time variations. The locations corresponding to a specific fragment relative to the SSF peaks are fairly constant across different lanes. We will use the term *size* to refer to this relative location, measured in basepair (bp). Although it has the same unit with the true fragment length, it is not identical. A size can have fractional values due to measurement errors, while the length is always an integer. Furthermore, as we will see shortly, the relationship between fragment length and size is not simple due to various biases.

An interpolation method must be used to determine the size based on sparsely located SSF. This procedure is called *sizing*. A curve is fitted to the SSF points (such as that shown in figure 2.2a), and the relationship between time and size can be looked up using the curve. The mapping between time and size can be used to resample the traces, as in figure 2.4b, or to assign sizes to the peaks in the traces with the original time scale. The latter is the more common practice, used by systems such as the ABI Genescan/Genotyper software. [This is because their approach is based on peak identification, where each trace is reduced to a set of abstract 'peaks', defined by a location and an intensity value (either the peak height or peak area).]

Various studies found that using SSF to size fragments is a reasonably precise method, with standard deviations typically less than 0.3 bp [Mansfield *et al* 1996, Idury and Cardon 1997, Wenz *et al* 1998]. In most cases, the precision is enough to distinguish alleles differing by 1 bp, although there are a few markers where some alleles might deviate by more than 0.45 bp [Idury and Cardon 1997], resulting in ambiguity of identification. In addition to the specific markers, the precision depends on the instrumentation [Deforce *et al* 1998] and, certainly, the electrophoresis conditions that affect fragment resolution (usually these have been well-optimized in practical genotyping protocols).

The choice of the curve fitting method also affects the precision [Ghosh *et al* 1997]. Of several sizing methods available in the ABI GeneScan software [ABI 1996], the local Southern method [Southern 1979b, Elder and Southern 1983, 1987] was found to be the most precise[1]. The problem of fitting a curve to SSF points can be considered a generic curve estimation problem, where many standard methods are known in other fields, e.g., Ramsay and Silverman [1997], Eubank [1999], Loader [1999], Hastie *et al* [2001]. Although the precision is limited by the underlying physical system, it is worthwhile to investigate alternative sizing methods that may contribute less to the imprecision.

[1] Note that only five methods are offered: 2nd and 3rd order polynomial least squares, local and global Southern, and cubic spline.

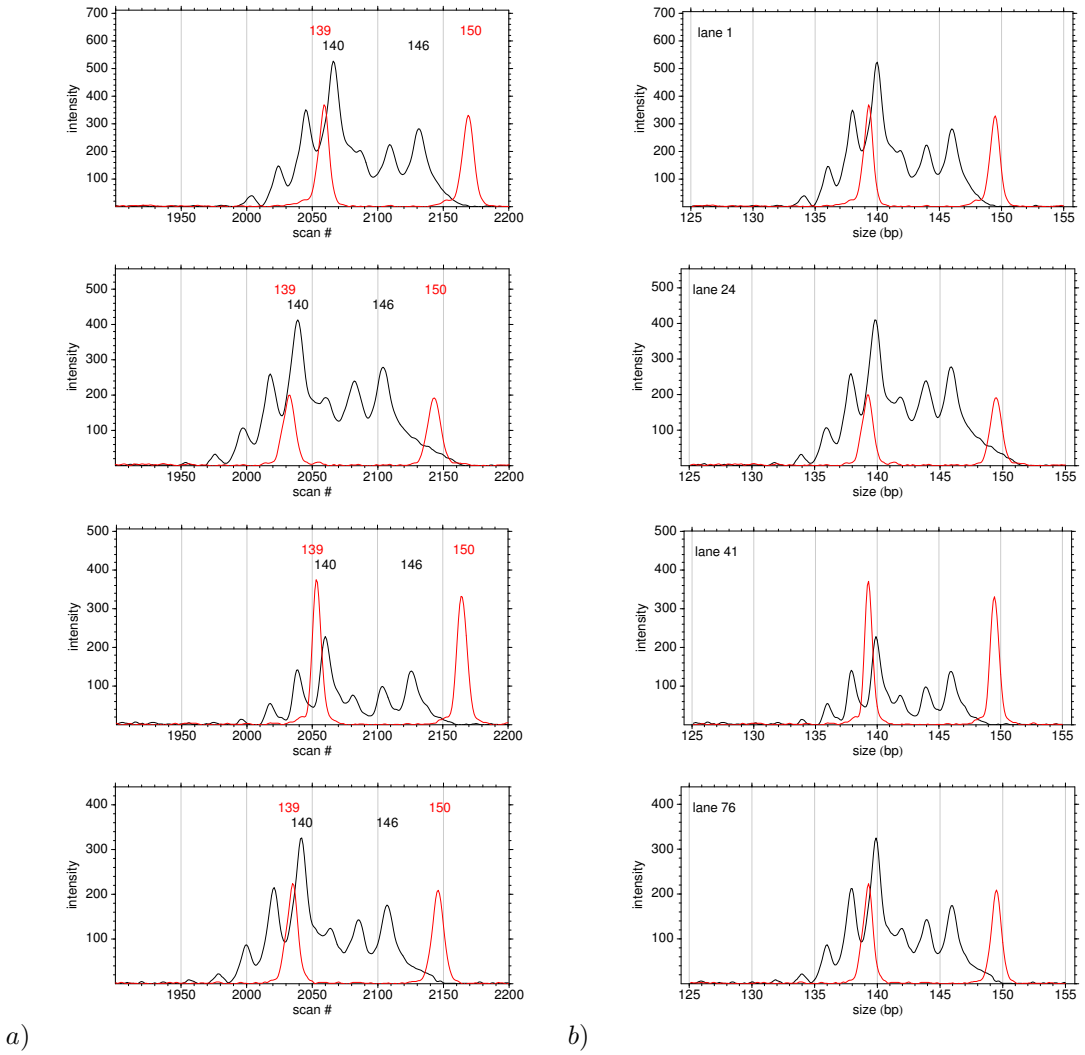


Figure 2.4: Using SSF to normalize migration time variations. The four black traces are different individuals with the same genotypes (140, 146) of the marker D18S452, while the red traces are some size-standard fragments (with known length of 139 and 150 bp, as indicated above the peaks). On panel *a*, the horizontal axes are the original time of measurements, and variations in time can be seen clearly. A sizing method is used to resample the SSF traces so that they are aligned (panel *b*). The unknown traces are also resampled co-ordinately. [The exact sizing method is not relevant to our point, and will be detailed later.]

2.1.3 Sizing bias

Although sizing based on SSF is adequately precise, in the sense that the same fragment can be identified across lanes, the method is not accurate, in the sense that the estimated size can be quite far from the true length (the number of nucleotides). In general, the bias can be length dependent. That is, the systematic deviation from the true length might vary across different fragment lengths. We will refer to this as a 'bias curve'. Various factors may contribute to sizing bias:

1. The fluorescent dyes attached to the unknown fragments are different from that attached to the SSF fragments. Because the dye molecules are bulky, they can significantly shift the mobility [Hahn *et al* 2001, Tu *et al* 1998]. Depending on the type of the dyes, size differences up to 7 bp might be encountered, with biochemically similar dyes, e.g. fluorescein derived or rhodamine derived, behaving similarly. Furthermore, the bias is not constant for all fragments but length dependent, following a certain curvature (figure 2 in Hahn *et al* [2001]).

2. The sequences of the unknown fragments are different from those of the SSF. Although under denaturing conditions the migration rate is roughly a function of the length, sequence-specific properties such as base compositions and secondary structure formation might still have significant effects. Weber and May [1989] found that the mobility of the complementary strands of microsatellite fragments, which has exactly the same length, differed. This was further investigated and confirmed by Saitoh *et al* [1998]. It is conceivable that, in general, different types of repeat, e.g. $(CA)_n$ vs. $(CTG)_n$ repeats, will have different bias curves.

3. Sequence-specific biases might also affect the SSF [Mayrand *et al* 1992]. For example, figure 2.2c shows that the sharp turn at the fragment 150 bp is systematic. This is reproducible even across different runs (data not shown), meaning that the effect is not due to the particular running conditions (which might be responsible for the overall curved trend), but due to idiosyncratic behavior of the SSF, particularly the 150 bp fragment. Because sizing of unknown fragments is based on the SSF points, one anomalous SSF point may affect all unknown fragments in the two intervals flanking the point, especially when the curve-fitting method is sensitive to this deviation. Using more carefully designed SSF might alleviate this problem [Mansfield *et al* 1996]. However, the ABI GS500 fragments are widely used SSF, including in genotyping data used for this project.

4. Different running conditions (such as temperature, batches of reagents,

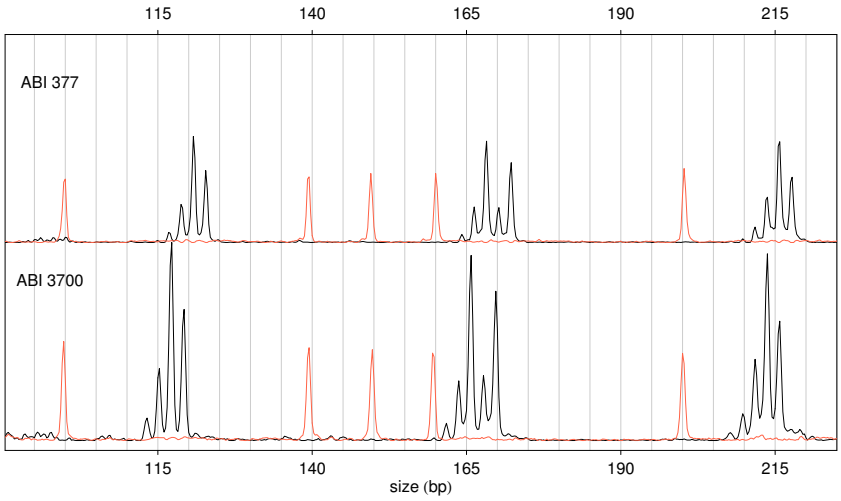
laboratory protocols, and instrumentation) may change the bias curve. An example is shown in figure 2.5, where the same fragments migrate differently, relative to the SSF, in capillary and slab gel electrophoresis.

5. The choice of the curve-fitting method may introduce artefactual biases. For example, straight-line regression (as shown in figure 2.2) certainly gives different values than linear interpolation (simply connecting the points by straight lines). Although no sizing method can remove the biases inherent in the physical design, an ideal sizing method will not introduce its own biases. A curve that follows each SSF very closely, for example interpolative methods such as Lagrange or splines, might be sensitive to some idiosyncratic fragments. On the other hand, 'smoothing' method that averages many points might increase the variance because faraway points can have significant influence, although physically they might behave independently. This is the issue of bias and variance trade-off, well-known in the area of curve estimation [Hastie *et al* 2001, page 37].

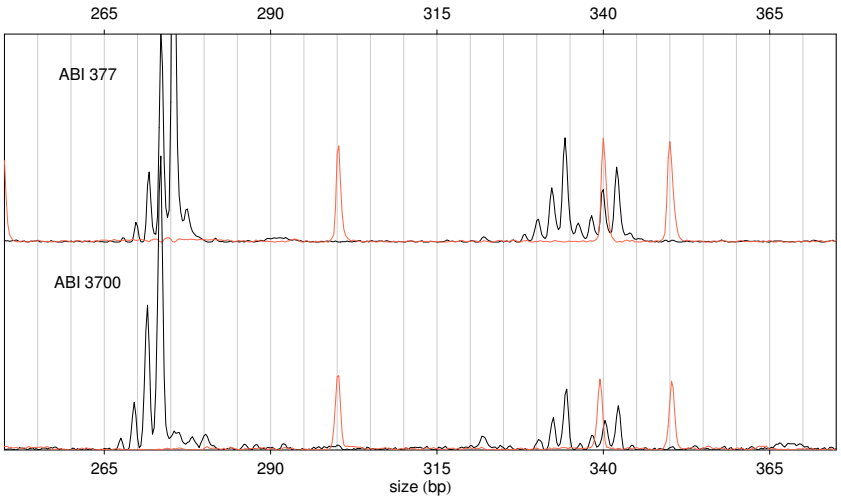
All of the effects above can be summarized by a marker-specific curve specifying length-dependent biases. The effect of this curve can be seen even in data from the same run. The mean size of a fragment may not be close to an integer (which should be if the unknowns and the SSF migrate in the same way). Furthermore, the distances between adjacent allelic peaks may not be integers either, because the bias curve is not constant. Also known as 'allelic drift' [Idury and Cardon 1997] or 'waving trends' [Zhao *et al* 1998], this phenomenon is illustrated in figure 2.6. The drift means that the unknown fragments migrate either slightly faster or slower than the SSF. The relative rate might also change, as shown by the curvature in figure 2.6c. A constant relative rate would have appeared as a straight line with a certain slope in the plot.

2.1.4 The implications of sizing errors to allele calling

The lack of sizing precision for some markers is an inherent loss of information and cannot be remedied. Calling errors can be avoided by flagging such markers, by looking at the estimated sizing variance [Idury and Cardon 1997]. A more refined method is by looking at the alleles. If they deviate by more than, say, 0.3 bp, the observations are discarded [Pálsson *et al* 1999]. For some markers, we can assume that alleles only occur at repeat-unit length, e.g. every 2 bp for dinucleotide repeats. Thus, deviations by more than 0.5 bp (but less than 1 bp) can be treated by rounding to the nearest multiple of repeat-unit length. However, when alleles differing by 1 bp do occur, they might be called erroneously. Markers that may have true alleles differing by non-repeat-unit



a)



b)

Figure 2.5: An example of instrument bias affecting the mobility of microsatellite fragments (black) relative the size-standard fragments (red). The same PCR products (of 5 different markers) are loaded into ABI-377 (slab gel) and ABI-3700 (capillary) instruments, along with GS500 size-standard fragments. The figure above shows the traces from the two electrophoresis runs. The two panels show different portions of same traces (100–225 bp and 250–375 bp). The horizontal axis has been transformed such that the SSF peaks are aligned, and the traces are "warped" accordingly by re-sampling and interpolation. In panel *a*, the relative migration is faster in the capillary than in slab gel. When the unknown fragments in the two runs are compared, a shift of ∼4 bp can be seen on the cluster of peaks around 120 bp. This shift progressively diminishes to about 2 bp around 215 bp. Further shift is seen for higher range (panel *b*). In fact, at 340 bp, the relative migration is slightly slower in ABI 3700. Note that this bias cannot be fixed by the choice of sizing methods, since they cannot change the order of fragment mobilities.

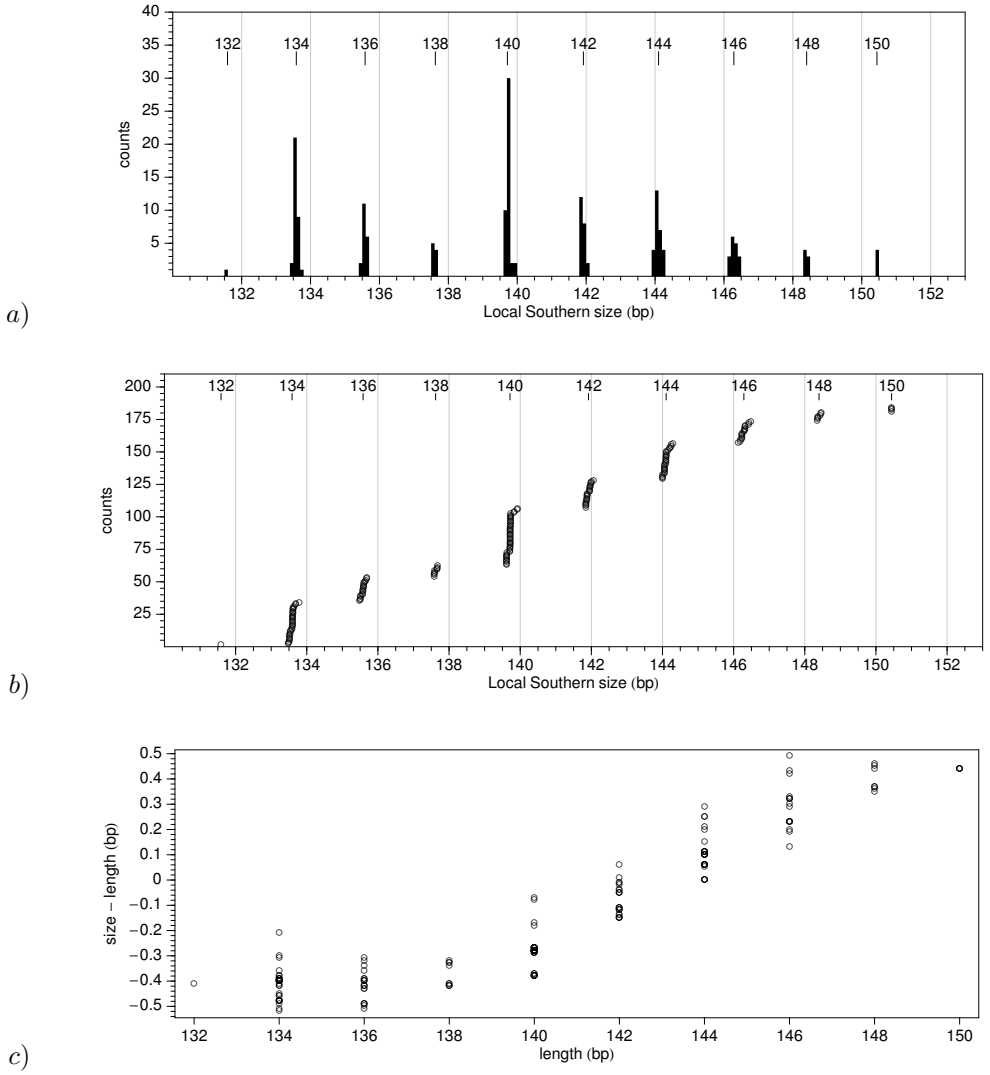


Figure 2.6: Bias curve and allelic drift. Panel *a* shows the histogram of 184 allelic sizes from marker D18452 (the stutter peaks have been removed and only the location of the centers of the allelic peaks are used). The length of the alleles is indicated above each 'cluster' of sizes. [Note that these lengths might differ from the true length by an arbitrary integer. For illustration, we can assume the labels are the true lengths.] The histogram's bin size is 0.1 bp. Panel *b* shows the empirical cumulative distribution (suggested by Li *et al* [2001], except that they flip the axes), which show the actual location of each points. On both panel *a* and *b*, we can see an example of "allelic drift", where the size drifted slightly from the length across the range. If the length are known (or presumed), the bias curve and the size variation for each allelic length can be seen more clearly by plotting size minus length against the length (panel *c*).

lengths may be recognizable from the frequency of occurrences of such alleles and their consistent locations.

The implications of sizing bias are more serious. In some applications, such as forensics and paternity testing, mis-identification of alleles is unacceptable. The problem of sizing bias is avoided by running an 'external standard' which is a mixture of all known alleles for a given marker. These 'allelic ladder' fragments are labeled using the same fluorescent dye and run on lanes next to the unknown lanes [Puers *et al* 1993, Smith 1995a]. Identification can be performed by directly comparing the sizes of unknown alleles with those in the allelic ladder. This approach, however, is too expensive for applications such as genome scans involving hundreds of markers[2] and many individuals. Allelic ladders must be painstakingly constructed and maintained for all markers. They also take up many electrophoresis lanes that are otherwise useful for genotyping the unknowns.

For other downstream applications such as disease mapping and marker-assisted breeding, the consequences of sizing bias are not too dire, because it is actually not important to know the exact allelic lengths. Most statistical genetic analyses consider allele labels as unordered enumeration of possible genetic states. As long as the alleles are labeled consistently in the whole data set, any labeling scheme will do (including arbitrary re-coding and permuting the labels). However, it is convenient if the allele labels are chosen to correspond to the true length. When new alleles are encountered, their labels have already been allocated (there is no need to 'squeeze in' out-of-order or inelegant labels). More importantly, combining data from different sources is impossible if each source uses its own arbitrary labeling scheme.

For most purpose it is sufficient to use the *relative length* of an allele. This is a number that differs from the true length by an unknown constant integer (but consistent throughout the data set). Assigning consistent relative length is more difficult that simply rounding the sizes to the nearest integer. For example, if such rounding procedure is applied to the allele sizes clustered around 133.5 bp in figure 2.6, some sizes will be rounded up while others rounded down, depending on whether they are more or less than 133.5. Proper rounding procedure should put all alleles around 133.5 under the same label, or 'bin'. Whether the label is 133 or 134 (or other integers, for that matter) is arbitrary, provided that the labels of other bins are spaced accordingly, such that the difference between adjacent allelic labels do not differ too much from the spacing between adjacent clusters of allelic sizes. In other words, the bias curve, such as that implied by figure 2.6c, should be as smooth as possible.

The procedure of identifying the periodic clusters of allele sizes is known

[2]The 'human identification' marker set used in forensics consists of only a few markers.

as 'binning'. Manual binning it is quite intuitive and straight-forward. Even before the advent of automated fluorescent technologies, binning could be done by directly 'counting the DNA ladder' seen in the photographic gel image. We have no access to such data, but the concept can be illustrated by constructing a 'pseudo-gel' image from the trace data of automated electrophoresis (figure 2.7a). When examining such image, a trained analyst would consider all lanes simultaneously, mentally constructing a 'DNA ladder' associated with periodic locations of the fragments. Although there is no lane-specific SSF and a sizing curve is not usually constructed[3], the alleles can be scored by counting the number of stutter peaks separating them. In traces with well-separated alleles, such as those near the bottom of the image in figure 2.7a, it is possible to correctly assign the relative length by counting the peaks in the intervening region, seen in other lanes. A similar procedure can be used for automated fluorescence trace data (the method is called 'direct counting' by Haberl and Tautz [1999]). An alternative way to view the fragment ladder is by plotting all traces overlaid as in figure 2.7b. This plot can be used to manually identify the 'bins'.

Several automatic binning procedures have been published [Idury and Cardon 1997, Ghosh *et al* 1997, Li *et al* 2001], in addition to proprietary algorithms from instrument vendors. These methods take as their input the allele sizes after they are called (such as those in figure 2.6a), instead of the raw trace data. This means that the periodicity of the stutter peaks, which may help identifying the bins, are ignored. However, the methods are easier to implement (because they operate on a set of points instead of on a data matrix) and the allelic sizes are easily available as the output of existing (semi-automatic) allele calling software.

In the approach taken by Idury and Cardon [1997] the bins are specified by a set of bin boundaries with a constant spacing (therefore assuming that the bias curve is a straight line). The two parameters (for bin spacing and offset) are estimated by minimizing the variance of allele sizes within each bin. This procedure will not be optimal for markers where the bias curve is not linear (such as the one in figure 2.6). A slightly different approach was proposed by [Mansfield *et al* 1994]. Binning is applied to all peaks including the stutter peaks (after the peak identification steps; thus not using all trace data). A histogram similar to that in figure 2.6a is constructed, and after some filtering steps, the highest peaks are used as the bin locations. We can therefore consider this as a non-parametric method. As opposed to that of Idury and Cardon [1997], the bin locations are not constrained by the requirement of linear biased curve. It

[3]In this technology, care is taken that the electrophoresis of all lanes are synchronized as much as possible, to allow direct comparison across lanes (manually).

is not clear how Mansfield's method perform in the presence of noisy peaks[4].

There is an inherent difficulty with binning based on the patterns found in the data. There are certain markers where the allelic distributions are 'disconnected'. That is, the alleles are distributed in separate clusters, intervened by significantly wide regions where no alleles are observed (see figure 2.8). Although the relative lengths within each cluster can be reliably estimated, the length difference between alleles from different clusters might not be, because there is no ladder of peaks that can be used to count the difference.

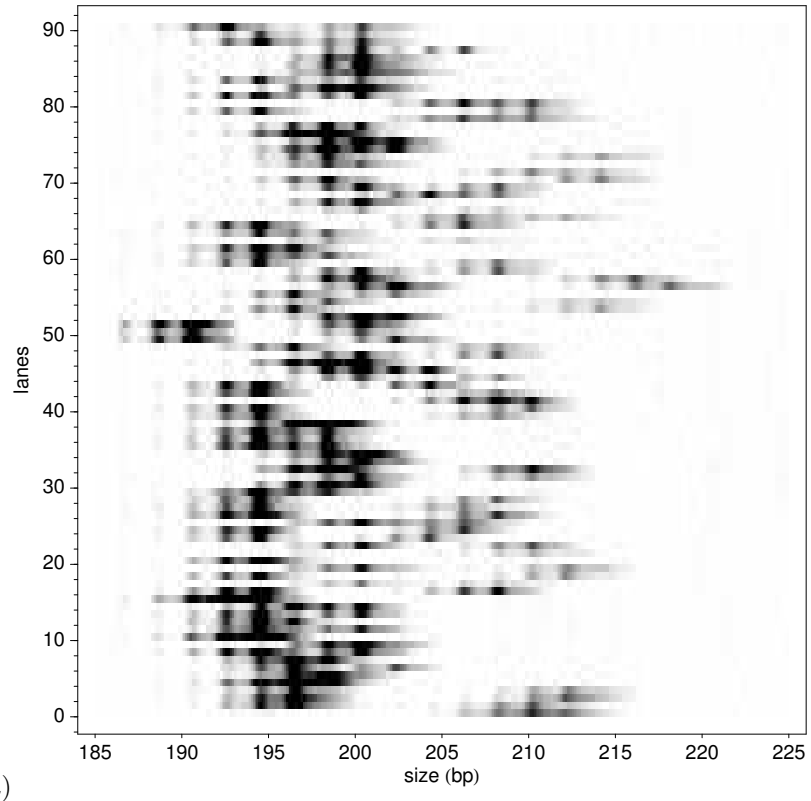
Large genotyping projects might be a collaboration of many labs using different measurement conditions, consequently with different biases. Even within the same lab and instrumentation, the environmental conditions (or subtle properties in the batches of reagents) might change during the course of a large project, which may last several years. Sizing bias can cause problem in merging data from different sources. This is also illustrated in figure 2.8, where the sizes of the same alleles differ between ABI-377 and ABI-3700 instruments. Binning certainly needs to be performed first on each data set (for most markers this can be done reliably, without the ambiguity exemplified in figure 2.8). Combining the integer-valued relative lengths of the alleles is easier than combining the real-valued sizes. The problem is how to resolve the unknown integer constant in each data set.

One possible way is by including a control individual in each run [Knowles *et al* 1992, Ghosh *et al* 1997]. One example is the commonly used CEPH family member #1347–02. The difference between the known lengths of the control and the length from binning is the constant adjustment to be made for all other alleles in the run. However, CEPH controls may occasionally fail (or migrate erroneously). In the cases where the binning within each run might be inconsistent (such as when the ladder is disconnected), the CEPH genotype may not be able to completely resolve the lengths, because it only has two alleles (which might even be homozygous). Alternatively, in many cases we may assume that the allelic frequency profiles of a marker are fairly similar across different runs[5], such as those in figure 2.8. Adjustment can thus be made based on pattern similarity between the profiles. If the binning in each run is consistent, a simple integer shift can be used to synchronize the allelic lengths, say by maximizing a similarity score between the shifted profiles.

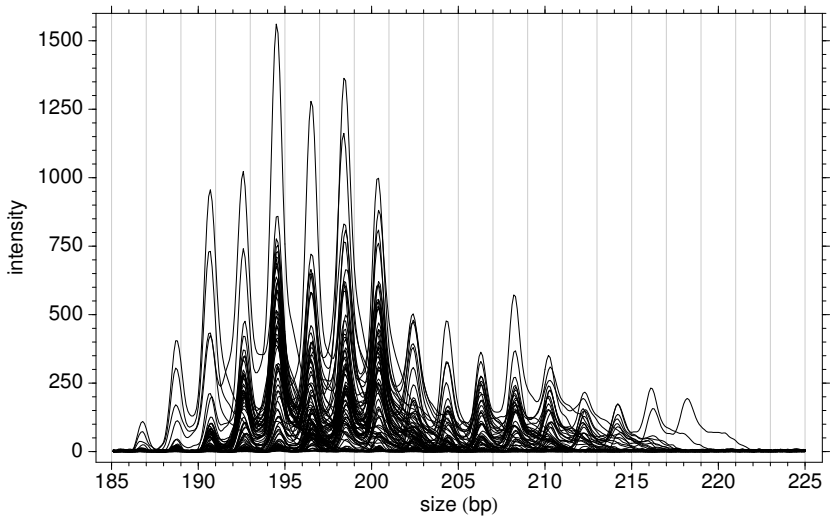
Whichever method is used to merge genotypes from different sources, allele

[4]The details of the algorithm were not published. It was a part of the, now defunct, Pharmacia ALF/ALP system and it is not clear whether the algorithm is still being used elsewhere today.

[5]In studies where allelic frequency between population might differs, such as in case and control studies, care must be taken when dividing the samples into separate runs.

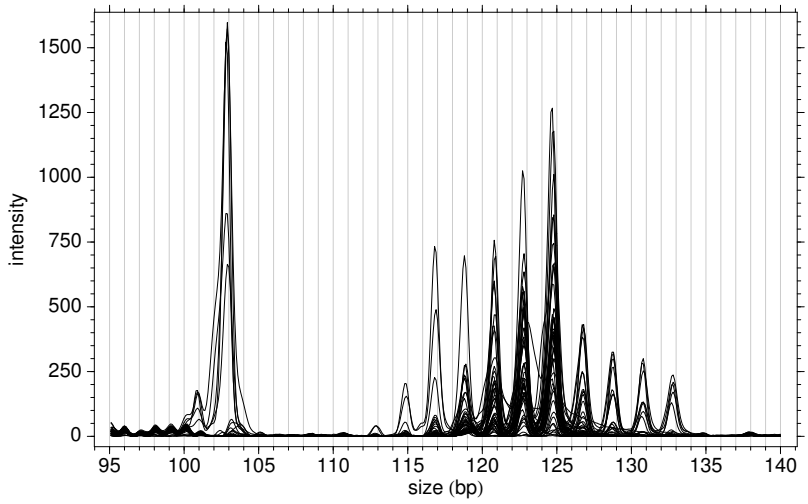


a)

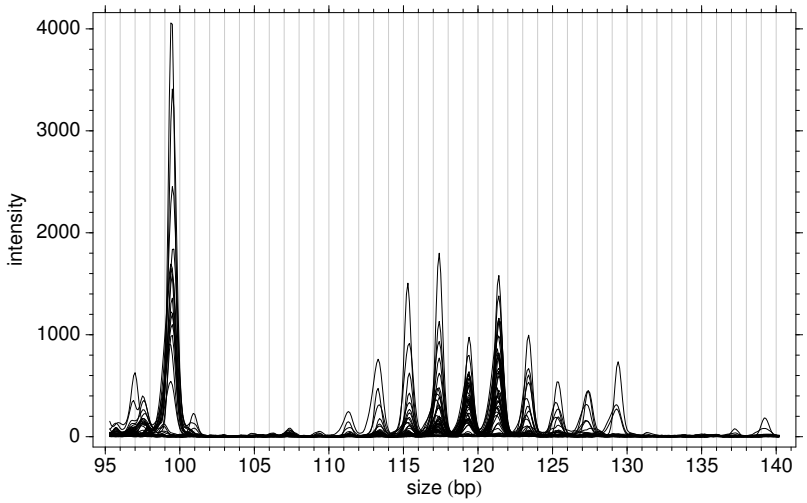


b)

Figure 2.7: DNA Fragment ladder. Panel *a* shows an image of a trace data matrix. The traces have been aligned based on their respective SSF. Panel *b* is the overlay plot of the same trace data.



a)



b)

Figure 2.8: A marker with 'disconnected' allelic distribution. The PCR products of the marker D1S2797 (different set of individuals) were run under a slab gel electrophoresis (ABI-377), panel *a*, and capillaries (ABI-3700), panel *b*. Peaks are absent from the region around 105–115 bp in panel *a* (or 103–112 bp in panel *b*). When the two data sets are binned separately, the size difference between the leftmost allele (103 bp in panel *a* and 99.5 bp in panel *b*) and the main cluster of peaks might be rounded in different directions.

label incompatibility is currently still a serious problem [Weeks *et al* 2002]. We think the main source of difficulty is binning inconsistency resulting from a 'disconnected allele ladder'. Including all peaks, including the stutter peaks, may improve the reliability of automatic binning procedure. If inconsistency is inevitable due to a lack of information in the data, then the procedure for merging genotypes from different data sets should use more flexible adjustments, by allowing local shifts instead of a single constant. Local similarity of the allelic frequency profiles might be used to optimize the alignment.

2.2 Formulation

Our proposed solutions to the sizing, binning and merging problems consist of two main parts. The first procedure performs sizing and binning simultaneously on the raw trace data, producing a matrix of aligned traces. This is done for each marker in each run. The output is passed on to the allele caller (to be described in the next two chapters). The second procedure combines the genotypes (the result of allele calling) from different runs, based on local similarity of the allele frequency profiles. This step is performed on the allele labels instead of the whole trace because the main assumption is that there is similarity between allele frequencies, which is not exactly the same with the combined trace patterns. Furthermore, combining traces from multiple runs to be analyzed together by the allele caller is not beneficial because each run might have its own systematic effects. We describe the merging procedure here because the algorithm is very similar to that for trace alignment.

Our aim is to estimate the mapping between migration time and fragment length, through the size relative to SSF and the relative lengths 'counted' from the observed fragment ladder. Let us define the domains more precisely:

1. $s \in \mathbb{R}$ denotes the time scale of the raw measurement. This corresponds directly to physical time. The unit is arbitrary and usually it is the number of sample points since the measurement is started (called scan numbers or scan#). Although the signals are discrete, we treat them as if they were continuous. The intensity values corresponding to fractional scan numbers are interpolated.

2. $u \in \mathbb{R}$ denotes the scale relative to the SSF. This scale depends on the curve-fitting method.

3. $t \in \mathbb{R}$ denotes the relative length of the fragments. Although the length of a fragment is an integer, it is convenient to treat t as continuous. The center of DNA fragment peaks in this scale are located at integer points. This scale specifies the fragment ladder or the 'binning scheme'.

4. $\tau \in \mathbb{R}$ denotes the true length, in the sense that alleles from different runs are directly comparable under this scale. It does not have to correspond to the absolute number of nucleotides in the PCR products, which is genetically irrelevant.

Input

A *trace* is a real-valued, continuous function of any one of the scales above. The range is limited to the interval of the marker in question. Each data set is a collection of n traces. The 'raw' traces (the input signals) are the results of preprocessing steps (lane-tracking and color-separation), denoted by $x_j(s)$, where the index $j = 1, \ldots, n$ identifies the lanes, and s is the scan number as described above. Although the raw data are discretely sampled (at the points where s is an integer), the changes in intensity are smooth enough for the values in between the measurement points to be interpolated, using a method to be detailed later. The same applies to the other scales.

The SSF information is assumed to be available for each lane as a set of pairs $\mathcal{S}_j = \{(u_1, s_1), (u_2, s_2), \ldots, (u_i, s_i), \ldots\}$. Each pair corresponds to an SSF fragment, where u_i is the known length[6] and s_i is the scan number. Another piece of information to be specified is the marker interval $(t_0, t_k]$ covering the range of all possible alleles (with some safety margins). $k = t_k - t_0$ is the interval width in basepairs, so that at most there are k DNA fragments. The open left boundary of the interval is for convenience when working with discrete data (thus the number of data points correspond easily with the interval width). The 'output resolution', denoted by T, needs to be specified. This is the number of data points per basepair in the resampled traces. Choosing $T = 10$ is sufficient (no loss of relevant information) and convenient. In the raw data that we used, the sampling rate varies. On average, it is around 11 points per bp for ABI-377.

It is not necessary to specify the repeat types, such as di- or trinucleotide repeats. All traces are treated as if they had a fragment ladder with 1 bp periodicity. This allows the same setting to handle all markers with non-repeat unit alleles as well as those with extensive plusA peaks. Stronger assumptions about the periodicity can be optionally specified, although all results in this project use 1-bp periodicity assumption. We would like to know how far this can be pushed, because it is convenient to specify minimal marker-specific information and the periodicity assumption may depend on allelic distribution of the population being studied.

[6] We use the symbol u instead of t because this is the known length of the SSF, which is a different scale from the length of the microsatellite fragments, due to dye bias.

Outline of the method

We want to find the mappings between the scales:

$$s \leftrightarrow u \leftrightarrow t \leftrightarrow \tau \tag{2.1}$$

which are bijective (invertible), monotone increasing, continuous and smooth. These properties are used to guide the estimation of the mapping.

The output of the method is a set of aligned traces $y_1(t), \ldots, y_j(t)$, for $t \in (t_0, t_k]$, Each trace y_j is a result the function composition:

$$y_j(t) = x_j \left\{ s_j[u_j(t)] \right\} \tag{2.2}$$

where:

- $x_j(s)$ is the raw trace data (with interpolation when necessary)
- $s_j(u)$ is based on the SSF points $\mathcal{S}_j$ interpolated according to a curve-fitting method.
- $u_j(t)$ is estimated from the observed fragment ladder according to

  $$u_j(t) = t + \phi(t) + \psi_j(t) \ . \tag{2.3}$$

  $\phi(t)$ represents marker-specific systematic bias, which applies to all lanes, and $\psi_j(t)$ is "jitter", or time variation specific to each lane.

$\phi(t)$ and $\psi_j(t)$ need to be estimated from the data. A much simplified version of the algorithm (for illustration) is as follow:

1. Assume $\phi(t) = 0$ and $\psi_j(t) = 0$, and resample $y_j^{(1)}(t) = x_j[s(t)]$.

   Summarize all lanes, e.g. by computing $y^*(t) = \max_j y_j^{(1)}(t)$.

   Find ϕ as a curve that minimizes the difference between the summary y^* and a periodic function, say $w(t) = \cos(2\pi t/T)$:

   $$\min_{\phi(t)} \int_{t_0}^{t_k} |y^*[t + \phi(t)] - w(t)|^p \ dt \ . \tag{2.4}$$

   where p specifies the appropriate norm.

2. Assume $\psi_j(t) = 0$, and resample $y_j^{(2)}(t) = x_j[s(t + \phi(t))]$.

   Summarize all lanes, e.g. by computing $y^*(t) = \max_j y_j^{(2)}(t)$.

   For each lane j, find $\psi_j(t)$ as a curve that minimizes the difference between the summary y^* and each trace $y_j^{(2)}$.

   $$\min_{\psi_j(t)} \int_{t_0}^{t_k} |y_j^{(2)}[t + \psi_j(t)] - y^*(t)|^p \ dt \ . \tag{2.5}$$

   where p specifies the appropriate norm.

3. Resample $y_j^{(3)}(t) = x\{\ s[t + \phi(t) + \psi_j(t)]\ \}$

The main idea of the approach is that two signals that are similar except for a small time distortion can be aligned (equation 2.4 and 2.5). This is a problem encountered in many fields [Ramsay and Silverman 1997, Sankoff and Kruskal 1983, Wang and Isenhour 1987, Rabiner and Juang 1993, Mott 1998] and can be solved by methods such as dynamic programming (also known as dynamic time warping or DTW). Usually both signals are observations, but here, in the first alignment (equation 2.4) we compare an ideal fragment ladder (y^*), which is made of periodic peaks at the expected locations (integer points of t), with the ladder obtain by combining all traces, which is essentially the outline (or "skyline projection") of the overlaid plots such as figure 2.7b. The discrepancy between the two is the systematic bias (or allelic drift). The second alignments (equation 2.5) reduce the variation between lanes and put the center of all peaks at integer points. Here each individual trace is aligned with the empirical fragment ladder constructed from the traces that have been corrected for ϕ.

We do not attempt to treat equation 2.3, 2.4 or 2.5 as rigorous statistical models. The method is somewhat *ad hoc* and the ultimate justification is comparison with binning made by human judgment. As the result of trial-and-error improvements, the actual algorithm contains many *ad hoc* modifications and constraints. These will be detailed in the next section. Some of the modifications are: 1) the use of enhanced traces (by frequency filtering and intensity thresholding) results in stronger features for alignment and resistance to noise, 2) absolute difference ($p = 1$) for the alignment scores, 3) second-order loess or locally weighted regression for the sizing curve, and 4) using the average of a few largest intensities for constructing the summary ladder (y^*), to make it more robust to noisy peaks.

Merging genotypes from different runs is based also on signal alignment. Here the signals are the allelic frequency profiles from each run. One run is arbitrarily chosen as the 'reference' and the others are aligned to this reference run. Letting $f(t)$ the allelic frequency profile of a run, and $g(t)$ of another, we want to find the set of local shifts $\xi(t)$ that minimizes:

$$\sum_{t=t_0}^{t_k} \{f(t) - g[t + \xi(t)]\}^2 \tag{2.6}$$

Dynamic time warping alignment is also used. We use a summation sign instead of an integral to highlight the discrete nature of the allele labels[7]. Different

[7]Dynamic programming alignment is always a discrete method. However, in the case of trace alignment, interpolation and smoothing are applied to the alignment curve ϕ and ψ_j to reflect the continuous nature of electrophoretic distortions.

constraints from those for trace alignment need to be used to reflect the nature of the distortions, which is binning inconsistency due to sparse fragment ladder.

2.3 Algorithm Descriptions

2.3.1 Trace resampling and interpolation

As indicated in the outline of the proposed method, we need to repeatedly resample $y_j[s_j(u)]$ for all j. This involves looking up the value of $x_j(s)$ and $s_j(u)$ for arbitrary real values of u. The interpolation method has to keep the relevant information intact, as well as be computationally efficient.

To ensure the fidelity of resampled signals, we can use the Shannon-Whittaker sampling theorem as a guideline (Mallat [1998, section 3.1], Press *et al* [1992, section 12.1]). Roughly, a signal can be represented by sampling at the interval of Δu, if it does not contain Fourier components with smaller periods than Δu. Choosing Δu to be $1/T = 0.1$ bp per data point should be more than sufficient for our purpose. $s_j(u)$ is a very smooth function (see figure 2.2, page 31) relative to Δu. The corresponding Δs is, on average[8], about the same with the sampling interval of the raw data from ABI-377 instrument (around 11 data points per bp under the typical gel concentration, electrophoresis voltage and scanning rate at 2400 per hour). In ABI-3700 instrument the sampling rate of the raw data is about 16 data points per bp. The additional sampling resolution does not contain more information other than background fluctuations. To avoid aliasing, a smoothing filter is applied to the raw traces before resampling them. The filter cutoff is chosen such that Fourier components with periodicity smaller than average Δs are attenuated. For efficiency and flexibility, a recursive smoothing filter is used (see Appendix A).

After smoothing the raw signal, the value of $x_j(s)$ for arbitrary (non integer) s can be looked up using first-order Lagrange (linear) interpolation. Letting s_a and s_b the nearest integer scan numbers of the raw data points ($s_a \le s \le s_b$), the interpolated intensity corresponding to s is:

$$x(s) = \frac{s_b - s}{s_b - s_a} x(s_a) + \frac{s - s_a}{s_b - s_a} x(s_b). \tag{2.7}$$

Because $s_b - s_a = 1$, this can be computed efficiently by

$$x(s) = x(s_a) + (s - s_a)\left[x(s_b) - x(s_a)\right] \ , \tag{2.8}$$

where s_a and $s - s_a$ can be obtained readily from the built-in machine instruction[9] that returns the integral and fractional value of a floating point number.

[8] Because of the time warping effects, the resampling rate is slightly non uniform.

[9] Accessible, for example, by the `fmod` function in ANSI C standard library.

In this way, the lookup operations can be done very efficiently. There is no need to perform interval search as usually done in generic interpolation methods, such as that in [Press *et al* 1992, section 3.1]. For the same reason, evaluation of $s_j(u)$ is done by pre-computing the values just once for a range of values of u, at $\Delta u = 1$ bp, based on whichever curve estimation method is chosen to fit the SSF points. This represents the curve $s_j(u)$ as a discrete signal with a very good approximation. Subsequently, repeated lookup operations are done using linear interpolation analogous to equation 2.8.

2.3.2 Sizing curve

We need to find the relationship between s and u for each lane, given the observed SSF points $\mathcal{S}_j = \{(u_1, s_1), (u_2, s_2), \ldots\}$. Given a set of scan numbers s_j from different lanes associated with the same fragment t, the sizing method must minimize the variance of the corresponding u_j's. Although sizing bias is inevitable due to physical factors, the sizing method should not add artificial fluctuations to the bias curve. It is less difficult to design bias correction or binning methods (which rely on the expected periodicity of the fragment ladder) if the bias curve is as flat (close to a constant) as possible, or at least as linear as possible, i.e. the fragment ladder is as evenly spaced as possible. Although our method for bias correction does not assume the bias curve to be linear or specified by other parametric forms, it has some smoothness constraints that limits the flexibility of the estimated bias curve. Relaxing these constraints risks making the estimate less reliable and more sensitive to noisy data.

We compared three different methods: linear interpolation (first-order Lagrange), the commonly used local Southern method, and second-order locally weighted polynomial regression [Cleveland 1979, Hastie *et al* 2001], which, to our knowledge, has never been applied to this problem. Other methods such as cubic splines and ('global') polynomial regressions have been shown to have larger variance than the local Southern, possibly because of its local nature. Linear interpolation is the most 'local' method (depending only on two flanking points). It would be interesting to compare the variance and bias of linear interpolation to those of the local Southern method, which is more complicated to compute[10].

The details of the local Southern as implemented in ABI GeneScan are not published. We found that it is possible to reproduce their output (up to 2 decimal digits, which all that they produce) by following the procedure detailed in the manual of an alternative fragment analysis software, Genographer AFLP[11]. This is similar in principle to the one outlined in Elder and Southern [1987].

[10] Interestingly, linear interpolation is not available in ABI GeneScan.

[11] `http://hordeum.oscs.montana.edu/genographer/help/lsouthern.html`

One important 'modification' is that ABI GeneScan treats the scan number as if it was mobility (which is what the original Southern's formula expects as input). Note that by definition, the mobility is the reciprocal of the scan number (migration time). In a sense, this is a mis-application of the local Southern method, which was originally intended for fixed-time electrophoresis that measures migration distance, which is proportional to mobility. However, empirically this does not cause grossly erroneous results. Further examination of the local Southern methods indicates that it is a generic interpolation method that does not depend too much on the original physical model of DNA fragment migration (the ABI GeneScan manual mentions that it is 'closely related' to cubic spline).

Both linear interpolation and the local Southern are 'interpolative' in the sense that the curve has to pass through all SSF points. This makes them more sensitive to sequence-specific SSF idiosyncrasies. On the other hand, methods based on regression consider the migration time s_i of a given known length u_i to contain some errors. It is difficult, however, to specify a parametric form for the expected migration time as a function of the length because changes in running conditions might affect the curvature systematically, in unpredictable ways. Nevertheless, we know that locally the curve should be smooth. It is reasonable, therefore, to try a smoothing or 'non-parametric regression' method, such as the locally-weighted polynomial regression (also known as loess curve).

Second-order polynomials are used to avoid the effect of flattened troughs and valleys found in first-order loess [Hastie *et al* 2001, page 171]. Dropping the subscript j for simplicity, we approximate the curve $s(u)$ for a given lane by

$$s(u) \simeq \beta_u^{(0)} + \beta_u^{(1)} u + \beta_u^{(2)} u^2 \ . \tag{2.9}$$

Here $\beta_u^{(r)}$ are the regression coefficients obtained from minimizing

$$\min_{\beta_u^{(r)}} \sum_{i=1}^{p} K_\lambda(u, u_i) \left[s_i - \beta_u^{(0)} - \beta_u^{(1)} u_i - \beta_u^{(2)} u_i^2 \right]^2 \tag{2.10}$$

where $u_1, \ldots, u_p$ and $s_1, \ldots, s_p$ are the known lengths and scan numbers of the p size-standard fragments. K_λ is the kernel than controls the locality of the approximation. We use the Gaussian kernel:

$$K_\lambda(u, u_i) = \exp\left\{ -\frac{(u - u_i)^2}{2\lambda^2} \right\} \ . \tag{2.11}$$

λ is chosen based on visual inspection on a wide range of observations (of ABI GS500 SSF in various runs). The value of 50 seems to be good enough. Lower values causes discontinuity in the curve (note that the spacing of SSF in ABI GS500 is roughly 50 bp), while higher values causes the local features to be missed. Values between 50 and 100 do not seem to produce significantly different curves.

The normal equation is used to solve the problem (using Cholesky decomposition):

$$A^{\mathrm{T}}W^{\mathrm{T}}WA\boldsymbol{\beta} = A^{\mathrm{T}}W^{\mathrm{T}}W\boldsymbol{s} \tag{2.12}$$

where

$$A = \begin{bmatrix} 1 & u_1 & u_1^2 \\ \vdots & \vdots & \vdots \\ 1 & u_p & u_p^2 \end{bmatrix}, \tag{2.13}$$

$W = \operatorname{diag}[W_\lambda(u-u_1), \ldots, W_\lambda(u-u_p)]$, and $\boldsymbol{s} = (s_1, \ldots, s_p)^{\mathrm{T}}$.

We need to solve the least-squares equation for three coefficients for every point u on every lane j. To save computation time, this is only done once for each lane. The values of $s(u)$ for $50 \leq u \leq 400$ are computed, with $\Delta u = 1$, and the values in between are linearly interpolated (as mentioned in subsection 2.3.1, page 48). The values outside the range (or near the endpoints) might be unreliable, but most microsatellite fragments are between 70 to 350 bp.

2.3.3 Signal enhancement

Although it is possible to use the original signal $y(t) = x\,(s[u(t)])$, we found that pre-treating the signal to highlight certain features improved the performance of the alignment. We need to highlight fluctuations which correspond to the DNA fragment ladder and suppress faster changing noise peaks, as well as slower changing ones, which might be "dye blobs" or other electrophoretic contaminants. There might also be occasional failures of the ABI GeneScan baselining algorithm, especially in the presence of artificially negative peaks due to color bleed.

We use a bandpass filter that lets frequency components from 0.025π to 0.125π (or periodicity of 4 bp to 0.8 bp) to go through, while attenuating the frequencies outside the range (see figure 2.9a). This range was found through trial and error. It is necessary to make the band a little wider than just around the periodicity of 1 bp, because some peaks are not very sharp (especially in the upper range of the electrophoresis where the peaks are more diffuse). Using a narrow frequency band would attenuate those peaks.

A Butterworth recursive filter is used [Antoniou 1993] to allow efficient computation. The transfer function is:

$$H(z) = \frac{.06745535 - .134911z^2 + .06745535z^4}{1 - 2.94453z + 3.38621z^2 - 1.84429z^3 + .412802z^4}\,. \tag{2.14}$$

Note that these coefficients are specific for the sampling rate of 10 points per bp. For other sampling rates, the coefficients can be computed easily using the

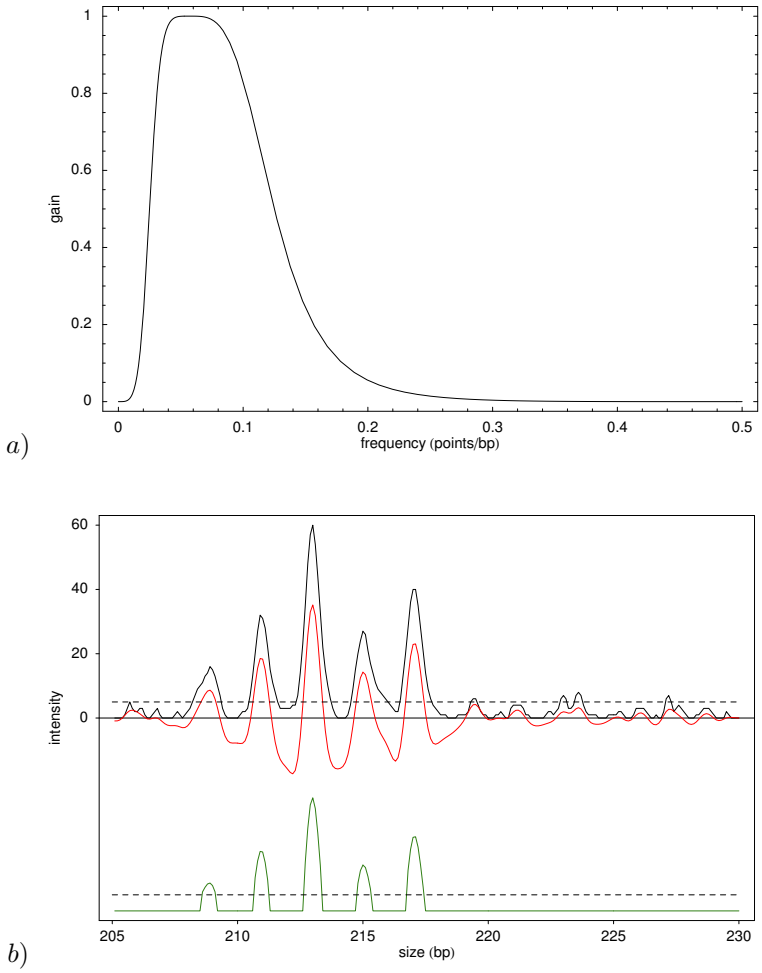


Figure 2.9: Signal enhancement before applying dynamic time warping. Panel *a* shows the frequency response of the bandpass filter used to highlight features in the trace data. Panel *b* shows an example result of applying the bandpass filtering and thresholding procedure. The black trace is the original signal, the red one is the bandpass-filtered signal, and the green one is the result of thresholding the filtered signal.

standard method for Butterworth filter design. The filtering is performed in both directions (cascaded) to obtain zero-phase response, so that the locations of the center of the peaks remain the same with those in the original signal.

After filtering, regions not containing any DNA fragment may have low background noisy. This will cause spurious alignment when the two signals are aligned according to the background fluctuations. To prevent this, these noisy fluctuations are removed by zeroing the intensity if it falls below certain cutoff (see figure 2.9b). The initial cutoff values were initially estimated from the distribution of the intensities (by fitting a mixture of two Gaussian densities for the signal and the noise). For trace data preprocessed by ABI GeneScan, we found that the cutoffs for various markers are consistently around the fluorescence unit of 5 (this is essentially the level of the background noise of the instrument). We use this fixed cutoff instead of estimating it from the data because it saves computation. The cutoff value has to be re-calibrated for other instruments.

This signal enhancement procedure is essentially a data reduction (or feature extraction) process, similar to the commonly used peak identification procedure where a combination of smoothing filters and derivatives (which is in effect, a bandpass filter) is used to choose candidate peak locations, followed by identifying the local maxima as the peaks. The difference is that we do not proceed to reduce each peak to a single 'spike' (a pair of location and intensity values). In this way, the traces are still represented as time series vectors and the peaks are still "fuzzy". This is sufficient for our purpose (and in fact, robust to cases where a peak is split into more than one maximum due to noise). Proper peak deconvolution is non-trivial [Li and Speed 2000]. We need to know, or to estimate, the peak locations, which is the problem that we are trying to solve using this trace alignment approach. Note that this enhancement step is done only to provide the appropriate input to the alignment algorithm. The final outputs are still the aligned versions of the original traces.

2.3.4 Fragment ladder summary

We need to estimate a periodic density of the DNA fragment ladder in a marker, based on a set of (enhanced) traces $y_1, \ldots, y_n$. The simplest way is to use the average across lanes at each time point:

$$\bar{y}(t) = \sum_{j=1}^{n} y_j(t) \tag{2.15}$$

The problem with this is that rare fragments (from rare alleles) might not appear in the estimate, although we still need to align the peaks. On the other hand,

using the maximum across lanes:

$$y_{\max}(t) = \max_{\forall j} y_j(t) \tag{2.16}$$

will ensure that the summarized intensity does not depend on the allele frequency. However, this might be sensitive to high-intensity contaminants.

Our solution is to use the average of a few top intensity values. Let $j_1, j_2, \ldots, j_n$ be lane indices such that:

$$y_{j_1}(t) \geq y_{j_2}(t) \geq \ldots \geq y_{j_n}(t) \tag{2.17}$$

The summary of the fragment ladder is:

$$y_q^*(t) = \sum_{i=1}^{q} y_{j_i}(t) \; . \tag{2.18}$$

When $q = 1$ this gives the maximum, while $q = n$ gives the mean. In practice, $q = 5$ seems to be optimal. An example is shown in figure 2.10.

2.3.5 DTW for trace alignment

Our problem is to find the correspondence between the two time scales t and u, by minimizing the difference between two signals $f(t)$ and $g(u)$ which are fairly similar, except for the "time warps". We set the signal $f(t)$ as the reference signal and fix the time scale t. The time scale u can be expressed as:

$$u(t) = t + \phi(t) \; . \tag{2.19}$$

We call $\phi(t)$ the 'alignment curve', to be estimated by minimizing:

$$\min_{\phi(t)} \int_{t_0}^{t_k} |g(t + \phi[t]) - f(t)|^p \; dt \tag{2.20}$$

where p specify the 'norm' of the difference. The appropriate value of p is to be determined empirically. Roughly it should reflect the distribution of the absolute difference between $g(u)$ and $f(t)$.

If a parametric form of $\phi(t)$ can be assumed, the problem above can be solved using generic optimization techniques. For microsatellite traces, it is hard to parametrically specify $\phi(t)$. We can only assume a certain degree of smoothness, in addition to monotonicity of $u(t)$. We therefore choose to use dynamic programming alignment or dynamic time warping (DTW) to solve the minimization problem. This approach is also computationally less expensive. There is no need to repeatedly resample and interpolate $g(t+\phi[t])$, which would be required if iterative optimization is used to solve equation 2.20.

The basic idea of DTW is that the difference (or similarity) between $g(u)$ and $f(t)$ are evaluated only at a finite number of points, usually an evenly

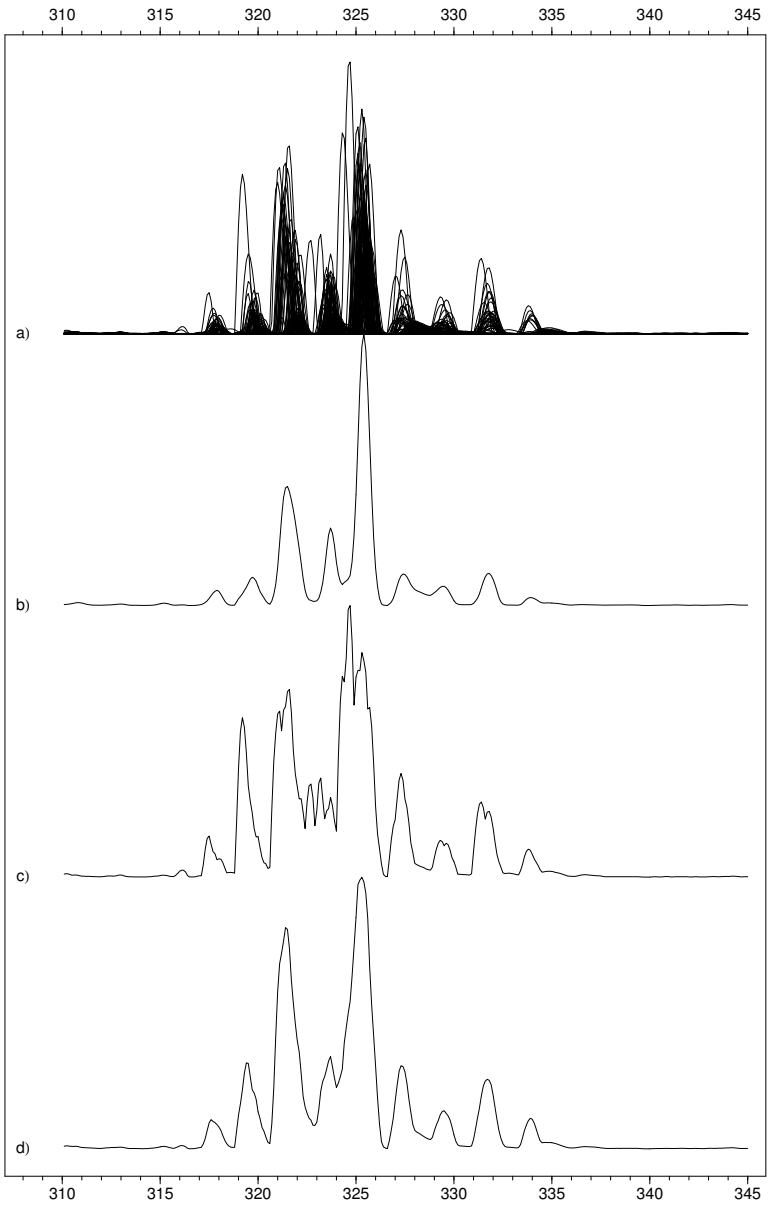


Figure 2.10: Fragment ladder summary. Panel *a* shows the overlay plot of all the enhanced traces of marker D17S944. Panel *b* shows the mean, where peaks are not well-represented (for example, at 318 bp and 334 bp). Panel *c* show the maximum, where the summary becomes noisy due to a few strong, out-of-alignment peaks. and panel *d* is the average of the five highest intensity values, which is a good compromise.

spaced grid in the plane spanned by u and t. $\phi(t)$ is thus found as a path that connects some points on the grid, having an optimal sum of the pointwise scores. Each possible path is made up of 'moves' or path segments, which are directional and have to obey the monotonicity condition. The path segments can be considered edges of a directed acyclic graph, and choosing the optimal path can be decomposed into choosing the optimal subpath leading to every grid point. This can be done using a recursion, and because the optimal subpath leading to a point does not depend on the remaining subpath to be completed, the computation can be done efficiently by caching the temporary results (the accumulated score and the choice of moves) at each grid point.

DTW is not a single specific method, but a framework with many options for specifying various parameters and constraints. These are systematically presented in Rabiner and Juang [1993, section 4.7]. Below we describe the design decisions for our problem.

Path region Usually the alignment grid covers all regularly sampled values of $u \in [u_0, u_k]$ and $t \in [t_0, t_k]$. This is unnecessary for our problem, because the distortions that we want to correct are only slightly off the main diagonal $u(t) = t$. We define the path region to be a band around the diagonal, bounded by $u(t) = t + \phi_{\max}$ and $u(t) = t - \phi_{\max}$. We also assume that the two traces are padded by zeroes outside the interval of interest. In this way, the alignment grid can be considered rectangular in the plane spanned by $[-\phi_{\max}, \phi_{\max}]$ and $[t_0, t_k]$ (see figure 2.11). The diagonal trajectory $u(t) = t$ becomes the constant $\phi(t) = 0$ in this plane. We will use this ϕ-versus-t plane instead of the usual u-versus-t because it is easier to visualize, as well as to implement, the computation. The spacing of the grid is the same as the sampling rate (10 data points per bp). A path may start at any point with $t = t_0$ and ends at any point where $t = t_k$.

Path segments The simplest path segments are made of three possible moves which for every increment of Δt, $\Delta\phi$ may be -1, 0, or $+1$ data points. This is, however, too flexible for our purpose. 'Stiffer' curve can be produced by requiring that a minimum number of $\Delta\phi = 0$ moves are taken before a change in $\phi(t)$ is allowed. Suppose three 'flat' moves are required, the total path is constructed from the subpaths shown in figure 2.12.

Recurrence equation Let $d(t, \phi)$ be the pointwise score, and $D(t, \phi)$ the accumulated score of a path. The minimization can be done recursively choosing

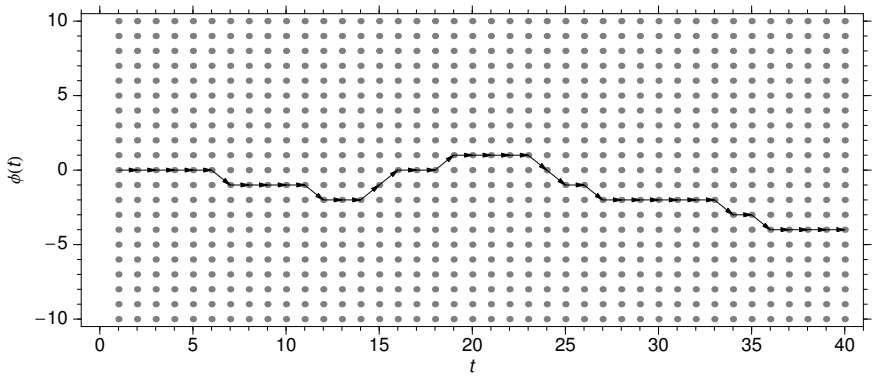


Figure 2.11: The curve $\phi(t)$ is approximated by a trajectory made up of linear path segments. Each dot is a node of the path where the score function is evaluated. The score of the path is sum of the score of the nodes.

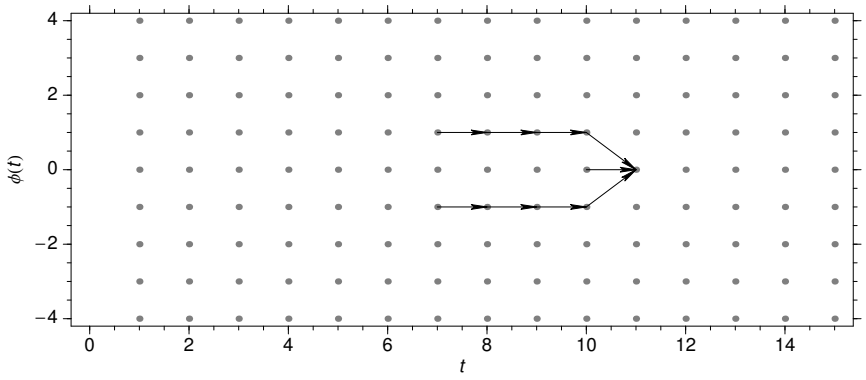


Figure 2.12: Possible path segments if $\Delta\phi = -1$ or $\Delta\phi = +1$ are constrained such that they can only happen after three consecutive occurrences of $\Delta\phi = 0$.

one of the three options:

$$D(t,\phi) = \min \begin{cases} \omega\, d(t,\phi) + D(t-\Delta t, \phi+\Delta\phi) \\ \quad d(t,\phi) + D(t-\Delta t, \phi) \\ \omega\, d(t,\phi) + D(t-\Delta t, \phi-\Delta\phi) \end{cases} \tag{2.21}$$

subject to the requirement that a change in ϕ (the top or the bottom choice) can only be taken after certain number of the middle choice. ω, called the 'slope weight' in Rabiner and Juang [1993], is a factor greater than one that can be used to further penalize changes in ϕ, and make the alignment curve stiffer.

The recursion is computed 'bottom-up'. The values of $D(t,\phi)$ is computed from $t = 0$ to $t = t_k$, processing every column of the alignment grid from left to right. For each (t,ϕ) along the path, the value of $D(t,\phi)$ is stored, together with the pointer to the previous node that minimizes equation 2.21. The end point is chosen among the nodes in the rightmost column, selecting the one with minimal $D(t,\phi)$. To complete the alignment path, backtracking is done by following the pointers recursively, all the way back to the first column.

Alignment score As indicated by equation 2.20, we choose to minimize the p-norm of the difference between $f(t)$ and the warped signal $g(t+\phi[t])$. Naturally,

$$d(t,\phi) = |y(t) - g(t+\phi)|^p \,. \tag{2.22}$$

However, in an effort to increase the smoothness of the alignment curve (which still fluctuates even after using the slope weighting ω and the consecutive $\Delta\phi = 0$ rule), we found that smoothing the score along the rows of the alignment grid results in a curve that is less variable. Thus, we use

$$d(t,\phi) = \int_{-\infty}^{+\infty} W(t-s)|y(s) - g(s+\phi)|^p \, ds \tag{2.23}$$

where $W(t-s)$ is a kernel of a smoothing filter. A cascaded, bidirectional exponential smoothing filter is used (see appendix A). The wider the kernel, the smoother (and less adaptive) the resulting alignment curve. In the extreme case, where the scores throughout the length of the rows are effectively averaged, the alignment algorithm simply finds a constant lag.

Smoothing the scores does not require much more computation. The value of $|y(t) - g(t+\phi[t])|^p$ need to be computed only once for each node and then stored in the alignment matrix. Smoothing is then performed for each row, overwriting the memory. This can be done in linear time and 'in-place' using a recursive smoothing filter. Afterward, the dynamic programming recursion (accumulating the scores and backtracking) can be done without modification, using the value of $d(t,\phi)$ stored in the matrix and then overwriting it by $D(t,\phi)$.

Preprocessing the input The two signals have to have the same baseline, which is zero. If some areas in the signal are blank (as encountered in microsatellite traces), the background noise should be removed, to avoid erratic alignment driven by the noisy peaks. These requirements are satisfied by pretreating the signals using the enhancement procedure in subsection 2.3.3. It is also important that the intensity scales of the two signals are similar. This can be done by standardizing the intensity, e.g. by making $\|f\|_2 = \|g\|_2 = 1$.

Smoothing the alignment path The alignment path found by DTW is 'jagged', due to the rectangular grid for discretizing the path, while the real sizing bias due to electrophoresis is smooth. The path can be smoothed using a lowpass filter. Smoothing also ensures that $u(t)$ is monotone increasing. Unsmoothed path segments may contain parts where t and $t + \Delta t$ are mapped to the same value of u, i.e., at the segment with $\Delta\phi = -1$. If this path is used to resample the trace, duplicate intensity values will be found adjacent to each other, interrupting the smoothness of the trace.

Specific applications The DTW algorithm is used in two different parts of the trace alignment method (see page 46): finding the systematic bias ϕ (equation 2.4) and finding the 'jitter' of individual lanes ψ_j (equation 2.5). Both are done using similar parameter settings. $p = 1$, or the absolute difference, is used for the alignment score. The slope weight ω is set to 1.5. The minimum number of horizontal moves ($\Delta\phi = 0$) need to be taken is three. The lowpass filter coefficient for smoothing the alignment score is 0.5, while the coefficient for smoothing the alignment curve is 0.85.

The range of the band, $[-\phi_{\max}, \phi_{\max}]$, is limited to ± 0.4 bp for jitter correction, because we do not want to change the identity of the alleles. For estimating ϕ, this band can be set to ± 2 bp. It is not likely that the bias will be curved by more than this range in a single marker.

In estimating ϕ, we are comparing the fragment ladder summary with an ideal periodic signal. We use the 'chopped cosine' function:

$$w(t) = \begin{cases} \cos(2\pi/T) & \text{if } \cos(2\pi/T) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.24}$$

Although a periodicity of 1 bp is assumed, this alignment still works fine on data with 2-bp periodicity (such as the one in figure 2.10). For perfect dinucleotide markers with no plusA peaks (like many markers in the commercial ABI linkage mapping sets), the stronger assumption of 2-bp periodicity can be specified easily by aligning against $\cos(\pi/T)$. However, undesirable results might be encountered when some odd alleles (or plusA peaks) are unexpectedly present. Adjacent peaks differing by 1 bp will be stretched into 2 bp in the aligned traces.

2.3.6 DTW for aligning allele frequency profiles

DTW is also used to merge genotypes (allele labels) from different sources. As mentioned in section 2.1.4, a simple shift by a constant integer does not always correct the labeling differences, because consistent binning across different runs might be impossible if the allelic frequency distribution is 'disconnected'. Nevertheless, if the allelic frequency profiles are similar between different runs, local shifts might be used to align the labels and thus, to merge the data. This is illustrated in figure 2.13.

Automatic alignment can be done using the DTW algorithm similar to that used for trace alignment. The parameterization of the algorithm is different because the nature of the signals is different. The alignment path should not be smoothed, because allele labels are discrete, and changes in the path are due to discrete 'gaps', instead of continuous stretching and shrinking. These gaps occur as the results of different binning decisions in separately analyzed data sets, and they appear in the regions lacking allelic peaks. Thus, an additional constraint is applied to the recurrence equation 2.21. A move with $\Delta\phi \neq 0$ cannot be chosen if it causes elimination of an allele label which has a non-zero count in either profile. The other settings are:

- Consecutive horizontal moves should span at least 3 bp before a gap can be introduced.
- No smoothing of the alignment score $d(t, \phi)$ needs to be done.
- For the norm, $p = 2$ (least squares) is better than $p = 1$, possibly because the signal is made of allelic frequencies instead of fluorescence intensities (which may have a few outlying high peaks better handled by $p = 1$).
- The slope weight, ω, is set to 1.5.
- The range of the alignment band is ± 4 bp.

In addition to merging data from different sources, this DTW algorithm can be used to match allele labels produced from the same trace data by different allele calling systems (manual or automatic), in order to see genuine calling discrepancies. This is the primary use of the DTW merging algorithm in this project.

2.3.7 Implementation

Trace data and SSF information are extracted from ABI sample files created by ABI GeneScan, which is also used to perform lane-tracking, color-separation and SSF identification. Reading the binary file format (also known as the 'ABIF'

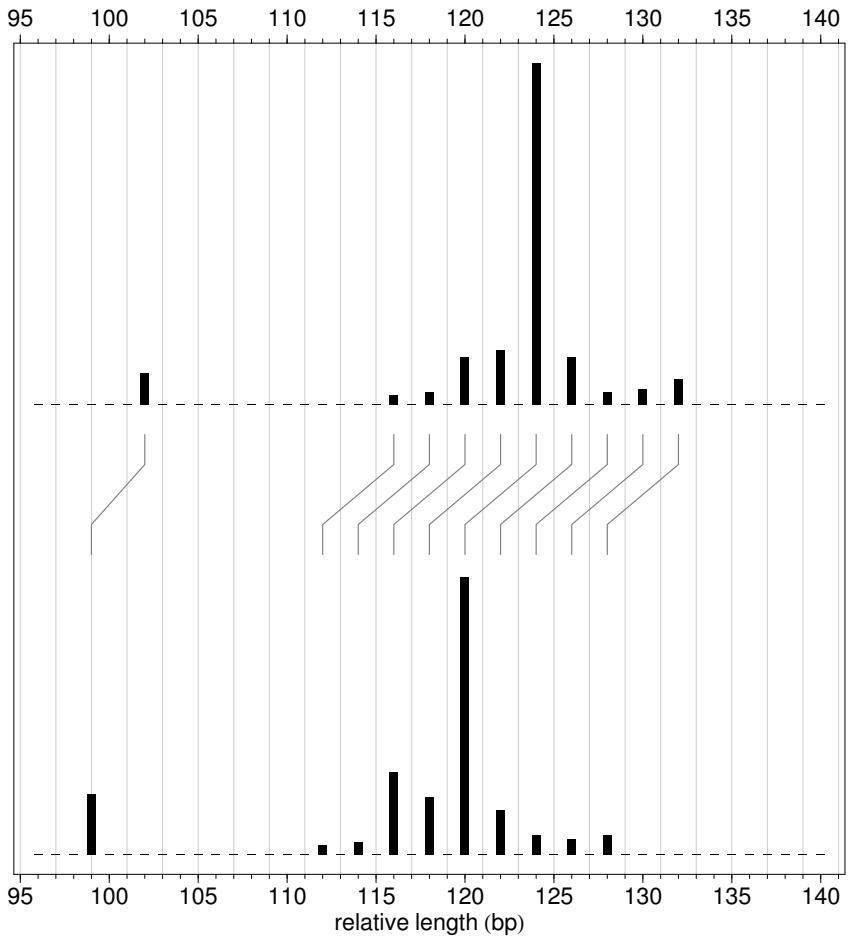


Figure 2.13: Alignment of allele labels based on the similarity between allele frequency profiles. The alleles are from the data set in figure 2.8. (Note that the trace intensity may vary between alleles and lanes, and thus does not correspond to the allelic frequency in a simple way.) Most allele labels (those between 111 bp and 131 bp) differ by 4 bp, while one allele is binned as 102 bp in the top profile and 99 bp in the bottom (3 bp difference).

format) is done using a custom perl script `exabif.pl`, based on the description of the sequencing data format in Tibbetts [1995]. The genotyping data format is similar, except for extra data tags containing peaks and SSF information.

Sizing using a second order loess curve is done by a stand-alone program written in C called `loessf.c`. It takes the SSF pairs and outputs a piecewise linear approximation of the curve, evenly spaced at 1 bp. The curves from all lanes are saved as a matrix. Programs for other sizing methods (local Southern and linear interpolation) were also written.

The main trace alignment method is implemented in C, in a program called `stral` (for 'short tandem repeat trace alignment'). It performs the steps on page 46 and creates an aligned data matrix for each marker. On a Pentium-III/450MHz running Linux, processing 96 lanes from a run, containing 17 markers (panel 1 of ABI Linkage Mapping Set v2) takes only 30 seconds (17 seconds on 700 MHz system). The memory footprint is between 1Mb to 3Mb (depending on the size of the marker intervals). It is easy to parallelize the computation because each marker is handled by a separate process. On a four-processor machine (with 700 MHz CPU speed), the markers were divided into four different sets (each processed serially). The wall-clock time to align all 17 markers was only about 5–6 seconds.

For merger allele calls, a separate program `dtwmerge.c` was written. It takes as input two genotype tables (from two data sources) and produces the mapping from allele labels in one set to the labels in the other. The input format is the same as that of ABI Genotyper tabular output.

2.4 Results and Discussion

The ultimate way to assess the performance of the trace alignment method is by testing the whole allele calling system and examining if some of the errors are caused by the trace alignment step. This will be presented in chapter 4. Here, we will only present some illustrations of the algorithm's behavior.

2.4.1 Comparison of some sizing methods

The alignment steps (equation 2.4 and 2.5, as well as allele frequency profile alignment) compensate for sizing bias and variations. Thus, the choice of the sizing method is not that critical, although extremely curved sizing bias and dispersed sizes (for a given fragment) will cause alignment difficulties and binning errors. As mentioned previously, we tested three different methods: local Southern, linear interpolation (lint1) and second-order loess (loess2). The local Southern method is the 'gold standard', which has been shown to have small variance [Ghosh *et al* 1997].

To assess the variance and bias of a sizing method, we use a plot similar to that in figure 2.6c. The allelic peaks (resulting from manual allele calling) are used. These calls have allele labels (based on manual binning), as well as the associated electropherogram locations of the peaks. The corresponding sizes are computed using the sizing method, and the values of size minus length for each allele are shown in figure 2.14. The standard deviations of size-minus-length for each bin and for all alleles pooled together are shown in table 2.1.

These results indicate that all methods have similar variance, but the bias curves are different. Note that the length in figure 2.14 is only relative, up to an unknown integer constant. Therefore, closeness to zero does not make a better method. It is desirable, however, to make the bias as close to a constant as possible. loess2 tends to give flatter curves, which means that the spacing of the fragment ladder is more uniform and closer to the expected periodicity. For some markers, such as D18S64 in figure 2.14, the bias curve is bent regardless of the sizing methods. This is possibly due to the inherent physical properties of the fragments.

The local Southern method is not significantly different from linear interpolation, although the latter is much easier to compute. Both are susceptible to the migration anomaly of the 150 bp fragment in the ABI GS500 SSF set (see figure 2.2, page 31). All marker intervals that cover 150 bp alleles exhibit the same artificial curving as that seen in D3S3681 in figure 2.14 (data not shown, but the large pooled standard deviations of those markers, shown in table 2.1, are due to this effect). A binning or alignment algorithm that attempts to correct this curve might need to be extra flexible (and sacrifice reliability in the presence of noise).

We concluded that loess2 is the method of choice. The sensitivity of the local Southern method to SSF anomalies is important to note because the method is widely used. In some cases, the phenomena of 'allelic drift' [Idury and Cardon 1997, Haberl and Tautz 1999] might be caused by the local Southern method.

2.4.2 Examples of trace alignment results

A set of traces from the marker D18S64 (from panel 24 of ABI LMS v2) is chosen to illustrate various aspects of the alignment method. This marker is somewhat atypical. The allele range is up to 340 bp, where electrophoresis under the typical conditions starts to lose precision. It has large sizing variability and a bent bias curve (see the bottom panel of figure 2.14), and can therefore illustrate more clearly the correction of warps and jitter by our proposed method.

The alignment algorithm resamples the traces three times (see page 46). Figure 2.15 shows the overlay plots from the various stages. In the first re-

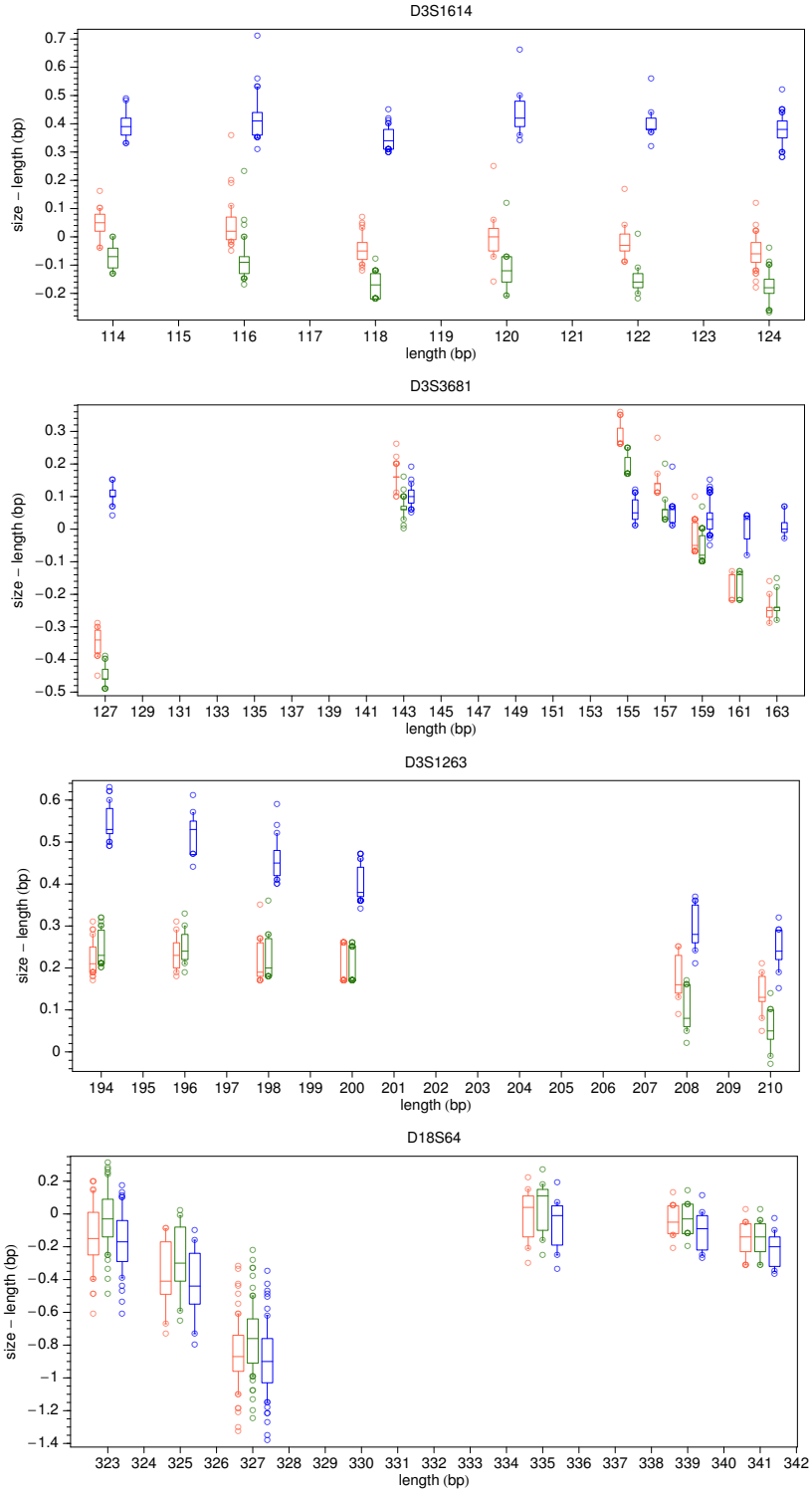


Figure 2.14: Sizing bias and variance for various sizing methods. From left to right, the red, green, and blue box plots correspond to local Southern, lint1 and loess2 sizing methods, respectively.

Table 2.1: Comparison of the sizing variation of three different sizing methods: local Southern, first order Lagrange interpolation (lint1), and second order loess (loess2). For each method, the standard deviation is computed for binned and pooled data. Numbers in boldface show large differences in the deviation between pooled and binned data when local southern or lint1 is used, but not when loess2 is used.

marker	range (bp)		standard deviations of (size − length)					
			local southern		lint1		loess2	
			binned	pooled	binned	pooled	binned	pooled
D17S928	70	115	0.09	0.10	0.09	0.11	0.09	0.10
D18S63	75	120	0.10	0.14	0.10	0.12	0.10	0.11
D4S392	79	119	0.05	0.10	0.05	0.11	0.06	0.06
D3S1271	83	113	0.05	0.05	0.05	0.07	0.05	0.05
D3S1614	95	136	0.06	0.08	0.05	0.08	0.06	0.06
D3S3681	119	175	0.04	**0.22**	0.04	**0.20**	0.04	**0.05**
D18S474	120	155	0.07	**0.28**	0.07	**0.28**	0.07	**0.07**
D18S452	125	155	0.07	**0.28**	0.07	**0.27**	0.06	**0.07**
D3S1311	132	165	0.05	**0.23**	0.05	**0.23**	0.05	**0.06**
D4S1534	140	177	0.04	**0.32**	0.04	**0.27**	0.04	**0.05**
D18S59	150	180	0.08	**0.26**	0.08	**0.22**	0.08	**0.10**
D18S53	155	190	0.09	0.14	0.09	0.13	0.09	0.11
D17S785	165	200	0.08	0.10	0.08	0.10	0.08	0.10
D3S1565	165	203	0.05	0.06	0.05	0.06	0.05	0.08
D3S1263	185	225	0.04	0.05	0.04	0.09	0.04	0.12
D17S921	190	220	0.06	0.06	0.06	0.07	0.06	0.08
D18S1161	215	250	0.07	0.09	0.07	0.09	0.07	0.11
D17S784	220	250	0.08	0.09	0.08	0.09	0.08	0.10
D4S414	230	258	0.07	0.08	0.06	0.08	0.06	0.09
D3S1285	233	261	0.07	0.07	0.07	0.08	0.07	0.07
D17S938	235	265	0.08	0.12	0.08	0.11	0.08	0.11
D4S406	241	277	0.06	0.15	0.06	0.14	0.06	0.11
D18S68	265	300	0.09	0.12	0.09	0.12	0.09	0.12
D4S1597	273	309	0.08	0.31	0.08	0.31	0.08	0.30
D4S405	279	317	0.06	0.21	0.06	0.20	0.06	0.20
D4S1575	287	315	0.09	0.09	0.09	0.09	0.09	0.09
D18S464	300	325	0.20	0.23	0.21	0.23	0.20	0.23
D18S64	305	350	0.18	0.39	0.18	0.38	0.18	0.39
D17S944	310	345	0.22	0.29	0.22	0.28	0.22	0.28

sampling, it assumes that $u(t) = t$ (no bias). The fragment ladder summary constructed from these 'temporary' traces is then aligned against the periodic 'chopped cosine' function (which has the same phase with the gridlines in figure 2.15). The alignment curve, $\phi(t)$, is used in the next round of resampling, assuming $u(t) = t + \phi(t)$. The effect correction is not easily seen in figure 2.15b because of the variability of the peak locations (we will present this more clearly below). Although still dispersed, the peaks are now centered around integer locations (the gridlines). The deviations from these integer locations are corrected by aligning each trace individually against the fragment ladder summary newly constructed from $y_j^{(2)}$. The reduction of variability in peak locations can be seen in figure 2.15c. Note that there is one peak at 324 bp that falls in between two densely populated bins. This peak might actually belong to either one of the flanking bins, but deviates too far making it impossible to decide which way to bin. The second alignment step is constrained such that $-0.4 \leq \psi_j(t) \leq 0.4$, so that 'stray' peaks will stay where they are, to be identified visually or by the downstream allele calling method.

The DTW algorithm is illustrated in figure 2.16. We can consider the alignment grid as a surface where the alignment scores specify the elevation of the surface. The peaks of the reference trace form vertical walls in the surface (the periodic green columns in figure 2.16a). The peaks of the other trace form angled walls. The places where the walls intersect are narrow passes (because the peaks cancel). The optimal alignment curve is the path going across the matrix from left to right, avoiding climbing the walls by "sneaking" through the passes.

The fragment ladder summary (figure 2.16b) shows more clearly the warps in the SSF-based scale, and the correction made by $\phi(t)$. The final alignment curves, $\phi(t) + \psi_j(t)$, are shown in figure 2.16c. The variation of each lane can be seen around the main trend. Another way to visualize the alignment curves, and their relationship with the patterns of peaks in the data, is shown in figure 2.17.

2.5 Summary

We have developed and implemented an efficient algorithm for aligning microsatellite traces. The algorithm produces a corrected trace data matrix suitable for the subsequent analysis. More rigorous assessment of the performance will be presented later, in conjunction with testing the allele calling algorithm. The algorithm illustrates the power of dynamic programming approach for solving problems related to time warping. We have also discovered that 2nd-order loess is a better sizing method than the widely used local Southern.

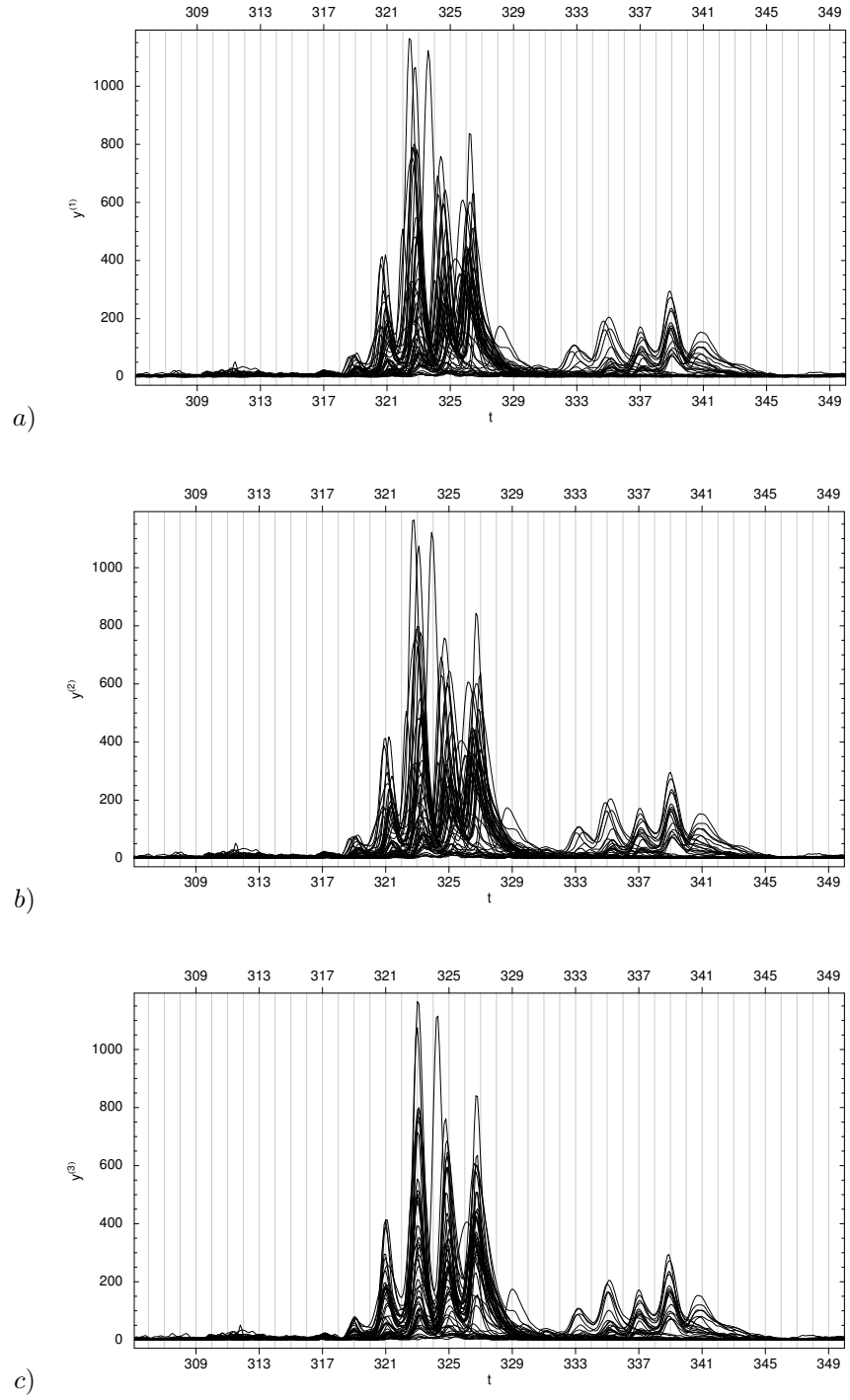


Figure 2.15: Overlay plots of several traces in the same marker throughout the three stages of alignment, $y_j^{(1)}$, $y_j^{(2)}$ and $y_j^{(3)}$, are shown in panel a, b and c, respectively.

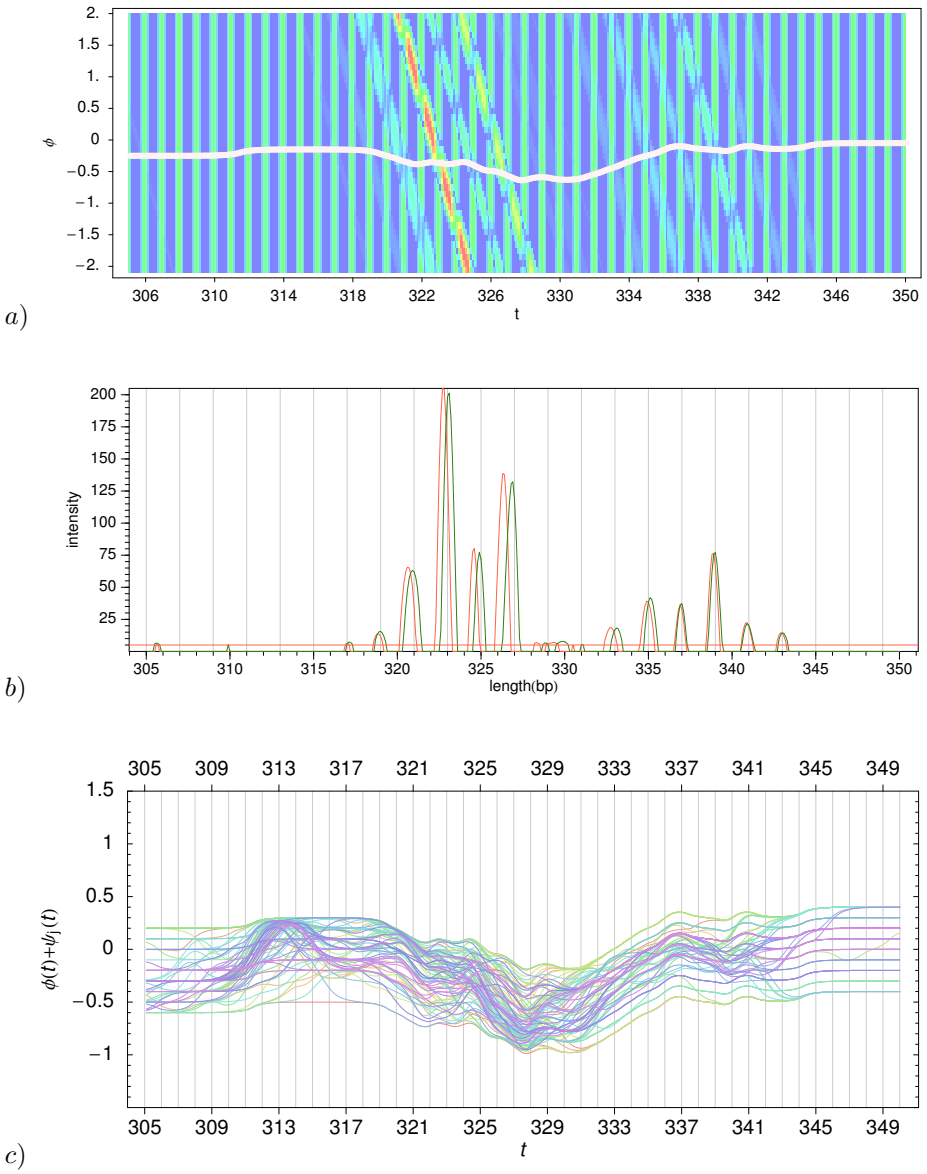
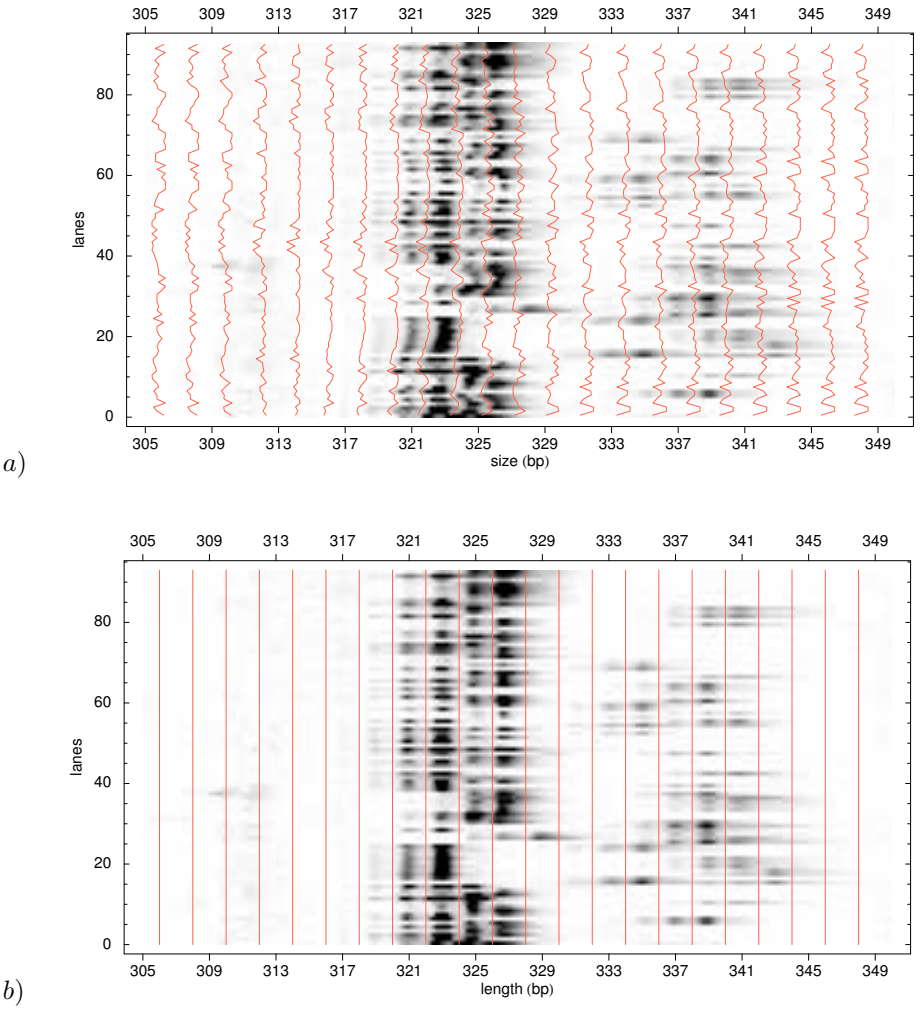


Figure 2.16: An illustration of DTW matrix and alignment curves. Panel *a* shows the dynamic programming matrix for minimizing equation 2.4 (see page 46). The rainbow color coding corresponds to the alignment score $d(t, \phi) = |y(t + \phi) - w(t)|$ (blue is low and red is high). The white curve is the smoothed optimal path $\phi(t)$. Panel *b* shows the fragment ladder summary before and after alignment (red and green, respectively). Panel *c* is the alignment curves, $\phi(t) + \psi_j(t)$, resulting from minimizing equation 2.5. The color coding corresponds to $\|\psi_j(t)\|^2$ (purple is low and yellow is high).



a)

b)

Figure 2.17: Panel a shows an image representation of our example trace data. The dark pixels correspond to high intensity. The horizontal scale is based on SSF only ($y^{(1)}$). The red gridlines are $\phi(t)+\psi_j(t)$ for $t = 306, 308, 310, \ldots$ drawn across the lanes. We can see that gridlines follow the trend of the fragments. Panel b shows the same data after the third resampling. The red gridlines correspond to $t = 306, 308, 310, \ldots$. Straightening the gridlines (by incorporating $\phi(t)$ and $\psi_j(t)$ into the horizontal scale) aligns the fragments.

Chapter 3

Allelic Pattern Estimation

3.1 Overview

The trace alignment procedure presented in the previous chapter removes electrophoretic 'time warps', which is just one of the measurement effects in microsatellite genotyping (figure 3.1). The resulting aligned traces (figure 3.1e) can be considered a multivariate data matrix, where each column (a trace position) can be compared directly across different lanes. The problem now is how to remove the remaining effects.

As reviewed in chapter 1, each allele has a characteristic pattern which is reproducible within an electrophoresis run. In homozygotes, the pattern of the whole trace is the same with the prototypical allelic pattern. In heterozygotes, the trace pattern is the linear superposition of the two allelic patterns [Perlin *et al* 1995, Stoughton *et al* 1997]. The combined effect of plusA, slippage and diffusion (the transformation from figure 3.1b to figure 3.1e) can be approximated by a linear model, where the characteristic patterns of the alleles are the basis vectors and the unseen allelic quantities are the coefficients to be found (figure 3.2). The knowledge that at most two of the coefficients can have non-zero values, and that none of them may be negative, is used to constrain the approximation. The best pair of alleles that minimizes the least-squares distance between the observed and predicted pattern is searched for using a model selection procedure.

In a previously proposed method [Perlin *et al* 1995], the allelic patterns have to be obtained from calibration data sets where the genotypes are known. A pattern library needs to be painstakingly created for each marker and changes in measurement conditions are likely to require re-calibration. Instead of building a library for each marker, we use a parametric model specifying how the patterns are generated during PCR amplification and electrophoresis. The unknown genotypes and the model parameters are simultaneously estimated, by

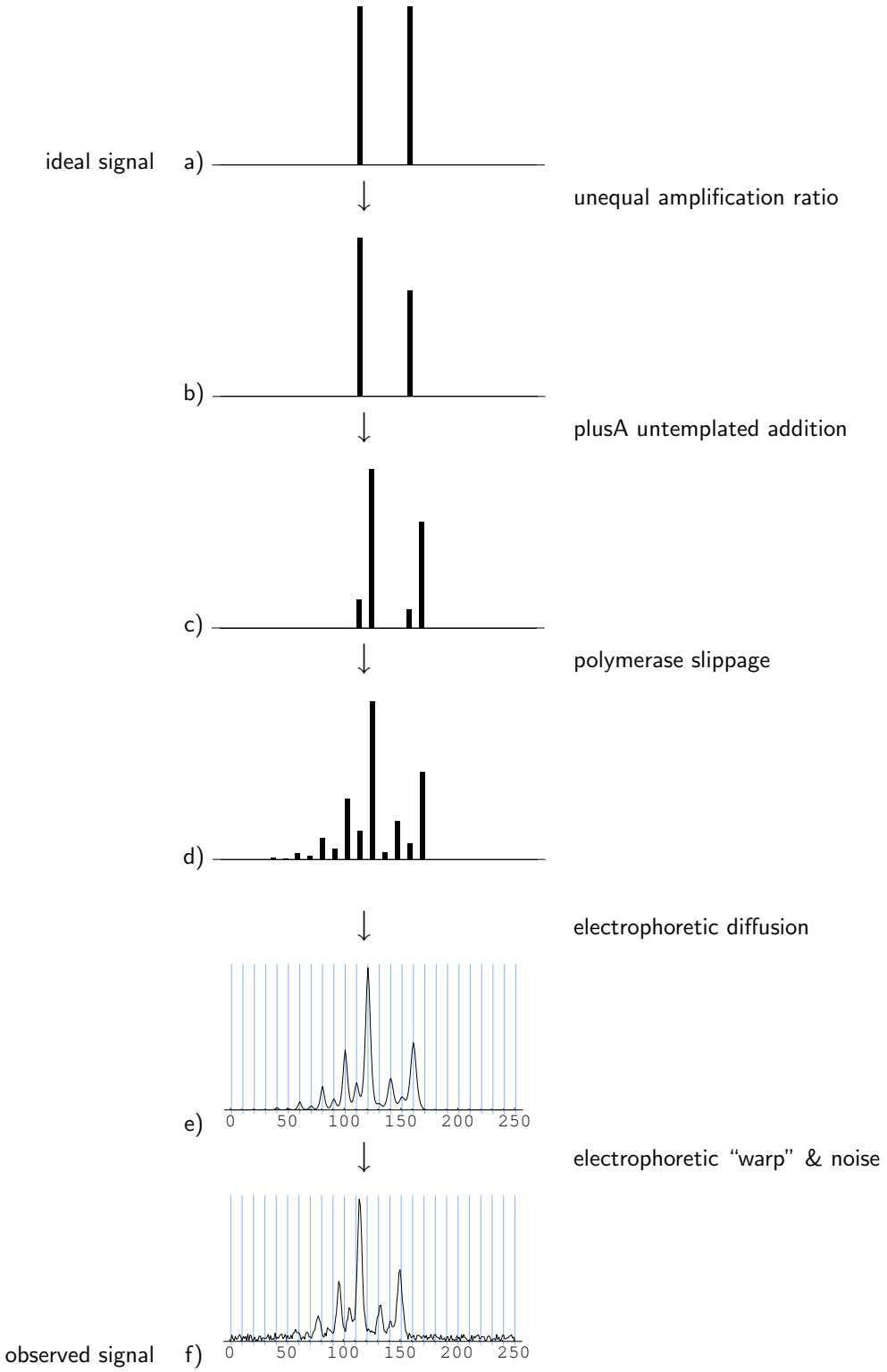


Figure 3.1: PCR and electrophoresis in microsatellite genotyping as a sequence of transformations. The trace alignment procedure (chapter 2) removes the "warp" effect.

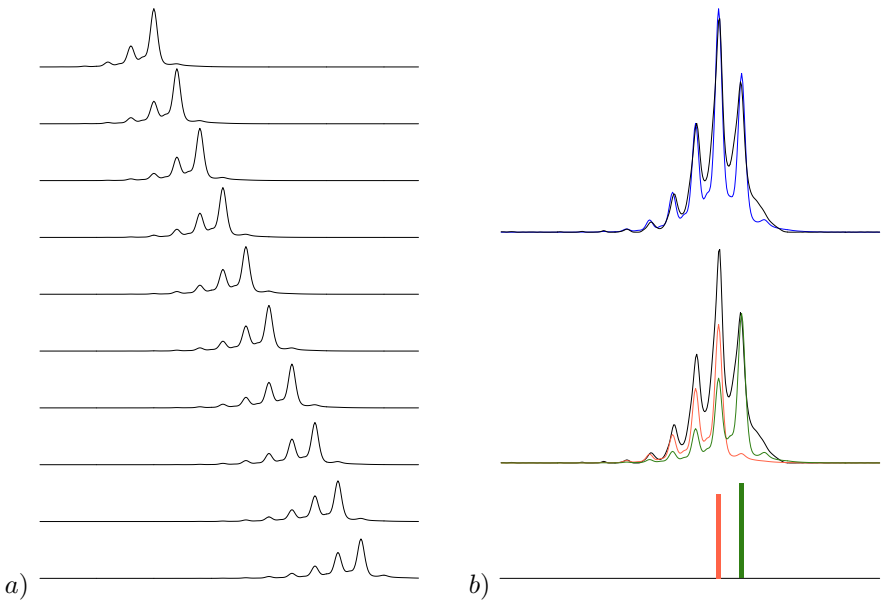


Figure 3.2: Explaining a trace data as a linear combination of two allelic patterns. Panel a is the set of patterns for possible alleles in a marker. Panel b is an example of how an observed trace (black line) can be approximated by the linear combination (blue curves) of two patterns (red and greed curves) that correspond to the true alleles. The regression coefficients (the two colored bars) are the estimates of the relative quantities of the allelic DNA fragments.

optimizing the fit of the reconstructed patterns to the data.

The main challenge in developing the model is how to make it adaptable for the wide range of possible patterns that might be encountered (for example, figure 1.5, page 11), while keeping the number of model parameters as small as possible to allow reliable estimation directly from the data. Our approach is to linearly decompose the patterns according to the underlying physical processes (as shown in figure 3.1). The linear operator at each transformation step is a simple 'convolution' [1], which can be parameterized by the extent of the local spreading of the DNA fragments. The computation of the convolutions can be done very efficiently, in linear time, using recursive filters. Finding the best-fit set of patterns can be seen as optimizing the filter coefficients.

Being able to estimate the allelic patterns and to reduce the trace data to pairs of allelic coefficients does not give the final answers. In weak signals (or traces contaminated by strong, spurious peaks such as those from non-specifically amplified fragments and dye crosstalks), two alleles are always chosen by the linear model to explain the observed pattern as much as possible, although the observed and reconstructed shape may be quite dissimilar. A procedure to detect these instances needs to be devised. Additionally, there is a problem with deciding whether to use only one or two allelic patterns to explain a homozygote or a heterozygote observation.

Due to background noise, a homozygous trace is always explained as a linear combination of two basis vectors in order to minimize the least-squares criteria, although one of the coefficients does not correspond to a true allele (and is often very small). Throwing away this 'false' allele cannot be done using a simple threshold applied uniformly to all genotypes, because the false coefficient in a homozygote might be larger than the true coefficient in some heterozygotes (see figure 3.3). Human analysts resolve this problem by relying on some regularity about the amplification ratio. In general, each pair of allele has its own specific ratio (give and take experimental noise). Typically, the larger the length difference between the two alleles, the larger the difference between their intensities, with the shorter allele having the stronger signal[2].

Knowing the typical ratio for a given pair of alleles will certainly help identifying false heterozygotes. The further the ratio of the coefficients deviates from the expected ratio, the more likely it is that the observation is a homozygote. After observing many markers, we found a simple model that relates the length difference and the allelic ratio. The model has a marker-specific parameter that can be estimated from the data.

[1] Not in the strict sense, because some effects might be time-varying.

[2] There are rare exceptions to this rule where in certain pairs, the longer allele is more strongly amplified in a reproducible way.

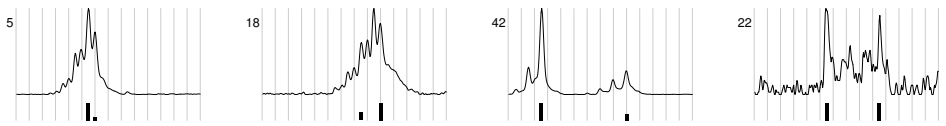


Figure 3.3: Problems with using the best two patterns in linear approximation as the genotype. Panel 5 and 18 should be called homozygotes, while panel 42 is a heterozygote and panel 22 should be rejected as a failed measurement. Note that the proportion of the true second allele in panel 42 is smaller than the false second allele in panel 18, complicating the rule for throwing away false heterozygotes.

Combining the fitness of the allelic pattern model and the deviation from the expected heterozygote ratio to call the genotypes turned out to be quite a complicated problem. The two metrics have to be weighted in a way that maximizes the calling performance (the trade-off between error and hit rate). The calling procedure, which also incorporates other quality measures to detect failures, will be described in the next chapter. Here, we simply describe the models of PCR and electrophoresis effects, and how to fit them. The main objective of modeling these effects is to extract features (distance metrics) that are marker-independent so that the subsequent allele calling procedure can deal with different markers in the same manner.

3.2 Methods

3.2.1 Formulation

We will use the index $a, b \in \{1, \ldots, k\}$ as allele labels. A genotype is a pair (a, b). By convention, $a \leq b$. There are $\frac{1}{2}k(k+1)$ possible distinct genotypes. We consider all allelic lengths $t_1, \ldots, t_k$ to be possible, although the actual allelic distribution might be more restricted (for example, in most dinucleotide repeats, all alleles are at either odd- or even-numbered lengths). In this way, the difference between allele indices is consistent with their relative difference in length (that is, $b - a = t_b - t_a$).

The input data is an aligned data matrix $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \cdots & \boldsymbol{y}_n \end{bmatrix}$. Here $\boldsymbol{y}_j \in \mathbb{R}^m$ is the trace data of lane j, in the length interval $(t_0, t_k]$. The use of the open lower bound of the interval, which exclude t_0 as a possible allele, is to simplify vector length arithmetics. Typically the dimension of the trace vector, m, is larger than k. If T is the number of data points per bp, then the dimension of

the trace vector is simply $m = kT$, with the first trace data point at the length position $t_0 + 1/T$. In practice the interval boundary is chosen to avoid alleles few basepair from the edges. The alignment algorithm in the previous chapter produces traces with $T = 10$, by default.

We will denote the coefficients of allele a and b (the spikes in figure 3.1b) by α and β, respectively. These corresponds to the quantity of DNA fragments from each allele. Each trace j can be approximated by:

$$\boldsymbol{y}_j \approx \alpha_j \boldsymbol{\mu}_{a_j} + \beta_j \boldsymbol{\mu}_{b_j}, \qquad \alpha_j \geq 0,\, \beta_j \geq 0\,, \tag{3.1}$$

where $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ are the set of all possible allelic patterns ($\boldsymbol{\mu}_a \in \mathbb{R}^m$). Each pattern is non-negative, and to ensure that α and β reflect the quantities of the allelic DNA fragments, we require that the area under the curve of any $\boldsymbol{\mu}_a$ equals one. That is, $\sum_{t=1}^{m} \mu_a^t = 1$, where μ_a^t the intensity of pattern a at the trace position t.

The patterns are assumed to be constant across different lanes in a given marker data from the same run (that is, there is no lane-specific effect). In general, the allelic patterns are time-dependent. That is, $\boldsymbol{\mu}_a$ is not a time-shifted version of $\boldsymbol{\mu}_b$. However, the patterns are rather similar and differ only in the extent of the stutter peaks. We use a parametric model to generate $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$. The highly regular shape of the allelic pattern allows a handful of parameters (seven in our model, detailed below) to specify the characteristic shapes of alleles in a marker.

Let $\boldsymbol{\theta}$ be the vector of the model's parameters, and $\boldsymbol{\mu}_a^\theta$ the pattern of allele a under a model parameters $\boldsymbol{\theta}$. Simultaneous estimation of the model parameters, the genotypes and the allelic proportions are done by the least-squares minimization:

$$\mathsf{RSS} = \min_{\boldsymbol{\theta}} \sum_{j=1}^{n} \min_{a_j, b_j, \alpha_j, \beta_j} \left\| \boldsymbol{y}_j - \alpha_j \boldsymbol{\mu}_{a_j}^\theta - \beta_j \boldsymbol{\mu}_{b_j}^\theta \right\|_2^2 \qquad \alpha_j \geq 0,\, \beta_j \geq 0\,. \tag{3.2}$$

We have two optimization problems. The inner optimization, for each lane j, is a linear least-squares minimization with a non-negativity constraint, and a restriction that at most only two of k possible basis vectors $\boldsymbol{\mu}_k^\theta$ have positive coefficients. We will call this 'genotypic least-squares approximation'. The outer optimization, over the model parameter $\boldsymbol{\theta}$, is a generic non-linear minimization problem, with constraints on the parameters $\boldsymbol{\theta}$.

3.2.2 Genotypic least-squares approximation

Here, we would like to find the best genotype, and the coefficients of of the alleles, given a trace and a set of allelic patterns determined by the current

value of $\boldsymbol{\theta}$. For clarity, we drop the trace subscript j and the superscript $\boldsymbol{\theta}$. The minimization to solve is:

$$\min_{a,b,\alpha,\beta} \left\| \boldsymbol{y} - \alpha\boldsymbol{\mu}_a - \beta\boldsymbol{\mu}_b \right\|_2^2 \qquad \alpha \geq 0,\, \beta \geq 0\,. \tag{3.3}$$

The simplest way is to enumerate all possible pairs (a, b). For each one of them, α and β can be obtained using the Cramer's rule:

$$\alpha = \frac{\begin{vmatrix} \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{y} & \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b \\ \boldsymbol{\mu}_b^{\mathrm{T}}\boldsymbol{y} & \boldsymbol{\mu}_b^{\mathrm{T}}\boldsymbol{\mu}_b \end{vmatrix}}{\begin{vmatrix} \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_a & \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b \\ \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b & \boldsymbol{\mu}_b^{\mathrm{T}}\boldsymbol{\mu}_b \end{vmatrix}} \qquad \beta = \frac{\begin{vmatrix} \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_a & \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b & \boldsymbol{\mu}_b^{\mathrm{T}}\boldsymbol{y} \end{vmatrix}}{\begin{vmatrix} \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_a & \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b \\ \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b & \boldsymbol{\mu}_b^{\mathrm{T}}\boldsymbol{\mu}_b \end{vmatrix}} \tag{3.4}$$

The enumeration of all possible genotypes is a potential computation bottleneck, especially since many dot product calculations need to be done. To save time, the values of the determinants can be cached for a given model parameter, as well as all possible dot products $\boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{\mu}_b$. Also, for each j, $\boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{y}_j$ needs to be computed only once. However, it still takes quadratic time to go over all possible pairs. To speed up the finding of the best pair, the following heuristic is used:

1. Find a^*, by iterating over all possible k indices:

   $$a^* = \arg\min_{a,\alpha} \left\| \boldsymbol{y} - \alpha\boldsymbol{\mu}_a \right\|_2^2 \tag{3.5}$$

   or equivalently,

   $$a^* = \arg\max_{a}\ \boldsymbol{\mu}_a^{\mathrm{T}}\boldsymbol{y} \tag{3.6}$$

2. Fixing a^*, find the best $b^* \neq a^*$ by iterating over all possible alleles, minimizing equation 3.3.

3. The pair (a^*, b^*) obtained by step 1 and 2 is not necessarily the best. This may happen when the distance $b-a$ is small, resulting in significant overlap of the two allelic patterns. The first allele chosen, a^*, might be neither of the true alleles, if the pattern $\boldsymbol{\mu}_{a^*}$ can explain the data better (that is, when the length of a^* is between the lengths of the two true alleles). However, the true alleles must be located in the vicinity of (a^*, b^*), based on the fact that the allelic patterns have non-zero intensity values localized around the main allele peak. To 'escape' from this suboptimal solution, whenever the distance $b - a$ is too small, say less than 5 bp, exhaustive search of possible pairs of $a, b \in [a^* - d, b^* + d]$ is performed, where d is a small value (in practice, $d = 3$ is sufficient).

This procedure speeds up the search by doing exhaustive enumeration on a few alleles, and only when the two allelic patterns in the data are not well separated. There is no prove yet that this heuristic is optimal, but it is satisfactory for the purpose of estimating the model parameters. Exhaustive enumeration of all possible pairs will be done later when the ultimate allele calling is performed. The main focus here is optimization of $\boldsymbol{\theta}$, and it is assumed that a few wrong genotypes, in a typical data set of 96 traces, do not have significant effect on the estimate of $\boldsymbol{\theta}$.

Because of the background noise, it is usually possible to find the second allele, even if the true genotype is homozygous. We have mentioned before that throwing away the small allele is not a trivial problem, but at least we can do so safely for very small second allele, to accelerate the search further. A pair is required to satisfy $0.1 \leq \dfrac{\alpha}{\alpha+\beta} \leq 0.9$. Additionally, most markers do not have genotypes with alleles differing by 1 bp, and therefore we require that $|a-b| > 1$. This constraint can be relaxed for a few markers that have many heterozygotes differing by 1 bp. These additional two rules are inserted in the loops of step 2 and step 3 in the algorithm above.

If the basis vectors $\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_k$ are close to the true allelic pattern, e.g. after successful convergence of equation 3.2 to the true parameters $\boldsymbol{\theta}$, and the noise level is very low, the best pairs (a_j, b_j) for each trace may serve as the estimate of the true genotype. Thus, this algorithm essentially performs allele calling, and will be referred to as GLSA caller (for 'genotypic least-squares approximation'). The least-squares distance between the observed and the reconstructed pattern is a good candidate for a quality indicator.

Although useful for ranking the genotypes associated with a trace, the least-squares distance cannot be used directly for rejecting or accepting the observations. The least-squares distance of a blank trace (which are mostly the background noise) is essentially the same as that of a good observation, which has the same level of background noise superimposed on the 'true' allelic pattern. We found that the following 'standardized' distance:

$$z^2_{a,b,j} = \frac{\min\limits_{\alpha,\beta\geq 0} \|\boldsymbol{y}_j - \alpha\boldsymbol{\mu}_a - \beta\boldsymbol{\mu}_b\|^2}{\|\boldsymbol{y}_j\|^2} \tag{3.7}$$

is more useful and can be used to rank lanes in the same marker, according to the signal quality (see figure 3.4). The values of z^2 is always between 0 and 1, and can be considered the ratio of the noise to the signal.

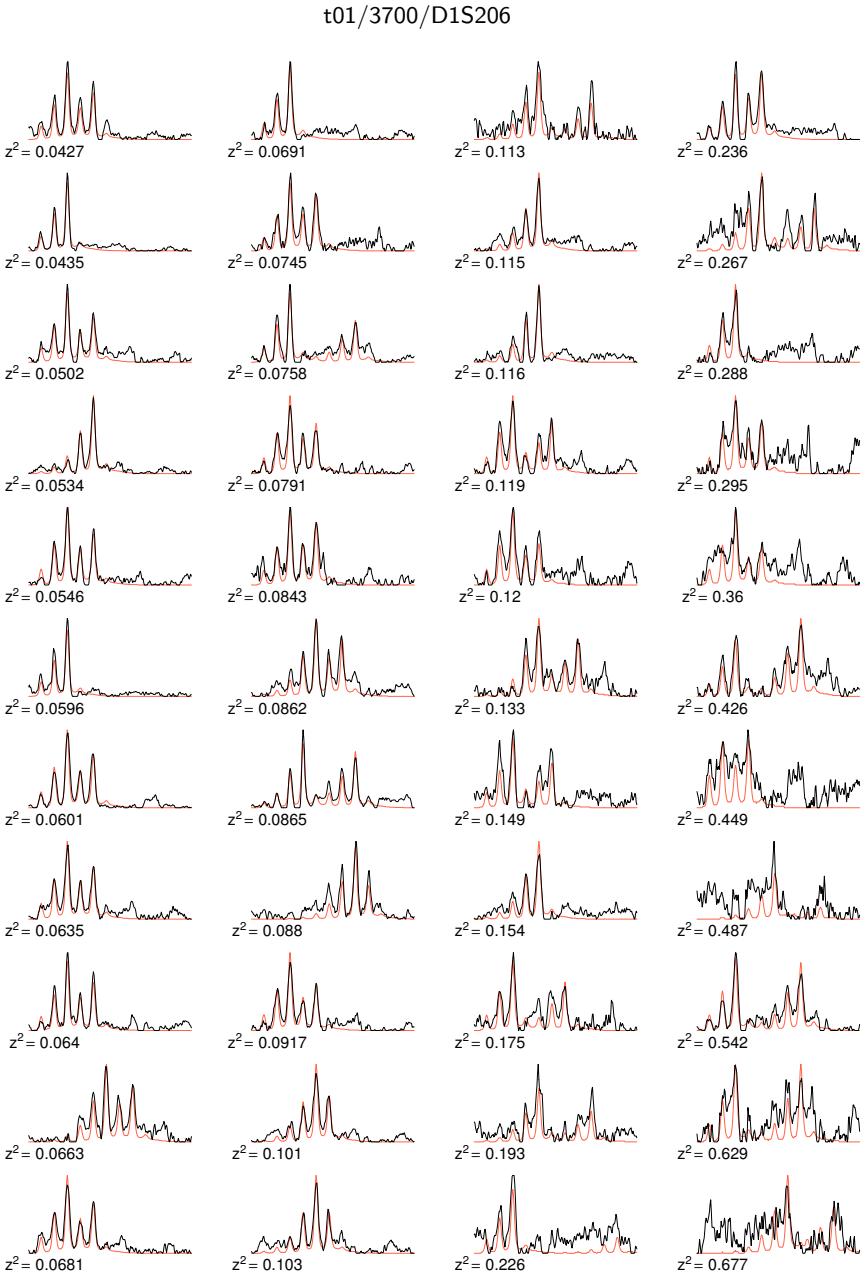


Figure 3.4: An example of a data set with varying quality between lanes. The black lines are the trace data, while the red lines are the reconstructed pattern of the best genotype (according to the z^2 value or the least-squares distance). Across lanes, the z^2 value increases as the signal becomes more noisy.

3.2.3 Allelic pattern model

Each of the plusA, slippage and diffusion effects is modeled as a linear operator. The allelic pattern $\boldsymbol{\mu}_a^\theta$ is the result of concatenating the operators:

$$\boldsymbol{\mu}_a^\theta = (\boldsymbol{DSA})_\theta \, \boldsymbol{\delta}_a \tag{3.8}$$

where $\boldsymbol{\delta}_a \in \mathbb{R}^k$ is an indicator vector, whose components are zero except the a-th which equals one. The operators $\boldsymbol{D}$, $\boldsymbol{S}$ and $\boldsymbol{A}$ correspond to diffusion, slippage and plusA effects, respectively. The transformation maps a pattern in the "basepair space" ($\mathbb{R}^k$) to the "electrophoresis pattern space" ($\mathbb{R}^m$).

The effects broaden the signal locally, and thus the column vectors of the matrices $\boldsymbol{D}$, $\boldsymbol{S}$, and $\boldsymbol{A}$ contain "impulse responses", or kernels, whose positive values are concentrated along the diagonal. When it can be assumed, it is convenient to model the linear operators as 'proper' convolutions (time invariant), because the number of parameters is smaller in addition to potentially faster computation. While the slippage effect has to be modeled by time-varying shapes, both the plusA and diffusion can be approximated by convolutions. Although this is not strictly true for electrophoretic diffusion (the peaks are more blurred for long fragments due to the more time spent inside the electrophoresis media), the change is negligible within the narrow marker interval.

There are many ways to specify the spread functions of $\boldsymbol{D}$ and $\boldsymbol{S}$ ($\boldsymbol{A}$ is quite trivial). Computational efficiency is an important consideration, because construction of the allelic patterns and dot product operations need to be performed repeatedly during the parameter optimization. The fastest way to compute a convolution is using recursive filters (see appendix A), provided that the impulse response can be described by a complex rational function in the Fourier domain.

One such filter is the exponential smoothing filter. Let y be the output of the filter and x the input. The filtering can be performed in either forward direction (causal) or backward (anticausal), by computing the recurrence:

$$\begin{array}{rcccll} y_t & = & (Fx)_t & = & a\,x_t + (1-a)y_{t-1} & \text{causal} \\ y_t & = & (Gx)_t & = & a\,x_t + (1-a)y_{t+1} & \text{anticausal} \end{array} \tag{3.9}$$

The operators F and G can be expressed in matrix forms (let $b = 1 - a$):

$$F = \begin{bmatrix} a & 0 & 0 & \cdots & 0 \\ ab & a & 0 & & 0 \\ ab^2 & ab & a & & 0 \\ \vdots & & & \ddots & \vdots \\ ab^{m-1} & ab^{m-2} & ab^{m-3} & \cdots & a \end{bmatrix} \tag{3.10}$$

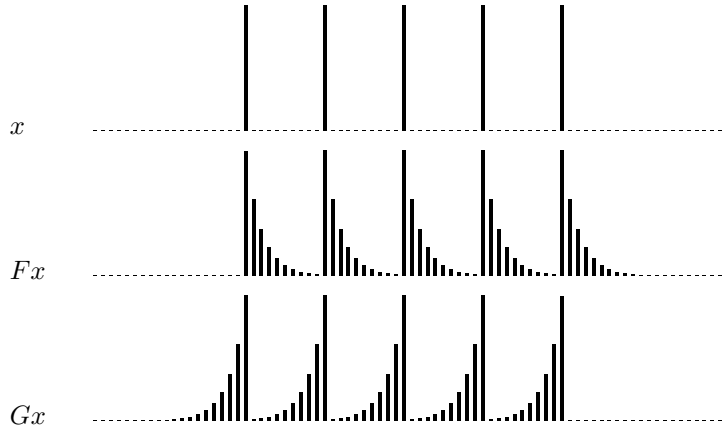


Figure 3.5: Exponential smoothing filter. Fx is the result 'forward' filtering of x, while Gx is the result of 'backward' filtering.

$$G = \begin{bmatrix} a & \cdots & ab^{m-3} & ab^{m-2} & ab^{m-1} \\ \vdots & \ddots & & & \vdots \\ 0 & & a & ab & ab^2 \\ 0 & & 0 & a & ab \\ 0 & \cdots & 0 & 0 & a \end{bmatrix} \tag{3.11}$$

We can see that the column vectors of F and G is the probability density function of a geometric distribution. Graphically, the results of filtering a pattern evenly spaced spikes are shown in figure 3.5. Instead of using the filter coefficient a which is rather non-intuitive, we parameterize the shape by the variance of a geometric distribution:

$$\sigma^2 = \frac{1-a}{a^2}, \tag{3.12}$$

where σ corresponds to the extent (or scale) of the spread. The filter coefficient is the positive root of the quadratic equation above:

$$a = \frac{-1 + \sqrt{1 + 4\sigma^2}}{2\sigma^2} \tag{3.13}$$

F and G can be cascaded to obtain impulse responses with desired spread, symmetry and sharpness. For example, the impulse responses of $(FG)^p$, is symmetric and progressively approaching Gaussian as p increases. Asymmetry is achieved by using different numbers of F's and G's, i.e. F^pG^q where $p \neq q$. The impulse response variance of the whole cascaded operators is the same with the sum of the variance of each operator (since each one is a convolution).

To model the time-dependent effect of polymerase slippage, we use an extension to the exponential smoothing filter:

$$\begin{array}{rcccll} y_t & = & (\tilde{F}x)_t & = & a_t\, x_t + (1-a_t)y_{t-1} & \text{causal} \\ y_t & = & (\tilde{G}x)_t & = & a_t\, x_t + (1-a_t)y_{t+1} & \text{anticausal} \end{array} \tag{3.14}$$

where a_t is the filter coefficient at time t. The time-varying filters can also be expressed in matrix form:

$$\tilde{F} = \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_1 b_2 & a_2 & 0 & & \\ a_1 b_2 b_3 & a_2 b_3 & a_3 & & \\ \vdots & & & \ddots & \\ a_1 b_2 \ldots b_m & \cdots & & & a_m \end{bmatrix} \tag{3.15}$$

$$\tilde{G} = \begin{bmatrix} a_1 & \cdots & & a_{m-1}b_{m-2}\ldots b_1 & a_m b_{m-1}\ldots b_1 \\ & \ddots & & & \vdots \\ & & a_{m-2} & a_{m-1}b_{m-2} & a_m b_{m-1} b_{m-2} \\ & & 0 & a_{m-1} & a_m b_{m-1} \\ 0 & \cdots & 0 & 0 & a_m \end{bmatrix} \tag{3.16}$$

where $b_t = 1 - a_t$. The parameterization of a_t depends on the problem. For our particular case, it will be detailed below.

Details of the operators

PlusA effect This effect is trivial to model because it simply splits a fragment into two peaks in certain proportions. Transformation by $\boldsymbol{A}$ can be applied by computing:

$$y_t = \theta_1 x_{t-1} + (1-\theta_1)x_t$$

which is equivalent to multiplication by a bidiagonal matrix with $(1-\theta_1)$ along the diagonal and θ_1 at the superdiagonal. Both the input and output signals are vectors in $\mathbb{R}^k$, where each data point coincides with a fragment length in basepairs.

Polymerase slippage Because the extent of the spread increases with the length of the allele, this operator is time-dependent (equation 3.14). Both insertion and deletion slippages occur, so the spreading needs to be applied in both directions, with the left tail significantly more extensive than the right one (which is often missing). Additionally, the left tail is more rounded than a

geometric impulse response, therefore more than one anticausal filter needs to be applied.

We assume that the underlying physical process, insertion and deletion slippage, are binomial events and the net result is a convolution of them. Although it is possible to model these events precisely (by considering the number of repeats and PCR cycles), it is sufficient to approximate the resulting patterns using a few time-dependent smoothing filters. The longer the allele, the higher the chance of slippage because of the larger number of repeats. Therefore, the impulse response variance should change linearly with the length, because the variance of a convolution is the sum of the variance of each kernel. Therefore, $d\sigma^2/dt$ is a constant.

The input and output vectors are in the k-dimensional space, where each data point corresponds to a fragment length. However, each slippage event changes the length of the fragment by a multiple of the repeat-unit length (for example, two for dinucleotide repeats). Thus, the previous output value in equation 3.14, y_{t-1} or y_{t+1}, is from the trace position differing by the repeat-unit length (not the immediately adjacent position in the DNA fragment ladder). Note that we still need to compute the output for every basepair position, to accommodate all possible alleles. The repeat-unit length needs to be specified manually for each marker. It is possible to automatically determine the repeat-unit length by trying various values (2, 3 or 4) in minimizing equation 3.2, but this might not worth the computational effort. The repeat-unit length of each marker can be stored in a database along with its interval and dye channel information.

After experimenting with several filter combinations, the following was found to be satisfactory:

$$\boldsymbol{S} = (\tilde{\boldsymbol{G}}^2 \tilde{\boldsymbol{F}})_{\theta_2,\theta_3,\theta_4} \tag{3.17}$$

where the parameters θ_2, θ_3, and θ_4 are used to specify, respectively, the scale (σ) at the midpoint of the marker interval, how fast the variance changes with length (a constant rate for $d\sigma^2/dt$), and how large the scale of the right tail is as a proportion of the scale of left tail. Examples of the impulse responses of $\boldsymbol{S}$ are shown in figure 3.6.

Electrophoretic diffusion

Unlike the previous two transforms, this operator maps a vector in $\mathbb{R}^k$ to $\mathbb{R}^m$. The pattern of the previous transform, $\boldsymbol{SA}$, is first 'up-sampled' by inserting $(T-1)$ zeroes in between each basepair position. Afterwards, the diffusion transform $\boldsymbol{D}$ is done in $\mathbb{R}^m$.

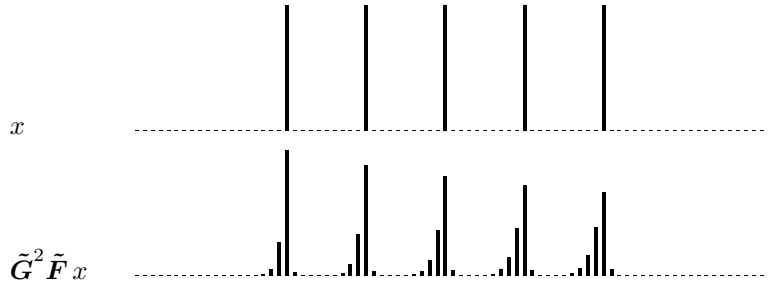


Figure 3.6: An example of the impulse response produced by the operator $\boldsymbol{S}$. The parameter values are $\theta_2 = 2$, $\theta_3 = 0.2$ and $\theta_4 = 0.15$.

The spread corresponding to each DNA fragment is roughly symmetrical and rounded, thus multiple smoothing is needed. The following is used to model this diffusion:

$$(\boldsymbol{FG})^6_{\theta_5} \,. \tag{3.18}$$

Another aspect of diffusion needs to be modeled is the "trailing blur" effect found in some gels (for example, marker D6S257* on figure 1.5, page 11). This effect is gel-specific and does not seem to be related to the marker itself. It appears that some portion of the DNA molecules lags behind the main clusters of stutter peaks. The main allelic patterns themselves are still fairly well defined, while the trailing patterns are blurred. This can be modeled by superimposing the original pattern with the blurred and lagged version of the signal using:

$$(1-\theta_7)\boldsymbol{I} + \theta_7\boldsymbol{F}^3_{\theta_6} \,. \tag{3.19}$$

The parameter θ_6 specifies the blurring of the lagged pattern, and θ_7 corresponds to how much of the molecules lag behind. The overall effect of electrophoresis is:

$$\boldsymbol{D} = \left\{(1-\theta_7)\boldsymbol{I} + \theta_7\boldsymbol{F}^3_{\theta_6}\right\}(\boldsymbol{FG})^6_{\theta_5} \tag{3.20}$$

3.2.4 Model parameter optimization

Minimization of the 7-parameter model (equation 3.2) is done using the Nelder-Mead downhill-simplex method [Nelder and Mead 1965]. The reason for this choice is it is simple to implement and reasonably fast, even in the absent of knowledge on the derivatives of the objective function. In fact, the RSS value might contain discontinuities due to the combinatorial nature of the genotypes. The objective function value might change smoothly when the parameter values are perturbed, if the genotypes remain the same. However, when the perturbation is such that one or more genotypes are switched, the RSS value might suddenly jump.

Such an objective function landscape naturally contains many local minima. To find a global solution, restarts are performed following the method in Press *et al* [1992, pp408–412]. After convergence is reached, all vertices of the simplex, except the best one, are randomized by adding a fraction of an identity matrix. The restarts need to be performed multiple times. We also found it useful to restart even before convergence, after every certain number of iterations (say 50 cycles).

Because different markers vary in interval lengths (data dimensionality), the RSS score in equation 3.2 needs to be adjusted so that the same tolerance for convergence give roughly the same degree of fitness in different markers. We use the 'standardized RSS':

$$\mathrm{SRSS} = \frac{\mathrm{RSS}}{\sum_{j=1}^{n} \|\boldsymbol{y}_j\|^2}, \tag{3.21}$$

which takes values between 0 and 1. The same convergence criteria can then be applied for all markers. For example, the difference between the SRSS values of the best and worst vertices (ΔSRSS) should be less than 0.005. The SRSS score is also a rough indicator of the quality of a particular marker data set. For example, a noisy data set will converge to a large SRSS value.

After experimenting with many markers, it was found necessary to put constraints on the values of the parameters. Obviously, parameters specifying proportions should be between zero and one, and those specifying the scales of impulse responses cannot be negative. Putting upper bounds on these parameters also helps reduce the risk of being trapped in local minima. The bounds are chosen to rule out "impossible" patterns, such as those with very wide spread that are never encountered in practice. The "bounding box" of the parameter space are shown in table 3.1, along with the initial values which are chosen from the mean of the optimal values in typical markers. On average, fewer iteration cycles are needed when these initial values are used.

The lower bounds of the parameters are set to zero to prevent meaningless negative values, except for θ_1, where it is set to 0.2, because $\theta_1 = 0$ and $\theta_1 = 1$ specify identical patterns (except that the allele label is shifted by one basepair). The commonly used PIG tailing primer modification increases the rate of plusA effect, thus making the pattern more consistent [Smith *et al* 1995b, Brownstein *et al* 1996]. Therefore, in many data sets the true θ_1 is very close to one. If $\theta_1 = 0$ is allowed, this might be found as the answer for the markers with $\theta_1 \approx 1$, resulting in inconsistent allele labeling. This constraint disallows $0 \leq \theta_1 < 0.2$. However, markers having such values have not been encountered so far.

Table 3.1: Constraints and initial values of the model parameters.

parameter	min	max	init	comment
θ_1	0.2	1	0.9	portion of plusA peaks
θ_2	0	3	1.8	slippage scale at the interval midpoint
θ_3	0	0.2	0.1	rate change of the slippage scale
θ_4	0	0.2	0.025	portion of the right tail
θ_5	0	0.75	0.35	scale of diffusion
θ_6	0	1	0.125	extent of the trailing blur
θ_7	0	1	0.2	portion of the trailing blur
ρ	-0.01	0.05	N/A	slope of the heterozygote curve (section 3.2.5)

3.2.5 Unequal amplification ratio

It has been widely observed that the shorter the allele, the more competitive it is when co-amplified with a longer allele [Pálsson *et al* 1999], although no quantitative model has been proposed for this phenomena. When the values of the proportion of the shorter allele, $\alpha_j/(\alpha_j+\beta_j)$, are plotted against the length differences, $b_j - a_j$, we can see that they follow a straight line (figure 3.7). The degree to which the two alleles compete, as a function of length difference, depends on the marker.

Although the trend follows a straight line, it is safer to fit the saturation curve:

$$H(a_j, b_j, \rho) = 0.5 + 0.45 \tanh\left\{\rho\left(b_j - a_j\right)\right\}, \tag{3.22}$$

to ensure that the curve never crosses $\alpha/(\alpha+\beta) = 1$. The parameter ρ determines the steepness of the curve. The offset 0.5 corresponds to the equal amplification efficiency of the paternal and maternal alleles when $a_j = b_j$. The factor 0.45 is to limit the saturation curve so that it does not reach $\alpha/(\alpha+\beta) = 1$. There might be markers where the heterozygote patterns of certain genotypes are inherently indistinguishable from the homozygote patterns of the shorter alleles [Ewen *et al* 2000], due to extreme difference in efficiency and the presence of background noise. Lowering the asymptote of the curve prevents fitting to false heterozygotes.

The parameter ρ is found by minimizing the least-squares fit:

$$\min_{\rho} \sum_{j=1}^{n} \left[H(b_j, a_j, \rho) - \frac{\alpha_j}{\alpha_j + \beta_j} \right]^2 . \tag{3.23}$$

This is a non-linear optimization problem, which is solved by searching the value of ρ in the interval $[-0.01, 0.05]$. Values outside this range do not seem to be

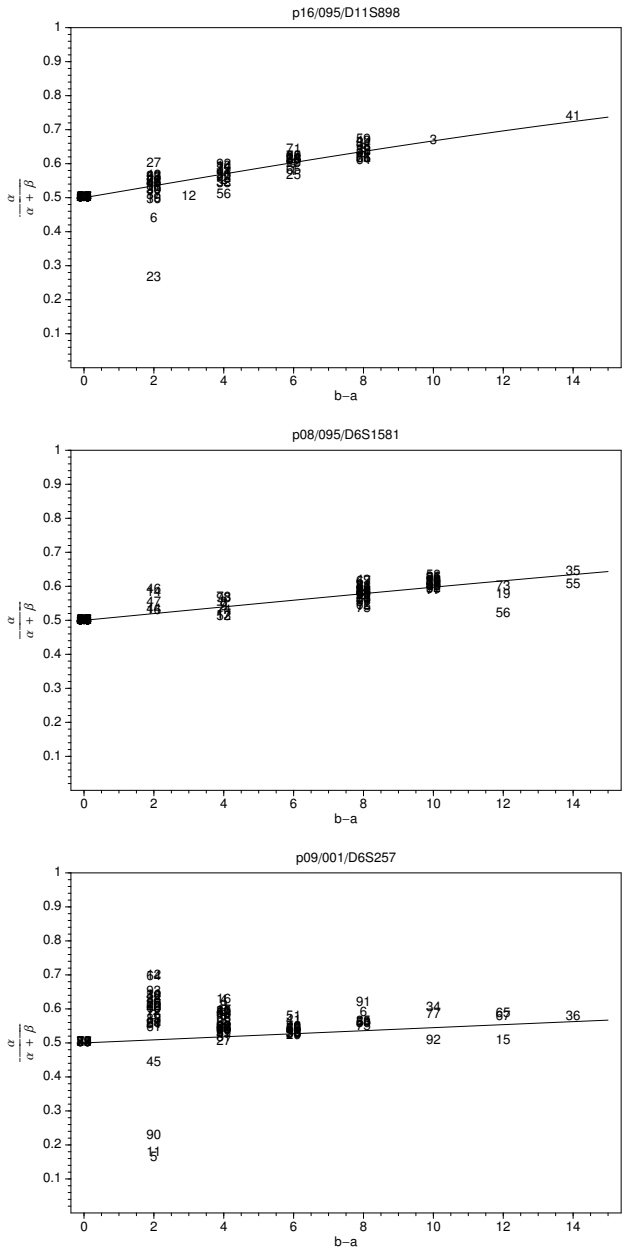


Figure 3.7: Examples of the relationship between the proportion of the shorter allele, $\alpha/(\alpha+\beta)$, and length difference, $b-a$, for the 'best' genotype that minimizes equation 3.2. The trend roughly follows a straight line, with intercept at $(0, 0.5)$, which is expected for homozygotes. The slope is marker specific. The outliers (points away from the main trend) are homozygotes with significantly high values of the second coefficient (usually falsely explaining the stutter artefacts as an allele).

reasonable (the ratio is too different, or violate the "shorter-is-stronger" rule). To make the fitting more resistant to false heterozygotes, robust regression is used (using iterative reweighting and Hampel's influence function described in Campbell [1980]). Examples of the best-fit curves are shown in figure 3.7.

3.3 Results and discussion

The ultimate assessment of the model's performance is the calling accuracy of the whole system, to be presented in chapter 4. Here, we only present some examples to illustrate the ability of the model to adapt to a wide range of markers.

3.3.1 Model fitting

Figure 3.8 illustrates the change of the SRSS throughout the Nelder-Mead iterations. The corresponding trace data and reconstructed models are shown in figure 3.9. On a 'good' marker (figure 3.8a and 3.9a), with a typical model parameter values, convergence is fast, and a low SRSS value can be achieved. Marker data with unusual allelic shapes (figure 3.8b and 3.9b) may take a while to fit, requiring several restarts. This data also illustrates the 'trailing blur' effect that necessitates the use of θ_6 and θ_7 parameters. On noisy data, such as those with very weak signal and therefore high background noise (figure 3.8c and 3.9c), the SRSS values cannot drop too far.

Manual examination of plots such as those in figure 3.9 indicates that the proposed model for allelic pattern generation works well for most markers. Although the patterns generated by the model does not always fit tightly (compare figure 3.9b and 3.9a), the best genotype chosen is still correct most of the time. The optimality of the fit does have some effects on the calling performance. More lenient convergence criteria (or no restart) produce less optimal allelic patterns, resulting in some erroneous choices of the best genotypes.

Currently, the iteration is allowed to proceed for 50 cycles (or $\Delta\text{SRSS} < 0.005$) before a restart is enforced, unless $\Delta\text{SRSS} < 0.001$, in which case no more optimization needs to be done. Four restarts might be done before the whole optimization terminates. On a Pentium II/450 MHz running Linux, it typically takes 15 to 30 seconds to process a marker data set with 96 lanes, depending on the size of the marker interval (the dimension of data vector, m, is typically between 200 to 500). We have not yet explored other global optimization strategies that might give similar or better fitness with fewer iteration cycles.

The distribution of optimal SRSS values of 354 different data set is shown in figure 3.10. A typical 'good' marker will have an SRSS value less than 0.05.

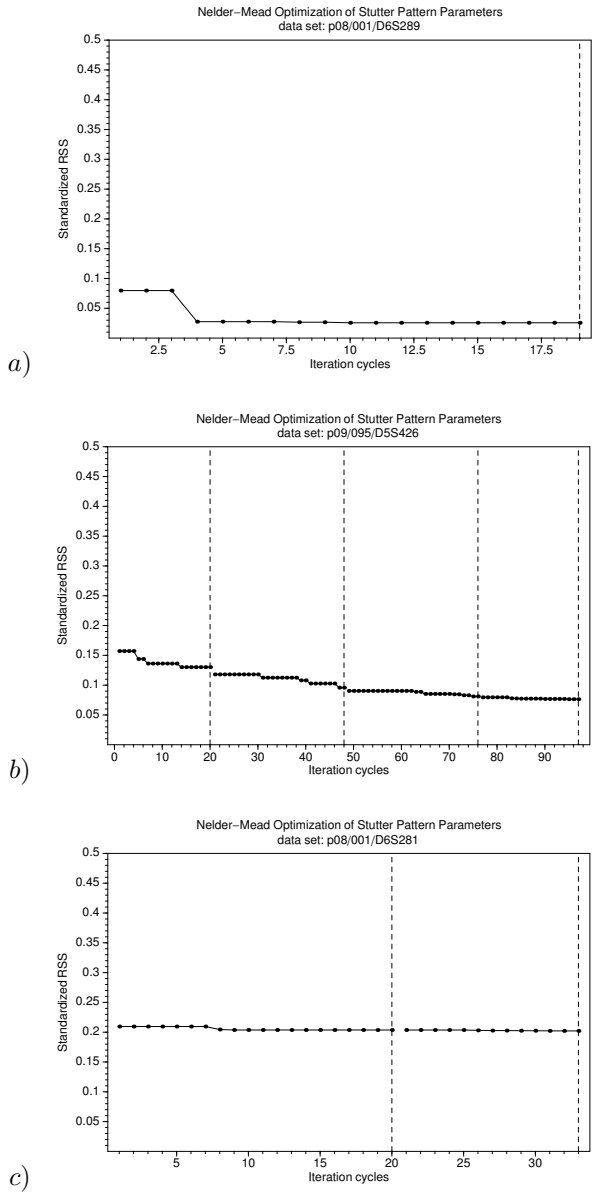
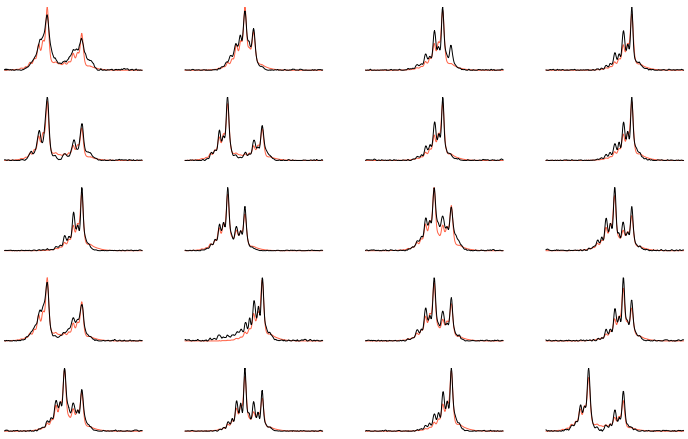
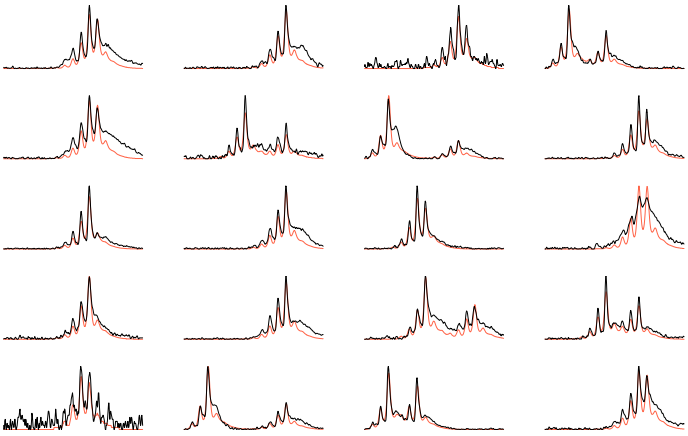


Figure 3.8: Examples of the drop of SRSS values (of the best vertex) during the Nelder-Mead optimization. The dashed vertical lines are restarts.

a) p08/001/D6S289, SRSS $= 0.026$, $\boldsymbol{\theta} = (0.74, 1.78, 0.06, 0.01, 0.40, 0.10, 0.26)$, $\rho = 0.023$



b) p09/095/D5S426, SRSS $= 0.076$, $\boldsymbol{\theta} = (0.97, 1.94, 0.11, 0.17, 0.43, 0.09, 0.55)$, $\rho = 0.024$



c) p08/001/D6S281, SRSS $= 0.202$, $\boldsymbol{\theta} = (0.86, 1.90, 0.05, 0.09, 0.38, 0.10, 0.26)$, $\rho = 0.050$

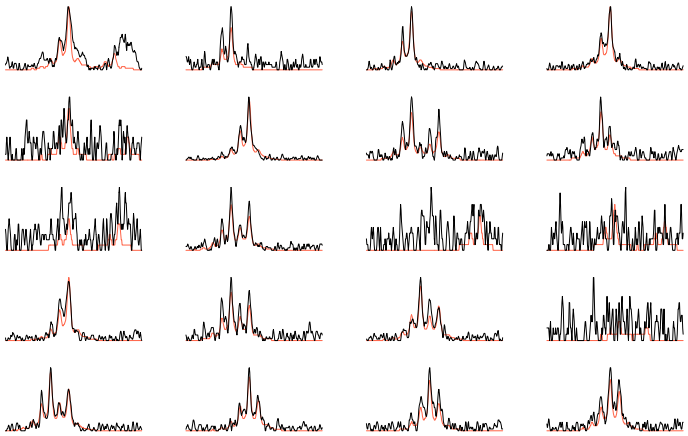


Figure 3.9: Illustrations of the fit between the data (black lines) and the model (red lines). The data sets are the same with those in figure 3.8.

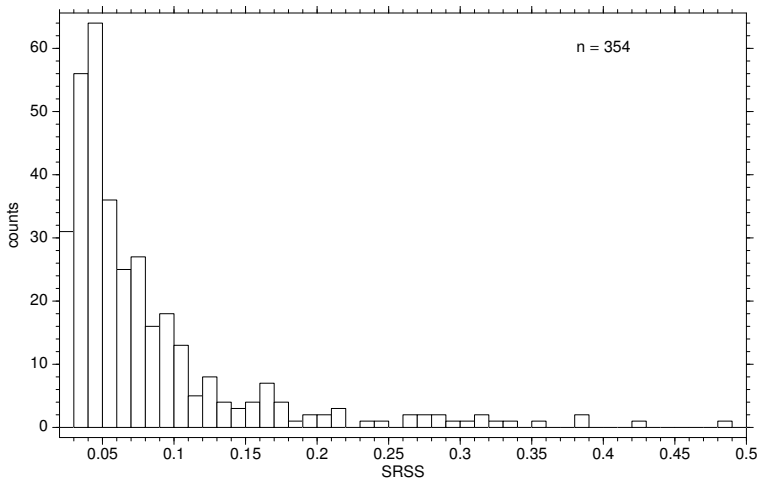


Figure 3.10: The distribution of SRSS found by the optimization procedure, from 354 markers. Note that 6 data points with SRSS > 0.5 are not shown.

Those with SRSS value > 0.2 are outliers which indicate bad data, such as those shown in figure 3.9c, or more rarely, very atypical allelic patterns. The distributions of marker parameters $\theta_1, \ldots, \theta_7$ and ρ are shown in figure 3.11. The main peaks of the distributions are contained well within the parameter constraints (table 3.1). The initial values also coincide with the optimal values of typical markers.

The knowledge about the parameter and SRSS values distribution can be used as diagnostics for detecting 'pathological' markers or runs. We have not yet incorporated the deviation from the expected values into the quality values (chapter 4). For the time being, the histograms can be used as guidelines for markers requiring special attention or extended optimization cycles.

Overall, we found that the proposed model can adaptively fit a wide range of marker data very well. The versatility might be attributable to our approach of decomposing the trace patterns into transformations that closely resemble the underlying physical processes. Although the model does not attempt to meticulously mimic the biochemical process, such as considering the probability of polymerase slippage per repeat per replication and its convolution throughout PCR cycles, the resulting patterns are similar enough to observed stutter patterns and sufficient for our purpose. Unlike the model proposed by Miller and Yuan [1997], which only deals with polymerase slippage, our model incorporates other artefacts such as plusA and electrophoretic diffusion. For the purpose of allele calling, these effects are equally important.

There are many advantages in being able to represent the measurement process compactly using a few parameters. Unlike in the method proposed by

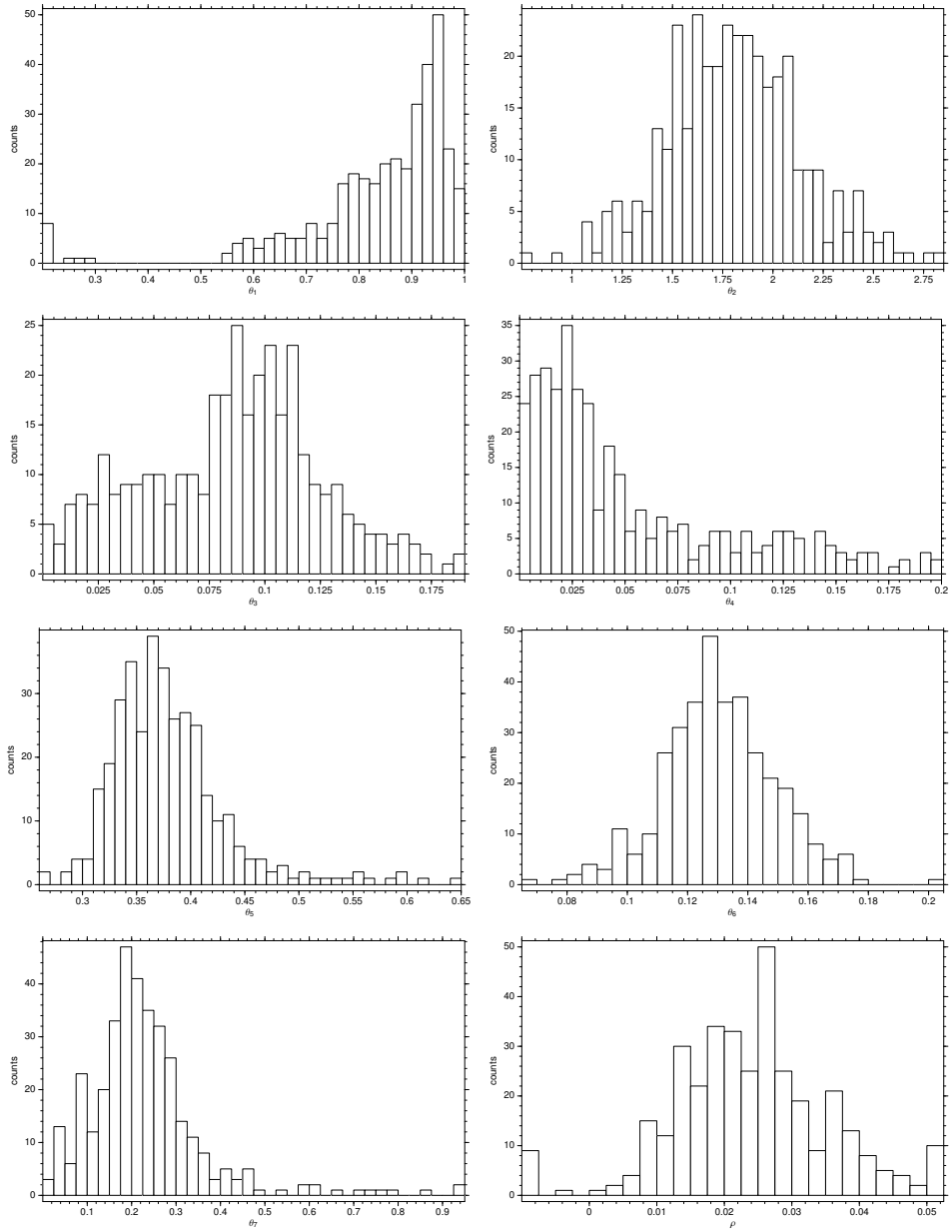


Figure 3.11: Distribution of the fitted parameter values on 354 data sets. See table 3.1 (page 85) for the descriptions of the parameters.

Perlin *et al* [1995], there is no need to use training sets to build the allelic pattern library. The reproducibility and the regularity of the allelic patterns, as well as the large number of lanes in a typical run, make it possible to estimate the patterns directly from the observations. Our model flexibly adapts to changes in experimental conditions, as long as the resulting variations in allelic shapes are well within the range of possibilities allowed by the model. Poor fits of the model (the outliers in figure 3.10) are usually attributable to many noisy traces in the data, due to weak amplification affecting the whole data set, or inherently problematic markers that are hard to score manually.

Our method differs from the data-adaptive approach by Stoughton *et al* [1997], which constructs the allelic shapes directly from the traces. Although their method can extract patterns from homozygotes and well-separated heterozygotes, it becomes problematic when some alleles are present only in overlap with other alleles. In contrast, our method can even predict the patterns of unobserved alleles.

Further improvement of the model is not impossible. The arrangement of the recursive filter cascades was chosen through a trial-and-error process, by visually inspecting the fit and intuitively changing the arrangements. It should be possible to parameterize the arrangement, and automatically search, using techniques of combinatorial optimization, for a better design that can fit a wide range of markers with, say, better SRSS value on average and less computation time. Note that this optimization needs to be done only once using a large number of traces with known genotypes. The resulting 'optimal pattern generator' would still have parameters to be estimated for each marker data set, as outlined above.

3.3.2 GLSA caller

It is useful to see the performance of this 'naive' allele caller, before adding more sophisticated discrimination rules. A detailed description of the validation methods will be presented in chapter 4. Roughly, we need to see the trade-off between calling error (which can be minimized by rejecting unreliable observations) and the number of correct calls (which is also reduced when more data are thrown away). In GLSA allele caller, the z^2 score (equation 3.7) is used as the quality indicator. The trade-off between the error rate (percent miscalls in the accepted observations) and the 'hit' rate (the number of correct calls) is shown in figure 3.12.

GLSA can correctly call up to 85% of the data (where manual calls are available for 96% of them). This is achieved at a cutoff level which contains around 9% error. The error rate goes down slowly as the quality requirement is

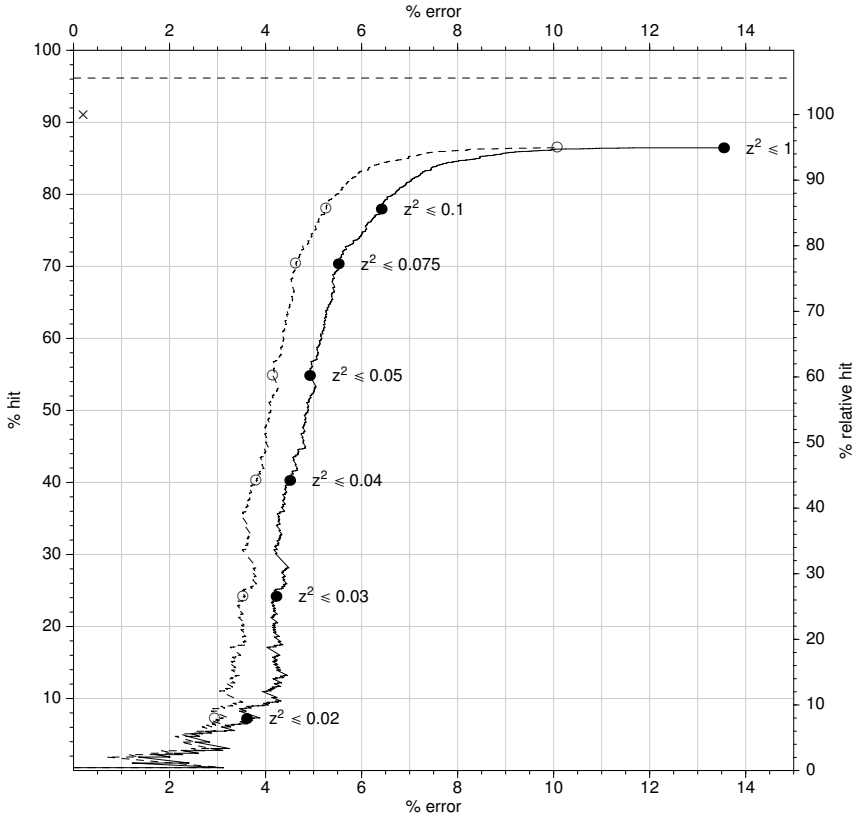


Figure 3.12: The performance of GLSA allele caller. The automated calls are compared with manual calls on 7792 traces (on 87 markers, 3 runs of different individuals for each marker). '% error' is the percent errors per number called (those in the data subset selected by the z^2 cutoff). '% hit' is the percent correct call (of the total number of traces). The trade-off between the two as the function of the z^2 cutoff is shown by the solid curve (for all types of discrepancies including those where GLSA calls but human discard) and dashed curve (for 'definite' miscalls, where both GLSA and human make the calls but the genotypes differ). 96% of the data has manually called genotypes, some of them based on repeated measurements, whose traces are not used by GLSA (the remaining 4% of the data cannot be called, even after repeated measurements). The '×' mark indicates the performance of human analysts on the same set of traces (without considering the repeated measurements), at 92% hit and (estimated) 0.2% error. '% relative hit' corresponds to GLSA hit rate relative to the human hit rate.

increased (by lowering the maximum acceptable z^2 value). The error rate never reaches below 1% required by most downstream genetic analysis. To bring the error rate below 4%, 90% of the data has to be discarded. The z^2 can throw away bad data which GLSA cannot call reliably (those with $z^2 > 0.1$), but it is not a good indicator of the error rate for $z^2 \leq 0.1$, as indicated by the almost constant error rate (between 4% and 6%) for most of the data. Nevertheless, GLSA can correctly call up 80% of the data at that error range, and thus the z^2 score should be useful as a 'feature variable' in a more sophisticated allele caller.

Somewhat similar performance was obtained for the 'TrueAllele' system [Perlin *et al* 1995, Perlin 2000], which was reported in Pálsson *et al* [1999]. The error rate overall was $719/7596 = 9.4\%$. Unfortunately, they did not report the performance using a varying cutoff for data rejection, thus we cannot see how it performs on the portion of the data with better quality. The algorithm is based on least-squares fitting of two best alleles, as in GLSA, except that the patterns are calibrated using training sets. Only after additional *ad hoc* rules (as implemented in the software 'DecodeGT') are applied, the performance can be improved to 1.12% miscalls (although it is not clear how much of the data needed to be discarded).

3.3.3 Unequal amplification model

The main purpose of modeling unequal amplification ratio is to obtain the expected proportions of the two coefficients, for any pair of alleles. The relationship between the allelic proportion in heterozygotes and their difference in length is surprisingly simple (figure 3.7 shows the plots for typical markers). Within one marker, the allelic proportion is largely determined by the length difference.

It is not clear yet how this can be explained by the underlying kinetics. Such a physical model should consider the change of the relative quantities of the fragments throughout PCR cycles, with each fragment having a 'competitiveness index' that might be length dependent. It is plausible to assume that the probability of a strand synthesis being completed in cycle decreases with the length of the strand. Better understanding of the unequal amplification phenomena will be very useful for analyzing quantitative assays using microsatellites: loss-of-heterozygosity detection [Sidransky 1994] and genotype pooling [Kirov *et al* 2000].

For our 'qualitative genotyping' purpose, the model above suffices. In addi-

tion to the z^2 score, now we also have the h^2 score:

$$h^2_{a,b,j} = \left[\frac{\alpha}{\alpha+\beta} - H(a_j, b_j, \rho)\right]^2 \tag{3.24}$$

where $H(a, b, \rho)$ is the heterozygote curve (equation 3.22). The next chapter describes how the two metrics produced by our model, the least-squares distances to the expected pattern and heterozygote proportion, are integrated into a discrimination rule for choosing the most likely genotype.

Chapter 4

Allele Calling and Quality Scores

4.1 Overview

In the previous chapter, we have seen that although the allelic pattern model can be fitted well, the performance of the GLSA caller is poor. In the TrueAllele/ Decode-GT system, Pálsson *et al* [1999] use a number of *ad hoc* rules to flag 'bad' traces, so that the automated calls that are likely to be erroneous can be separated from the 'good' ones. The new allele caller from Applied Biosystem, the ABI GeneMapper[1], uses many different metrics called 'process-based quality values' to flag traces that need to discarded, checked manually or accepted. However, neither uses a quality indicator that has predictive ability, i.e. corresponds to the error rate within the subset of the data selected by a certain threshold, such as the PHRED quality score that has been found to be very useful for quality control in DNA sequencing [Ewing and Green 1998, Richterich 1998]. Devising such quality score for microsatellite genotyping is understandably more difficult, due to the heterogeneity of marker-specific behaviors. The trace alignment and allelic pattern model presented in the previous two chapters are estimates of the marker-specific characteristics. It is therefore reasonable to treat the deviations from the model as ingredients for a quality score, which is hopefully marker-independent and predictive.

We use the same quality score for choosing the most likely genotype and throwing away bad observations. This quality score is computed for each possible genotype (a, b), not just for the 'best' genotype found by the GLSA algorithm. Thus, instead of only flagging bad observations, the quality score ranks alternative genotypes according to their closeness to the true genotype. The best scoring genotype is chosen as the call. If the data is ambiguous, it is possible that the top few genotypes are very similar. Presenting the alternatives will help manual editing as well as downstream analysis methods that can take

[1] `www.appliedbiosystems.com`

ambiguous genotypes (instead of simply declaring the genotype as 'unknown'). The quality score should also be comparable across traces (not just for ranking genotype within one trace), therefore it can be used to rank the traces in a marker data set based on the quality of the best genotype. This will help manual editing by putting higher priority on problematic traces. Ultimately, if the score is comparable across markers and runs, it will be a useful tool for data handling and quality control in large genotyping projects.

There are many ways to devise a score based on features derived from the measurements. We have explored a simple one, which uses a linear combination of feature variables that are assumed to be deviations from the expected 'good' value. This is illustrated by the following example. Both the z^2 and h^2 scores are indicators of 'error' from the model, thus it is natural to combine the z^2 and h^2 scores into the weighted sum:

$$Q_{a,b,j} = w_z \, z^2_{a,b,j} + w_h \, h^2_{a,b,j} \tag{4.1}$$

where w_z and w_h are the weights that correspond to the relative contribution of each distance. The 'Q-score' is then used to select the best genotype. An example is shown in figure 4.1 and 4.2. In the plane spanned by z^2 and h^2, the weights define a direction that determines the ranking of the genotypes. How to find the optimal weights is one of the methods proposed in this chapter.

In addition to the heterozygote ratio, there are other 'features' that can be used to improve discrimination. The signal intensity has so far been ignored because we focused on the trace pattern. The information about the intensity is partially contained in the z^2 score because weak signals tend to be more noisy. Explicitly incorporating the heights of the main allelic peaks into the quality was found to improve the ranking of alternative genotypes. Other features that were found to be useful are the sharpness of the allelic peaks and the amount of shift introduced to align the trace (chapter 2).

All these features are combined in a manner similar to combining z^2 and h^2 in the example above. The sum of squared errors[2] used for the Q-score is reminiscent of a χ^2 statistic. However, the arbitrary weighting (which is to be empirically derived) does not allow strict interpretation of the Q-scores as χ^2 statistics with a certain degree of freedom. Furthermore, each individual feature may not be χ^2_1-distributed. Nevertheless, the resulting Q-score is roughly gamma distributed, although the scale and shape might be specific to each marker data set. To obtain a more marker-independent score, the distribution of the second-best Q-scores in a marker data set is used as the "noise" distribution.

[2]The z^2 itself is a standardized regression residual, which is the sum of squared deviations at all data points of a trace. If the background noise at each data point is normally distributed (i.i.d.), then the z^2 is a (scaled) χ^2 statistic.

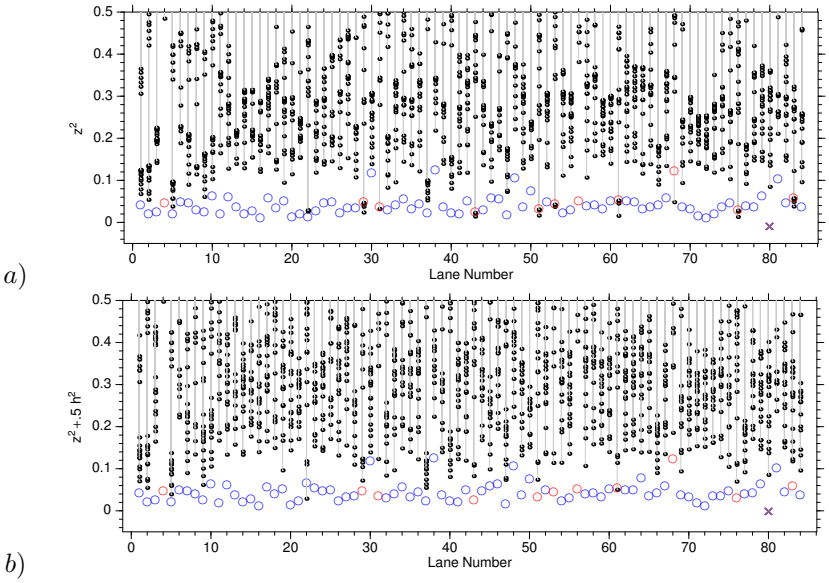


Figure 4.1: Panel *a*) shows the z^2 score of all possible genotypes (not all visible) for each trace (identified by the lane number). Those corresponding to the true genotypes are plotted in colored circle, blue for heterozygotes and red for homozygotes. The black dots are the values for incorrect genotypes. '×' indicates a trace that should be discarded. Note that for homozygotes, there are usually one or more (incorrect) genotypes with smaller z^2 score. Panel *b*) shows similar plot, but $z^2 + 0.5\,h^2$ is used to rank the genotypes. Most of the homozygotes genotypes now have the best score. There is one heterozygote genotype (lane 22) called incorrectly under this score. The marker data set is `t05/103/D4S406`.

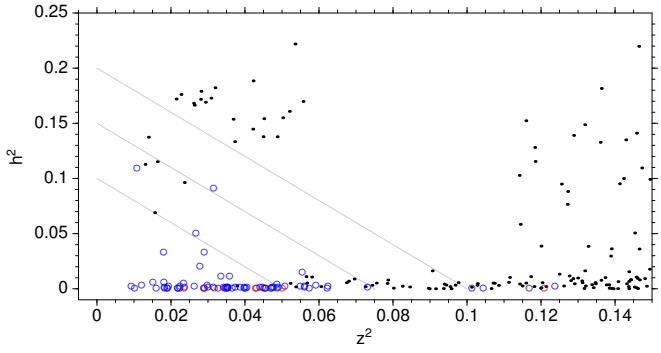


Figure 4.2: Another way to visualize the use of the combined score. All points on the same gray line have the same value of $z^2 + 0.5\,h^2$ (the values for the three lines are 0.05, 0.075 and 0.1). The combined score corresponds to locations along the direction perpendicular to the gray lines. Discrimination between true and false genotypes is much improved compared to using z^2 alone (which is equivalent to using vertical lines). The data set is the same as that in figure 4.1.

A new quality indicator called the *L-score* is derived from the density fitted to the second-best Q-scores in each marker data set.

We do not assume any probability model as the basis for this approach. Here, we present our discrimination rule as an *ad hoc* method, justified empirically. In the future, it might be possible to treat this problem more properly using a formal probabilistic approach. The rest of this chapter assesses the performance of the proposed allele calling rules and the ability of the quality score to predict the error rate. Based on the test results, a suggestion is made on how to use the algorithm in a genotyping system.

4.2 Methods

4.2.1 Formulation

Selecting the best genotypes for the trace $\boldsymbol{y}_j$ is done by ranking all possible genotypes (a, b), according to a quality score $Q_{a,b,j}$. A certain threshold of this score will also used to reject bad quality traces (at least within the same marker and run). A perfect trace will have $Q_{a,b,j} = 0$ for the true genotype (a, b).

The Q-score is derived by combining several features. Each feature is a distance measure, which can be computed for all possible pair (a, b) given a trace $\boldsymbol{y}_j$. Let $d_i(a, b, j)$ denotes this distance for feature i. The z^2 and h^2 mentioned above are examples of feature variables. The quality score is the weighted sum:

$$Q_{a,b,j} = \sum_{i=1}^{p} w_i \; d_i(a, b, j) \qquad\qquad w_i \geq 0 \; . \tag{4.2}$$

The features are assumed to be marker-independent (after the lengthy procedures to remove marker-specific effects described in the previous two chapters). This allows the same weight vector $\boldsymbol{w} = (w_1, \ldots, w_p)$ to be used in different markers. This means that $\boldsymbol{w}$ can be calibrated using a set of markers that might be different from those to be analyzed automatically. Marker-specific Q-scores could be more powerful, but the calibration requires more effort and newly encountered markers can not be handled. Here, we are exploring how far a marker-independent rule can be pushed.

The weights specify the contribution of each feature to the score. As illustrated in the introduction, the weight vector $\boldsymbol{w}$ defines parallel hyperplanes in the feature space. The direction of these hyperplanes can change the ranking of alternative genotypes in a trace, and one particular hyperplane (a certain value of Q) can be chosen to accept or reject a trace if the best genotype of that trace still lies outside the hyperplane, i.e. not on the same side as the origin.

The direction of the hyperplane is optimized based on the desired trade-off between the error and hit rate. In practice, the trade-off at a specific error rate is more important than averaged performance. A caller that performs badly on the whole data but can guarantee less than, say, 1% error for a portion of the data might be more useful than that with a better overall performance but yields fewer correct calls for the same error rate. With the former, manual recheck can be skipped altogether for some of the data, while with the latter, re-examination needs to be done for a larger portion of the data (although less editing is required overall). We therefore decided to find $\boldsymbol{w}$ that maximizes the number of correct calls for the acceptable error rate, which is 1% in our case. This is the objective function of a generic, constrained optimization problem.

Although the weights optimized using a large number of markers may be applied to different markers, the actual magnitude of the Q-score might not be comparable. To obtain a score with the same meaning across markers, we used another score derived from the Q-score. This score, called the L-score, is the cumulative distribution function of the second-best Q-scores.

4.2.2 Feature variables

In addition to the z^2 and h^2 mentioned before, we use three other features. They are chosen based on several types of common miscalls which are not reflected in the z^2 and h^2 score alone. Signal strength needs to be explicitly included. These are the heights of the main allelic peaks. Our approach does not perform peak detection step but directly compare the whole high-resolution trace pattern. There are often high-intensity "blurs" and "blobs", which obviously do not correspond to DNA fragment peaks, but the least-squares fitness criteria will attempt to explain them. Such contaminants usually have dull and rounded signal at the location of the main allelic peaks (and they might not even be local maxima). A bandpass filter can highlight true DNA fragment peaks (see figure 2.9, page 52). The intensity of the filtered signal, relative to the original one, may indicate if a region of the trace is 'peaked' enough for a DNA fragment. Lastly, the trace alignment algorithm in chapter 2 shifts the location of peaks so that they can be easily compared across lanes. If a peak strays too far from the expected location, say by nearly ± 0.5 bp, it might be shifted in the wrong direction.

The features need to be transformed into non-negative values that increase with decreasing quality. We have not yet explored thoroughly the effect of their distributions on the discrimination ability. We found that squaring the deviations make them roughly χ^2_1 or gamma distributed with longer tail (with shape parameters less than 0.5). Without implying a χ^2 model, we re-formulate

the Q-score as:

$$Q = \sum_{i=1}^{p} w_i \frac{[f_i(a,b,j)]^2}{\sigma_i^2} \qquad w_i \geq 0. \tag{4.3}$$

where f_i is an 'appropriately transformed' feature variable. The variables are standardized by σ_i^2 to make the scales of different features roughly in the same order of magnitude. The actual magnitude of σ_i^2 is not so important for the score because the optimized weights can absorb arbitrary scales. However, standardized feature variables make the values of the weights more intuitive (they can be seen roughly as proportions of 'contribution' to the decision making) and experimenting with initial values and constraints for the optimization procedure is easier.

The distributions of the features for the true genotypes in some markers are shown in figure 4.3 and their definitions and transformations are detailed below.

1. **Allelic pattern fit**

   This is based on the z^2 score (equation 3.7).

   $$d_1(a,b,j) = \frac{z_{a,b,j}^2}{\sigma_1^2}, \tag{4.4}$$

   where σ_1^2 is chosen to be 0.05. This is based roughly on the order of magnitude of the sample variance of some marker data sets, rounded to a 'convenient' number. See figure 4.3a to compare with the distributions in various markers. σ_i^2 for other features are chosen in similar way.

2. **Deviation from the expected ratio**

   This is based on the h^2 score (equation 3.24).

   $$d_2(a,b,j) = \frac{h_{a,b,j}^2}{\sigma_2^2} \tag{4.5}$$

   where $\sigma_2^2 = 0.005$.

3. **Peak heights**

   Let t_a and t_b the size of allele a and b in basepairs. $y_j(t_a)$ is the trace intensity at position t_a of trace $\boldsymbol{y}_j$. In ABI trace files, the peak heights range from 0 to 8192 (13-bit digitized fluorescence intensity), with the background noise level fluctuating around ± 10 units. A peak height of 50 is often used as the minimum for a meaningful peak (for example, in

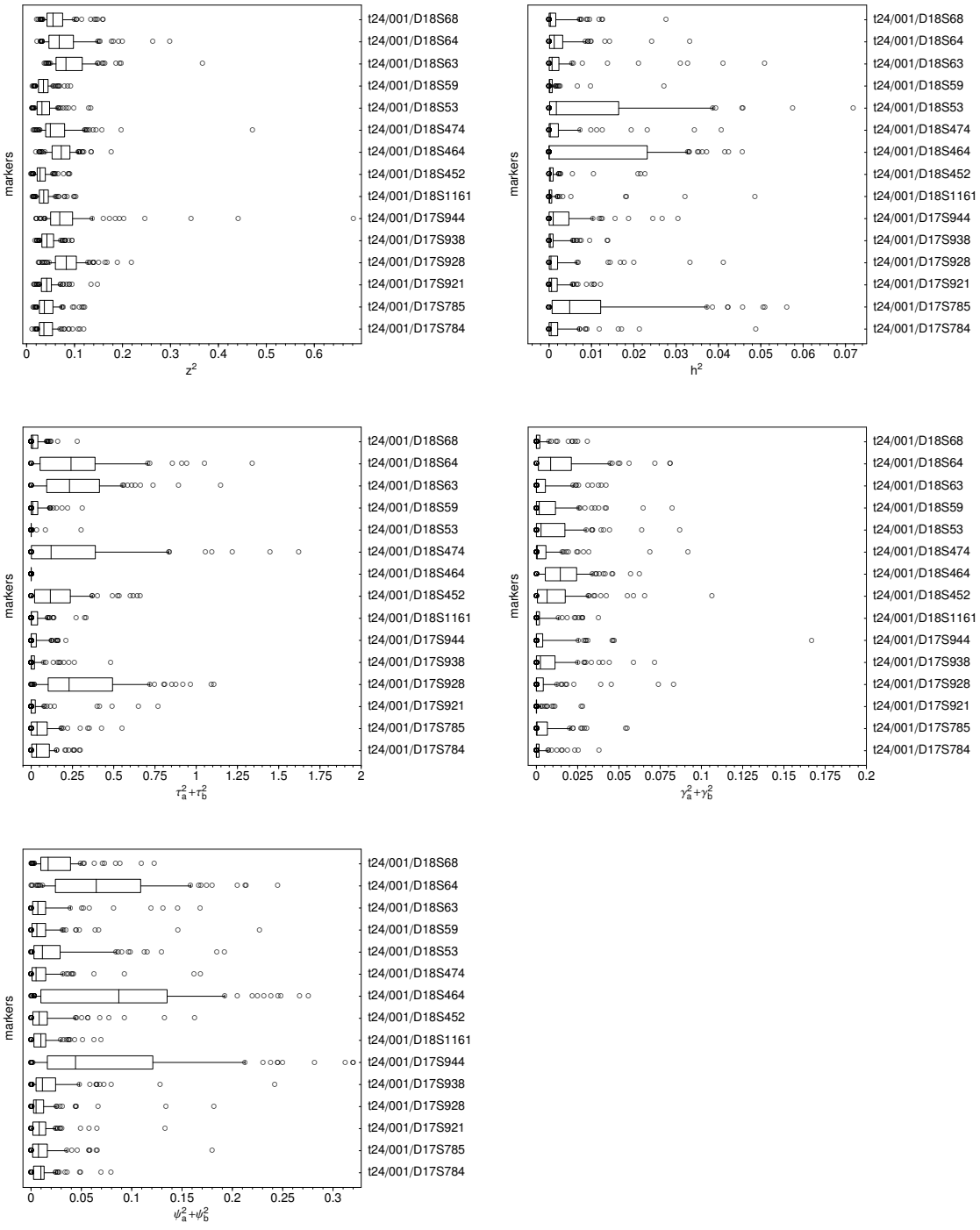


Figure 4.3: The distributions of the feature values for the true genotypes in several markers. They are long-tailed and symmetrical, and the distributions of each variable are fairly similar across markers. Those with the median shifted to the right and more spread out tails are known as 'problematic' markers. For example, the $\psi_a^2 + \psi_b^2$ values of D1S64, D1S464 and D1S944 are more variable because the ranges of these markers are above 300 bp, where electrophoretic migration is more "jittery".

Pálsson *et al* [1999]). We use this score for each allele:

$$\tau_{a,j}^2 = \begin{cases} 0 & \text{if } y_j(t_a) > 500 \\ [\log_{10}\{y_j(t_a)\} - \log_{10}(500)]^2 & 10 \le y_j(t_a) \le 500 \\ +\infty & \text{otherwise} \end{cases} \tag{4.6}$$

For the genotype (a, b):

$$d_3(a, b, j) = \frac{\tau_{a,j}^2 + \tau_{b,j}^2}{2\sigma_3^2} \tag{4.7}$$

where $\sigma_3^2 = 0.125$. Note that this transformation is very specific to the fluorescence intensity unit in ABI trace files. The scales seem to be fairly consistent across different runs and ABI 377 machines.

4. **Peak sharpness**

   Similar to peak highlighting method in chapter 2 (figure 2.9), we can bandpass filter the trace $\boldsymbol{y}_j$ to produce a signal $\boldsymbol{v}_j$ with highlighted peaks:

$$\boldsymbol{v}_j = \boldsymbol{B}\boldsymbol{y}_j \tag{4.8}$$

   The bandpass filter $\boldsymbol{B}$ is a cascade of single-pole lowpass and highpass filters (see appendix A), parameterized such that the gain is 0.5 at the frequency 0.5 and 3 bp per cycles, respectively. The value of $v_j(t_a)/y_j(t_a)$ is the 'sharpness index', which is independent of the signal strength.

   The squared deviation from the expected sharpness is:

$$\gamma_{a,j}^2 = \begin{cases} 0 & v_j(t_a)/y_j(t_a) \ge 0.3 \\ \left[0.3 - \dfrac{v_j(t_a)}{y_j(t_a)}\right]^2 & \text{if } 0 < v_j(t_a)/y_j(t_a) < 0.3 \\ +\infty & \text{otherwise} \end{cases} \tag{4.9}$$

   For the genotype (a, b):

$$d_4(a, b, j) = \frac{\gamma_{a,j}^2 + \gamma_{b,j}^2}{2\sigma_4^2} \tag{4.10}$$

   where $\sigma_4^2 = 0.01$. Note that the transformation is specific to the particular trace data (and the parameterization of the filter $\boldsymbol{B}$). The constant 0.3 is the typical value of $v_j(t_a)/y_j(t_a)$ for good peaks.

5. **Alignment shift**

   The trace alignment algorithm (chapter 2) estimates the curve that maps the expected DNA fragment size t to the observed size u:

   $$u(t) = t + \phi(t) + \psi_j(t) \tag{4.11}$$

   where $\phi(t)$ is the systematic warping and $\psi_j(t)$ is lane-specific, random "jitter". Large $\psi(t_a)$ or $\psi(t_b)$ indicates allelic peaks that stray too far from the expected location. The amount of shift can be used directly:

   $$d_5(a, b, j) = \frac{\psi_j^2(t_a) + \psi_j^2(t_b)}{2\sigma_5^2} \tag{4.12}$$

   where $\sigma_5^2 = 0.01$.

Ad hoc rules

There are also a number of *ad hoc* rules that were found to improve the performance significantly. These can be seen as features that have the value of either 0 or $+\infty$ depending on whether the condition is satisfied. The rules are:

1. The weaker allele should not be too small in proportion to the large one. If $\alpha/(\alpha + \beta) < 0.025$ or $\alpha/(\alpha + \beta) > 0.975$, then $Q_{a,b,j} = +\infty$.

2. The stutter peaks are occasionally explained better by a false allele. The following rule reduces the frequency of these incidents. if $\alpha < \beta$ and $b - a \leq 2$, use $2h_{a,b,j}^2$ instead of $h_{a,b,j}^2$. That is, if the two alleles are close together, and the shorter allele is weaker, then the deviation is increased to penalize the genotype.

3. If we allow the two alleles to differ by one bp, many homozygotes are falsely described by a pair where $b - a = 1$ or $b - a = 3$. For most dinucleotide markers, such pairs are never observed. The following rule is used: if $b - a < 4$ and $b - a$ is odd, throw away the genotype. Some markers are known to contain pairs with alleles differing by odd numbers. The rule above can be dropped for these markers (identified manually). Automatic detection of these 'odd' markers is still being investigated. The following rule seems to work for many cases: if the frequency of odd $b - a$ for $b - a > 4$ is higher than 10%, the marker might be a genuine 'odd' marker.

4.2.3 Weight optimization

The weights in equation 4.2 need to be chosen so that the hit rate is maximized for a given error rate. This is done empirically based on a 'training' data set. The objective function is computed as follows:

- Given a weight vector $\boldsymbol{w}$, compute $Q(a, b, j)$ for all j. In practice, it is sufficient to use the top twenty genotypes based on z^2 alone, because it is the dominant feature. This cuts down the computational expense dramatically.
- For each trace j, choose the best $Q(a, b, j)$ as the called genotype.
- Sort the traces according to $Q(a, b, j)$, and loop through them in ascending order, accumulating the number of error and correct calls. Stop when a specified error rate is reached, and return the cumulative counts of correct calls as the objective function value.

We consider failed measurements as errors, because we want them to have large Q-score. The error rate chosen is 1%, which is close to the 'critical' rate for genotyping data [Weeks *et al* 2002]. The performance curve tends to be somewhat smooth, so the hit rates for similar error rates are maximized as well. Using smaller error rate leads to a "bumpy" objective function landscape because of the smaller number of traces taken into account. Using larger error rate produces suboptimal ranking in the lower error range.

The optimization of the weights is done using the Nelder-Mead downhill simplex method [Nelder and Mead 1965]; the same tool we use for model fitting in chapter 3. The rationale is similar: there is no information about the objective function's derivatives (and it might be non-differentiable in some parts due to the discrete nature of the hit counts). Furthermore, $\boldsymbol{w}$ needs to be constrained and the number of parameters to be optimized is small. For p weights, only $p - 1$ of them needs to be optimized, because we are interested only in their relative magnitudes, i.e. the direction of the hyperplane, not how far it is from the origin. The weight for the z^2 score is fixed to one, and thus only four parameters need to be optimized. To escape local minima, we use several runs with completely randomized starting vertices (including the best vertex). For each run, multiple restarts as described in chapter 3 are also performed.

4.2.4 The L-score

The Q-score optimized as above is not necessarily comparable across different markers. Although within each marker it ranks the observations reasonably consistently (unreliable traces tend to have larger score), the same cutoff might not correspond to the same error rate. Firstly, the Q-score of the best genotype might not be equally distributed across markers. Secondly, the ability to discriminate depends not only on the scores for the best genotype, but also on the distribution of the scores for the second-best genotype. If the two are very close

or overlapping, calling errors are more likely. Figure 4.4 shows two examples of marker-specific distribution of the best and second-best Q-scores.

As shown in the picture, we fit a gamma distribution to the second-best distribution. A new quality score, called the L-score, is defined as follows:

$$L_{a,b,j} = -\log_{10} \int_0^{Q_{a,b,j}} G(q;\boldsymbol{\theta})\, dq \tag{4.13}$$

where $G(q;\boldsymbol{\theta})$ is the p.d.f. of the gamma distribution. The integral is the value of the cumulative distribution function at $Q_{a,b,j}$. The negative of the log is used because it is more intuitive (the higher the score, the better the quality) and the meaningful range, to be demonstrated later, is conveniently between 0 (the worst quality, unusable data) and 10 (extremely good quality). We don't have any theoretical rationale for using the c.d.f. value (although this is somewhat analogous to using the p-value of a null hypothesis as a test statistic). The c.d.f. is simply considered a monotone transform of the Q-score (adjusted to each data set), and the relationship between the L-score and the error rate is to be determined empirically.

The parameter vector $\boldsymbol{\theta}$ (the shape and scale of the gamma distribution) is estimated from the data using the maximum likelihood criterion. This has to be done iteratively, and although better method exists, it is quite simple to do so using the Nelder-Mead downhill simplex method[3], directly using the data likelihood as the objective function to maximize the two parameters of gamma distributions, with non-negativity constraints on the parameters. Data points with $Q > 10$ are considered outliers and thrown away before fitting the p.d.f.

4.2.5 Training and test data sets

Ideally, data sets with known "true" genotypes should be used. The ultimate way to look at the number of microsatellite repeats is by sequencing, but this is expensive and available only for a very few samples (such as the CEPH family member controls). There are also control samples that have been genotyped independently many times and used for testing the overall performance of genotyping labs [Weeks *et al* 2002]. We had no immediate access to such data, and furthermore, our method relies on having a large number of individuals per marker to estimate the model parameters. We therefore decided to use data set from the archive of the AGRF. These are traces and manual calls from the daily operation of the genotyping center. The advantage of using these data sets, in addition to their very large sizes, is that they reflect the reality of the genotyping operation. Some traces might be of bad quality because of poor DNA sample

[3] Already implemented in the allele calling program for other purposes.

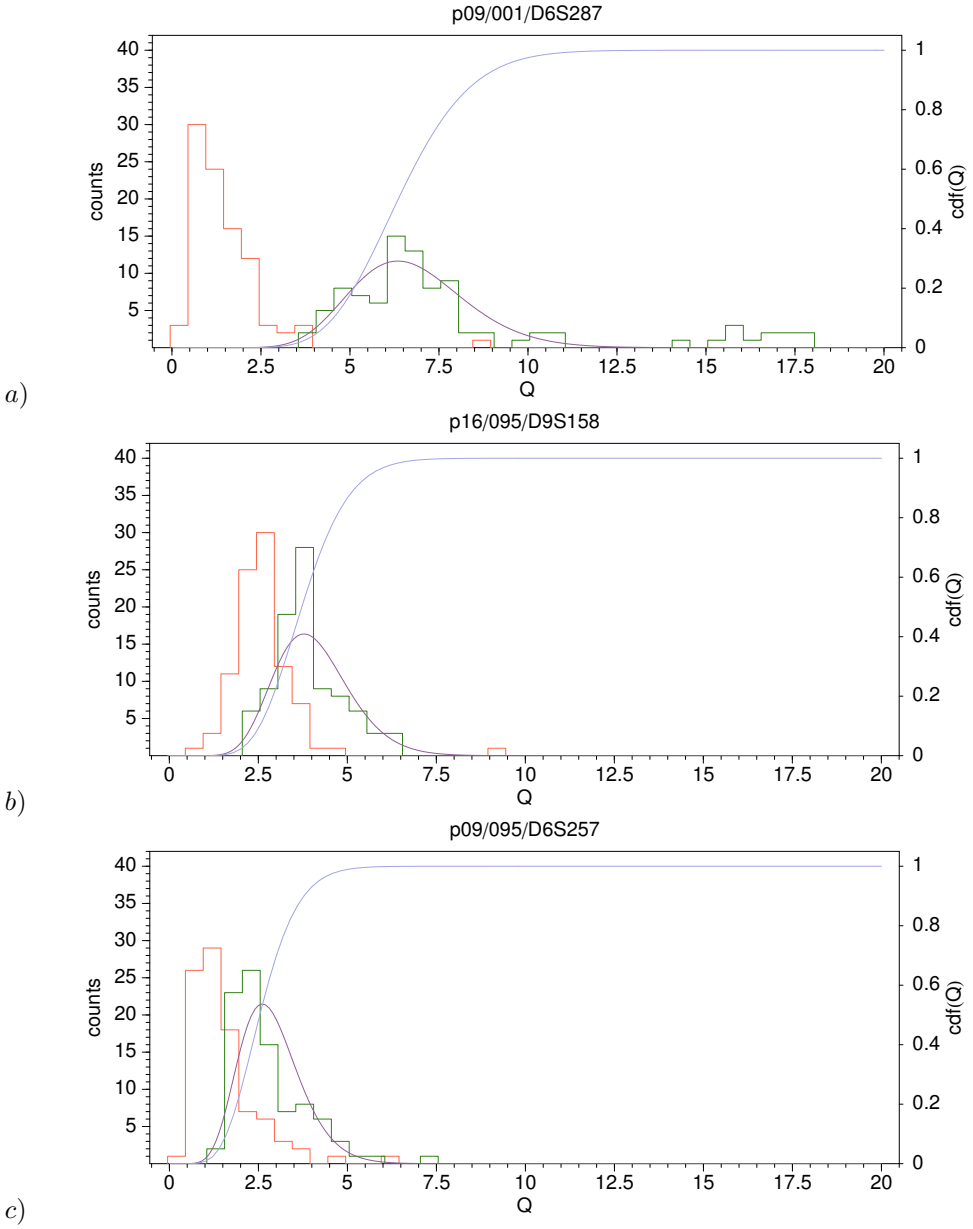


Figure 4.4: Examples of marker-specific distributions of the Q-score. In all plots, the histograms of the Q-scores for the best genotypes are shown in red, while those for the second-best genotypes are shown in green. In panel *a* the distributions of the best and second-best scores are well separated. Panel *b* and *c* shows overlapping distributions. The purple bell-shaped curves are gamma densities fitted to the second-best scores. The blue sigmoidal curves are the c.d.f. of the density up to Q (the values are indicated by the scales on the right vertical axis of each plot).

quality and certain markers might be inherently problematic (weakly amplified, prone to fluorescence cross-talk, or having atypical patterns).

We arbitrarily chose genotyping data from 10 different panels in the ABI Linkage Mapping Set v2.0. For each panel, 3 different runs of different individual samples were used. Each run has approximately 90+ lanes (ignoring the positive and negative controls). Two panels, panel 5 and 24, were used as the training set (with 29 distinct markers), while the rest (panel 8, 9, 10, 11, 12, 16, 19, and 20) were used as the test set, containing 118 distinct markers. The total number of traces[4] are 7,792 for the training set and 33,003 for the test set. All electrophoresis runs were performed on ABI 377 machines. Preprocessing (lane tracking, dye separation, baselining and SSF assignment) was done using the ABI GeneScan software. The complete list of the markers (and their marker intervals) can be found in appendix B.

Genotyping and manual allele calling were performed according to the procedure described in Ewen *et al* [2000]. If a trace can not be called or is ambiguous, measurement is repeated once before the genotype is declared to be unknown. We have two different set of calls available. The 'original' calls are those based on the first round of measurement, and the 'final' calls based some repeated traces. For training and testing our algorithm, only the traces from the first round of measurement are used, because large-scale testing is easier to do[5]. Moreover, we are interested in the performance on a fresh trace data, not on those that have been manually handpicked for repeated measurements. However, when comparing the automated and the manual calls, we use the 'final' calls because they are closer to the 'truth'.

4.2.6 Assessing the performance

There are several issues that need to be considered when using the manual calls as the 'true' genotypes. Firstly, some calling errors might be present in the manual calls. This has been estimated [Ewen *et al* 2000] to be somewhat small ($\leq 0.2\%$), and should be sufficient to test a calling system that is expected to have larger error rate. For brevity, we will use the term 'error' to mean 'disagreement'. The actual error rate of the calling algorithm might be slightly lower than the disagreement rate (it is unlikely that both calls are wrong when they agree).

Secondly, not all traces have their genotypes available. Although some are definitely measurement failures, others are not called because they are considered 'ambiguous' by the guidelines for the manual calling procedure. We

[4] A 'trace' here means a portion of an electrophoresis trace associated with a specific marker.

[5] We have not integrated the algorithm to the LIMS system at the time the investigation was conducted. The 'original' traces are conveniently organized into folders in the filesystem.

therefore need to distinguish between 'definite miscalls' (genotypes are assigned differently) and failures to reject what was considered ambiguous by human judgment. It is possible that some of the automated calls for these might turned out to be correct when compared to the (unknown) true genotypes. Discarding traces with unknown genotypes makes the comparison simpler, but the test does not reflect how the automated method would perform in real situations, where rejecting or accepting a trace is as difficult a decision as picking the best genotypes, if not more so.

Lastly, the allele labels produced by manual calls might differ from the automated ones due to different sizing and binning schemes, although each might be internally consistent (each allele is always given the same label across lanes). In such cases, the automated and manual calls refer to exactly the same peak in the chromatograms as the allelic peak, but different labels are used. These discrepancies should not be considered as errors, unless the labeling discrepancies are due to inconsistency within a binning scheme (for example, the same allele is binned differently in different traces). The different binning schemes of the automated and the manual calling system should be 'normalized' by finding a mapping between allele labels of the the two using the allele frequency profile (which should be very similar, with a few calling discrepancies, because they come from the same DNA samples). This is illustrated in figure 4.5.

Comparison procedure

For each trace, manual calling produces a pair of integer allele labels (chosen according to a binning scheme) and a pair of peak sizes estimated according to the local Southern method. The peak sizes can be inverted back into raw trace positions (or scan numbers). Thus, for the same trace we can see if the automated and the manual calling procedure refer to the same peak, regardless of the allele labels. Additionally, we are interested in assessing the ability of the trace alignment procedure (chapter 2) to bin alleles consistently across lanes and merge allele labels from different runs. Hence, the comparison needs to be done at the level of allele labels.

The first step that needs to be done in comparing the automated and the manual calls is to resolve the differences in allele labels due to labeling schemes. This is the same problem with combining calls from multiple sources with different sizing and binning methods, which was addressed in chapter 2 (section 2.3.6), using dynamic programming to align allele frequency profiles. The mapping in figure 4.5 is actually produced using the algorithm. The adjustments for most markers are constant one or two basepair shifts, although occasionally gaps need to be introduced. On the test set, out of 354 markers, 308 (87%) do not require

AGRF (manual)		STRAL/FA		
allele1	allele2	allele1	allele2	Q-score
315	319	314	318	1.26
?	?	318	325	5.59
325	329	324	327	2.72
319	325	318	324	2.75
325	327	324	325	4.72
315	325	314	324	2.30
315	327	314	325	2.77
315	333	314	332	1.93
327	327	325	325	2.36
327	327	325	325	3.14
325	327	324	325	3.51
325	327	324	325	1.84
325	327	324	325	4.00
327	329	325	327	1.83
325	329	324	327	2.87
327	329	325	327	3.31
327	329	325	327	2.51
327	327	325	325	3.38
319	327	318	325	3.13
315	319	314	318	1.54

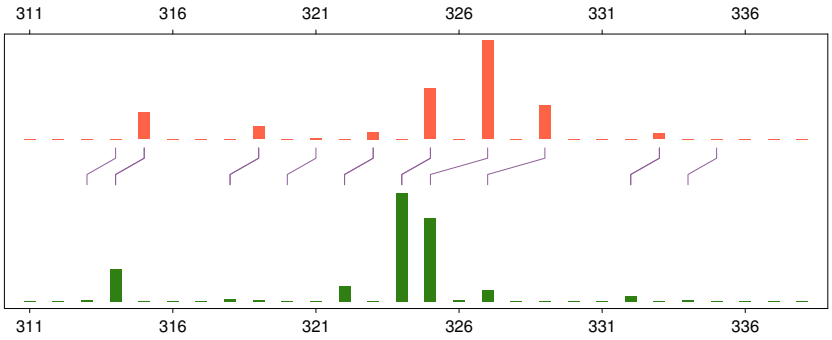


Figure 4.5: Comparing allele labels from different binning schemes. This is an (extreme) example of different allele labels used by the AGRF and STRAL/FA (the proposed methods). The table shows some of the calls made for some traces (the rows) from a marker data set (not all are shown). The figure below shows the histograms of the allele labels (the top one is for AGRF calls and the bottom for STRAL/FA). The purple lines are the appropriate mapping, which differs by 1 bp for most alleles except for those at 327 and 329 (in the AGRF scale). The migration behaviors of the two alleles are anomalous and the AGRF binning procedure shift them up using the assumption that alleles in dinucleotide repeats differ by 2 bp (while STRAL does not use this assumption by default, see chapter 2, page 59).

a gap, 41 (11.5%) require 1 gap, and the remaining 5 (1.5%) require 2 gaps. It is fair to question whether this adjustment procedure shifts some alleles that are inherently different calls into the same one, thus lowering the error counts. To address this, we require that identical calls should also match the peak locations (with a certain tolerance, say ± 0.5 bp) in addition to matching the allele labels.

After allele label alignment, one the following 'discrepancy types' are recorded for each comparison:

0 Perfect match: both allele labels and peak locations agree.

1 One-allele miscall. In one allele, the peak location and label disagree; while in the other, they match perfectly.

2 Two-allele miscall. Same as above but affecting both alleles.

3 One-allele binning disagreement: allele labels differ, but the peak locations agree, affecting one allele.

4 Two-allele binning disagreement. Same as above, but affecting both alleles.

5 'Unknown'. Manual call is not available.

The automated method always calls, so there is no case where it explicitly declares an observation a failure, although the quality value (the L-score) might be very low.

To have a general assessment of the performance, we need to count the trade-off between the number of errors and correct calls under various thresholds of the L-score. This is done by sorting the observations according to descending order of L-scores, and cumulatively counting the errors and hits. Rather than looking at all cases of discrepancies listed above, it is simpler to summarize the error types into the following:

Type A The combined counts for all type of discrepancies, including case 5 (missing manual call). This count reflects also the failure to reject traces that are deemed unreliable by human analysts. However, some of the automated calls might be correct when compared to the (unknown) true genotypes.

Type B The combined counts of discrepancy type 1, 2, 3, and 4. These are 'definite errors'.

Type C The combined counts of discrepancy type 1 and 2. Binning errors are excluded from this type so that we can assess the trace alignment method.

Assessment statistics

The statistics which is particular interest is the 'error rate':

$$\% \text{ error} = 100 \times \frac{\text{number of errors}}{\text{number of calls}} , \tag{4.14}$$

which is directly related to the measure used as error requirements by downstream analysis [Weeks *et al* 2002][6]. This is computed for all of the three error types (A, B or C).

Note that this definition of 'error rate' is different from that used in receiver operating characteristic (ROC) curve where the error rate is the number of false positives relative to the total number of negatives (which is unknown in our case). The error rate in ROC curve decreases monotonically when the threshold is made more stringent (because it is relative to a constant), whereas our error rate may increase even when the acceptance threshold is increased, if miscalls occur with good quality values (thus their frequency increases when correct calls are thrown away).

The trade-off of using a stringent quality value requirement is a reduced number of correct calls. We use a 'hit rate' measure relative to the total number of traces, including those that fail:

$$\% \text{ hit} = \frac{\text{number of correct calls}}{\text{total number of traces}} . \tag{4.15}$$

This gives us a clear idea about the yield for the whole genotyping process, but might not be a fair indicator of the caller's performance, because it is possible that one particular data set contains many inherently unusable traces (such as those due to bad DNA samples). To assess the algorithm itself, we use the hit rate relative to what human judgment can do for the traces:

$$\% \text{ relative hit} = \frac{\text{number of correct calls}}{\text{number of manual calls}} . \tag{4.16}$$

Note that for the 'number of manual calls', we counted only those available in the 'original' genotype tables, not in the 'final' ones (see the last paragraph of section 4.2.5). Thus it is possible that the relative hit rate exceeds 100% if the algorithm can get more correct genotypes than human seeing only the same traces. The 'original' calls are somewhat conservative because there is an option to repeat measurements for ambiguous data (which might still be less costly than making wrong calls). Nevertheless, the number of 'original' calls is a rough indicator of the portion of the data that are 'usable' (or the total number of 'positives'). Thus, the relative hit is approximately similar to the 'true positive rate' in ROC analysis. The algorithm performance on different data sets with varying qualities can therefore be compared more easily.

[6] The 'error rate' in PHRED quality score [Ewing and Green 1998] is also defined similarly.

4.3 Results and discussion

4.3.1 FAL1 allele caller

The optimization was done using multiple random restarts. There are many slightly different local optima that result in similar performance curves. The following value was found to be the best so far for our training data set: $\boldsymbol{w} = (1, 0.069919, 0.133511, 0.055029, 0.224701)$. We will refer to the new algorithm, with these specific weights, as 'FAL1' caller (for 'find allele using L-score, version 1').

The overall performance curve for the training set is shown in figure 4.6. The performance of FAL1 is dramatically better than the GLSA. An error rate[7] of less than 1% is achievable for up to 65% of the data (or up to 70% of the usable data based on the relative hit rate). 90% of the data is called correctly if we are willing to accept 4% error rate (the same level at which GLSA only gives 10% hit).

The performance is still much less than that of human analysts, which called 91% of the data at the (estimated) error rate of $\leq$ 0.2%. However, if this performance is similar for other data sets and the L-score can predict the error rate, a hybrid calling system can be devised, where a cutoff of $L \geq 3$ is used to choose traces that are called with less than 1% error. 60% of the data can be called automatically. Although all of the remaining 40% have to be examined, only 5% needs to be corrected. It is easier to confirm correctly pre-assigned genotypes than to edit them.

The relationship between the L-score and the probability of errors is illustrated in figure 4.7. Errors are more frequent on the left side, as the L-score decreases. Furthermore, the errors are also associated with the actual values of the L-score, not just with the ranking in a marker-specific way, as can be seen from the association of the dots with the color. The color map can be used to assist manual genotyping[8]. A human analyst can navigate by starting from the boundary between the green and cyan ($L = 3.0$) and manually examine the traces in the order prioritized by the L-score, from high to low (see Figure 4.8). Cleaner traces will be encountered first and can be scanned quickly. The examination can be stopped when the L-score becomes too low (say, less than 0.5 or so), discarding the rest of the traces without having to look at them.

[7] We will assume type A errors unless otherwise indicated. Note that this might be a bit conservative.

[8] Of course the dots are missing in a new data set.

$n = 7792$

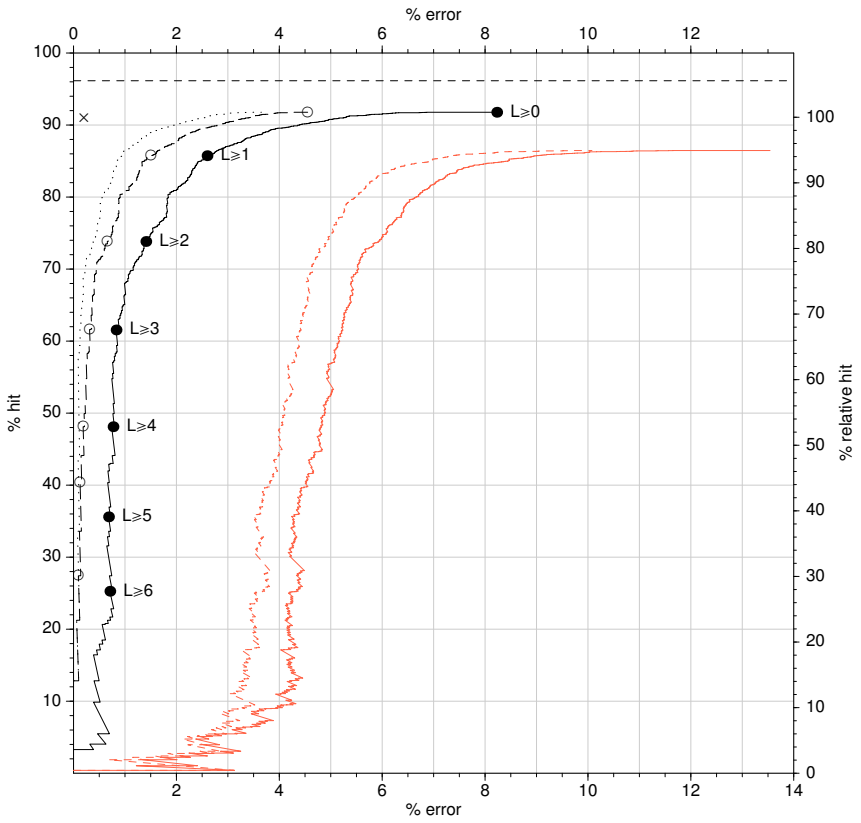


Figure 4.6: The performance of FAL1 on the training set are shown by the black solid, dashed and dotted curves (for error type A, B and C, respectively). The cutoffs based on the L-score are shown by the dots on the curves. The horizontal dashed line (at 96% hit) indicates the number of 'final' calls; while the '×' at 92% is the number of genotypes in the 'original' calls. The red curves are the performance of GLSA caller (error type A and B), which are the same as the ones in figure 3.12 and reproduced here for convenient comparison. FAL1 performs significantly better than GLSA.

Figure 4.7: A diagram illustrating the relationship between the L-score and various error types. The color map is made of small boxes each corresponding to a single trace. Each row contains traces from the same marker, ranked according to increasing L-score from the left to the right. The color corresponds to the L-score (the color scale is shown at the top left corner). If a box does not contain any dot, then the automated and the manual calls agree. White dots correspond to missing manual call (type 5 discrepancies, see page 4.2.6), red dots to type 1 or 2 (miscalls), and blue dots for type 3 or 4 (binning errors). Each box typically contains two dots. The top one is comparison against the 'original' calls, while the bottom against the 'final' calls. The numbers to the left of the map is the average L-score for the whole row (red if < 2.5). The columns N, G2, G1, M, E, B, ? are, respectively, the number of traces, 'final' calls, 'original' calls, matches, miscalls, binning errors and unknown genotypes. Note that only half of the training set is shown (panel 24 only).

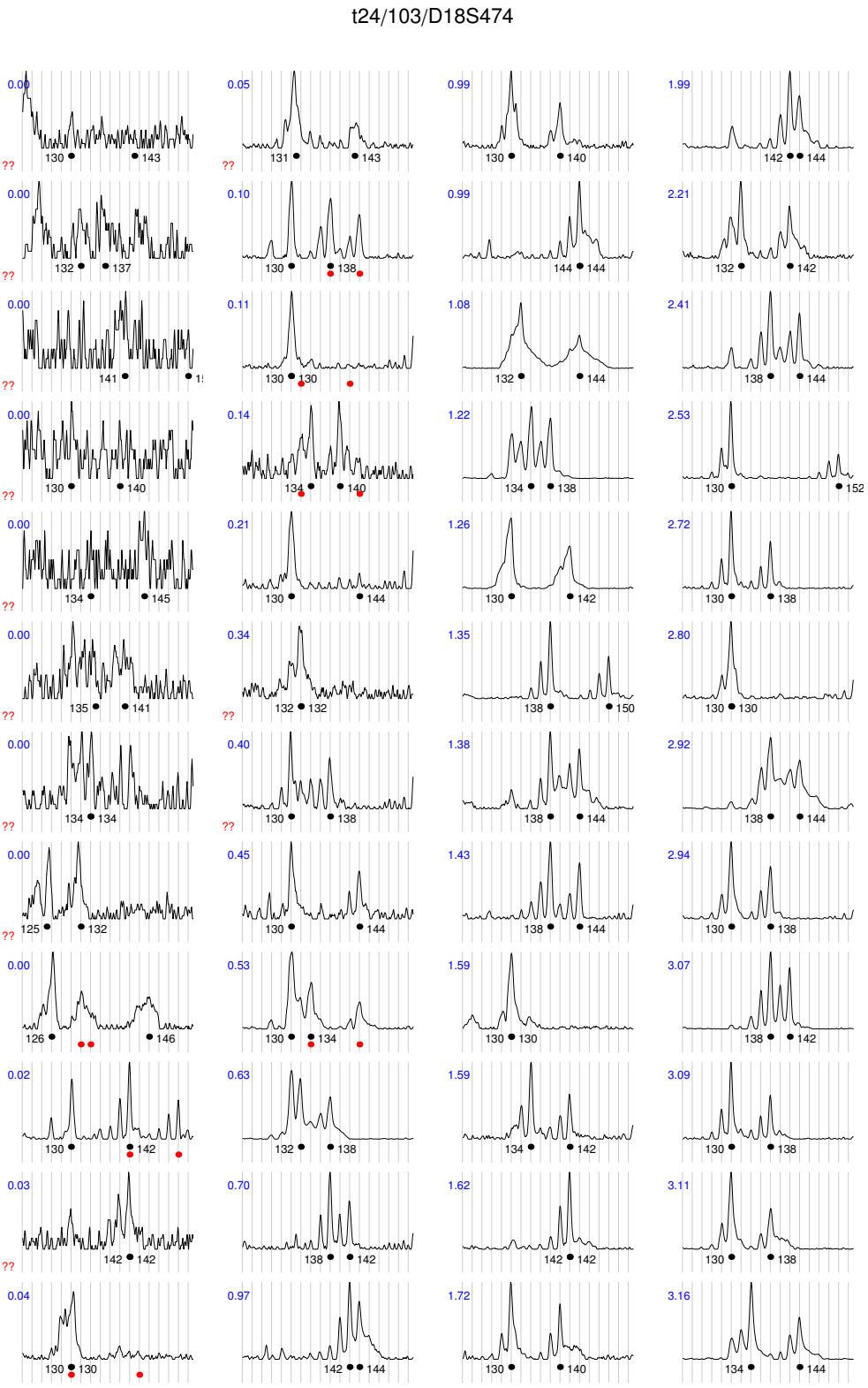


Figure 4.8: Trace data of the marker t24/103/D18S474 (row #67 in figure 4.7). The blue numbers are the L-score. The black, numbered dots are the automated calls. If a miscall occur, the true genotypes are shown by red dots. If the true genotype is unknown (white dots in figure 4.7), two red question marks are shown. Otherwise the calls match.

4.3.2 Performance on the test set

Examples of 'quality color maps' from the test set are shown in figure 4.9 and 4.10. As in the training set, the L-score is closely associated with the occurrence of errors. In figure 4.9, there are three sets of problematic traces (from the same marker `p16/*/D9S158`) where most of the calls are wrong. Figure 4.10 is a panel where the data quality is the worst overall, as shown by the low counts of manual calls (numbers in column G1), with some markers discarded entirely. In both examples, the L-scores are closely associated with the calling errors.

The performance curves on the test set are shown in figure 4.11. Overall, the calling performance on the test set is fairly similar to that on the training set. The curves for the test set are slightly more rounded, with a lower percentage of hits than the training set. However, the test set contains more bad quality traces, as indicated by its having fewer manual calls (82%) than the training set (92%). Many of these are in panel 12 (figure 4.10), which is known to be problematic even for human analysts (Wayne Ward, AGRF, personal communication).

Section C.2 (appendix C, page 144) shows the performance curves stratified according to the panels, as well as the complete quality maps of the test data set. We can see that the performance is closely related to the overall data quality as indicated by the number of traces callable by human. Panel 09, 10, 16 and 19 are particularly good. The error rates are actually better than that of the training set. In the other panels (particularly panel 12), the type A curves (including 'unknown' discrepancies) are worse, but the type B curves (definite errors) are not too different. This suggests that the L-score (which is calibrated from the particular training set we chose) is better at predicting definite miscalls than pointing out what human analysts would consider to be 'unknowns'.

If the L-score is to be used to partition the data into those that are automatically called or discarded, we need to assess its ability to predict the error rate. Figure 4.12 shows the error and hit rate as a function of the L-score for both data sets. In panel *a*, the error rate for the test is roughly similar. For L-score cutoffs less than 3, the type A error is higher in the test set. However, this is the range of quality that needs manual editing because the error rate is predicted to be more than 1%. Thus the underestimation is not harmful. At the recommended cutoff itself ($L = 3$), the error rate is very closely predicted. Interestingly, for L-scores greater than 3, the test set is better. Furthermore, increasing L-scores still correspond to a decreasing error rate in the test set, as opposed to the flat error rate for the training set.

For type B errors the test and training results are very close. Thus, for the good quality traces where manual calls can be made (in both training and test

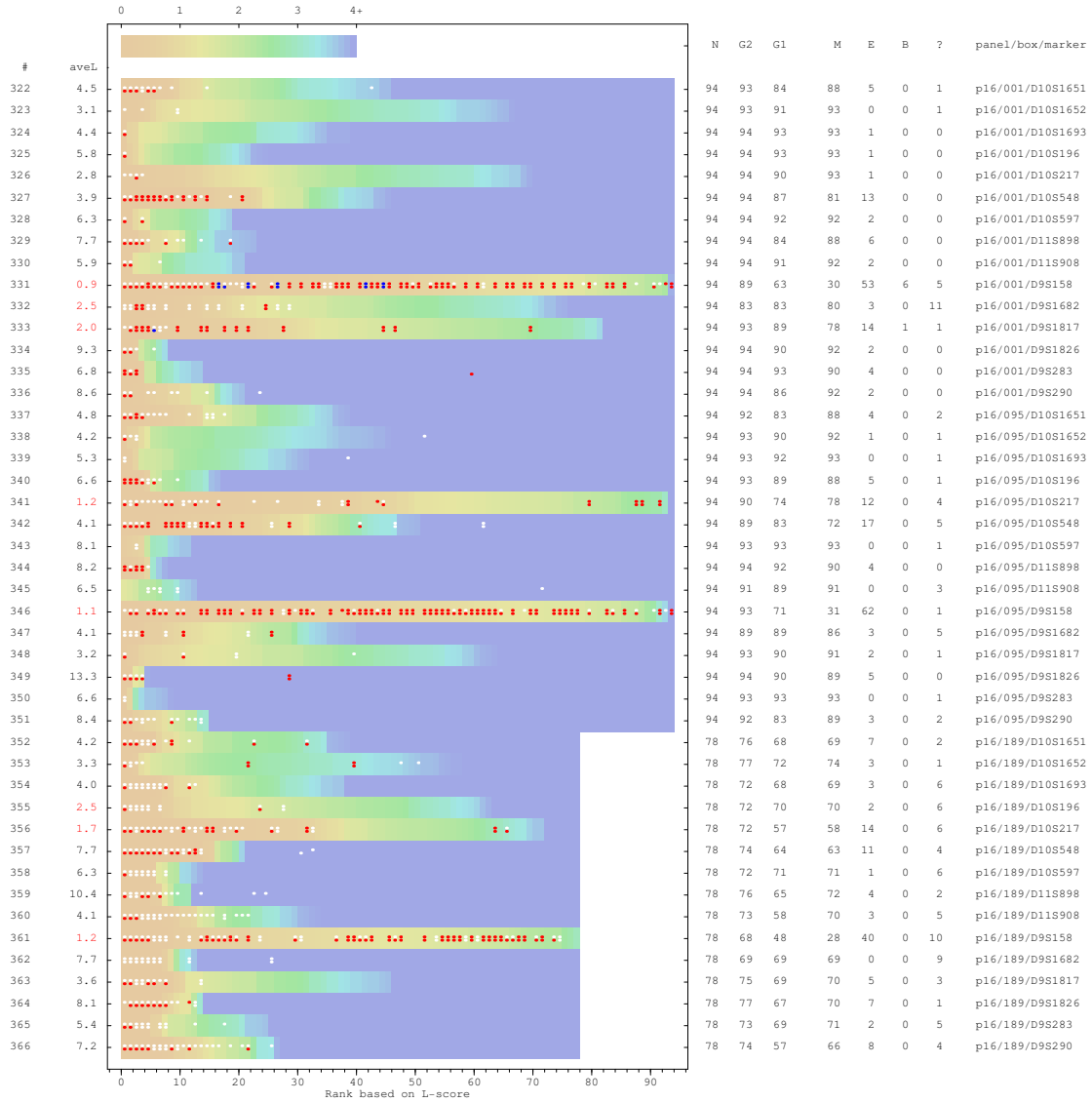


Figure 4.9: The quality map for panel 16 from the test set. See figure 4.7 (page 115) for descriptions.

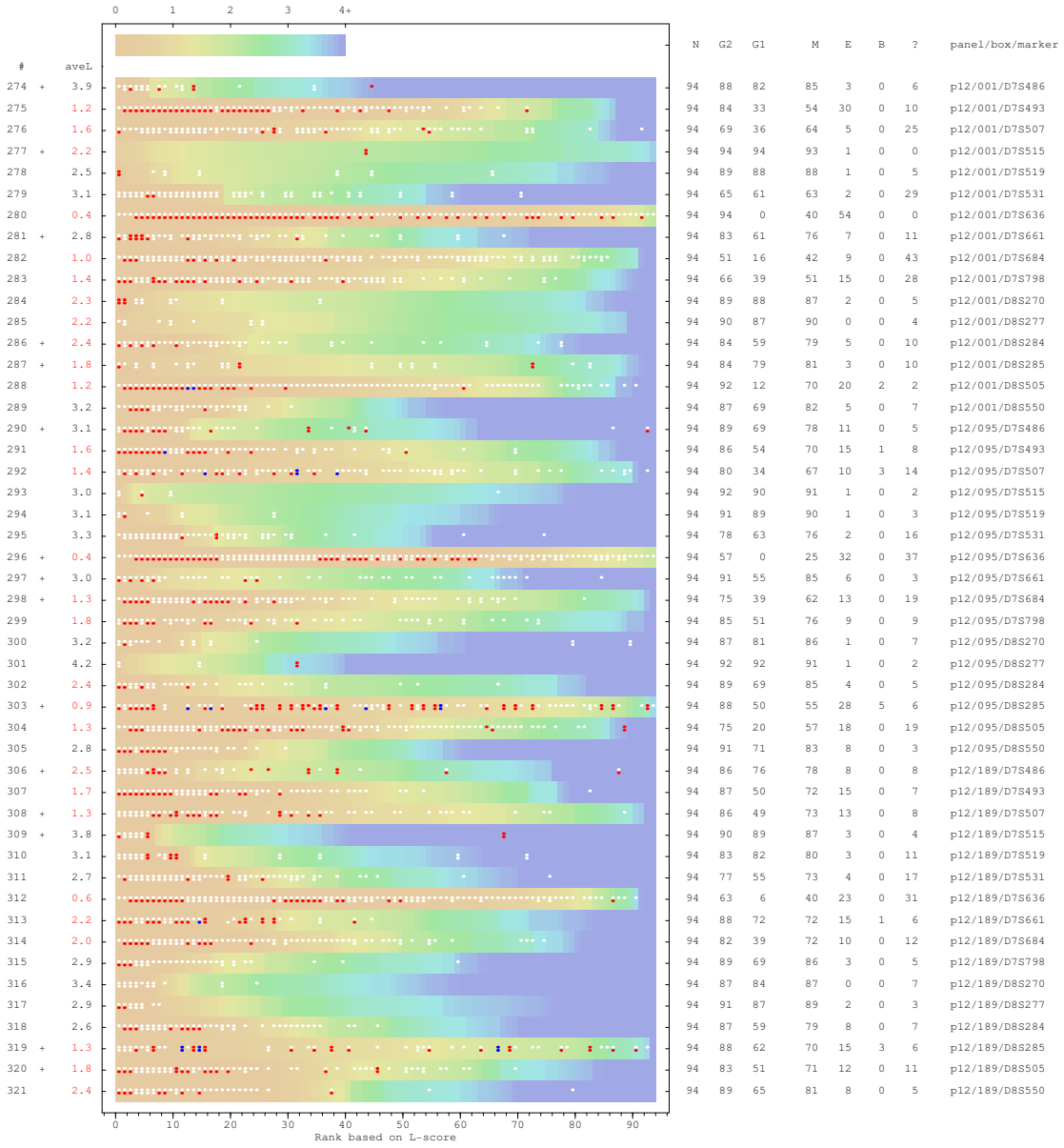


Figure 4.10: The quality map for panel 12 from the test set. See figure 4.7 (page 115) for descriptions.

$n = 33,003$

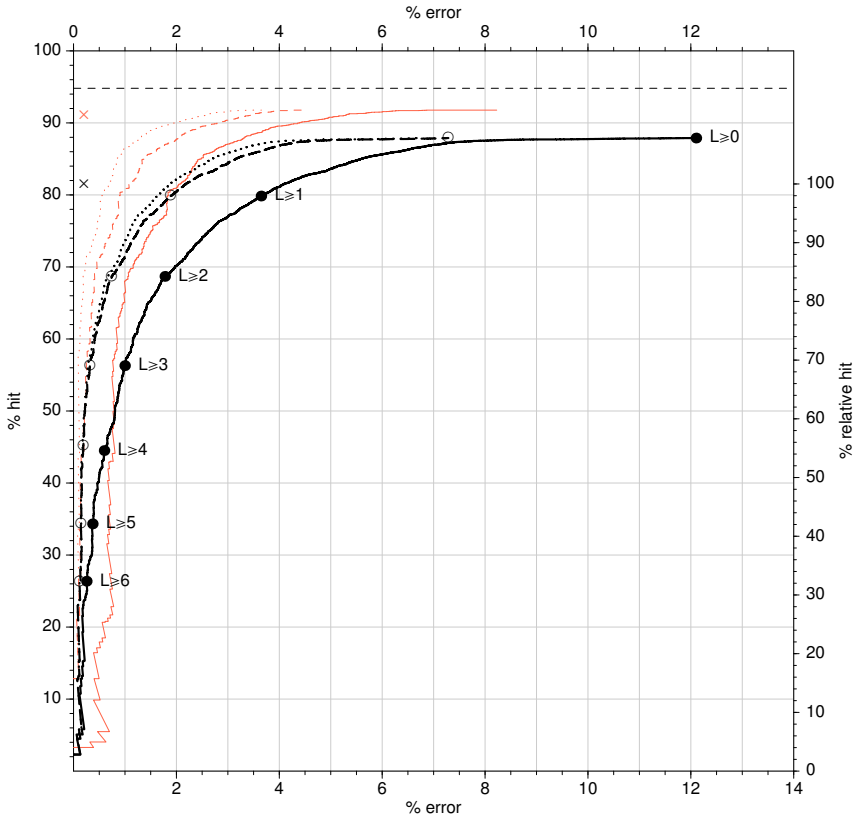


Figure 4.11: The automated calling performance on the test set (black curves) and the training set (red curves, same as the black curves in figure 4.6). Figure description is the same with that of figure 4.6. Note that only 82% of the test set can be called by human (the '×' mark), compared to nearly 92% of the training set. Both are fairly similar.

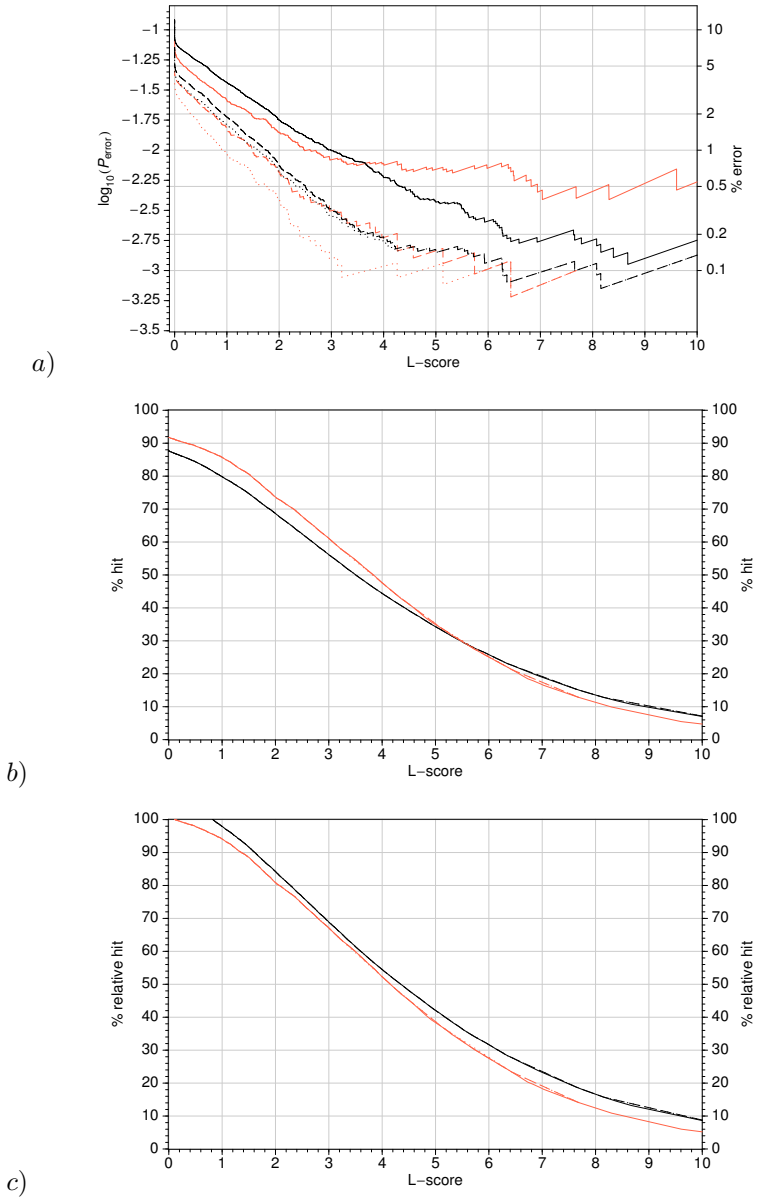


Figure 4.12: An alternative way to view the allele caller performance. The black and red curves correspond to the test set and training set, respectively. The solid, dashed and dotted curves corresponds to error type A, B and C. Panel a, b and c respectively show the error, hit and relative hit rate as functions of the L-score.

set), the L-score is associated with the same error rate. However, for bad quality traces discarded by human callers, which are more abundant in the test set, the L-score might underestimate the true error rate. This might be due to the lack of examples, in the training set, of traces that should be discarded. This might be improved by re-training the algorithm using the test set.

Type C errors are counted to assess the binning performance of the trace alignment algorithm. The difference between the type B and C curves is the proportion of binning errors. It is fairly low in the test set and curiously higher in the training set. When the trace data of the training set is examined, most of the binning errors occur in a particular marker (`t05/001/D4S405`, see quality map on page 142), which is prone to cross-talk from adjacent fluorescence channels. There are many spurious peaks shifts that are out of phase with the DNA fragment ladder in this marker, interfering with estimation of the alignment curves. Although the time scales are adjusted wrongly, resulting in labeling discrepancies, the subsequent allele calling algorithm still identifies the correct peaks based on the patterns.

Figure 4.12*b* shows the hit rate as a function of L-score. The two curves are quite similar, differing by ±5%. As in figure 4.12*a*, the hit rate of the test set is worst in the low range of L-score. The L-score is not expected to be an accurate predictor of the portion of correct calls (relative to the total number of traces), because this is largely determined by the quality of the data. The curves for the relative hit rate (figure 4.12c) are more similar, with the rate in the test set being slightly better overall, especially in the range of high L-scores.

We mentioned previously, based on the training set, a recommendation for a hybrid calling system where automated calls with $L \geq 3$ are accepted without re-examination, with less than 1% error and 60% of the data correctly called. If this is applied to the test set, slightly higher error rate is found (1% instead of 0.8%) and 56% of the data is correctly called. The relative hit rate is similar at nearly 70% of callable data.

Overall, the results of the test set closely resemble those of the training set. This is remarkable considering the markers are all different and the size of the training set is only 23% of the test set. We expect that the L-score will behave similarly on new data sets. Further improvement might be achieved if larger data set is used to re-calibrate the weight vector $\boldsymbol{w}$, although it is not clear yet how significantly. Our experience with developing the FAL1 caller from the GLSA caller, by incrementally adding features and rules, indicated that significant improvements required extra features, in addition to a larger training set.

One source of common errors yet to be incorporated into the L-score is the fluorescent dye cross-talk. Spurious peaks appear as the results of very

strong peaks in other color channels, which saturate the detector and violate the linearity assumption in the color separation algorithm. To get a feature variable that detect this instance, we would need to look at the raw fluorescence data. These are available in the same trace data file, but more investigation is required to find the appropriate transformation (there are four variables from all dye channels per allelic peak) to produce a feature that can be incorporated into the Q-score formula (equation 4.3).

It is difficult to compare the performance of STRAL/FA with those of existing methods. As mentioned in chapter 1 (page 19), there are only a few published results from other automated calling systems. Weber and Broman [2001] reported 94% accuracy, which is similar to that of our GLSA caller (it is not clear if they have a quality measure to subset the data). Pálsson *et al* [1999] reported a test result for 6912 genotypes. Their algorithm (TA/DecodeGT) selects 5806 of the as 'good' and 78 miscalls were found in this subset. This means an error rate of $78/5806 = 1.34\%$ and the rate of correct calls is 82.8%. On our test set, the yield for the same error rate is only 62% (type A) and 75% (type B). However, for individual panel, the yield might be better, or worse, depending on the data quality (see section C.2, page 144). Note that the data set used by Pálsson *et al* [1999] might be significantly better because problematic markers with odd alleles had been removed from their test set. It is also not clear how much of their data set can be called manually. If all of their traces are callable, then the data set is very good (no missing genotypes) and our method performs better on such data sets (see the curves for panel 09, 10, 16, 19). Furthermore, comparing the yield at 1.3% error is not really meaningful because manual examination still needs to be done on the calls. If the errors are evenly distributed within the 'good' subset, then all of them have to be examined. Their method does not provide a single quality measure like the L-score that can be varied to get the desired trade-off and can be used to rank the traces.

Without implying that our method is better or worse than the existing ones, we would like to conclude by stressing that proper benchmarking can only be done using exactly the same data set, using the same assessment criteria. Such studies are currently difficult to conduct because of the lack of openly shared trace data sets. We are planning to make our methods and some of the data sets publicly available in the near future to address this issue.

4.3.3 Summary

We have developed an allele calling algorithm based on a quality score, which also can roughly predict the error rate. Fully automated genotyping can be performed on 55% of the data (with $L \geq 3$) with less than 1% error. Manual

examination needs to be performed on about 20-30% of the traces with lower quality ($1 < L < 3$), which contain only up to 4% error. The remaining data (with $L < 1$) can be discarded. The yield after combining the automated and edited calls should be comparable to that of the manual method. The L-score can also make the editing software easier to use by ranking the traces.

The performance above are average results on the whole test set, with traces from different panels and runs pooled together. The statistics for individual panels or runs show some variability. We have not yet conducted a detailed analysis on how precisely the L-score can predict the error rate. Nor have we investigated how much the guidelines for a hybrid system can save on costs. We are currently still focusing on improving the calling algorithm. Under the framework that we have established for automatic optimization of quality scores and large-scale testing, it is likely that significant improvements can be made in the near future.

Chapter 5

Summary and Conclusions

5.1 Summary of the proposed method

The method is divided into three main steps: trace alignment, allelic pattern fitting, and calling by quality values. The first two steps remove run-, marker- and allele-specific variations, producing several feature variables that can be treated uniformly by the calling step. Calibration using a training set can therefore be done in a marker-independent manner. The outline of the steps is as follows.

1. Trace alignment (chapter 2)

   The objective is to normalize variations in the time domain of electrophoresis traces, so that the subsequent steps can treat the aligned traces from the same marker as a multivariate data matrix.

   - The input is a set of preprocessed trace data files from the same electrophoresis run. Tracking, color separation and identification of size-standard fragments still needs to be done by external software.
   - Each marker interval is processed separately; multiple lanes are analyzed simultaneously.
   - To correct for various biases, the alignment algorithm relies on the size-standard fragments (using a 2nd order loess curve) and the periodic pattern DNA fragment ladder (using dynamic programming alignment).
   - The main output is a matrix of resampled and aligned trace data. The alignment curves are also available.

2. Allelic pattern estimation (chapter 3)

The objective is to compensate for marker-specific PCR and electrophoresis artefacts, making it possible to use the same discrimination rules for all markers in the subsequent allele calling step.

- The input is an aligned trace data matrix.
- The expected shape of all possible alleles in marker data is modeled parametrically.
- The model has several components, corresponding to physical processes that occur during PCR and electrophoresis: unequal amplification ratio, untemplated 3' addition, polymerase slippage and electrophoretic diffusion. Eight marker-specific parameters are used.
- The model parameters are estimated from the data, using a least-squares criterion and optimized using the Nelder-Mead downhill simplex method.
- The model is used to produce two features for each possible genotype: allelic pattern fitness and deviation from heterozygote ratio.

3. Allele calling by quality values (chapter 4)

   - The input is a set of features (for every possible genotype):
     - Fitness between the trace data and the expected patterns
     - Deviation from the expected heterozygote ratio
     - Intensity of the main allelic peaks
     - Sharpness of the main allelic peaks
     - The amount of shift needed to align the main allelic peaks
   - A quality indicator, the Q-score, is derived as a weighted sum of the transformed features. The genotype with the lowest score is the most likely to be the true genotype.
   - The weights are optimized by maximizing the hit rate for a given error rate, say 1%.
   - The quality indicator is tuned further in a marker-specific way, using the distribution of the Q-score of the second-best genotype. The negative log of the c.d.f. (the L-score) produces identical calls with the Q-score, but the same quality threshold might be use across different markers.

Implementation The core of the methods have been implemented in UNIX (Linux) operating system, as a collection of programs and scripts written in C and `perl`. Currently, the input trace files have to be in ABI sample file format,

after preprocessing using ABI GeneScan software to remove color interference and identify the size-standard fragment peaks. Processing data a whole run of 96 lanes of a typical panel (15 markers) takes less than five minutes on a Pentium II/450MHz Linux machine. GUI software for browsing and editing the traces and genotypes are still being developed.

5.2 Results

Benchmark tests were performed on 33,003 genotypes from 24 gels (8 panels, 3 runs per panel with different individual samples), taken from the daily operations of the AGRF. Manual calls are available for up to 95% of the data (some from repeated genotyping). Error rates of $< 1\%$ can be achieved, at a data rejection threshold that still yields 55% correct calls. Up to 85% of the data can be correctly called if 5% error is acceptable. The performance on the test set is roughly the same as that of the training set, suggesting similar results for new data sets. Furthermore, the L-score can reasonably predict the error rate, particularly around the critical range of 1% error.

Based on the test set, we propose a partially-automated genotyping system:

- Accept all automated calls with $L \geq 3$. This portion is estimated to contain 1% error, and correct calls for around 50-60% of the data.
- Use the L-score to rank the traces (within each marker), and manually examine and edit them in the order of decreasing L-score.
- Discard the traces with $L < 1$.

This procedure can be used to produce genotypes with the same yield and error rate as the current system, but with less effort.

5.3 Future work

This project built a framework for further improvements. A mechanism for accessing the data and to benchmark the performance of new or modified algorithms has been established. This makes it easy to conduct further developments, such as:

- Calibration of the quality scores using a larger data set.
- Automatic tuning (based on a large calibration data set) of algorithm parameters that are currently fixed, such as the parameters for smoothing filters and dynamic programming in trace alignment, and the recursive filter combination in the allelic pattern model.

- More detailed examination of the wrong calls, and devising the possible improvements. For example, most errors are made in deciding between a homozygote and a heterozygote with similar patterns. The model of the unequal amplification ratio might need to be refined to include a length dependent variance.

- Implementation of a graphical user interface for browsing and editing the calls. Because the quality values are available for all possible genotypes, it useful to display not just the best one (the default call), but also the top few, most likely genotypes. Often, when the best genotype is erroneous, the true genotype is the second or the third one. Editing will be facilitated if ranked alternatives are provided.

- Improvement of model fitting by re-optimization using partially called data. The optimization of model parameters should be more reliable if the true genotypes are available for a few lanes. When the automated calls are being edited manually, the algorithm can be re-run incorporating the corrected genotypes on a few lanes, which might automatically fix similar errors in other lanes.

- Marker-specific parameters. Although the main philosophy of the approach is marker independent recognition, having marker-specific parameters (possibly implemented as "prior" parameters) might improve the performance. Unlike some other existing systems based on libraries of whole allelic patterns, it might be sufficient to store a handful of marker-specific parameters and use the fully adaptive approach if such parameters are not available.

5.4 Concluding remarks

A prototype of automated allele calling system has been developed. Unlike other automated allele calling methods, our algorithm is marker-independent and uses a predictive quality value. Although the performance is still not equal to that of human analysts, it may significantly facilitate manual allele calling.

This project highlights the power of generative models in adaptive recognition systems. The model compresses the systematic variability in data from heterogeneous sources (markers) into a handful of parameters that can be reliably estimated from the data itself. The extensive repository of manually called trace data, now unlocked by this project, opens the door for a more challenging problem: automatic development of the calling algorithm itself, by searching the space of possible models and analysis steps.

Bibliography

ABI (1996) GeneScan Analysis Software version 2.1. User's manual. Perkin-Elmer Applied Biosystems, Foster City, CA.

ABI (2001a) Genotyper Software version 2.5. User's manual. Perkin-Elmer Applied Biosystems, Foster City, CA.

ABI (2001b) ABI PRISM Linkage Mapping Set version 2.5. Panel Guide. Perkin-Elmer Applied Biosystems, Foster City, CA.

Ahmadian A, Lundeberg J (2002) A brief history of genetic variation analysis. *Biotechniques* **32**:1122-1128.

Antoniou A (1993) Digital filters: analysis, design and applications. *2nd ed.* McGraw-Hill, New York.

Beckmann JS, Weber JL (1992) Survey of human and rat microsatellites. *Genomics* **12**:627–631.

Berno AJ (1996) A graph theoretic approach to the analysis of DNA sequencing data. *Genome Res* **6**:80–91.

Beuzen ND, Stear MJ, Chang KC (2000) Molecular markers and their use in animal breeding. *Vet J* **160**:13–14.

Brownstein MJ, Carpten JD, Smith JR (1996) Modulation of non-templated nucleotide addition by Taq DNA polymerase: primer modifications that facilitate genotyping. *Biotechniques* **20**:1004-1010.

Campbell NA (1980) Robust procedures in multivariate analysis I: robust covariance estimation. *Appl Stat* **29**:231–237.

Carey L, Mitnik L (2002) Trends in DNA forensics analysis. *Electrophoresis* **23**:1386–1397.

Carrano AV, Lamerdin J, Ashworth LK, Watkins B, Branscomb E, Slezak T, Raff M, de Jong PJ, Keith D, McBride L, et al (1989) A high-resolution,

fluorescence-based, semiautomated method for DNA fingerprinting. *Genomics* **4**:129–136.

Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots.

Cutler A, Breiman L (1994) Archetypal analysis. *Technometrics* **36**:338–347.

Deforce DLD, Millecamps REM, Van Hoofstat D, Van den Eeckhout EG (1998) Comparison of slab gel electrophoresis and capillary electrophoresis for the detection of the fluorescently labeled polymerase chain reaction produtcs of short tandem repeat fragments. *J Chromatogr A* **806**:149–155.

Dekkers JCM, Hospital F (2002) The use of molecular genetics in the improvement of agricultural populations. *Nat Rev Genet* **3**:22–32.

Dib C, Faure S, Fizames C, Samson D, Drouot N, Vignal A, Millasseau P, Marc S, Hazan J, Seboun E, Lathrop M, Gyapay G, Morissette J, Weissenbach J (1996) A comprehensive genetic map of the human genome based on 5,264 microsatellites. *Nature* **380**:152–154.

Domnisoru C, Zhan X, Musavi M (2000) Cross-talk filtering in four dye fluorescence-based DNA sequencing. *Electrophoresis* **21**:2983–2989.

Edwards A, Caskey CT (1991) Genetic marker technology. *Curr Opin Biotechnol* **2**:818–822.

Elder JK, Southern EM (1983) Measurement of DNA length by gel electrophoresis II: comparisons of methods for relating mobility to fragment length. *Anal Biochem* **128**:227–231.

Elder JK, Southern EM (1987) Computer-aided analysis of one-dimensional restriction fragment gels. pp 165–172. In Bishop MJ, Rawlings CJ (eds) "Nucleic acid and protein sequence analysis—A practical approach". IRL Press, Oxford.

Eubank RL (1999) Nonparametric regression and spline smoothing. 2nd ed. Marcel Dekker, New York.

Ewen KR, Bahlo M, Treloar SA, Levinson DF, Mowry BF, Barlow JW, Foote SJ (2000) Identification and analysis of error types in high-throughput genotyping. *Am J Hum Genet* **67**:727–736.

Ewing B, Green P (1998) Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Res* **8**:186–164.

Ewing B, Hillier L, Wendl MC, Green P (1998) Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. *Genome Res* **8**:175–185.

Ghosh S, Karanjawala ZE, Hauser ER, Ally D, Knapp JI, Rayman JB, Musick A, Tannenbaum J, Te C, Shapiro S, Eldridge W, Musick T, Martin C, Smith JR, Carpten JD, Brownstein MJ, Powell JI, Whiten R, Chines P, Nylund SJ, Magnuson VL, Boehnke M, Collins FS (1997) Methods for precise sizing, automated binning of alleles, and reduction of error rates in large-scale genotyping using fluorescently labeled dinucleotide markers. FUSION (Finland-U.S. Investigation of NIDDM Genetics) Study Group. *Genome Res* **7**:165–178.

Gill P, Koumi P, Allen H (2001) Sizing short tandem repeat alleles in capillary array gel electrophoresis instruments. *Electrophoresis* **22**:2670–2678.

Gordon D, Abajian C, Green P (1998) *Consed*: a graphical tool for sequence finishing. *Genome Res* **8**:195–202.

Haberl M, Tautz D (1999) Comparative allele sizing can produce inaccurate allele size differences for microsatellites. *Mol Ecol* **8**:1347–1349.

Hahn M, Wilhelm J, Pingoud A (2001) Influence of fluorophor dye labels on the migration behavior of polymerase chain reaction—amplified short tandem repeats during denaturing capillary electrophoresis. *Electrophoresis* **22**: 2691–2700.

Hall JM, LeDuc CA, Watson AR, Roter AH (1996) An approach to high-throughput genotyping. *Genome Res* **6**:781–790.

Hamada H, Petrino MG, Kakunaga T (1982) A novel repeated element with Z-DNA-forming potential is widely found in evolutionarily diverse eukaryotic genomes. *Proc Natl Acad Sci* **79**:6465–6469.

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning: data mining inference and prediction. *Springer, New York.*

Hauge XY, Litt M (1993) A study of the origin of 'shadow bands' seen when typing dinucleotide repeat polymorphisms by the PCR. *Hum Mol Genet* **2**:411–415.

Hite JM, Eckert KA, Cheng KC (1996) Factors affecting fidelity of DNA synthesis during PCR amplification of d(C-A)n.d(G-T)n microsatellite repeats. *Nucleic Acids Res* **24**:2429–2434.

Hudson TJ, Engelstein M, Lee MK, Ho EC, Rubenfield MJ, Adams CP, Housman DE, Dracopoli NC (1992) Isolation and chromosomal assignment of 100

highly informative human simple sequence repeat polymorphisms. *Genomics* **13**:622–629.

Idury RM, Cardon LR (1997) A simple method for automated allele binning in microsatellite markers. *Genome Res* **7**:1104–1109.

Kim KS, Yeo JS, Choi CB (2002) Genetic diversity of north-east Asian cattle based on microsatellite data. *Anim Genet* **33**:201-204.

Kirov G, Williams N, Sham P, Craddock N, Owen MJ (2000) Pooled genotyping of microsatellite markers in parent-offspring trios. *Genome Res* **10**:105–115.

Knowles JA, Vieland VJ, Gilliam TC (1992) Perils of gene mapping with microsatellite markers. *Am J Hum Genet* **51**:905–508.

Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* **401**:788–791.

Levitt RC, Kiser MB, Dragwa C, Jedlicka AE, Xu J, Meyers D, Hudson JR (1994) Fluorescence-based resource for semiautomated genomic analyses using microsatellite markers. *Genomics* **24**:361–365.

Li L, Speed TP (1999) An estimate of the crosstalk matrix in four-dye fluorcence-based DNA sequencing. *Electrophoresis* **20**:1433–1442.

Li L, Speed TP (2000) Parametric deconvolution of positive spike trains. *Ann Stat* **28**:1279–1301.

Li JL, Deng H, Lai DB, Xu F, Chen J, Gao G, Recker RR, Deng HW (2001) Toward high-throughput genotyping: dynamic and automatic software for manipulating large-scale genotype data using fluorescently labeled dinucleotide markers. *Genome Res* **11**:1304–1314.

Litt M, Lutty JA (1989) A hypervariable microsatellite revealed by in vitro amplification of a dinucleotide repeat within the cardiac muscle actin gene. *Am J Hum Genet* **44**:397–401.

Loader C (1999) Local regression and likelihood. *Springer, New York*.

Mallat S (1998) A wavelet tour of signal processing. Academic Press, San Diego.

Mansfield DC, Brown AF, Green DK, Carothers AD, Morris SW, Evans HJ, Wright AF (1994) Automation of genetic linkage analysis using fluorescent microsatellite markers. *Genomics* **24**:225–233.

Mansfield ES, Vainer M, Enad S, Barker DL, Harris D, Rappaport E, Fortina P (1996) Sensitivity, reproducibility, and accuracy in short tandem repeat genotyping using capillary array electrophoresis. *Genome Res* **6**:893–903.

Mayrand PE, Corcoran KP, Ziegle JS, Robertson JM, Hoff LB, Kronick MN (1992) The use of fluorescence detection and internal lane standards to size PCR products automatically. *Appl Theor Electrophor* **3**:1–11.

Miller MJ, Yuan B (1997) Semiautomated resolution of overlapping stutter patterns in genomic microsatellite analysis. *Anal Biochem* **251**:50–56.

Mott R (1998) Trace alignment and some of its applications. *Bioinformatics* **14**:92–97

Nelder JA, Mead R (1965) A simplex method for function minimization. *Computer Journal* **7**:308–313.

Odelberg SJ, White R (1993) A method for accurate amplification of polymorphic CA-repeat sequences. *PCR Methods Appl* **3**:7–12.

Pálsson B, Pálsson F, Perlin M, Gudbjartsson H, Stefansson K, Gulcher J (1999) Using quality measures to facilitate allele calling in high-throughput genotyping. *Genome Res* **9**:1002–1012.

Perlin MW, Burks MB, Hoop RC, Hoffman EP (1994) Toward fully automated genotyping: allele assignment, pedigree construction, phase determination, and recombination detection in Duchenne muscular dystrophy. *Am J Hum Genet* **55**:777–787.

Perlin MW, Lancia G, Ng SK (1995) Toward fully automated genotyping: genotyping microsatellite markers by deconvolution. *Am J Hum Genet* **57**:1199–1210.

Perlin MW (2000) Methods and system for genotyping. United States Patent No. 6,054,268. *US Patent and Trademark Office.*

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C. Cambridge University Press, Cambridge, MA.

Puers C, Hammond HA, Jin L, Caskey CT, Schumm JW (1993) Identification of repeat sequence heterogeneity at the polymorphic short tandem repeat locus HUMTH01[AATG]$_n$ and reassignment of alleles in population analysis by using a locus-specific allelic ladder. *Am J Hum Genet* **53**:953–958.

Rabiner WR, Juang BH (1993) Fundamentals of speech recognition. *Prentice Hall, Englewood Cliffs, N.J.*

Ramsay JO, Silverman BW (1997) Functional data analysis. Springer, New York.

Richterich P (1998) Estimation of errors in "raw" DNA sequences: a validation study. *Genomer Res* **8**:251–259.

Riekkola ML, Jönsson JA (2001) Terminology for analytical capillary electromigration techniques. IUPAC Provisional Recommendations. Project 530/10/95. To be published. (online document: `www.iupac.org`).

Roberts S, Everson S (2001) Independent component analysis: principles and practice. *Cambridge University Press.*

Saitoh H, Ueda S, Kurosaki K, Kiuchi M (1998) The different mobility of complementary strands depends on the proportion AC/GT. *Forensic Sci Int* **94**:155–156.

Sankoff D, Kruskal J (1983) Time warps, string edits, and macromolecules: the theory and practice of sequence comparisons. *CSLI publications (1999 reprint).*

Sidransky D (1994) Nucleid acid-based methods for the detection of cancer. *Science* **278**:1054–1058.

Smith RN (1995a) Accurate size comparison of short tandem repeat alleles amplified by PCR. *Biotechniques* **18**:122–128.

Smith JR, Carpten JD, Brownstein MJ, Ghosh S, Magnuson VL, Gilbert DA, Trent JM, Collins FS (1995b) Approach to genotyping errors caused by non-templated nucleotide addition by Taq DNA polymerase. *Genome Res* **5**:312–317.

Southern EM (1979a) A preparative gel electrophoresis apparatus for large scale separations. *Anal Biochem* **100**:304–318.

Southern EM (1979b) Measurement of DNA length by gel electrophoresis. *Anal Biochem* **100**:319–323.

Stoughton R, Bumgarner R, Frederick WJ 3rd, McIndoe RA (1997) Data-adaptive algorithms for calling alleles in repeat polymorphisms. *Electrophoresis* **18**:1–5.

Tautz D (1989) Hypervariability of simple sequences as a general source for polymorphic DNA markers. *Nucleic Acid Res* **17**:6463–6471.

Tibbetts C (1995) Raw data file formats, and the digital and analog raw data streams of the ABI PRISM 377 DNA sequencer. Unpublished. Available online at: `www.cs.cmu.edu/afs/cs/project/genome/ftp/other/377_Raw_Data.ps`

Toth G, Gaspari Z, Jurka J (2000) Microsatellites in different eukaryotic genomes: survey and analysis. *Genome Res* **10**:967–981.

Tu O, Knott T, Marsh M, Bechtol K, Harris D, Barker D, Bashkin J (1998) The influence of fluorescent dye structure on the electrophoretic mobility of end-labeled DNA. *Nucleic Acid Res* **26**:2797–2802.

Walsh PS, Fildes NJ, Reynolds R (1996) Sequence analysis and characterization of stutter products at the tetranucleotide repeat locus vWA. *Nucleic Acids Res* **24**:2807–2812.

Wang CP, Isenhour TL (1987) Time-warping algorithm applied to chromatographic peak matching gas chromatography/Fourier transform infrared/mass spectrometry. *Anal Chem* **59**:649–654.

Weber JL, May PE (1989) Abundant class of human DNA polymorphisms which can be typed using the polymerase chain reaction. *Am J Hum Genet* **44**:388–396.

Weber JL (1990) Informativeness of human (dC-dA)$_n$·(dG-dT)$_n$ polymorphisms. *Genomics* **7**:524–530.

Weber JL, Broman KW (2001) Genotyping for human whole-genome scans: past, present and futures. *Adv Genet* **42**:77–96.

Webster MT, Smith NG, Ellegren H (2002) Microsatellite evolution inferred from human-chimpanzee genomic sequence alignment. *Proc Natl Acad Sci USA* **99**:8748–8753.

Weeks DE, Conley YP, Ferrell RE, Mah TS, Gorin MB (2002) A tale of two genotypes: consistency between two high-throughput genotyping centers. *Genome Res* **12**:430–435.

Weissenbach J, Gyapay G, Dib C, Vignal A, Morissette J, Millasseau P, Vaysseix G, Lathrop M (1992) A second-generation linkage map of the human genome. *Nature* **359**:794–801.

Wenz H, Robertson JM, Menchen S, Oaks F, Demorest DM, Scheibler D, Rosenblum BB, Wike C, Gilbert DA, Efcavitch JW (1998) High-precision genotyping by denaturing capillary electrophoresis. *Genome Res* **8**:69–80.

Zhao C, Heil J, Dickinson W, Ott L, Hamm M, Vaske D, Christensen C, Weber JL (1998) Improvement in accuracy on allele binning in large-scale genotyping. Unpublished report. The Center for Medical Genetics, Marshfield Medical Research Foundation.

Ziegle JS, Su Y, Corcoran KP, Nie L, Mayrand PE, Hoff LB, McBride LJ, Kronick MN, Diehl SR (1992) Application of automated DNA sizing technology for genotyping microsatellite loci. *Genomics* **14**:1026–1031.

Appendix A

Recursive Linear Filters

Filtering in the frequency domain are required by various parts of the allele calling system, both for smoothing (low-pass filtering) and enhancing or sharpening features (band-pass filtering). This is equivalent to applying a time-invariant linear operator (or transforming by a Toeplitz matrix). Such operation is also known as convolution.

Fast Fourier Transform can be used to perform the task with $O(n \log n)$ complexity, instead of $O(n^2)$ required by naive matrix multiplication. Faster computation, in $O(n)$ might be achieved, for certain types of convolution kernels, using recursive or infinite impulse response filter (Antoniou [1993], Press *et al* [1992, pp558–564]). If the desired filter response can be described by the rational transfer function:

$$\mathcal{H}(z) = \frac{\sum_{i=0}^{M} c_i z^i}{1 - \sum_{j=1}^{N} d_j z^j}, \tag{A.1}$$

then the filtering can be performed by computing:

$$y_t = \sum_{i=0}^{M} c_i x_{t-i} + \sum_{j=1}^{N} d_j y_{t-j} \tag{A.2}$$

Only small number of coefficients (fewer arithmetic operations) are needed to obtain filters with sharp change in the frequency response. The filtering can also be done "in-place", overwriting the input by the output, requiring only a short queue. The disadvantage is a more complicated design procedure to find a rational polynomial that fit the required response. Additionally the phase response is non-linear, which distorts the symmetry of the peaks in the signal and changes their locations.

In chromatographic analysis, the location of the peaks are important and needs to be preserved. To obtain zero-phase response (which preserve the locations of the peaks), a cascade of causal (moving forward in time) and anti-causal (moving backward in time) filtering may be used. This is possible for our problem, where the input signals are available in memory and the data points can be accessed in both directions.

We use a handful of building blocks that can be cascaded to achieve sharper frequency response. For smoothing, the simplest is the exponential filter:

$$y_t = ax_t + (a-1)y_{t-1}\,, \qquad 0 \le a \le 1. \tag{A.3}$$

When forward and backward filters are combined, a symmetric impulse response is obtained. Cascading several forward-backward pairs produces a more rounded impulse response that tend to a Gaussian shape as the "order" of the cascade is increased. This filter is used for smoothing alignment scores and paths (chapter 2). The non-negative impulse response (convolution kernel) is also useful for modeling chromatographic spread function (chapter 3). This is not really "filtering" in the usual sense, but a convenient way to construct easily parameterized shapes as basis vectors in linear least-squares approximations. An extension to the time-invariant exponential filter is the time-varying filter where the coefficient a changes smoothly and monotonically with time, which is used in modeling the slippage patterns that depend on the allelic length (detailed in chapter 3).

Other building blocks are one-pole highpass and lowpass filters, with the transfer functions:

$$\mathcal{H}(z) = \frac{1+a}{2}\frac{(z-1)}{(z-a)} \qquad \text{(highpass)} \tag{A.4}$$

and

$$\mathcal{H}(z) = \frac{1-a}{2}\frac{(z+1)}{(z-a)} \qquad \text{(lowpass)} \tag{A.5}$$

which can be realized by:

$$\begin{aligned} y_t &= \frac{1+a}{2}(x_t - x_{t-1} + ay_{t-1}) && \text{(highpass)} \\ y_t &= \frac{1-a}{2}(x_t + x_{t-1} + ay_{t-1}) && \text{(lowpass)} \end{aligned} \tag{A.6}$$

The two can be combined to produce a bandpass filter. Sharper frequency response (narrow band in the Fourier domain) can be obtained by cascading the filters.

A more complex filter that we use is a Butterworth (or maxflat) filter for highlighting the 1 bp periodicity prior to trace alignment (chapter 2). Any textbook on digital signal processing, e.g. Antoniou [1993], can be consulted for designing Butterworth filters.

Appendix B

Panels of the Data Set

The markers are a subset of the ABI Linkage Mapping version 2 (medium density genome scan set). 10 panels were selected. Two were used for the training set, while the rest were used for testing. The two tables below list the markers and the intervals. The size ranges are slightly larger than those specified in ABI Panel Guide. Fluorescent dye 1, 2, 3 are 6-FAM, HEX and NED, respectively.

Table B.1: Markers used in the training set to optimize the weights $\boldsymbol{w}$. 29 distinct markers are used, each comprising three runs of different individuals. The total number of traces is 7,792.

panel 5			
marker	dye	size range	
D4S392	1	79	119
D3S1311	1	132	165
D3S1565	1	165	203
D4S406	1	241	277
D4S1575	1	287	315
D3S1271	2	83	113
D3S3681	2	119	175
D4S414	2	230	258
D4S405	2	279	317
D3S1614	3	95	136
D4S1534	3	140	177
D3S1263	3	185	225
D3S1285	3	233	261
D4S1597	3	273	309

panel 24			
marker	dye	size range	
D17S938	1	235	265
D18S464	1	300	325
D18S474	1	120	155
D18S53	1	155	190
D17S784	2	220	250
D17S921	2	190	220
D18S59	2	150	180
D18S63	2	75	120
D18S64	2	305	350
D17S785	3	165	200
D17S928	3	70	115
D17S944	3	310	345
D18S1161	3	215	250
D18S452	3	125	155
D18S68	3	265	300

Table B.2: 118 markers in 8 panels were used for testing. Each panel has three runs of different individuals, totaling 33,003 traces.

panel 8

marker	dye	size range	
D5S407	1	85	120
D6S289	1	158	190
D6S1610	1	199	223
D6S1581	1	256	280
D6S422	1	295	327
D5S644	2	81	121
D6S281	2	131	161
D6S262	2	166	196
D5S424	2	205	242
D5S419	2	253	295
D5S433	3	64	104
D5S422	3	113	145
D5S406	3	163	201
D5S400	3	216	248
D6S309	3	302	338

panel 9

marker	dye	size range	
D6S264	1	108	140
D6S1574	1	150	181
D6S276	1	200	242
D5S408	1	248	294
D6S308	1	323	361
D6S287	2	105	149
D6S292	2	153	185
D6S434	2	201	255
D5S426	2	274	308
D5S1981	3	116	136
D6S257	3	165	203
D6S446	3	217	239
D5S641	3	296	346

panel 10

marker	dye	size range	
D5S2027	1	175	211
D5S436	1	230	267
D6S460	1	275	310
D5S410	1	327	359
D6S462	2	103	129
D5S2115	2	141	190
D5S418	2	207	237
D5S428	2	243	271
D5S630	2	275	380
D6S470	3	120	155
D6S441	3	161	205
D5S471	3	236	266
D5S416	3	284	306
D5S647	3	323	373

panel 11

marker	dye	size range	
D7S484	1	97	125
D8S264	1	130	169
D8S260	1	190	226
D7S517	1	242	270
D8S1784	1	275	303
D7S2465	1	316	350
D8S549	2	73	93
D7S530	2	104	132
D8S258	2	141	165
D7S669	2	171	203
D8S272	2	210	270
D7S502	2	286	316
D7S630	2	323	361
D7S510	3	80	106
D7S640	3	110	160
D7S513	3	167	207
D8S514	3	211	241
D7S657	3	243	279
D7S516	3	303	333
D8S1771	3	340	374

panel 12

marker	dye	size range	
D7S507	1	80	120
D7S515	1	135	211
D7S486	1	221	245
D7S519	1	255	293
D7S661	1	300	344
D7S798	2	71	103
D8S505	2	109	133
D8S277	2	149	193
D7S493	2	202	244
D8S284	2	270	314
D7S684	2	339	371
D8S270	3	102	128
D7S636	3	136	182
D8S550	3	186	226
D7S531	3	276	305
D8S285	3	310	338

panel 16

marker	dye	size range	
D10S217	1	95	129
D11S898	1	139	171
D10S548	1	181	207
D9S1826	1	210	238
D9S290	1	241	273
D9S1817	1	278	322
D9S158	1	328	362
D10S196	2	104	124
D9S1682	2	147	167
D11S908	2	170	196
D10S1693	2	200	236
D10S597	2	273	303
D9S283	3	89	125
D10S1651	3	205	237
D10S1652	3	268	304

panel 19

marker	dye	size range	
D13S158	1	116	144
D13S159	1	152	204
D13S173	1	232	262
D12S364	1	295	333
D13S265	2	88	136
D12S352	2	151	181
D12S326	2	205	241
D12S310	2	243	261
D13S153	3	89	131
D13S171	3	174	212
D12S324	3	233	265

panel 20

marker	dye	size range	
D14S292	1	80	110
D14S275	1	140	169
D14S258	1	193	223
D14S280	1	238	268
D14S70	2	98	124
D14S283	2	127	167
D14S63	2	175	203
D14S985	2	235	263
D14S74	2	296	330
D14S65	3	124	166
D14S288	3	190	225
D14S276	3	236	260
D14S261	3	271	313
D14S68	3	316	356

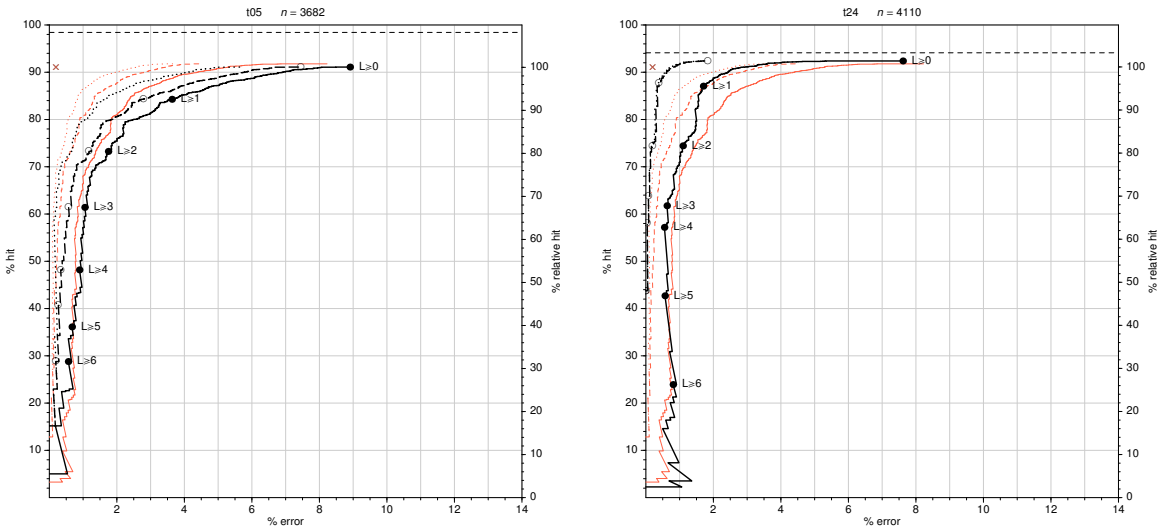
Appendix C

Complete Results

C.1 The Training Set

C.1.1 Panel-specific performance curves

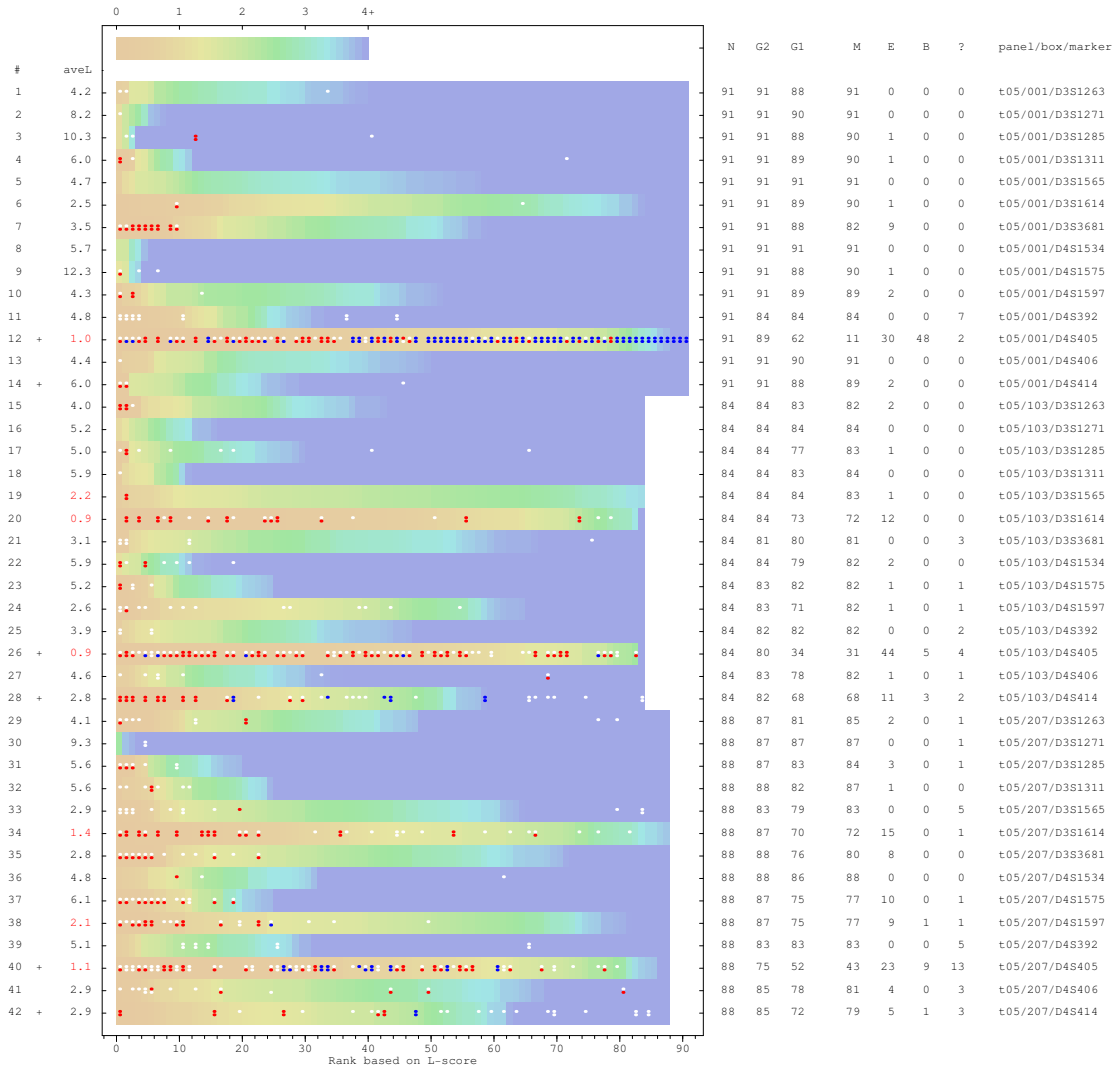
The performance curve for each panel are shown below. The figure description is the same with that of figure 4.11 (page 120). The red curves are for the whole training set.



C.1.2 Quality maps

The quality maps and the discrepancies for the training sets are shown below. The figure description is the same as that of figure 4.7 (page 115).

Panel 05 (training set)



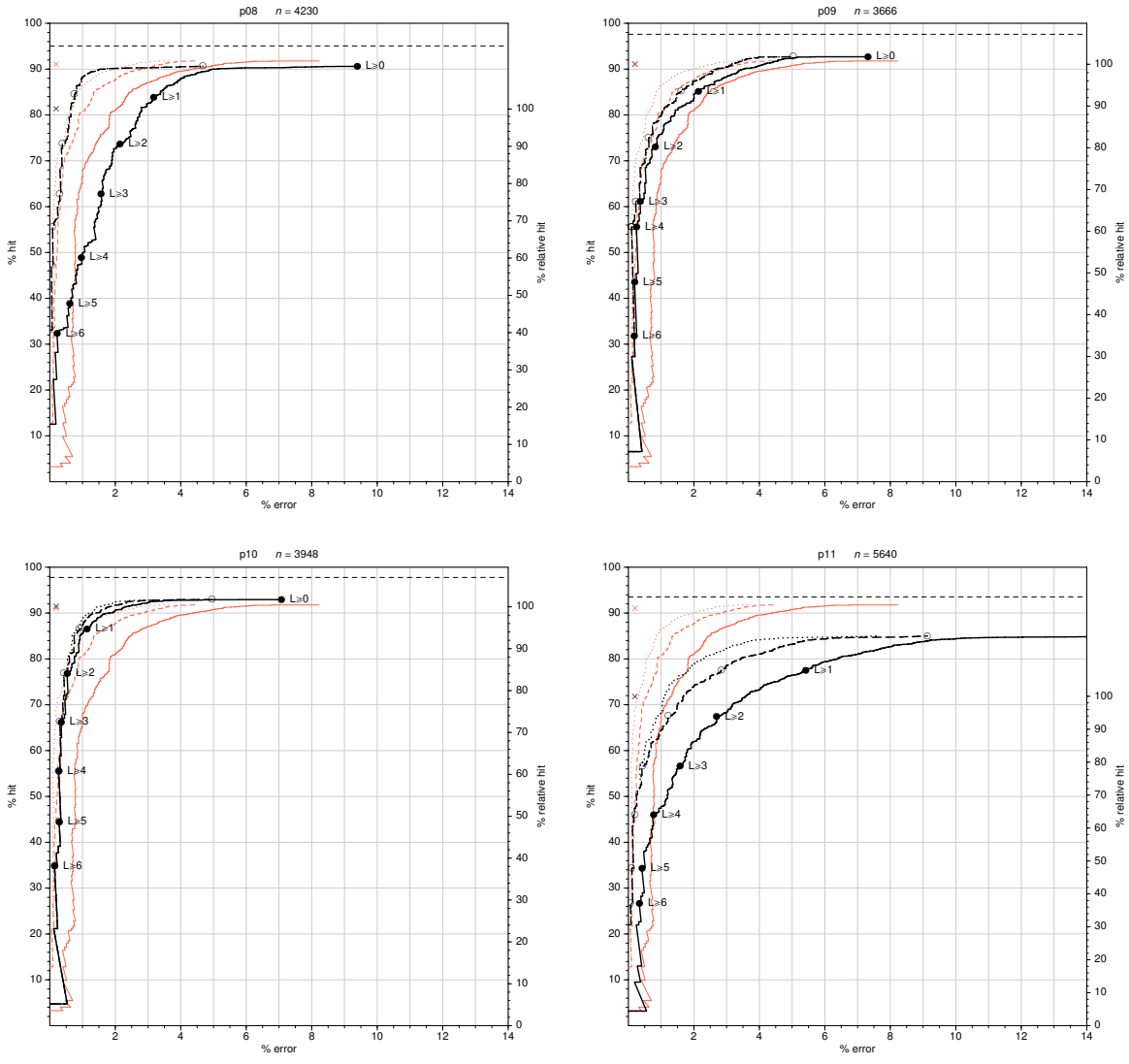
Panel 24 (training set)

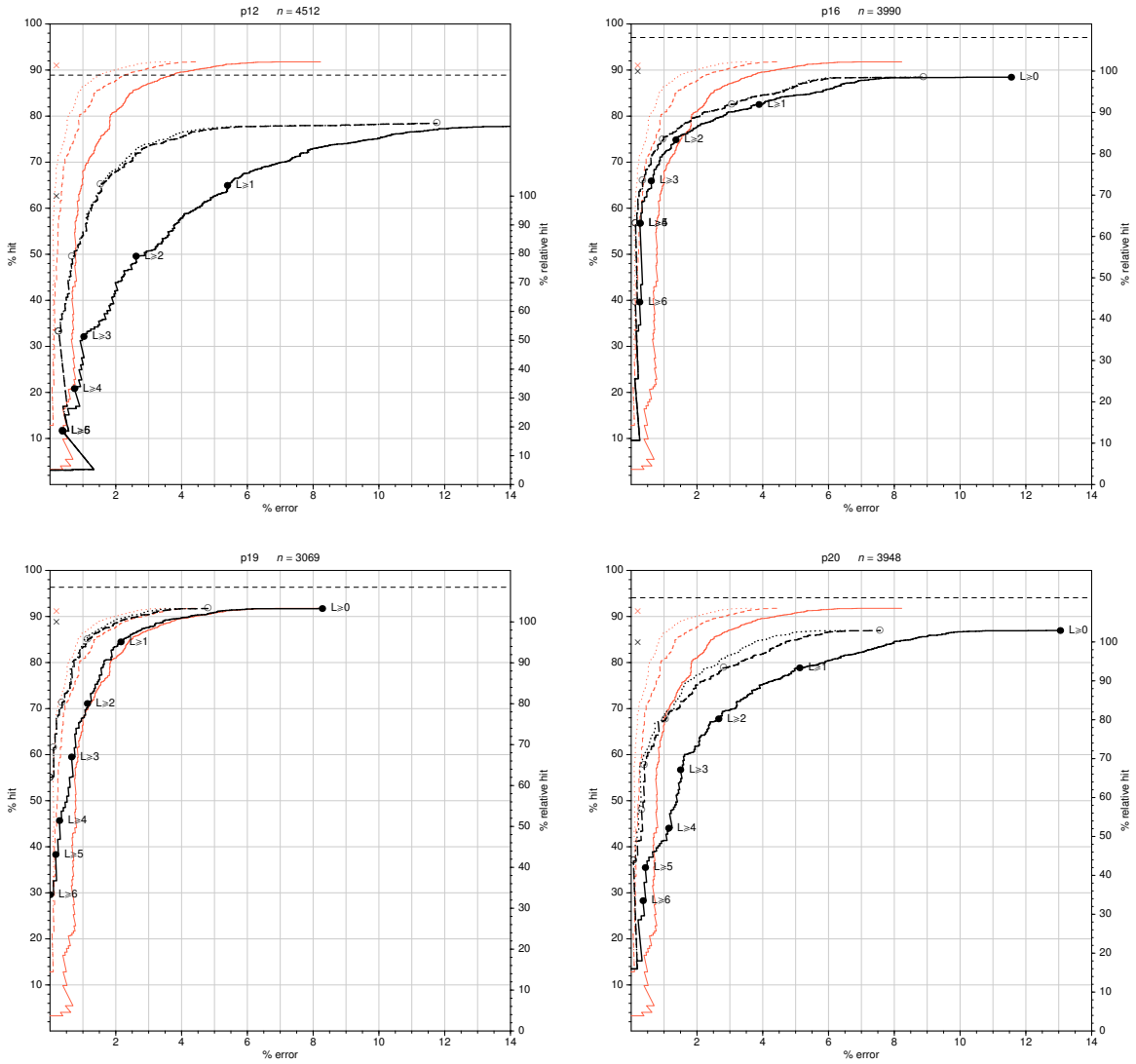
#		aveL	N	G2	G1	M	E	B	?	panel/box/marker
43		4.9	93	92	92	92	0	0	1	t24/001/D17S784
44		4.1	93	87	87	87	0	0	6	t24/001/D17S785
45	+	7.0	93	91	91	91	0	0	2	t24/001/D17S921
46		2.5	93	91	89	90	1	0	2	t24/001/D17S928
47		3.4	93	92	92	92	0	0	1	t24/001/D17S938
48		2.5	93	90	86	89	1	0	3	t24/001/D17S944
49		4.0	93	92	92	92	0	0	1	t24/001/D18S1161
50		6.1	93	92	92	92	0	0	1	t24/001/D18S452
51		2.2	93	92	89	92	0	0	1	t24/001/D18S464
52		4.5	93	91	85	90	1	0	2	t24/001/D18S474
53		4.2	93	92	91	88	4	0	1	t24/001/D18S53
54		7.6	93	92	92	92	0	0	1	t24/001/D18S59
55		1.4	93	89	87	88	1	0	4	t24/001/D18S63
56		2.0	93	88	86	87	1	0	5	t24/001/D18S64
57		4.4	93	91	91	91	0	0	2	t24/001/D18S68
58		4.8	90	81	78	78	3	0	9	t24/103/D17S784
59		5.2	90	81	77	79	2	0	9	t24/103/D17S785
60	+	6.9	90	80	75	79	1	0	10	t24/103/D17S921
61		3.1	90	81	78	80	1	0	9	t24/103/D17S928
62		3.8	90	80	77	76	4	0	10	t24/103/D17S938
63		3.3	90	80	78	79	1	0	10	t24/103/D17S944
64		4.7	90	80	79	79	1	0	10	t24/103/D18S1161
65		4.6	90	81	79	78	3	0	9	t24/103/D18S452
66		7.1	90	81	80	80	1	0	9	t24/103/D18S464
67		2.9	90	78	68	71	7	0	12	t24/103/D18S474
68		5.5	90	81	76	79	2	0	9	t24/103/D18S53
69		5.7	90	81	80	80	1	0	9	t24/103/D18S59
70		1.3	90	81	78	78	3	0	9	t24/103/D18S63
71		2.0	90	75	64	71	4	0	15	t24/103/D18S64
72		3.7	90	81	79	79	2	0	9	t24/103/D18S68
73		4.7	91	88	87	88	0	0	3	t24/207/D17S784
74		3.6	91	86	80	79	7	0	5	t24/207/D17S785
75	+	8.2	91	88	87	87	1	0	3	t24/207/D17S921
76		2.9	91	87	86	85	2	0	4	t24/207/D17S928
77		3.9	91	88	83	85	3	0	3	t24/207/D17S938
78		2.3	91	83	80	82	1	0	8	t24/207/D17S944
79		3.8	91	88	87	88	0	0	3	t24/207/D18S1161
80		5.7	91	87	84	87	0	0	4	t24/207/D18S452
81		3.2	91	88	85	87	1	0	3	t24/207/D18S464
82		4.7	91	86	84	85	1	0	5	t24/207/D18S474
83		4.8	91	88	83	87	1	0	3	t24/207/D18S53
84		5.2	91	88	88	88	0	0	3	t24/207/D18S59
85		2.0	91	87	83	84	3	0	4	t24/207/D18S63
86	+	2.0	91	85	76	79	3	3	6	t24/207/D18S64
87		4.1	91	87	84	87	0	0	4	t24/207/D18S68

C.2 The Test Set

C.2.1 Panel-specific performance curves

The performance curve for each panel are shown below. The figure description is the same with that of figure 4.11 (page 120). The red curves are for the training set.

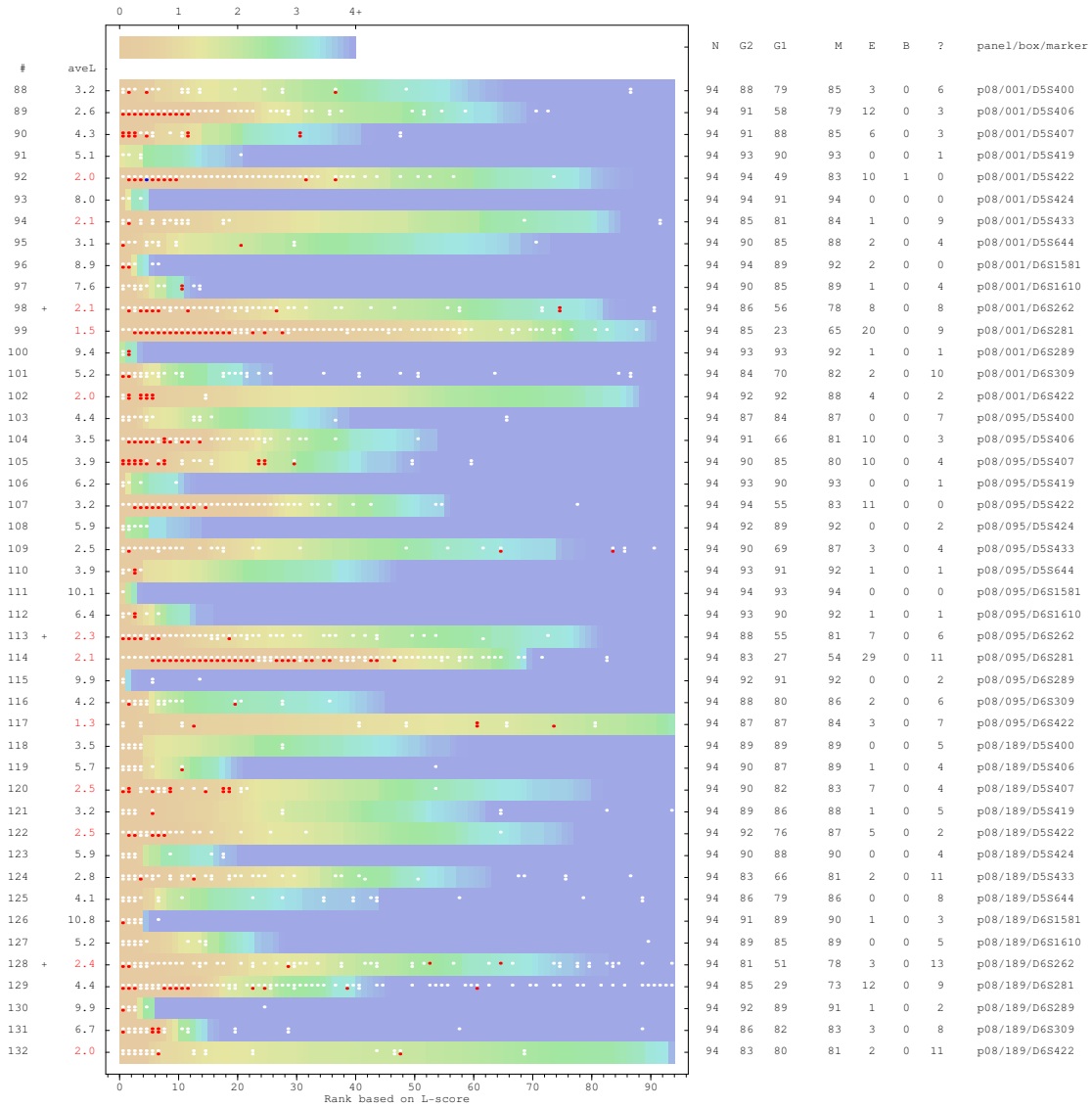




C.2.2 Quality maps

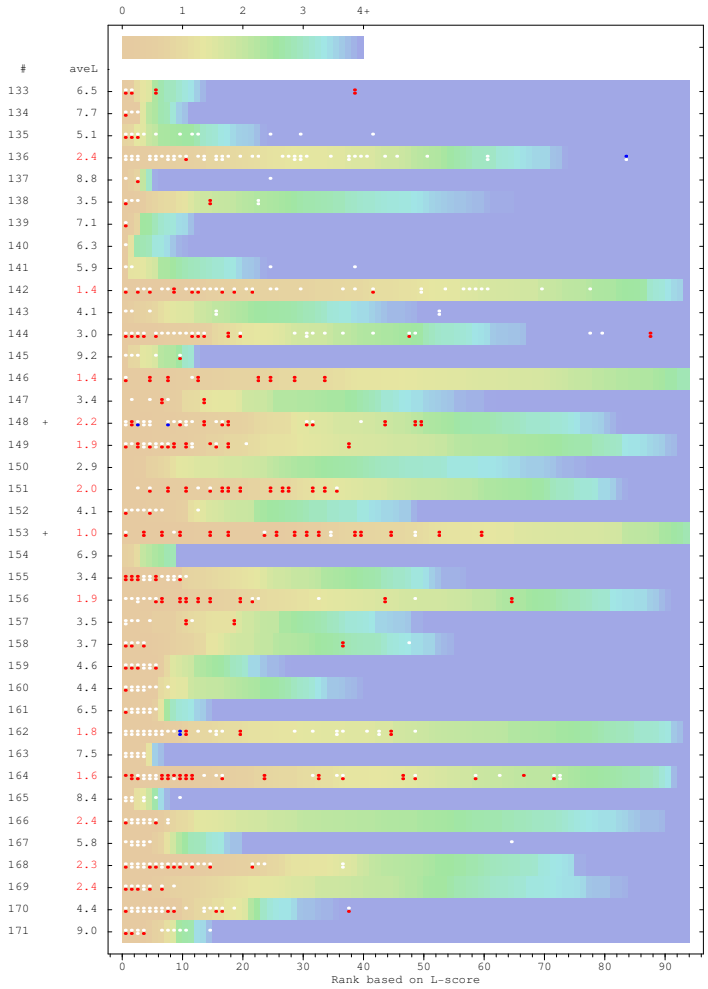
The quality maps and the discrepancies for the training sets are shown below. The figure description is the same as that of figure 4.7 (page 115).

Panel 08 (test set)



#		aveL	N	G2	G1	M	E	B	?	panel/box/marker
88		3.2	94	88	79	85	3	0	6	p08/001/D5S400
89		2.6	94	91	58	79	12	0	3	p08/001/D5S406
90		4.3	94	91	88	85	6	0	3	p08/001/D5S407
91		5.1	94	93	90	93	0	0	1	p08/001/D5S419
92		2.0	94	94	49	83	10	1	0	p08/001/D5S422
93		8.0	94	94	91	94	0	0	0	p08/001/D5S424
94		2.1	94	85	81	84	1	0	9	p08/001/D5S433
95		3.1	94	90	85	88	2	0	4	p08/001/D5S644
96		8.9	94	94	89	92	2	0	0	p08/001/D6S1581
97		7.6	94	90	85	89	1	0	4	p08/001/D6S1610
98	+	2.1	94	86	56	78	8	0	8	p08/001/D6S262
99		1.5	94	85	23	65	20	0	9	p08/001/D6S281
100		9.4	94	93	93	92	1	0	1	p08/001/D6S289
101		5.2	94	84	70	82	2	0	10	p08/001/D6S309
102		2.0	94	92	92	88	4	0	2	p08/001/D6S422
103		4.4	94	87	84	87	0	0	7	p08/095/D5S400
104		3.5	94	91	66	81	10	0	3	p08/095/D5S406
105		3.9	94	90	85	80	10	0	4	p08/095/D5S407
106		6.2	94	93	90	93	0	0	1	p08/095/D5S419
107		3.2	94	94	55	83	11	0	0	p08/095/D5S422
108		5.9	94	92	89	92	0	0	2	p08/095/D5S424
109		2.5	94	90	69	87	3	0	4	p08/095/D5S433
110		3.9	94	93	91	92	1	0	1	p08/095/D5S644
111		10.1	94	94	93	94	0	0	0	p08/095/D6S1581
112		6.4	94	93	90	92	1	0	1	p08/095/D6S1610
113	+	2.3	94	88	55	81	7	0	6	p08/095/D6S262
114		2.1	94	83	27	54	29	0	11	p08/095/D6S281
115		9.9	94	92	91	92	0	0	2	p08/095/D6S289
116		4.2	94	88	80	86	2	0	6	p08/095/D6S309
117		1.3	94	87	87	84	3	0	7	p08/095/D6S422
118		3.5	94	89	89	89	0	0	5	p08/189/D5S400
119		5.7	94	90	87	89	1	0	4	p08/189/D5S406
120		2.5	94	90	82	83	7	0	4	p08/189/D5S407
121		3.2	94	89	86	88	1	0	5	p08/189/D5S419
122		2.5	94	92	76	87	5	0	2	p08/189/D5S422
123		5.9	94	90	88	90	0	0	4	p08/189/D5S424
124		2.8	94	83	66	81	2	0	11	p08/189/D5S433
125		4.1	94	86	79	86	0	0	8	p08/189/D5S644
126		10.8	94	91	89	90	1	0	3	p08/189/D6S1581
127		5.2	94	89	85	89	0	0	5	p08/189/D6S1610
128	+	2.4	94	81	51	78	3	0	13	p08/189/D6S262
129		4.4	94	85	29	73	12	0	9	p08/189/D6S281
130		9.9	94	92	89	91	1	0	2	p08/189/D6S289
131		6.7	94	86	82	83	3	0	8	p08/189/D6S309
132		2.0	94	83	80	81	2	0	11	p08/189/D6S422

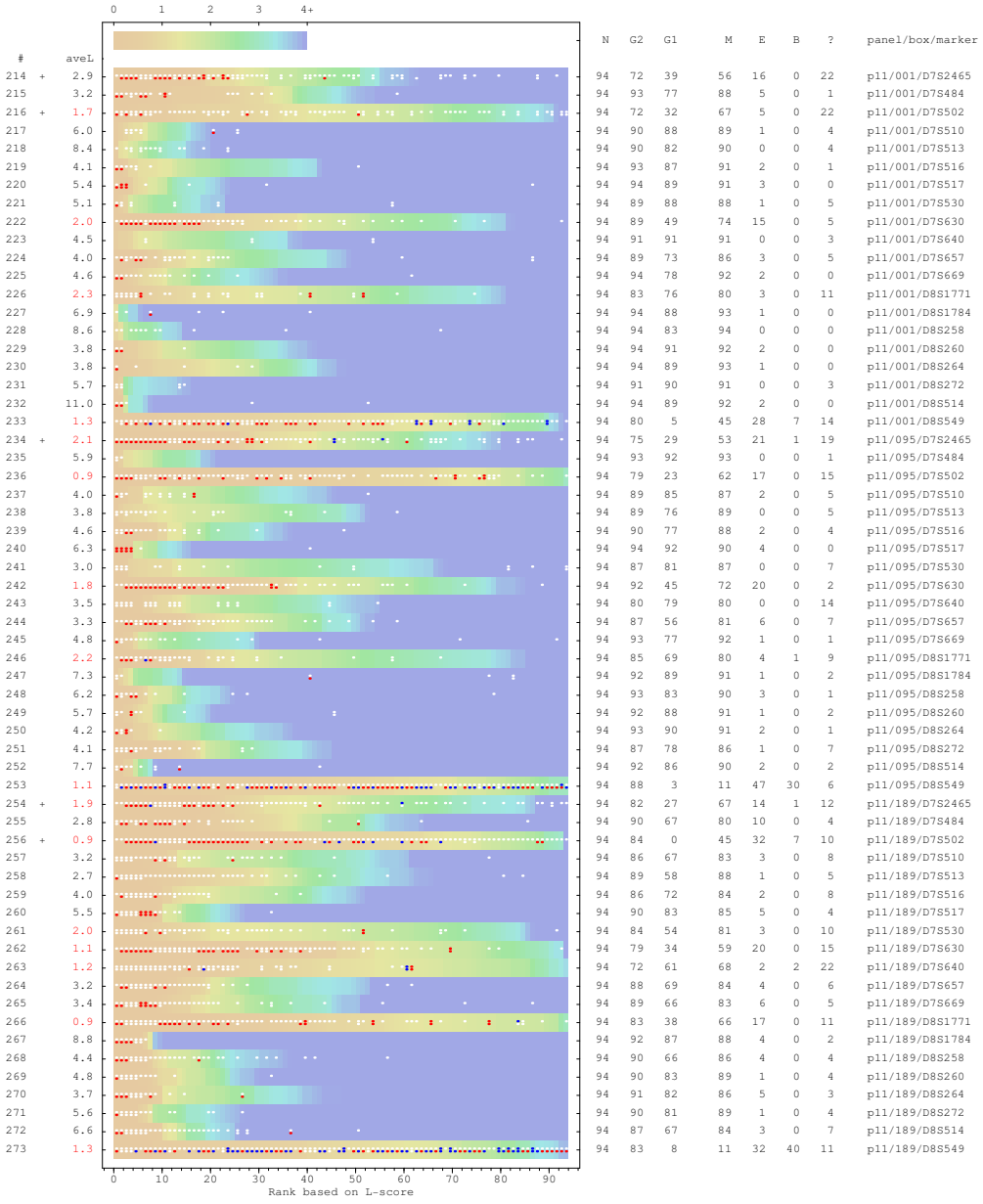
Panel 09 (test set)



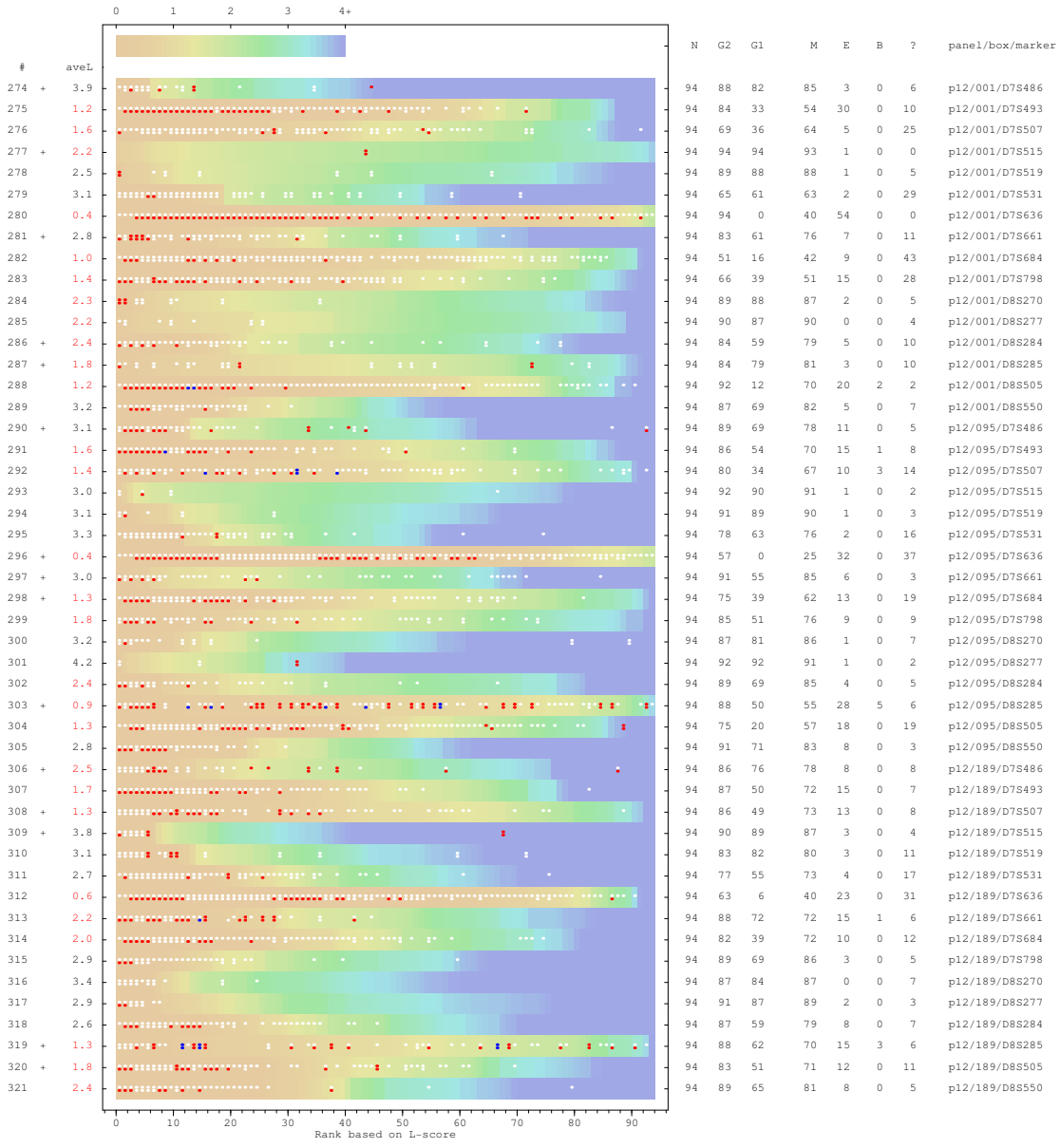
#		aveL	N	G2	G1	M	E	B	?	panel/box/marker
133		6.5	94	94	92	90	4	0	0	p09/001/D5S1981
134		7.7	94	94	91	93	1	0	0	p09/001/D5S408
135		5.1	94	94	83	91	3	0	0	p09/001/D5S426
136		2.4	94	79	62	78	1	0	15	p09/001/D5S641
137		8.8	94	94	91	93	1	0	0	p09/001/D6S1574
138		3.5	94	93	90	91	2	0	1	p09/001/D6S257
139		7.1	94	94	93	93	1	0	0	p09/001/D6S264
140		6.3	94	94	93	94	0	0	0	p09/001/D6S276
141		5.9	94	94	90	94	0	0	0	p09/001/D6S287
142		1.4	94	93	63	83	10	0	1	p09/001/D6S292
143		4.1	94	92	89	92	0	0	2	p09/001/D6S308
144		3.0	94	93	70	81	12	0	1	p09/001/D6S434
145		9.2	94	94	89	93	1	0	0	p09/001/D6S446
146		1.4	94	94	92	86	8	0	0	p09/095/D5S1981
147		3.4	94	94	91	92	2	0	0	p09/095/D5S408
148	+	2.2	94	90	80	78	10	2	4	p09/095/D5S426
149		1.9	94	91	84	81	10	0	3	p09/095/D5S641
150		2.9	94	94	94	94	0	0	0	p09/095/D6S1574
151		2.0	94	94	89	81	13	0	0	p09/095/D6S257
152		4.1	94	94	86	92	2	0	0	p09/095/D6S264
153	+	1.0	94	92	89	76	16	0	2	p09/095/D6S276
154		6.9	94	94	94	94	0	0	0	p09/095/D6S287
155		3.4	94	90	87	85	5	0	4	p09/095/D6S292
156		1.9	94	93	84	83	10	0	1	p09/095/D6S308
157		3.5	94	93	89	91	2	0	1	p09/095/D6S434
158		3.7	94	94	89	90	4	0	0	p09/095/D6S446
159		4.6	94	92	88	88	4	0	2	p09/189/D5S1981
160		4.4	94	91	87	90	1	0	3	p09/189/D5S408
161		6.5	94	90	88	89	1	0	4	p09/189/D5S426
162		1.8	94	84	74	80	3	1	10	p09/189/D5S641
163		7.5	94	90	90	90	0	0	4	p09/189/D6S1574
164		1.6	94	89	79	74	15	0	5	p09/189/D6S257
165		8.4	94	91	89	91	0	0	3	p09/189/D6S264
166		2.4	94	89	87	87	2	0	5	p09/189/D6S276
167		5.8	94	91	88	91	0	0	3	p09/189/D6S287
168		2.3	94	89	75	80	9	0	5	p09/189/D6S292
169		2.4	94	93	87	88	5	0	1	p09/189/D6S308
170		4.4	94	87	78	81	6	0	7	p09/189/D6S434
171		9.0	94	93	84	90	3	0	1	p09/189/D6S446

Panel 10 (test set)

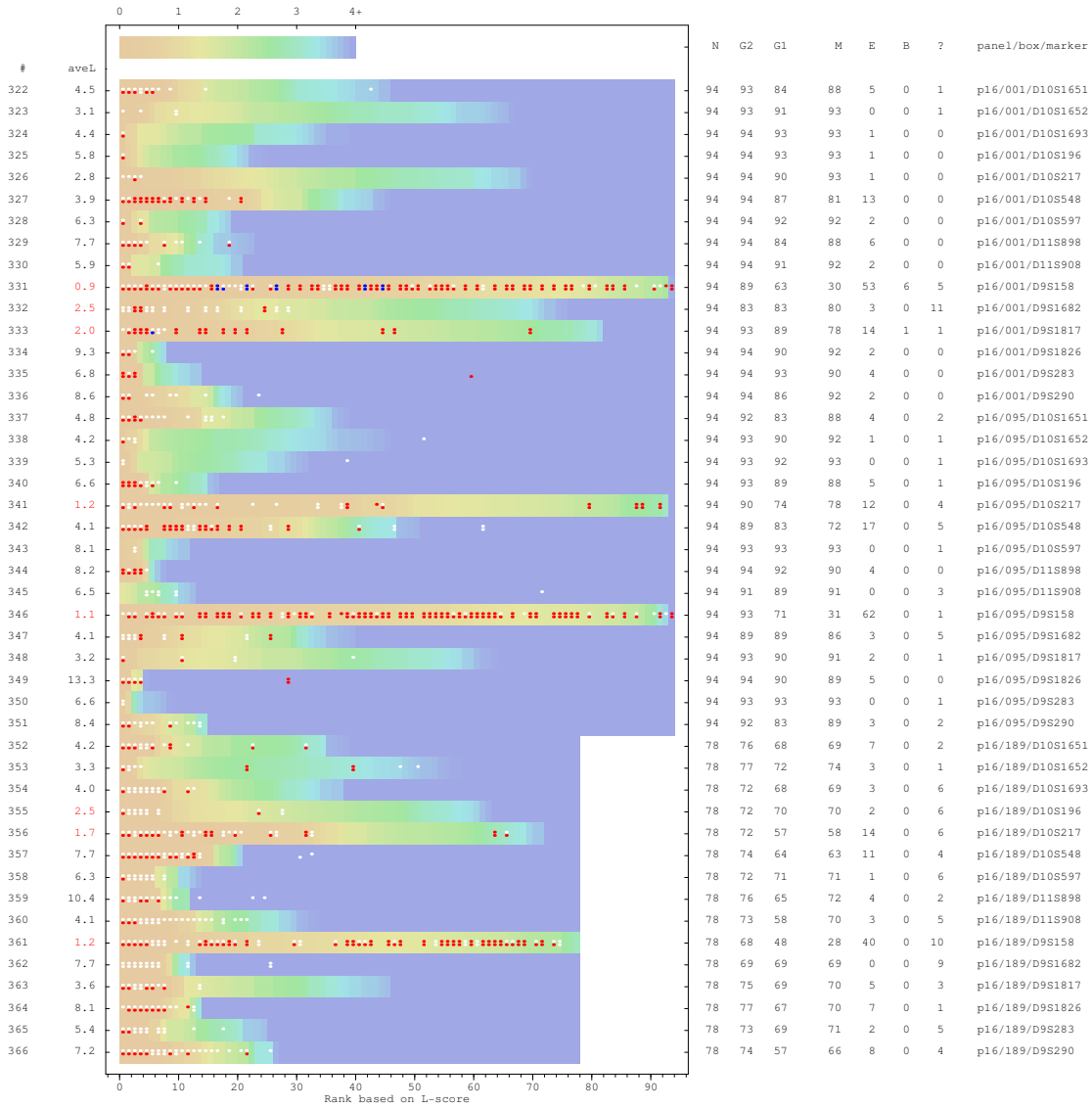
Panel 11 (test set)



Panel 12 (test set)

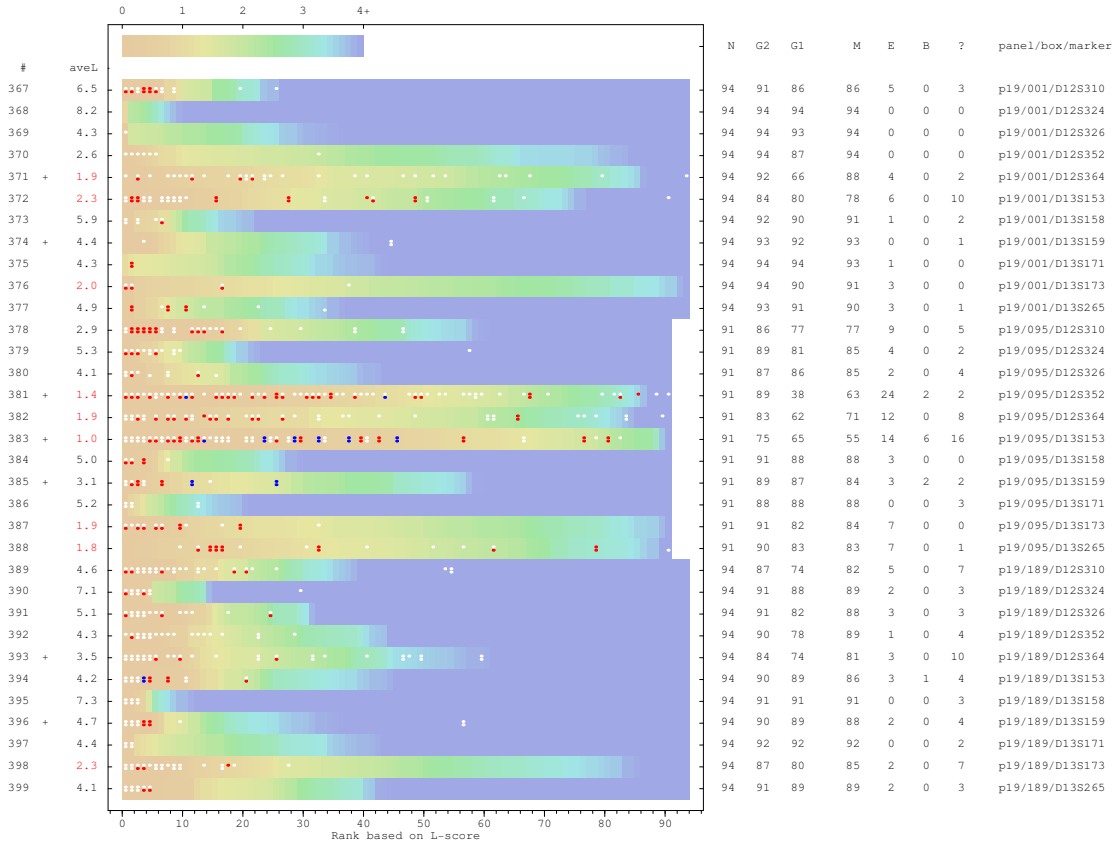


Panel 16 (test set)



#	aveL	N	G2	G1	M	E	B	?	panel/box/marker
322	4.5	94	93	84	88	5	0	1	p16/001/D10S1651
323	3.1	94	93	91	93	0	0	1	p16/001/D10S1652
324	4.4	94	94	93	93	1	0	0	p16/001/D10S1693
325	5.8	94	94	93	93	1	0	0	p16/001/D10S196
326	2.8	94	94	90	93	1	0	0	p16/001/D10S217
327	3.9	94	94	87	81	13	0	0	p16/001/D10S548
328	6.3	94	94	92	92	2	0	0	p16/001/D10S597
329	7.7	94	94	84	88	6	0	0	p16/001/D11S898
330	5.9	94	94	91	92	2	0	0	p16/001/D11S908
331	0.9	94	89	63	30	53	6	5	p16/001/D9S158
332	2.5	94	83	83	80	3	0	11	p16/001/D9S1682
333	2.0	94	93	89	78	14	1	1	p16/001/D9S1817
334	9.3	94	94	90	92	2	0	0	p16/001/D9S1826
335	6.8	94	94	93	90	4	0	0	p16/001/D9S283
336	8.6	94	94	86	92	2	0	0	p16/001/D9S290
337	4.8	94	92	83	88	4	0	2	p16/095/D10S1651
338	4.2	94	93	90	92	1	0	1	p16/095/D10S1652
339	5.3	94	93	92	93	0	0	1	p16/095/D10S1693
340	6.6	94	93	89	88	5	0	1	p16/095/D10S196
341	1.2	94	90	74	78	12	0	4	p16/095/D10S217
342	4.1	94	89	83	72	17	0	5	p16/095/D10S548
343	8.1	94	93	93	93	0	0	1	p16/095/D10S597
344	8.2	94	94	92	90	4	0	0	p16/095/D11S898
345	6.5	94	91	89	91	0	0	3	p16/095/D11S908
346	1.1	94	93	71	31	62	0	1	p16/095/D9S158
347	4.1	94	89	89	86	3	0	5	p16/095/D9S1682
348	3.2	94	93	90	91	2	0	1	p16/095/D9S1817
349	13.3	94	94	90	89	5	0	0	p16/095/D9S1826
350	6.6	94	93	93	93	0	0	1	p16/095/D9S283
351	8.4	94	92	83	89	3	0	2	p16/095/D9S290
352	4.2	78	76	68	69	7	0	2	p16/189/D10S1651
353	3.3	78	77	72	74	3	0	1	p16/189/D10S1652
354	4.0	78	72	68	69	3	0	6	p16/189/D10S1693
355	2.5	78	72	70	70	2	0	6	p16/189/D10S196
356	1.7	78	72	57	58	14	0	6	p16/189/D10S217
357	7.7	78	74	64	63	11	0	4	p16/189/D10S548
358	6.3	78	72	71	71	1	0	6	p16/189/D10S597
359	10.4	78	76	65	72	4	0	2	p16/189/D11S898
360	4.1	78	73	58	70	3	0	5	p16/189/D11S908
361	1.2	78	68	48	28	40	0	10	p16/189/D9S158
362	7.7	78	69	69	69	0	0	9	p16/189/D9S1682
363	3.6	78	75	69	70	5	0	3	p16/189/D9S1817
364	8.1	78	77	67	70	7	0	1	p16/189/D9S1826
365	5.4	78	73	69	71	2	0	5	p16/189/D9S283
366	7.2	78	74	57	66	8	0	4	p16/189/D9S290

Panel 19 (test set)



Panel 20 (test set)