Title
Modeling and Analysis of Oligonucleotide Microarray Data for Pathogen Detection

Author
McLoughlin, Kevin Shane

Publication Date
2013

Peer reviewed|Thesis/dissertation

Modeling and Analysis of Oligonucleotide Microarray Data for Pathogen Detection

by

Kevin Shane McLoughlin

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Biostatistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Terence P. Speed, Chair
Professor Sandrine Dudoit
Professor Lior Pachter

Spring 2013

Modeling and Analysis of Oligonucleotide Microarray Data for Pathogen Detection

Abstract

Modeling and Analysis of Oligonucleotide Microarray Data for Pathogen Detection

by

Kevin Shane McLoughlin

Doctor of Philosophy in Biostatistics

University of California, Berkeley

Professor Terence P. Speed, Chair

Microarrays have emerged during the past decade as a viable platform for detection of DNA from microorganisms in clinical and environmental samples. These microbial detection arrays occupy a middle ground between low cost, narrowly focused assays such as multiplex PCR and more expensive, broad-spectrum technologies like high-throughput sequencing. The Pathogen Bioinformatics Group at Lawrence Livermore National Laboratory is one of several teams that are actively working to develop arrays for clinical diagnostics, biologic product safety testing, environmental monitoring and biodefense applications.

Statistical algorithms that can analyze data from microbial detection arrays and provide easily interpretable results are absolutely required in order for these efforts to succeed. Several researchers have developed methods to determine what organisms are present in a microbial detection array sample. The algorithms developed so far operate mainly within a hypothesis testing framework, and are not motivated by a physical model of the process by which microbial DNA hybridizes to DNA probes on the array. Therefore, they only provide probabilities for the absolute presence or absence of an organism, and lack the ability to infer the abundances of the microbes in the sample. They also have limited capacity to handle samples containing complex mixtures of microorganisms.

This dissertation describes an approach to developing a quantitative algorithm for microbial detection array data analysis, capable of both identifying the organisms present in a sample and inferring their concentrations. After reviewing the most promising array designs and analysis algorithms that have been developed to date, I present a physical model for predicting probe signals on an array given a set of target organisms present in a sample and their concentrations. I describe the experiments that were performed to fit the key parameters in this model. Finally, I present an approach to solving the inverse problem, in which the probe signals are observed and used to infer the targets present and their concentrations.

To Susan, the source of all light in the universe.

Contents

Contents **ii**

List of Figures **iii**

List of Tables **vi**

1 Introduction: Application of microarrays to microbial detection **1**
1.1 Background: Microbial detection and discovery 1
1.2 Microbial detection microarrays 2
1.3 Analysis of detection arrays 7
1.4 Conclusion: the need for improved analysis methods 13

2 Predictive models of microbial detection array probe intensities **14**
2.1 Introduction and basic framework 14
2.2 Characteristics of microbial detection array data 18
2.3 Adjusting intensities for measurement process effects 21
2.4 Kinetics of probe hybridization 29
2.5 Equilibrium models of microarray hybridization 42

3 Target identification and quantitation from microbial detection array data **61**
3.1 Introduction 61
3.2 Estimation of scale factors using spike-ins and positive control probes 62
3.3 Quantitation of targets known to be present in a sample 71
3.4 Solving the general identification/quantitation problem 78
3.5 Conclusions and future directions 91

Bibliography **94**

List of Figures

1.1 Cartoon schematic of a microarray, showing single-stranded DNA oligo probes attached to substrate, with fluorescently labeled target DNA strands bound to selected oligos. Image credit: Sabrina Fletcher, LLNL. 3

2.1 Log intensity distributions for four classes of probes on an LLMDA array for respiratory syncytial virus 20
2.2 Quantile-quantile plot of log intensities for nonspecific and negative control probes on the RSV array 21
2.3 Nonspecific probe log intensity distribution from RSV array, with density estimates plotted on log scale to show power law behavior in tails 22
2.4 Intensity distributions for probes against six organisms present in different concentrations, in two scans of the same array, at 50% gain setting (left) and 20% gain (right). Density curves are colored according to the organism targeted by the probes. 23
2.5 Section of an array image from NimbleScan, showing scan bleedover and checkerboard arrangement of features 26
2.6 Scatter plot of pairwise comparisons of probe intensities from scans of the same array at three different PMT settings 28
2.7 Loss W for simulated data for two probes having same k_i^a and different k_i^d values, as function of k_i^a and k_i^d. 33
2.8 Rate constants k_i^a and derived variables (affinity constants and relaxation times) fitted to simulated intensity data, compared to the input values used to generate the data 35
2.9 Loss as function of k_i^d at input k_i^a, for two simulated probes; input and fitted k_i^d values are plotted as vertical lines. 36
2.10 Loss function contributions from one array for two simulated probes, as contours for combinations of k_i^a and k_i^d, and as function of k_i^d for the input k_i^a. 37
2.11 Scatter plot of $\log k_i^a$ and $\log k_i^d$ values, fitted by penalized least squares to real data 39
2.12 Penalized fit values for $\log k_i^a$ vs simulated inputs 40
2.13 Penalized fit values for $\log k_i^d$ vs simulated inputs 41

2.14 Left: Distribution of τ_{ij} computed from rate constants fit by pLS, assuming 4 μg DNA in sample. Right: Distributions of equilibrium intensity fractions with 4 μg DNA, at each time point. 43
2.15 Distribution of affinities fit by penalized least squares 44
2.16 Example nearest-neighbor model free energy calculation for a short probe-target DNA duplex 45
2.17 Effect of single mismatches on probe intensities as a function of position 46
2.18 Layout of tiling array probes relative to target genomes 47
2.19 Comparison of fitted g_d free energy parameters by dimer pair, using QR decomposition with pivoting vs minimum norm solution 51
2.20 Comparison of g_d free energy parameters fitted using QR and minimum norm strategies to SantaLucia solution-phase parameters 51
2.21 Fitted additive position effect function for log ratio data, using a B-spline basis with 13 knots 53
2.22 Smoothed scatterplot of ratio of inferred to predicted free energy changes from log ratio data, and penalized spline fit for multiplicative position effect with 13 knots 54
2.23 Fitted multiplicative position effect function for free energy changes, as in Figure 2.22, scaled to show variation with position 54
2.24 Predicted log intensities plotted against observed values, and residuals vs predicted intensities, based on initial ΔG parameters estimated from log ratio data 55
2.25 Residual sum of squares as a function of iteration count, for fitting perfect match dimer pair free energies to log intensity data. 56
2.26 Refitted ΔG parameters for PM dimer pairs, plotted against initial values estimated from log ratios 56
2.27 Predicted vs observed values, and residuals vs predicted values, of log intensities calculated using ΔG parameters fitted to the full PDNN model. 57
2.28 Smoothed scatterplots of background corrected intensities from two channels on each tiling array 58
2.29 Smooth scatterplot of background corrected intensities from same sample strain in replicate pairs of tiling arrays 60

3.1 Normal quantile-quantile plot of replicate probe log intensity deviations on a typical LLMDA version 5 array, showing heavy tails of distribution 64
3.2 Affinities fitted from simulated data vs input affinities used to generate the data, plotted on log scale. Plot symbol shape and color indicates the number of mismatches between the probe and target sequences. 65
3.3 Scale factors fitted from simulated data vs input scale factors used to generate data 66
3.4 Ratios of affinities fitted from simulated data to input affinities, as a function of the input affinities, for two simulated datasets 67
3.5 Median ratio of fitted to input affinities from simulated data, as a function of the input K_0 value, for 20 simulated datasets 68

3.6 Fitted vs observed intensities at concentration 16 pM for two simulated datasets 68
3.7 Top: Log affinities for *Thermotoga* probes fitted to real data, as a function of predicted free energy. Bottom: Differences between log affinities fitted to real data and predicted free energy / RT, as a function of fitted log affinity. Plot symbol and color indicates the number of mismatches between the probe and target sequences. 70
3.8 Fitted and predicted affinities for six perfect match *Thermotoga* probes and the first seven mismatch probes derived from them, as a function of the number of mismatches added. 71
3.9 Observed log intensities for positive control probes vs predicted log intensities and free energies, for an example array with *Thermotoga* DNA spiked in at 16 pM concentration. Red points in the right hand panel represent the predicted log intensity values. 72
3.10 Scale factors fitted from simulated data vs input scale factors used to generate data. Plot symbols indicate the concentration of *Thermotoga* DNA spiked into the sample to use as a calibration reference. 75
3.11 Target concentrations fitted from simulated data vs input concentrations used to generate data 76
3.12 Target concentrations fitted from simulated data vs input concentrations used to generate data, when Gaussian noise (with SD 1.0) was added to the log affinities used for the simulation 77
3.13 Concentrations fitted to intensities from 12 arrays from Latin square experiment vs actual concentrations of each target on the respective arrays. Values from replicate arrays are plotted with different colors and symbols. 79
3.14 Fitted concentrations for candidate targets from Latin square experiment data, plotted against actual concentrations for the true targets that were present from each family. Separate concentrations were fitted for the genome elements of Ba (chromosome and 2 plasmids) and Bt (two chromosomes). 84
3.15 Kernel density plots of fitted $\log_2$ concentration distributions for decoy targets in selected families not represented in the Latin square experiment samples; plus overall distribution for all 13 unrepresented families. Axis labels indicate untransformed concentrations; scale varies between plots. 86
3.16 Mean binding rates over 581 LLMDA version 5 arrays for probes containing G homopolymers, as a function of polymer length. 87
3.17 Comparison of affinities K_{ij} for probes matching *B. anthracis* str. Ames (target Ba) and decoy strains Sterne and A0174. 88
3.18 Fitted mixing proportions ϕ_{ij} for probes against targets Bt (top) Ft (middle), and Ba (bottom) on array 7 in the Latin square data set. Each bar represents the ϕ_{ij} values for one probe, with segments colored by target: dark blue for the correct target, cyan for the "unbound" contribution, and shades between yellow and red for decoy targets in the same family as the true target. 89

3.19 L_1 penalized likelihood fits of candidate target concentrations to intensities on array 7 in Latin square data set, plotted against the penalty parameter α. Correct targets are indicated by solid lines, decoys by dashed lines. A horizontal dashed line indicates the actual concentration of the true target. 92

List of Tables

2.1 Fitted scanner offset and PMT scaling values for 15 arrays 29
2.2 Covariates for *E. faecalis* hybridization experiments 30
2.3 Residual sum of squares, variance and SE from the tiling array data, using a succession of models of increasing complexity 57
2.4 Within-array variances for perfect match probes against both strains on array . 59
2.5 Between-array variances for perfect match probes 59

3.1 Layout for quantitation test experiment, showing concentrations of five known targets in each sample. Each sample was run on two replicate arrays. 73
3.2 Root mean square deviation of concentrations fitted to 25 simulated array data sets generated using 9 different concentrations of spiked in *Thermotoga* DNA. Concentrations were fitted using either the input affinities or with noise added to the log affinities, to simulate the effect of errors in the affinity estimates. 74
3.3 Actual (bold) and fitted concentrations for top targets in each bacterial or viral family represented in Latin square dataset, for one replicate array hybridized to each of samples 2 through 6. 85

Acknowledgments

I performed the research described here as a member of the Pathogen Bioinformatics Group at Lawrence Livermore National Laboratory, headed by Tom Slezak. I want to thank Tom and the other members of the team for their support and encouragement and for their fabulous contributions to the success of the Lawrence Livermore Microbial Detection Array. Tom has been a shameless promoter of my work, with a special talent for hiring great people and keeping them motivated. Shea Gardner designed the probes for multiple generations of the LLMDA, sparked ideas through many late afternoon discussions, and rekindled my enthusiasm on many a day when my fire was dwindling. Crystal Jaing ran the microarray laboratory, rounded up collaborators to provide us with interesting samples to analyze, kept us focused on manuscripts and grant proposals, and with Tom helped maintain the funding pipeline. All of them have been great friends and traveling companions. James Thissen and Nicholas Be performed most of the recent experiments, with utmost care and attention to detail.

I'd also like to thank Dave Nelson for demonstrating that one could get a Ph.D. this late in life; Pauline Gu and Ed Elhauge for co-developing the TriTool analysis software; Marisa Lam Torres, Clinton Torres, Mark Wagner and others for the KPATH database; Michelle Alegria-Hartmann and Chitra Manohar for initial LLMDA experiments; Nisha Mulakken for some of the LLMDA analysis; Peter Williams for help with Livermore Computing; Henrik Bengtsson for his explanation of scanner biases; Wenyi Wang for discussions on SNP detection; and Amy Rasley, Sahar El-Etr, Holly Franz, Jonathan Allen, Ann Loraine, Wenjing Zheng, Jennifer Weller, Susan, Sara, Joan, everyone else previously mentioned and the countless others I forgot to mention for inspiration and encouragement.

Some of our many collaborators outside LLNL who were especially helpful with LLMDA research include Lena Erlandsson and Maiken Rosenstierne from the Statens Serum Institut in Copenhagen; Eric Delwart and Joe Victoria from Blood Systems Research Institute; Arifa Khan from the FDA; and Hendrik Poinar from McMaster University.

Needless to say, this entire enterprise would have been impossible without the help of Terry, Sandrine, and Lior, to whom I'm eternally grateful.

The LLMDA research was supported by funding from the Laboratory Directed Research and Development (LDRD) Program at LLNL, the US Department of Homeland Security, the Defense Threat Reduction Agency, and the Food and Drug Administration.

Chapter 1

Introduction: Application of microarrays to microbial detection

This dissertation describes statistical models and algorithms I developed to analyze data from DNA microarrays for the purpose of microbial detection. In this chapter, I aim to provide the background material required to understand the goals and challenges of analyzing data from these arrays, and to summarize previous work on microbial detection array design and analysis. I will start by discussing the main technologies used to detect nucleic acids from bacteria, viruses and other microbes. I will then present an overview of microarray technology, and describe the approaches to array design used by several teams, including my colleagues at Lawrence Livermore National Laboratory (LLNL). I will compare the merits of these design approaches, focusing on those that are relevant for understanding the analysis problems. Finally, I will discuss the methods developed by other researchers to analyze detection arrays, and present the case for developing better methods.

1.1 Background: Microbial detection and discovery

Infectious diseases pose a growing threat to public health, due to increased rates of population growth, international trade and air travel, climate change, bacterial antibiotic resistance, and a wide range of other factors. In addition, global conflicts over the past decade have raised concerns that pathogenic agents might be released deliberately by terrorist organizations or other entities. Public and private funding agencies have responded to these concerns by investing heavily in the development of new assays for microbial surveillance and discovery. The majority of these new methods involve direct detection of microbial nucleic acids. Ideally, these methods should be effective both as *detection* assays (for identification of known microbes) and as *discovery* techniques (for revealing the presence of novel, previously uncharacterized organisms).

Most currently available methods for microbial detection and discovery using nucleic acid samples are based on one of three technologies. In order of increasing cost, these are

the polymerase chain reaction (PCR) [Mullis 86], oligonucleotide microarrays [Schena 95], and DNA sequencing [Sanger 77]. These platforms have different strengths and weaknesses. While sequencing provides the most in-depth, unbiased information, and is able to reveal completely novel organisms, it is at present still too costly and time-consuming for routine use - particularly when the resources required for data processing and analysis are taken into account. Although multiplex sequencing of bar-coded samples reduces the cost per sample, it also decreases the coverage and thus the sensitivity of the analysis; this may be an issue when the organism of interest has low abundance and the sample has not been treated beforehand to remove host and/or background DNA.

At the other end of the cost spectrum, PCR assays are very fast and sensitive, but have limited capacity for multiplexing [Bej 90, Vandenvelde 90]. When an assay is required to test for the presence of several organisms simultaneously, many PCR reactions may be needed, erasing any cost benefit. They are also highly specific; this is an advantage for detecting a microbe whose sequence is precisely known, but a great disadvantage for discovery of novel species, or for detecting variant strains of a known species.

Microarrays occupy a middle ground with respect to cost, processing time, sensitivity, specificity, and ability to detect novel organisms. The high-density arrays available today are able to test for the presence of thousands of different organisms simultaneously, at a cost less than US$100 per sample. Arrays can be designed with a combination of high-specificity probes and probes designed against conserved regions, so that they can be used in both detection and discovery modes. While most array designs select probes from fully sequenced genomes in GenBank and other databases, cross-hybridization between probes and similar but non-identical sequences allows detection of novel species, provided that they are closely related to those that were used for probe design. A limitation of microarrays is that, except for so-called universal arrays, probe designs must be updated periodically to include the ever-increasing number of microbial genome sequences being added to GenBank. Nevertheless, for many applications, microarrays offer an ideal balance of capabilities for broad-spectrum microbial surveillance.

1.2 Microbial detection microarrays

1.2.1 Microarray overview

A microarray is a miniaturized device containing short (25- to 70-mer) single stranded DNA oligonucleotide probes (or "oligos") attached to a solid substrate, as shown in Figure 1.1. The probes are designed to have sequences complementary to segments of one or more target organism genomes. Oligos may be spotted onto the array by mechanical deposition [Schena 95], or synthesized *in situ* either by spraying nucleotides from an inkjet printer head [Hughes 01] or through a series of photocatalyzed reactions [Pease 94]. Probes are arranged in a rectangular grid in which each spot or "feature" contains 10^5 to 10^7 copies of the same oligo. The density of features on the array varies between platforms, from 20,000 spots per

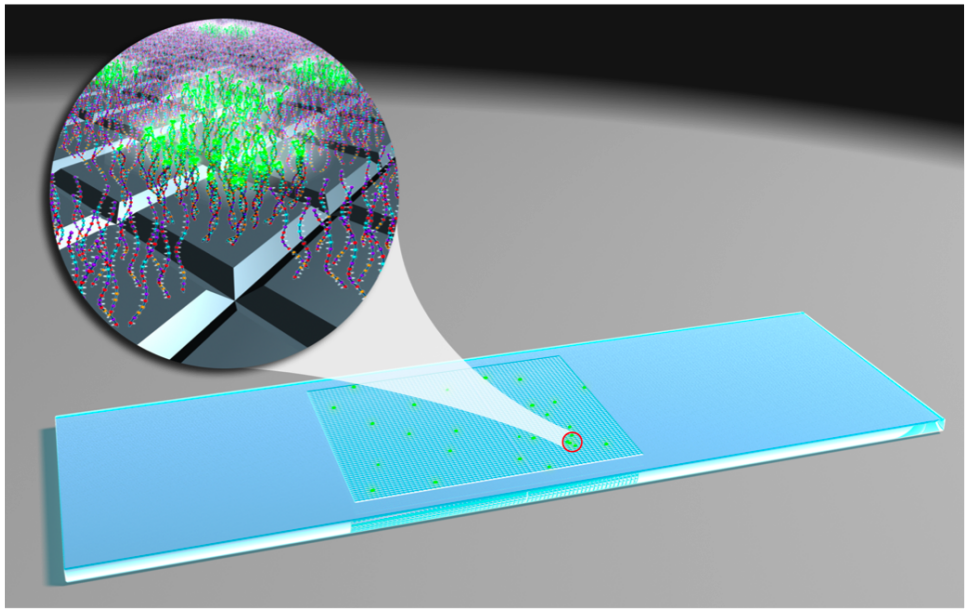

Figure 1.1: Cartoon schematic of a microarray, showing single-stranded DNA oligo probes attached to substrate, with fluorescently labeled target DNA strands bound to selected oligos. Image credit: Sabrina Fletcher, LLNL.

slide for a typical spotted array, to several million for platforms such as NimbleGen and Affymetrix that use photocatalytically synthesized oligos. Arrays may be subdivided with a gasket into subarrays, allowing multiple samples to be tested on one slide. Replicate features, scattered randomly across the array, may be used to allow correction for scratches and other localized artifacts. On some arrays, negative control probes with random sequences are included, to provide threshold intensity measurements for background noise correction.

To perform a microbial detection array analysis, total nucleic acids are first extracted from the sample. If the targets of interest include viruses with RNA genomes, a reverse transcription step is used to convert viral RNA to cDNA. The DNA is amplified if its initial concentration is too low; the DNA strands are then fragmented and fluorescently labeled. The labeled DNA is incubated on the array surface for several hours, allowing enough time for the DNA fragments to hybridize to complementary or nearly complementary probes, if they exist on the array. The array is then washed to remove unbound DNA and scanned to produce a file of fluorescence intensities for each feature. In the resulting image, bright features will correspond to probes that are complementary to the DNA in the sample.

1.2.2 Microbial detection array designs

Several groups have applied microarray technology to microbial detection. Their approaches may be distinguished according to the range of organisms targeted, the probe design strategy,

and the array platform used. Each group has also developed analysis algorithms targeting its own array platform, which I will discuss in Section 1.3.

ViroChip

The first microarray designed for detection of a wide range of microbes was the ViroChip [Wang 02]. The initial version of the ViroChip contained 1,600 probes derived from the 140 complete viral genomes available in GenBank when the array was designed. Later versions of the array were developed to cover a wider range of viruses as additional genomes were published [Wang 03]. The most recent version covers all viruses that had been sequenced through December 2010, and contains over 60,000 probes. Early versions of the ViroChip were fabricated by mechanically spotting synthesized oligonucleotides on a glass slide; the more recent versions are produced using Agilent inkjet technology [Chen 11]. The oligos are 70-mers, usually selected to match sequences common to a taxonomic family, but not found in other families. For some families, oligos were instead selected at the genus level. Since the probes were designed against conserved sequences, the ViroChip can be used to identify novel viruses within the same family as a known, sequenced virus. This capability was used to characterize the virus responsible for the 2003 SARS outbreak as a novel coronavirus [Ksiazek 03].

The advantages of the ViroChip platform include its ability to detect novel viruses within a known family. Its disadvantages are its lack of coverage of bacteria and other microbes, and the relatively small number of probes covering each virus.

Resequencing pathogen microarrays

Another approach to pathogen detection uses resequencing microarrays [Lin 06, Malanoski 06]. These arrays contain short probes (25- or 29-mers) tiled along selected genes of the target pathogen species. Four probes are designed for each location in a target gene: one with a perfect match base at the central position of the probe, and one for each of the 3 alternative bases. Hybridization and analysis of these arrays yields a sequence for each target gene homolog present in the sample; the sequence is then matched to a species and strain by comparison against a sequence database, using BLAST [Altschul 90].

The prototype Respiratory Pathogen Microarray (RPM v1.1) was manufactured as a custom Affymetrix array. It contained probes for several common human respiratory viruses and bacteria, including influenza, adenovirus, coronavirus, and rhinoviruses, together with bacteria such as *Bordetella pertussis* and *Streptococcus pneumoniae*. More recent designs based on this approach, such as the resequencing pathogen microarray for tropical and emerging infectious agents (RPM-TEI v1.0) [Leski 09], contain probes for a wider variety of pathogens and known toxin genes, focusing on the category A, B and C select agents defined by the US Centers for Disease Control (CDC).

The use of short oligonucleotide probes, with a large number of oligos per target gene, gives the RPM arrays very high specificity for strain-level identification of target organisms.

The disadvantages of the RPM approach are its lower sensitivity (due in part to the use of short oligos), the limited range of organisms that can be covered on a single array (because of the large number of probes required for each target), and its lack of ability to detect novel organisms.

Universal detection array

A sequence-independent "universal" microarray was described in [Belosludtsev 04]. Rather than selecting oligos from sequenced microbial genomes, the authors created an array containing 14,283 unique 12-mer and 13-mer probes with randomly generated sequences. Probes were excluded from the array design if they differed from previously selected probes at fewer than four positions (for 12-mers) or five (for 13-mers). The authors further refined the probe list by building prototype arrays, and excluding from further analysis probes that did not give reproducible signals when data from replicate hybridizations were compared. Oligos were synthesized on arrays using a photocatalytic process, similar to that used on the NimbleGen platform. Hybridizations were performed at low temperatures (23° C) to compensate for the extremely short probe length.

By hybridizing genomic DNA from several bacterial species, the authors demonstrated that their array produced reproducible patterns of probe intensities (or "signatures") that distinguished between one species and another. Probe intensities were not correlated either with occurrences of the oligo sequence in the target bacterial genome or with predicted free energies of hybridization. Thus, identification of an unknown target using this array relies on comparison of the observed intensity pattern to a compendium of signatures acquired by hybridizing known targets to the array. The unique advantage of the universal array approach is that the probes are not specific to genomes that have already been sequenced, so that the array design does not need to be updated as new genomes become available. The principal disadvantage is the lack of any means of predicting signatures for organisms of known sequence; these must be obtained experimentally for every species of interest. The large number of experiments required makes this approach impractical for broad-spectrum detection, especially for agents that must be handled using Biosafety Level 3 (BSL-3) and BSL-4 procedures.

GreeneChip

The "GreeneChip" arrays represent a broader-spectrum approach to microbial detection [Palacios 07, Quan 07]. These are high-density oligonucleotide arrays, fabricated using the Agilent inkjet system. GreeneChipVr version 1.0 contains 9,477 probes for viruses infecting vertebrates. GreeneChipPm v1.0 is a panmicrobial array design, containing all of the GreeneChipVr probes, together with probes for several thousand pathogenic bacteria, fungi and protozoa, comprising a total of 29,495 60-mer oligos. Viral probes were designed to target a minimum of three genomic regions for each family or genus of virus. Typically, one highly conserved region was chosen, along with two or more variable regions. Probe

sequences were selected so that every vertebrate virus in the ICTV database (International Committee on Taxonomy of Viruses) or in GenBank was represented by at least one probe with five or fewer mismatches. Bacterial, fungal and protozoan probes were selected by a similar strategy, except that the target sequences were only chosen from the 16S ribosomal RNA (rRNA) genes of bacteria and 18S rRNAs of fungi and protozoa.

When they were tested with virus-infected cell cultures and clinical samples from virally infected patients, the GreeneChip arrays correctly identified the virus at the species level. Performance with bacterial samples was poorer, due to the choice of 16S rRNA as the target gene; probes for these targets tended to cross-hybridize across taxa, so that some bacteria could only be identified at family or class resolution. The sensitivity of these arrays was comparable to that of the ViroChip series, due to the use of long (60-mer) oligos.

Lawrence Livermore Microbial Detection Array

The most comprehensive microbial detection arrays reported to date were designed by our team at Lawrence Livermore National Laboratory [Gardner 10, Jaing 08]. Initial versions of the Lawrence Livermore Microbial Detection Array (LLMDA) contained target probes for all bacteria and viruses (pathogenic and otherwise) for which full genome sequences were available. More recent versions also include probes for pathogenic fungi and protozoa. Probe lengths on a single array vary between 50 and 65 nucleotides (nt) and are adjusted so that all probes on the array have roughly equivalent affinities for their complementary target DNA molecules.

As on the GreeneChip, probes are selected from target genomes by one of two strategies. "Discovery" probes match genome regions that are unique to a taxonomic family or subfamily, but are shared by the species within that group. By targeting sequences that evolve more slowly within families, the discovery probes are optimized for detection of novel species within a known family. "Census" probes target highly variable regions that are unique to an individual species or strain. They are optimized for forensic use, to identify the specific strain of organism in a sample as precisely as possible.

The LLMDA designs were originally deployed using NimbleGen technology and are currently being migrated to the Agilent platform. Similar to Affymetrix, NimbleGen uses a photocatalytic process to synthesize oligos *in situ* on the array surface; unlike Affymetrix, it employs a digital micromirror device, rather than a set of photolithographic masks, to produce the light pattern that catalyzes the addition of nucleotides within each feature. Compared to spotted oligo or Agilent inkjet arrays, NimbleGen achieves higher probe densities, with up to 2.1 million features per array. Even at the lower densities offered by Agilent, the LLMDA has the capacity to target each sequenced microbial genome using between 10 and 50 or more distinct probes, depending on the array format and the types of microbes targeted. The probes targeting each genome element include oligos with mismatches as well as perfect matches; since 50- to 65-mer oligos bind sufficiently well even with some mismatches, a probe is included if it aligns to the target with at least 85% identity over the length of the probe, and with a 29 nt perfect match subsequence. On LLMDA version 5, an average of 130

probes satisfy the representation criteria for each target genome element, and over 67,000 targets from almost 6,000 microbial species are covered by the 360,000 LLMDA probes. A major difference between the LLMDA and other microbial arrays is that bacterial probes may be selected from anywhere in the genome, rather than only from 16S rRNA genes; thus, many bacteria can be identified at the strain level.

The LLMDA was shown to correctly identify a variety of previously characterized viral and bacterial cultures with high sensitivity and specificity [Gardner 10], and also confirmed sequencing-based detection of porcine circovirus DNA in a pediatric rotavirus vaccine [Victoria 10]. Recently, LLNL collaborated with the Statens Serum Institut in Denmark to develop a diagnostic platform using unbiased random amplification and the LLMDA to identify viral pathogens in clinical samples. [Erlandsson 11]. This work demonstrated the potential of the microarray technique for broad spectrum pathogen detection in human samples. It also showed that the LLMDA could detect both DNA and RNA viruses as well as bacteria and plasmids present in the same sample, and in many cases can differentiate different subtypes of the same viral species.

The advantages of the LLMDA over other array platforms are its broad coverage of bacteria and viruses, as well as of eukaryotic pathogens, with large numbers of probes per target sequence. It shares with other reference sequence based array designs a disadvantage relative to "universal" arrays: the array design must be updated periodically to include probes for new genomes deposited in GenBank.

1.3 Analysis of detection arrays

1.3.1 General problems of detection array analysis

Much of the initial work on microarray data analysis focused on the use of these arrays to measure gene expression [Smyth 03]; that is, to infer changes in messenger RNA (mRNA) concentration in cells or tissues, resulting from changing experimental conditions, by hybridizing labeled copies of mRNAs to arrays containing probes for specific gene transcripts. Algorithms for expression data analysis had to deal with the fact that, in most experiments, the true mRNA concentrations were unknown. Therefore, most work on expression analysis was aimed at background correction [Kooperberg 02], normalization [Bolstad 03], and estimation of concentration ratios (fold changes) between different conditions [Irizarry 03].

Detection array analysis offers the opportunity to understand microarray behavior in much greater detail, because the samples analyzed are produced from genomic DNA. Sequences are known for many microbial genomes, and standard laboratory techniques exist to measure the concentration of DNA in a sample. Therefore, one can design experiments in which sample DNA molecules with known degrees of complementarity to probe sequences are present, at a wide range of known concentrations. The wealth of information available in these experiments makes it possible to develop detection algorithms based on models, in which the probe signal given the presence of a target organism at some concentration is

predicted from the probe and target genome sequences. After fitting model parameters from experiments with known samples, one can solve the inverse problem to find the targets that best explain the observed array data for an unknown sample.

Nevertheless, detection arrays present many of the same analysis issues as other types of microarrays. Probe signals must be corrected for background fluorescence of the glass array substrate [Kooperberg 02], and have additional noise contributions due to transient hybridizations with noncomplementary or partially matching DNA molecules [Zhang 03]. Probe and target DNAs may form hairpins or other secondary structures that prevent hybridization between expected partners [Ratushna 05, Gibas 10], or enhance hybridization between unexpected probe-target pairs. Chemical saturation, in which most or all of the oligos in a probe feature are bound by target DNAs, creates a nonlinear relationship between target concentration and probe intensity [Burden 04]. Another source of nonlinearity is optical saturation [Dodd 04], which occurs when the scanner converts the analog probe intensity to a 16-bit digital value; all intensities greater than some threshold are converted to the maximum value (65535). If the scanner photomultiplier tube (PMT) gain is set too high, a substantial amount of information about the true probe intensities may be lost.

1.3.2 The GreeneLAMP algorithm

Many current algorithms for detection array analysis follow purely empirical approaches, without trying to model the physical processes underlying hybridization, washing and scanning. The algorithm developed for GreeneChip analysis, "log-transformed analysis of microarrays using p-values" (GreeneLAMP) [Palacios 07], is one such approach to the species identification problem. The GreeneLAMP algorithm makes several key assumptions about array experiments:

- Probe intensities are log-normally distributed.
- Probe intensities represent independent measurements of target genome concentrations.
- The number of probes for any species having positive signals is limited, on the order of 100 or fewer.

When pairs of probe sequences are 95% or more identical, the independence assumption is clearly violated. In this case probes are clustered into equivalence groups and their signals are pooled, in a manner not specified by the authors. Probes are associated with target taxa using BLAST; the score threshold for association is also not specified by the authors, and the algorithm does not differentiate between probes with strong and weak similarity.

To analyze an array experiment, the GreeneLAMP software first subtracts background levels from the probe intensities, for probes more than two standard deviations from the mean. The background levels are derived from matched control samples when they are

available, and from random 60-mer control probes on the same array otherwise. The software then centers the log intensities, divides them by the standard deviation to form Z-scores and computes tail probabilities (p-values) under the log-normality assumption. It then categorizes the probes as positive or negative according whether the p-values exceed a fixed threshold: 0.1 for arrays with matched controls, and 0.023 otherwise.

Finally, individual p-values for positive probes associated with each taxon are combined using the QFAST algorithm [Bailey 98]. This step depends crucially on the independence assumption. The product of the n p-values is used as a test statistic; its tail probability assuming independence of the p-values can be shown to be:

$$\mathbb{P}[\prod_i p_i > p] = 1 - p \sum_{k=0}^{n} \frac{(-\log p)^k}{k!}$$

The candidate taxa are then ranked by this combined p-value.

As mentioned in our discussion of the GreeneChip design, the GreeneLAMP algorithm was moderately successful in the analysis of viral samples, providing correct identification at the species level. Since the algorithm has only been applied to GreeneChip data, it is difficult to assess its performance independently from that of the chip design. The failure of the GreeneChip platform to precisely identify bacteria can be partially explained by the cross-reactivity of ribosomal RNA probes. However, an algorithm design that accounted for the greater affinity of probes for highly similar target sequences might have been able to overcome the limitations of the array design. A more severe limitation of GreeneLAMP is its inability to deal with complex mixtures, such as those found in clinical and environmental samples. Since the output of the algorithm is a single ranked list of taxa, there is no means to identify a combination of taxa that if present would best explain the observed intensity data.

1.3.3 The E-Predict algorithm

Another empirically motivated method is the E-Predict algorithm [Urisman 05], which was developed for analyzing ViroChip arrays. E-Predict computes a "theoretical hybridization energy profile" for each complete viral genome, by using BLAST to align probes to the genome sequence, and then computing a predicted hybridization free energy for each probe with a significant alignment. Free energies, which are related to the affinity for a probe to bind to a target genome fragment, are computed using a nearest-neighbor stacking energy method [SantaLucia 04], and are then scaled to produce a vector within the unit hypercube (using quadratic normalization by default):

$$\Delta G_{ij}^{(norm)} = \frac{\Delta G_{ij}^2}{\sum_{i=1}^{n} \Delta G_{ij}^2}$$

Here n is the number of probes on the array and ΔG_{ij} is the raw free energy for probe i hybridizing to target j. Probes with no BLAST hit to the target genome are assigned free

energies equal to zero, so the sum in the denominator need only be computed over probes with hits to target j.

To identify the target hybridized to an array, E-Predict by default normalizes the probe intensities y_i to have total sum equal to one:

$$y_i^{(norm)} = \frac{y_i}{\sum_{i=1}^{n} y_i}$$

Alternative (sum, quadratic, and unit vector) methods may be used to normalize both intensities and free energies. E-Predict then computes a similarity score for the normalized intensity and free energy vectors for each viral sequence in a candidate target database, using one of several similarity functions: the dot product, centered or uncentered Pearson correlation coefficient, Spearman correlation coefficient, or a function based on Euclidean distance. The target scoring highest is identified as most likely to be present in the sample.

E-Predict associates p-values with scores by comparing them to an empirical probability distribution derived from 1,009 microarray experiments, which was found to be approximately log-normal. The authors assume that the underlying null distribution is exactly log-normal, and estimate its parameters by iteratively trimming the highest score values for each virus until the remaining log scores show the least deviation from normality, according to a Shapiro-Wilk test. The mean and variance are computed from the remaining untrimmed values.

To be useful for analyzing clinical and environmental samples, a detection algorithm must be able to identify multiple organisms within a sample. This problem is addressed with an iterative version of E-Predict. After identifying the most likely target as described above, E-Predict sets the intensities of the probes matching that target to zero, renormalizes the intensity vector, recomputes the similarity scores for the remaining targets, chooses the highest scoring target, and computes its p-value. This process can be iterated until no remaining targets have p-values below a specified threshold.

At first glance, E-Predict appears to be motivated by a thermodynamic model, since it uses free energies to represent probe-target similarities or affinities. However, the authors don't present a physical justification for their choices of normalization and scoring functions; these were instead chosen because they provided the best separation in between-family comparisons and the least separation within families, for a particular test dataset. Therefore, one might be concerned that the algorithm might not generalize well to a wider range of datasets. Nevertheless, E-Predict has been successfully applied to identify or characterize viruses in thousands of ViroChip experiments; notably, it was used in 2003 to identify the infectious agent of severe acute respiratory syndrome (SARS) as a novel coronavirus [Wang 03].

1.3.4 VIPR

VIPR (Viral Identification using a PRobabilistic algorithm) [Allred 10] is a technique developed for analysis of viral diagnostic microarrays. It is essentially a naïve Bayes classifier, based on the assumption that probe intensities follow a log-normal distribution with one of

two sets of parameters for each probe, according to whether it is predicted to bind the target in the sample (is "on") or not ("off"); i.e. the log intensities Y_i are distributed as follows:

$$Y_i|\text{on} \sim N(\mu_{i,\text{on}}, \sigma^2_{i,\text{on}}); \quad Y_i|\text{off} \sim N(\mu_{i,\text{off}}, \sigma^2_{i,\text{off}})$$

The binding predictions are obtained by calculating free energies ΔG with a nearest-neighbor approach, and treating probes with ΔG below a fixed threshold as "on". To give accurate results, the VIPR model must be trained using data from positive control arrays to estimate the parameters of the "on" and "off" intensity distributions for each probe. Priors for the latent variables in the model (the on/off states) are derived by considering the fraction of probes $\mathbb{P}_{\text{pred}}[\text{on}]$ predicted to bind a target T, along with the numbers of targets $n(\text{state})$ sharing the same state for a given probe, and applying Bayes' rule:

$$\mathbb{P}_{\text{marg}}[\text{on}] = \frac{\mathbb{P}[T|\text{on}]\mathbb{P}_{\text{pred}}[\text{on}]}{\sum_{\text{state=on,off}} \mathbb{P}[T|\text{state}]\mathbb{P}_{\text{pred}}[\text{state}]}$$
$$\mathbb{P}_{\text{marg}}[\text{off}] = 1 - \mathbb{P}_{\text{marg}}[\text{on}]$$

Here $\mathbb{P}[T|\text{state}]$ is assumed to be uniform, i.e. equal to $1/n(\text{state})$.

Given the prior probabilities, the on/off distribution parameters for each probe, and the observed log intensities y_i, VIPR computes posterior probabilities for each probe i:

$$\mathbb{P}[\text{on}|Y_i = y_i] = \frac{\mathbb{P}[Y_i = y_i|\text{on}]\mathbb{P}_{\text{marg}}[\text{on}]}{\sum_{\text{state=on,off}} \mathbb{P}[Y_i = y_i|\text{state}]\mathbb{P}_{\text{marg}}[\text{state}]}$$
$$\mathbb{P}[\text{off}|Y_i = y_i] = 1 - \mathbb{P}[\text{on}|Y_i = y_i]$$

Finally, VIPR combines these probabilities to calculate a posterior probability for each target in a list of candidate viruses (assuming conditional independence of the probe states). Let $\mathcal{B}(T)$ be the set of probes predicted to bind T; then the probability that T is present is:

$$L(T) = \prod_{i \in \mathcal{B}(T)} \mathbb{P}[\text{on}|Y_i = y_i] \prod_{i \notin \mathcal{B}(T)} \mathbb{P}[\text{off}|Y_i = y_i]$$

When compared to E-Predict and other published algorithms, VIPR had greater accuracy in identifying viruses hybridized to a custom hemorrhagic fever virus array. Like GreeneLAMP, VIPR is not designed to deal with complex samples, where a mixture of targets might be present. Also, the requirement that parameters be fitted to data from arrays hybridized to each candidate target limits its usefulness for broad-spectrum microbial detection arrays. However, its ability to "learn" from additional training data means that VIPR may be more accurate than other algorithms when applied to specialized diagnostic arrays, designed to test for a limited range of species.

1.3.5 DetectIV

DetectIV [Watson 07] is a software package, written in the R language [R D 11], which provides simple visualization, normalization and significance testing functions for detection

array data. Unlike most of the methods discussed here, it is not tightly coupled to any particular array platform, and runs in any computing environment that supports the R language, including Mac OS X, Unix/Linux and Windows. To normalize probe intensities, DetectiV divides them by a reference intensity, which may be either the mean of a set of designated control probes, the global median intensity for the array or the intensity of the corresponding probe on a reference array; the logarithm of the intensity ratio is then reported for each probe. Significance testing is performed by selecting groups of probes sharing common family, species, or other annotations, and computing a one-sample t-test, with the null hypothesis that the log intensity ratio for each group is zero.

The DetectiV software leaves interpretation of the log ratio and t-test results to the user; it does not correct p-values for multiple testing, nor does it define any threshold values for "detection" of a particular species. Typically a user will rank families or species by p-value, and examine the log intensity ratios for the top n groups to decide which species are most likely to be present. DetectiV may thus be regarded as a useful package for exploratory data analysis, rather than a rigorous statistical tool. Nevertheless, the authors found that, when applied to two ViroChip data sets used in the E-Predict study [Urisman 05], DetectiV gave better prediction performance than E-Predict.

1.3.6 PhyloDetect

PhyloDetect [Rehrauer 08] is one of the few analysis methods that deals explicitly with the genomic similarity between taxonomically related organisms, and the consequent tendency of some probes to cross-hybridize to multiple organisms. Given a "match matrix" $M = [m_{ij}]$, in which $m_{ij} = 1$ if probe i matches target j and 0 if not, PhyloDetect groups targets into a nested hierarchy, based on the similarity of their column vectors in the match matrix. Targets that are indistinguishable (because their match vectors are identical) are collapsed. This grouping is done once for each array design and candidate target set. To analyze an array, PhyloDetect reduces the probe intensities to binary indicators (e.g., by thresholding against the median plus two standard deviations of the background intensities), and performs a series of hypothesis tests, one for each group in the hierarchy. Interestingly, the null hypothesis in each test is that an organism in the group is present; the alternative is that no organism in the group is present. The test statistic is based on the number of probes matching the group that have zero indicators, and a probe-independent false negative rate γ. If there are n probes matching the group, the probability of observing r or more probe intensities below the detection threshold is the complement of the cumulative binomial distribution,

$$\mathbb{P}[m \geq r] = \sum_{k=r}^{n} \binom{n}{k} \gamma^k (1-\gamma)^{n-k}$$

This probability is compared against a significance threshold α, and the group is predicted to be absent (at significance level α) if the probability is below α. The test is repeated for every group at every level in the hierarchy, and the scores are displayed in a tree structure format.

PhyloDetect is designed to work with data for any detection array, provided that one can construct a match matrix for its probes against a list of candidate targets; in fact, it is available as a web application provided by the authors. However, this implementation does not scale well for high-density microarrays and/or large candidate target sets, because the match matrix must be instantiated as a dense array data structure and transmitted to the application server, using large amounts of memory and bandwidth. In addition, the false negative rate parameter must be chosen carefully for each array design. The greatest strength of PhyloDetect is that its results can be easily interpreted, in the common situation where the sample contains an organism related but not identical to one or more of the candidate targets.

1.4 Conclusion: the need for improved analysis methods

Several promising approaches to microbial detection array design and analysis have been tested during the past decade. The array platforms vary widely in terms of fabrication cost, range of organisms targeted, and in sensitivity and specificity of detection. An essential component of any microbial detection platform is an analysis algorithm that can make sense of the noisy data produced with current array technology and yield easily interpretable results. Analysis and visualization software will become especially important as microbial detection arrays move from the research environment to widespread medical, industrial and military use. Most of the analysis algorithms described in this review can be adapted to handle data from a variety of array types, and each algorithm has its unique merits.

All of the analysis methods described above operate within a hypothesis testing framework, providing estimates of probabilities for absolute presence or absence of each candidate microorganism. Most of the algorithms have limited capacity to deal with samples containing complex mixtures of microbes, such as soil, water or human microbiome samples. For diagnostic and risk assessment purposes, and for the study of complex microbial communities, it is important to be able to measure not just the presence of particular organisms, but also their abundances in a sample. In order to assess concentrations of microbial nucleic acids in clinical and environmental samples, we need to have a physical model of the process by which microbial DNA hybridizes to oligos on an array and produces a set of probe intensity measurements.

In Chapter 2, I develop a predictive physical model for the hybridization and measurement process, and show results from testing it against data generated in our laboratory. In Chapter 3, I apply the model to the inverse problem of assessing the organisms present in a microarray sample and estimating their abundances.

Chapter 2

Predictive models of microbial detection array probe intensities

2.1 Introduction and basic framework

Analysis of detection array data involves solving two types of problems: predicting individual probe intensities when a sample contains known targets at specified concentrations; and inferring the targets present in an unknown sample, along with their abundances, from the probe intensities observed on an array. The prediction problem must be addressed first, and will be the subject of this chapter; the inference problem will be discussed in Chapter 3.

To solve the prediction problem, I developed physical models for the probe hybridization and intensity measurement processes. I then fit the model parameters using data from several experiments, in which samples containing genomic DNA at known concentrations from organisms with known genome sequences were hybridized to arrays. In this chapter, I'll present the predictive modeling approach, and assess its performance in experiments with known samples.

2.1.1 Physical models of probe-target hybridization

The models I wish to examine are based on an understanding of the hybridization process by which probes bind to labeled targets, the scanning process that produces light emission from the fluorescent labels, and the measurement process that yields an intensity value for each feature on the array. I'll begin by discussing hybridization models.

On a high-density microarray such as the LLMDA, each feature is a square 15 μm wide or smaller, containing approximately 500,000 to 1,000,000 oligos. The oligos are distributed randomly within features, and neighboring oligos are separated by an average distance comparable to their length, about 20 nm.

When a sample is hybridized to the array, some fraction θ_i of the oligos in feature i bind to labeled target DNA. After a specified hybridization time, the array is washed to remove unbound target. The fraction of bound oligos θ_i is assumed to depend on the target DNA

concentration and some measure of the probe's affinity for the targets in the sample. The exact form of the dependence varies according to the physical model, but in general the bound fraction increases with both concentration and affinity. In a simple two-state equilibrium model of hybridization of a probe i to a single target species j with concentration c_j, the affinity is a constant K_{ij} and the bound fraction is given by the Langmuir equation:

$$\theta_i = \frac{K_{ij} c_j}{1 + K_{ij} c_j} \tag{2.1}$$

In a typical affinity model, the affinity depends on the free energy ΔG_{ij} of hybridization between probe i and target sequence j, through the Boltzmann-Gibbs relation:

$$K_{ij} = K_0 e^{-\Delta G_{ij}/RT} \tag{2.2}$$

where K_0 is a constant with units of inverse concentration, T is the temperature and R the universal gas constant. The free energy in turn depends on the complementarity of the interacting portions of the probe and target sequences, and on the specific sequence of base pairs formed by the aligned probe and target.

In actual microarray experiments, there are typically multiple probes capable of binding a given target DNA, and multiple targets in the sample capable of binding to a given probe, with varying affinities. For a given probe i that can bind m different targets, the bound fraction at thermodynamic equilibrium is given by the full form of the Boltzmann-Gibbs distribution:

$$\theta_i = \frac{\sum_{j=1}^{m} c_j K_{ij}}{1 + \sum_{j=1}^{m} c_j K_{ij}} \tag{2.3}$$

$$= \frac{K_0 \sum_{j=1}^{m} c_j \exp(-\Delta G_{ij}/RT)}{1 + K_0 \sum_{j=1}^{m} c_j \exp(-\Delta G_{ij}/RT)} \tag{2.4}$$

Various techniques have been developed to predict the free energies ΔG_{ij} from the interacting sequences. Most of these are variations on the nearest-neighbor (NN) model discussed in [SantaLucia 04]. In a nearest-neighbor model, the free energy of hybridization for two aligned DNA sequences is parameterized as a sum of contributions from neighboring nucleotide pairs in the alignment. These models will be discussed in further detail in section 2.5. Most of these free energy parameters were derived experimentally from reactions in which both the probe and target were in solution. Since in microarrays oligos are anchored at one end to a planar substrate, the configurations available for target DNA molecules to bind to probes are more restricted. As I'll discuss later and as previous authors have found [Held 03, Hooyberghs 09], the free energies predicted by solution-phase models appear to be much more negative (by an order of magnitude) than those inferred from microarray experiments.

However, we cannot automatically assume that the system of probes and targets reaches thermodynamic equilibrium within typical hybridization times. For example, [Sartor 04]

found that, in many array experiments, hybridization times up to 66 hours were required to reach equilibrium, while the typical hybridization time in our laboratory is 17 hours. Thus, an ideal hybridization model should take reaction kinetics into account. In a two-state kinetic model, in which only target j is present in the sample, the bound fraction changes according to the following differential equation [Burden 04]:

$$\frac{d\theta_i(t)}{dt} = k^a_{ij} c_j [1 - \theta_i(t)] - k^d_{ij} \theta_i(t) \tag{2.5}$$

Here k^a_{ij} and k^d_{ij} are rate constants for the forward (adsorption) and backward (desorption) reactions. With initial condition $\theta_i(0) = 0$, this has the solution

$$\theta_i(t) = \frac{K_{ij} c_j}{1 + K_{ij} c_j} (1 - e^{-t/\tau_{ij}}) \tag{2.6}$$

where $K_{ij} = k^a_{ij}/k^d_{ij}$ and $\tau_{ij} = \frac{1}{k^a_{ij} c_j + k^d_{ij}}$. By comparing with equation 2.1, we see that the kinetic model converges to the Langmuir equilibrium model when t is much greater than the characteristic relaxation time τ_{ij}. We also see that the rate constants are related to the equilibrium affinity constant K_{ij}, and thus to each other through the hybridization free energy. By equation 2.2,

$$k^d_{ij} = \frac{k^a_{ij}}{K_{ij}} = \frac{k^a_{ij}}{K_0} e^{\Delta G_{ij}/RT} \tag{2.7}$$

The major difficulty in applying kinetic models to microarray analysis is that there are no methods available for predicting the rate constants from the DNA sequences of the reactants, and experimental measurements have been made on only a few of the many possible sequence pairs. Several experimental studies (reviewed in [Gibas 10]) suggest that, while both the adsorption and desorption rate constants depend on the probe and target sequence lengths, the salt concentration and the reaction temperature, only the desorption rate depends strongly on the specific probe and target DNA sequences. Thus, according to these studies it should be possible to approximately predict hybridization kinetics by treating the k^a_{ij} as being the same for probes with mismatches as for probes perfectly matching the target. The desorption constant k^d_{ij} is then computed from k^a_{ij} and from the predicted free energy using equation 2.7.

The model that emerges from this type of approach is one in which target DNA strands diffuse among the probe oligos and bind randomly to them for varying lengths of time, characterized by the corresponding k^d_{ij} values. Under this model, probes with high affinity to the target require more time than weakly bound probes to approach their equilibrium intensities [Dai 02]. This conclusion is supported by experimental evidence from Dai et al., and is expected because, for a probe with high affinity, a larger fraction of its oligos must be bound before the rate of target dissociation $k^d_{ij}\theta_i(t)$ matches the rate of association $c_j k^a_{ij}[1 - \theta_i(t)]$.

More complex kinetic models have been proposed [Gibas 10], involving transitions between more than two states; for example, a target DNA strand may be folded into a hairpin,

in which case it must unfold before it can bind to a probe oligo; or the target can bind partially to a probe oligo and then dissociate rather than hybridizing completely. [Hooyberghs 10] presents experimental hybridization data that cannot be explained under a two-state model, and introduces a model with a third state, in which the target is bound to a probe oligo over part of its length; the bound pair may then dissociate rather than hybridizing completely. Interestingly, the authors suggest that the nonequilibrium behavior of this model can be approximated using an equilibrium Langmuir model, with an increased value for the temperature. This may explain the discrepancy seen in [Held 03] and in our own data, to be described below, in which the free energies appeared to be much lower than predicted by the nearest neighbor model.

2.1.2 Scanning and measurement process effects

In a standard microarray scanner, a short-wavelength laser beam sweeps over the array surface in a raster scan pattern and excites fluorescent molecules, which respond by emitting photons of longer wavelengths. Some portion of the emitted photons are captured and analyzed by a detector, as described below. Most of the light is emitted by "labeled" target DNA molecules, which are produced by synthesizing copies of template DNA in the sample and attaching them to fluorescent dye molecules (fluorophores), such as cyanine 3 ("Cy3"). Each target DNA molecule is bound to a single fluorophore, which emits a certain number of photons according to its quantum efficiency and the number of photons it absorbs from the laser, which in turn depends on the laser power and the raster scan rate.

In addition, background fluorescence is emitted by the sample medium and the array substrate, by unbound target DNA remaining after the washing step, and by the probes themselves.

Thus, the intensity of light emitted by feature i, y_i, can be modeled as a sum of background fluorescence y_b, plus a signal proportional to the fraction of oligos θ_i in the feature that are bound to labeled target DNA:

$$y_i = y_b + \gamma\theta_i \tag{2.8}$$

where γ is a scale factor, assumed to be the same for all features on the array, but varying between arrays and scans. On high-density oligonucleotide microarrays, the background intensity can generally be treated as uniform over the surface of the array, although it may vary between arrays; it also depends on the emission wavelength, an important consideration for two-color array experiments.

The emitted photons are detected by a photomultiplier tube (PMT), producing an analog electrical signal, which is then mapped to an unsigned 16-bit integer by an analog-to-digital (A/D) converter. The signal produced by the PMT increases exponentially with the voltage v applied across electrodes within the tube, according to the equation

$$S = S_0 v^{\alpha n_d} \tag{2.9}$$

where n_d is the number of electrode stages (dynodes) in the PMT, and α is a constant determined by the material of the dynodes, ranging between 0.7 and 0.8 [Hamamatsu 06]. For example, in a 10-stage PMT with $\alpha = 0.7$, the increase in gain produced by raising the voltage from 650 to 750 is $(750/650)^7$, or about 2.73-fold. Experimenters typically adjust the PMT gain so that the measured signals span the full dynamic range of the A/D converter, while keeping the proportion of optically saturated pixels (discussed below) within acceptable limits.

The PMT output is affected by various sources of noise, the greater part of which is multiplicative; higher intensities are accompanied by proportionately larger measurement errors. Generally these errors can be fit reasonably well to a log-normal distribution, so that log-transforming the data removes the dependency of the variance on the intensity.

Usually, the scanner produces an image with finer resolution than the spacing between features, so that each feature is represented by between 9 and 64 pixels. Image processing software is then used to align the feature grid to the image and aggregate the individual pixel intensities into a combined intensity for each feature. It is possible that one or more pixels in a feature will have intensities that reach or exceed the maximum value that can be processed by the A/D converter. When this happens, the A/D converter outputs its maximum value ($2^{16} - 1 = 65,535$), and the pixel is said to be *optically saturated*. Note that, because of the multiplicative noise component, only a subset of pixels in a feature may be saturated, so that the feature intensity computed by averaging over all pixels may be reduced by saturation effects even when it is well below 65,535. Since optical saturation leads to loss of information, experimenters generally try to minimize the fraction of saturated pixels.

2.2 Characteristics of microbial detection array data

At LLNL, our group has designed and tested a wide range of microarrays based on the NimbleGen platform, including several versions of the Lawrence Livermore Microbial Detection Array (LLMDA) [Gardner 10], genome tiling arrays for bacterial SNP detection, genotyping arrays for known SNPs in bacteria and viruses, and functional gene arrays for assessing virulence gene presence in bacterial genomes [Jaing 08]. In nearly all our experiments, the sample consisted of genomic DNA from one or more species of microorganisms.

In order to assess the performance of our arrays, we ran a number of experiments with each array design in which we hybridized a sample of an organism of known genome sequence to the array. These experiments provided an excellent opportunity to develop and refine models of the hybridization and measurement processes and to fit parameters to the models. In the remainder of this chapter, I'll discuss the results of some of these experiments.

2.2.1 Probe types in the Lawrence Livermore Microbial Detection Array

In order to develop useful predictive models for LLMDA probe intensities, it is helpful to understand the strategy used to design probes, and the types of relationships that may result between probe and target sequences. With this in mind, we can then begin to interpret the observed distribution of intensities in a single-target experiment, for probes having each type of relationship to the known target.

On the LLMDA, each microbial genome is targeted by 10 to 50 probes, of length between 50 and 65 nt. These probes are either perfect matches to some targeted sequence, or have a small number of mismatches; in the latter case they are required to have a perfect match subsequence of length 29 or greater, and an overall 85% or greater nucleotide identity to the target [Gardner 10]. In addition, each target has a variable number of probes with weaker similarity. Often these probes are perfect or near-perfect matches to some related target in the same family. Due to the length of the probes used in the LLMDA, some of these weakly similar probes are able to bind to the target, though with a lower affinity than the perfect match probes.

For a given target, we can thus distinguish four classes of probes: target-specific probes meeting the aforementioned design criteria; weakly similar probes, having BLAST hits to the target but not meeting the criteria; nonspecific probes, lacking hits to the target; and negative control probes with randomly generated sequences.

2.2.2 Probe intensity distributions

To characterize the performance of each of these probe types, we performed several experiments in which known quantities of DNA from a single viral isolate of known sequence were hybridized to the LLMDA. The $\log_2$ intensity distributions for probes belonging to each of the four classes were then plotted, using the standard Gaussian kernel density estimator implemented in the R `density()` function. Figure 2.1 shows a typical example of one of these plots, for an array hybridized to DNA from a known respiratory syncytial virus (RSV) isolate. Since the sample consists of genomic DNA from one virus strain, the complementary target sequences for each probe are present in roughly the same concentration. Thus, the variability in intensities for different probes is due entirely to differences in probe-target affinities.

Several typical characteristics of LLMDA data are notable in this plot. First, the distribution of target-specific probe intensities is clustered near the maximum possible intensity ($2^{16} - 1$). This occurs because the LLMDA probes are designed with high affinity to their targets, so that probe features tend to be chemically saturated whenever their targets are present at sufficient concentration. In addition, the array was scanned at a high PMT gain setting, so that many probes are affected by optical saturation.

Second, the probes with weak similarity to the target have intensities spanning a wide range. The majority of them are at the low end of the scale, together with the nonspecific and

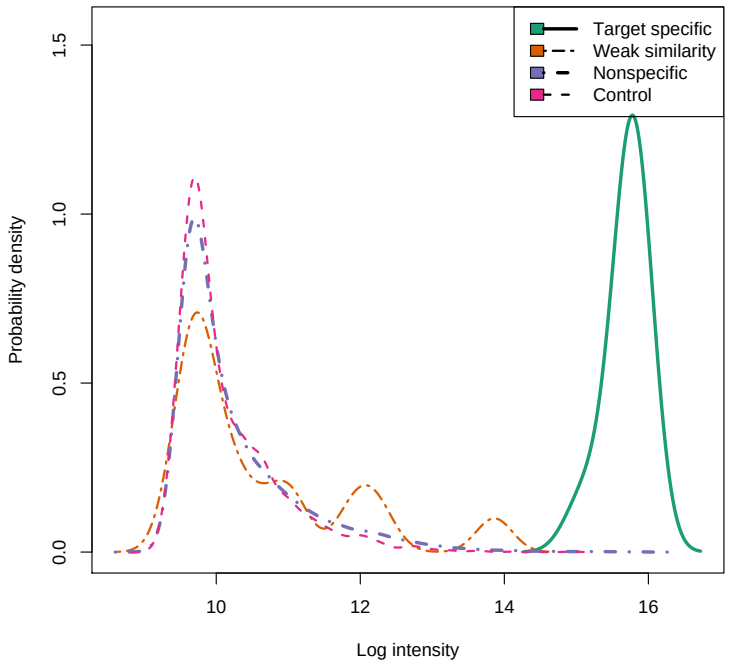

Figure 2.1: Log intensity distributions for four classes of probes on an LLMDA array for respiratory syncytial virus

control probes; however, the upper end of their distribution overlaps with that of the target specific probes, indicating that weak similarity is sometimes enough to produce substantial cross-hybridization.

Third, the distribution for nonspecific probes closely resembles that of the negative control probes. The resemblance is seen more clearly in Figure 2.2, in which I have plotted corresponding quantiles of the nonspecific and control probe intensities against one another. This validates my use of the control probe distribution as a reference standard in experiments where the target is unknown, to assess the respective chances that a probe signal arose from target-specific or from nonspecific hybridization.

Finally, the log intensity distributions of nonspecific and control probes have heavy tails on the right side. In fact, by plotting the density estimates on a log scale, as shown in Figure 2.3, the intensities are seen to follow power-law behavior in both tails, with log-normal behavior close to the mode. This pattern is reminiscent of the double Pareto-lognormal distribution [Reed 03], which has been used recently to describe a variety of data types such as incomes, particle sizes and stock prices.

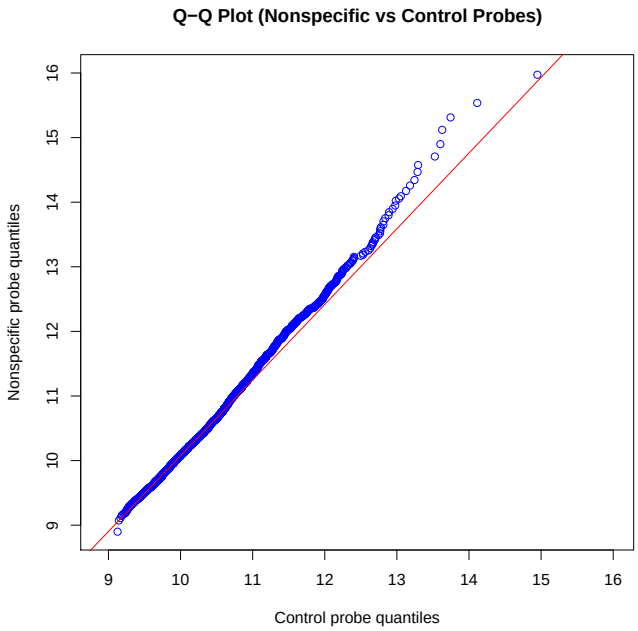

Figure 2.2: Quantile-quantile plot of log intensities for nonspecific and negative control probes on the RSV array

2.3 Adjusting intensities for measurement process effects

As I detailed in section 2.1.2, the process of scanning an array to measure fluorescence intensities produces measurements that are biased by the effects of scanner saturation, background fluorescence, and variations in PMT gain settings. In this section, I'll describe the procedures I developed to compensate for these biases.

2.3.1 Optical saturation correction

Pixels are optically saturated when their intensities exceed the maximum value that can be registered by the scanner's analog-to-digital converter. Generally, our technicians perform scans at the highest PMT gain they can use without having the proportion of saturated pixels exceed 0.05 percent across the whole array. Some scanners automatically iterate through a number of gain settings to find the optimal level. Unfortunately, optical saturation at any level introduces bias into the data, whereby average intensities of features containing saturated pixels are under-estimated. The 5 μm Axon 4000B scanner used for our older array experiments represents each feature with only 9 pixels; so even a small number of saturated pixels has a strong effect on the aggregate feature intensity. Our current Roche MS-200

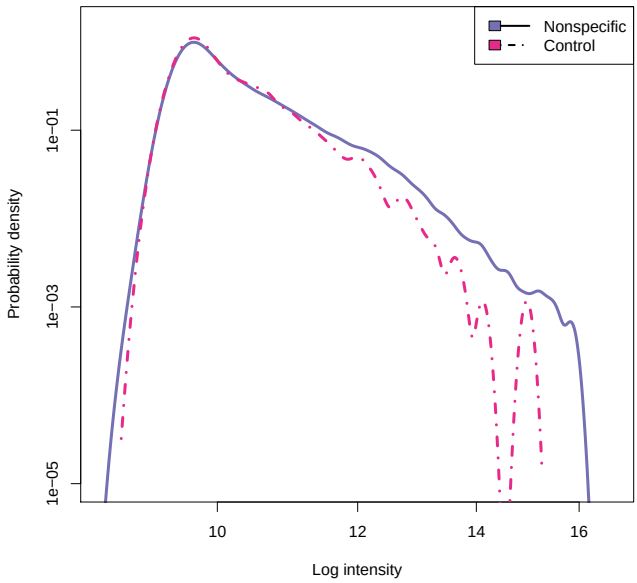

Figure 2.3: Nonspecific probe log intensity distribution from RSV array, with density estimates plotted on log scale to show power law behavior in tails

scanner, with 2 μm resolution, generates 49 pixels per feature and thus is less affected by small numbers of saturated pixels. However, its autogain software often saturates a large fraction of the pixels, so manual adjustment to a much lower setting is frequently required.

The effect of optical saturation on probe intensity measurements is illustrated by Figure 2.4, which shows log intensity distributions from two scans of the same array performed at different PMT gain settings. The array was hybridized to a sample containing DNA from 6 different targets, denoted Tm, Ba, Bt, Ft, Av, and Vv, each at a different concentration. Separate density curves are plotted for probes sensitive to each target. The saturation at the higher gain setting is manifested by the compressed distribution of intensities for the most concentrated targets (Ba and Bt). For both targets, most probes have $\log_2$ intensities between 15 and 16 (the maximum), despite Bt being present at four times the concentration of Ba. At the lower gain setting, the intensities are spread over a much wider range, reflecting the varying affinities of the probes for their targets and the higher concentration of Bt. Thus, optical saturation results in loss of information about probe affinities and concentrations of sample components.

The best approach to deal with optical saturation is to avoid it, by making sure that arrays are not scanned at too high a gain setting. If saturation is discovered early enough (within a few days or weeks of hybridization) and the arrays are preserved, they can be rescanned at lower gain without too much loss of signal. When this is not an option (e.g., when analyzing legacy data), we must compensate for saturation during data analysis.

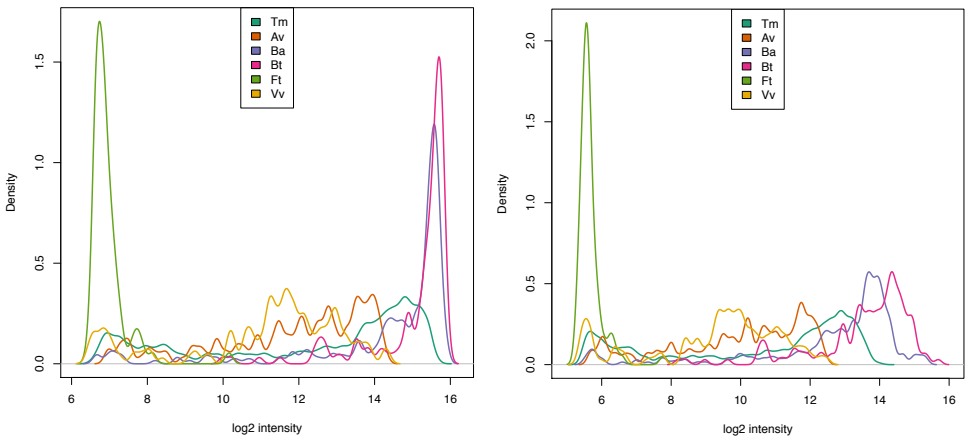

Figure 2.4: Intensity distributions for probes against six organisms present in different concentrations, in two scans of the same array, at 50% gain setting (left) and 20% gain (right). Density curves are colored according to the organism targeted by the probes.

To adjust for the effects of optical saturation, I implemented a method to compute a maximum likelihood estimate of the true mean feature intensity and its standard deviation. For any feature, I model the individual true pixel intensities x_i as following a normal distribution, $X_i \sim N(\mu, \sigma^2)$; by examining Q-Q/normal plots of pixel intensities for features without saturated pixels, I found the normal to be a reasonable approximation of the true distribution. Given the maximum scanner reading $y_{max} = 2^{16} - 1$, the observed intensity is $y_i = \min(y_{max}, x_i)$. The problem then reduces to estimating the parameters of a normal distribution from a set of measurements that are right censored at a known threshold. A more general version of this problem was solved over 60 years ago [Cohen 50]; the derivation below is specialized to the particular case at hand.

Because individual pixel intensities are not available from the NimbleScan array processing software, I wrote code to extract them from the TIFF image produced by the scanner, after aligning a feature grid to the image using the NimbleScan "auto-align" command. The intensities were used to generate a table listing, for each feature, the number of saturated pixels n_{sat} and the mean $\bar{y}$ and standard deviation s of all pixel intensities.

As it turns out, these are sufficient statistics for the maximum likelihood estimator. Suppose first that we are given the pixel intensities $\{y_i, i = 1...n\}$ for a feature. The likelihood function given these data is:

$$L(\mu, \sigma) \propto \prod_{i:y_i<y_{max}} \frac{1}{\sigma} e^{-\frac{(y_i-\mu)^2}{2\sigma^2}} \prod_{i:y_i=y_{max}} \left(1 - \Phi(\frac{y_{max} - \mu}{\sigma})\right) \tag{2.10}$$

so that the log likelihood is:

$$\log L(\mu, \sigma) = -(n - n_{sat}) \log \sigma - \sum_{i:y_i<y_{max}} \frac{(y_i - \mu)^2}{2\sigma^2} + n_{sat} \log \left(\Phi(\frac{\mu - y_{max}}{\sigma})\right) \tag{2.11}$$

Here $\Phi(x)$ is the normal CDF, satisfying $\Phi(-x) = 1 - \Phi(x)$. We can transform this into an expression involving the sufficient statistics instead of the pixel intensities:

$$\begin{aligned}
\log L(\mu,\sigma) + (n-n_{sat})\log\sigma - n_{sat}\log\left(\Phi(\frac{\mu - y_{max}}{\sigma})\right) &= -\sum_{i:y_i<y_{max}} \frac{(y_i-\mu)^2}{2\sigma^2} \\
&= -\frac{1}{2\sigma^2}\sum_{i:y_i<y_{max}} (y_i^2 - 2\mu y_i + \mu^2) \\
&= -\frac{1}{2\sigma^2}\left[n\bar{y^2} - n_{sat}y_{max}^2 - 2\mu(n\bar{y} - n_{sat}y_{max}) + (n-n_{sat})\mu^2\right] \\
&= -\frac{1}{2\sigma^2}\left[n(\bar{y^2} - \bar{y}^2 + \bar{y}^2 - 2\mu\bar{y} + \mu^2) - n_{sat}(y_{max}^2 - 2\mu y_{max} + \mu^2)\right] \\
&= -\frac{1}{2\sigma^2}\left[n(\bar{y^2} - \bar{y}^2) + n(\bar{y}-\mu)^2 - n_{sat}(\mu - y_{max})^2\right] \\
&= -\frac{1}{2\sigma^2}\left[(n-1)s^2 + n(\bar{y}-\mu)^2 - n_{sat}(\mu - y_{max})^2\right]
\end{aligned}$$

so that

$$\begin{aligned}
\log L(\mu,\sigma) = -\frac{1}{2\sigma^2}&\left[(n-1)s^2 + n(\bar{y}-\mu)^2 - n_{sat}(\mu - y_{max})^2\right] \\
&- (n-n_{sat})\log\sigma + n_{sat}\log\left(\Phi(\frac{\mu - y_{max}}{\sigma})\right)
\end{aligned}$$

To adjust feature intensities for optical saturation, I used numerical optimization to maximize the log likelihood with respect to μ and σ; the adjusted intensity is then $\hat{\mu}_{MLE}$. Note that, when $n_{sat} = 0$, $\log L(\mu,\sigma)$ reduces to the standard log likelihood for a normal distribution, with maximum likelihood estimates $\hat{\mu} = \bar{y}$ and $\hat{\sigma} = s$; i.e. no adjustment is required in this case.

One weakness of the maximum likelihood approach is that, when all pixels in a feature are saturated, the MLE grows to infinity and no correction can be performed. An alternative method would involve defining a prior distribution for the feature intensities and computing a maximum *a posteriori* estimate, given the measured pixel intensities and the number of saturated pixels. Since completely saturated features occur rarely in our data, I have not yet pursued this Bayesian approach; instead, I simply exclude these features from further analysis.

2.3.2 Background correction

In order to apply the linear model for probe intensities described by equation 2.8, we need a way to estimate the background signal y_b. The background is due to several components: autofluorescence of the array substrate, DNA probes, and sample medium; streaks and

bubbles; unhybridized target DNA left over from incomplete washing; the dark current of the photomultiplier tube; bleedover from adjacent features; and offsets added by the scanner circuitry. My definition of background does not include signal components due to "nonspecific hybridization", since nonspecific hybridization is simply hybridization of probe oligos to target DNA with little sequence similarity, which is already described by the hybridization model.

Background estimates can either be global for an entire array, localized to subregions, or targeted to individual features. While more localized estimates can capture nonuniformities in array hybridization or washing, they are also more liable to add noise to the corrected intensity estimate. For my analyses, I have corrected intensities using a global background estimate for each array.

The first step of my background correction procedure is to identify "empty cells" on the array, i.e. cells in between features, that don't contain any oligos for target DNA to bind to. On typical NimbleGen arrays, the features are arranged in a checkerboard pattern, with every other cell empty, as seen in Figure 2.5. The figure shows a close-up of of one of our array scans, centered on a cluster of bright fiducial spots (used for aligning the feature grid), in which the checkerboard pattern is apparent. Several dimmer features are also visible, while features designed against targets not present in the sample are dark. Empty cells are identified by examining the array design file, which gives the positions of features in the grid, and choosing the adjacent positions on each row.

Secondly, I exclude from the list of empty cells any which are adjacent to a feature with mean intensity over a fixed threshold (10,000 units, in my current implementation). I do this in order to eliminate from my background estimate contributions from bleedover, which can be seen in Figure 2.5 as the grey pixels extending leftward from the top and bottom of the upper left fiducial, and rightward from its center. Bled-over pixels are often seen adjacent to very bright features, and are caused by the slow decay of fluorescence after the scanning laser excites a region that is saturated with fluorophores. When this happens, the PMT continues to pick up photons from the feature even after the laser has moved on to the adjacent empty cell. Bled-over pixels appear on alternate sides of a bright feature, because the laser alternates directions as it scans successive rows of pixels.

Finally, I use the intensities measured in empty cells to estimate the background distribution for the array, and correct the feature intensities to remove the background contribution. To avoid negative signal estimates (which are physically meaningless, and prevent us from log-transforming the data), we need to do more than simply subtract the average background intensity from the observed feature intensity. Instead, I adapted the "normexp" background correction algorithm [Ritchie 07, Silver 09], which was in turn adapted from the first step of the RMA algorithm [Irizarry 03, Bolstad 04]. In this algorithm, the observed intensity is modeled as the sum of an exponentially distributed signal component and a normally distributed background component, truncated at zero: $Y = S + B$, with $S \sim \text{Exp}(\lambda)$ and $B = \max(0, X)$, where $X \sim N(\mu, \sigma^2)$. The corrected signal is estimated as the conditional

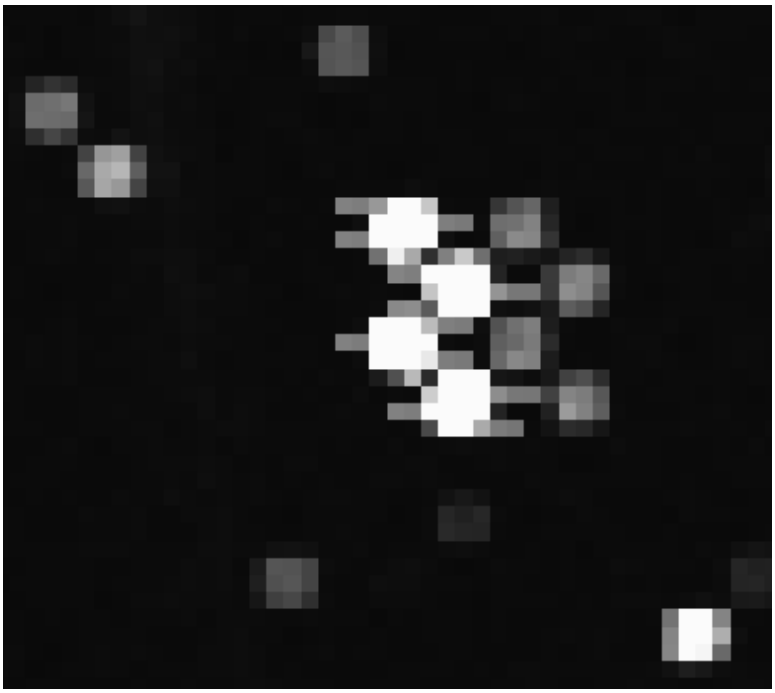

Figure 2.5: Section of an array image from NimbleScan, showing scan bleedover and checkerboard arrangement of features

expectation of S given the observed intensity y:

$$\mathbb{E}\left[S|Y=y\right] = a + \sigma \frac{\phi(\frac{a}{\sigma}) - \phi(\frac{y-a}{\sigma})}{\Phi(\frac{a}{\sigma}) + \Phi(\frac{y-a}{\sigma}) - 1} \tag{2.12}$$

where $a = y - \mu - \lambda\sigma^2$, and $\phi()$ and $\Phi()$ are respectively the standard normal density and CDF functions.

To correct microarray data with the normexp method, one must estimate three array-specific parameters: the μ and σ parameters for the normal component, and the rate parameter λ for the exponential component. The standard version of the normexp algorithm, implemented in the R `limma` package, subtracts the local background intensity from the foreground intensity measured for each feature, and fits the three parameters to the difference using a maximum likelihood procedure. Assuming that the empty cell intensities more accurately reflect the true background contribution, I instead estimated $\hat{\mu}$ and $\hat{\sigma}$ directly from the mean and standard deviation of the empty cell intensities. Following the example of the RMA algorithm, I then fitted the rate parameter λ to the $n_>$ feature intensities that exceed the mean background estimate, using maximum likelihood:

$$\frac{1}{\hat{\lambda}} = \frac{1}{n_>} \sum_{i:y_i>\hat{\mu}} (y_i - \hat{\mu}) \tag{2.13}$$

One problem with estimating the parameters of the background distribution using the empty cell intensities is that they don't reflect the contribution from autofluorescence of the DNA probes. I estimated the DNA autofluorescence component by examining an array that our technician scanned at a high gain setting without hybridizing a sample or performing any other processing on it beforehand. When I compared distributions of intensities, I found that cells containing DNA probes had intensities on average about 10% higher than empty cells.

To incorporate the DNA autofluorescence effect into the background correction algorithm, I used the negative control probe intensities rather than empty cell intensities to estimate the parameters of the normal-exponential distribution. Since the negative control intensities include both background effects and nonspecific hybridization effects, I used the maximum likelihood method implemented in the `limma` package to estimate the three parameters μ, σ and λ simultaneously, and then estimated the true signal for the target probes according to equation 2.12. This method performed well, and has the advantage of being easier to implement, since it does not require extracting the empty cell intensities from the TIFF image file.

2.3.3 Normalization for scanner PMT gain

Gene expression microarray data is typically normalized after background correction, to remove systematic variations of probe intensities between different arrays used in an experiment, and to give a similar intensity distribution across all arrays. Normalization based on statistical properties of the data, such as the quantile normalization method commonly used with RMA [Irizarry 03], is usually not appropriate for detection arrays, because there is no reason to expect different samples to produce the same numbers of bright probes. Therefore, we would prefer a normalization method based on a model of the scanning process.

The largest source of systematic variation in our data is the PMT gain setting used during scanning. As mentioned previously, our technicians typically adjust the PMT voltage to the highest value that doesn't produce an excess of saturated pixels, so that the output signal covers the full dynamic range of the A/D converter. It has been observed [Bengtsson 04] that the intensity values reported by the scanner are the sum of two components: an amplified input signal, whose gain increases with the PMT voltage; and an offset that depends on the scanned emission wavelength, but is independent of the PMT gain and is largely constant for a given scanner. Note that, since the offset does not increase with PMT gain, it is distinct from the signal produced by background fluorescence and other optical artifacts.

To estimate the gain and offset effects for the older Axon 4000B scanner used in our lab, I followed a procedure suggested by H. Bengtsson (personal communication). (A similar process could be applied to the Roche NimbleGen MS-200 scanner, for which the PMT gain setting is reported as a percentage of some arbitrary value rather than a voltage.) I used part of a short-hybridization dataset, to be described in more detail in section 2.4.1, generated from samples containing different concentrations of *Enterococcus faecalis* genomic DNA. The samples were hybridized for either 5 or 60 minutes to arrays containing probes for *E. faecalis*

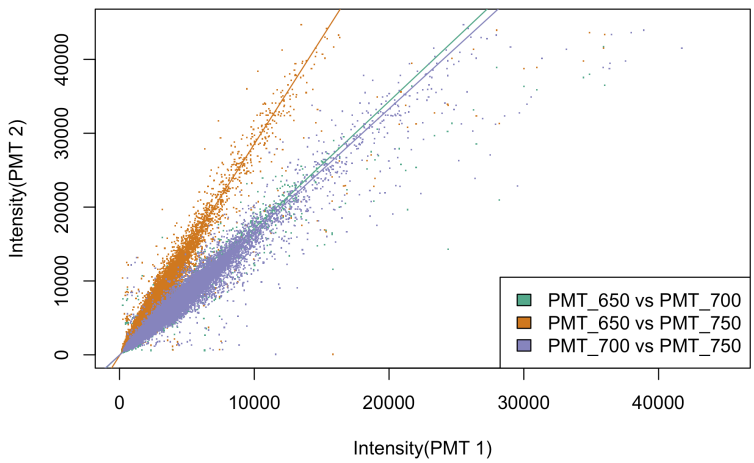

Figure 2.6: Scatter plot of pairwise comparisons of probe intensities from scans of the same array at three different PMT settings

and 3 other bacteria. Each of the 16 arrays was scanned three times, at PMT voltages 650, 700 and 750. One array was removed from the dataset because one of its scanned images missed a corner of the array, leaving 45 scans of 15 arrays in the dataset.

Since the Cy3 dye used with these arrays is not appreciably photobleached by the first and second scans under normal operating conditions, we can assume that the light signal x_i entering the PMT from feature i is the same for each scan. I fit the following affine model to the measured signals for the scan at PMT voltage $v \in \{650, 700, 750\}$:

$$y_{iv} = a + b_v x_i + \epsilon_{iv} \tag{2.14}$$

Here a is the offset, b_v is a slope parameter representing the effect of the PMT gain, and ϵ_{iv} is a noise term. To make the model parameters identifiable, I fixed $b_{650} = 1$, so that b_{700} and b_{750} are ratios of the gain at voltage 700 or 750 to the gain at 650 V.

I fit the remaining parameters to the data with iteratively reweighted principal component analysis (IWPCA), using the `fitIWPCA()` function from the R package `aroma.light` [Bengtsson 08]. Only features with no optically saturated pixels in any scan were used to fit the parameters.

Figure 2.6 shows overlaid pairwise scatter plots of the probe intensities from the three scans of an example array, together with the fitted lines transformed to (y_i, y_j) coordinates for each pair of scans (i, j). The parameters were fit separately for each array, and are shown in Table 2.1. We see that the gain ratio estimates are very close for all 15 arrays. The slope and offset values that deviate most from the medians are found for the arrays with the smallest DNA concentrations and hybridization times, which have narrower ranges of intensity values and thus provide less reliable fits.

From equation 2.9 with $b_{650} = 1$, we obtain the following expression for the PMT gain

DNA (μg)	Hyb time (min)	Replicate	a	b_{700}	b_{750}
1	5	1	146.7	1.79	3.24
1	60	1	81.8	1.67	2.74
1	60	2	60.0	1.69	2.76
2	5	1	103.9	1.74	2.99
2	5	2	86.2	1.71	2.91
2	60	1	73.1	1.70	2.81
2	60	2	37.7	1.70	2.75
5	5	1	79.0	1.74	2.93
5	5	2	51.0	1.72	2.83
5	60	1	61.0	1.68	2.77
5	60	2	50.4	1.69	2.82
10	5	1	63.6	1.71	2.87
10	5	2	58.9	1.70	2.81
10	60	1	44.9	1.66	2.68
10	60	2	44.5	1.71	2.77
Median			61.0	1.70	2.81
MAD			20.9	0.02	0.08

Table 2.1: Fitted scanner offset and PMT scaling values for 15 arrays

ratio:

$$b_v = \left(\frac{v}{650}\right)^{\alpha n_d} \tag{2.15}$$

Inverting this using the median slope estimates for b_{700} and b_{750} gives similar values for the exponent αn_d, 7.16 and 7.22 respectively. Therefore, I conclude that the affine model given by equation 2.14, with b_v given by equation 2.15, can be used to correct probe intensities for different PMT voltage settings, once the parameters a and αn_d have been determined for a particular scanner.

In subsequent analyses of data sets in which arrays are scanned with different PMT gains, I normalized the probe intensities to a common PMT setting of 650 V, using the median fitted values for the offset a and the gain ratio exponent $\eta = \alpha n_d$, by combining equations 2.14 and 2.15:

$$y^{(norm)} = a + \left(\frac{v}{650}\right)^{\eta} (y - a) \tag{2.16}$$

2.4 Kinetics of probe hybridization

I now wish to address the concerns raised in section 2.1.1 about whether equilibrium thermodynamic models are adequate to describe probe-target hybridization, given the typical array incubation times used in our laboratory. These issues are related to other questions we

sought to answer while studying the LLMDA's potential as a rapid diagnostic tool: what is the minimal hybridization time that can still yield accurate results; and can we compensate for short reaction times by increasing the sample concentration? To explore these issues in detail, I decided to fit the rate parameters k_i^a and k_i^d for a representative set of probes and determine their characteristic equilibrium times, given typical DNA concentrations in the hybridization reaction.

2.4.1 Hybridization kinetics dataset description

The dataset used to fit the rate parameters contained one set of scans from the short-hybridization *E. faecalis* experiment mentioned in section 2.3.3, together with data from 3 arrays run earlier with longer hybridization times (4, 8, and 16 hours). Probe intensities from all arrays were corrected for optical saturation and background-adjusted, using the procedures discussed earlier. Because the long- and short-hybridization arrays were scanned at different PMT settings (600 and 650 V, respectively), the long-hybridization probe intensities were adjusted using the affine normalization technique described in section 2.3.3. Each probe sequence was tiled in 5 replicate features; however, I excluded intensities from features in which all pixels were saturated on the array. The resulting dataset contained median corrected intensities for $n = 491$ probes with sequences taken from the *E. faecalis* genome. The experimental covariates for this dataset are shown in Table 2.2.

DNA quantity (ug)	Hyb time (min)	PMT	Replicates
1	5	650	1
1	60	650	2
2	5	650	2
2	60	650	2
5	5	650	2
5	60	650	2
10	5	650	2
10	60	650	2
4	240	600	1
4	480	600	1
4	960	600	1
4	1200	600	1

Table 2.2: Covariates for *E. faecalis* hybridization experiments

2.4.2 Fitting the rate parameters

The intensity data from this series of experiments was used to fit the parameters of the two-state kinetic model described by equations 2.6 and 2.8. Let y_{ijr} denote the corrected intensity

of replicate feature r for probe sequence i on array j. Given these data and the covariates for the arrays (the DNA concentration c_j and hybridization time t_j), the $2n+1$ parameters to be estimated are the intensity scale factor γ (from equation 2.8) and the probe-specific adsorption and desorption rate constants k_i^a and k_i^d, for $i = 1, ...n$. Since the target sequence is the same for all arrays in this dataset, the rate constants are also the same, and thus don't require an array index.

I fit the parameters by minimizing the L_2 loss:

$$W = \sum_{i,j,r} \left(\log y_{ijr} - \log(\gamma\theta_{ij})\right)^2 \tag{2.17}$$

where

$$\theta_{ij} = \frac{k_i^a c_j}{k_i^a c_j + k_i^d}(1 - e^{-t_j/\tau_{ij}}) \tag{2.18}$$

and $\tau_{ij} = \frac{1}{k_i^a c_j + k_i^d}$ is the characteristic time for relaxation to equilibrium. To avoid fitting zero or negative values for the parameters, which would be physically meaningless, I minimized W with respect to the log-transformed parameters $\lambda_i^a = \log k_i^a$, $\lambda_i^d = \log k_i^d$, and $\phi = \log\gamma$. Since ϕ is a linear parameter, I performed minimization in two stages. First, I estimated the nonlinear parameters λ_i^a and $\lambda_i^d, i = 1, ...n$ using the R nonlinear least squares function `nls()`, with ϕ fixed. Secondly, I estimated ϕ by solving $\frac{\partial W}{\partial \phi} = 0$ for ϕ, using the estimates of λ_i^a and λ_i^d to compute $\hat{\theta}_{ij}$. If N is the total number of intensity data points, the estimate is:

$$\hat{\phi} = \frac{1}{N}\sum_{i,j,r}\left(\log y_{ijr} - \log(\hat{\theta}_{ij})\right) \tag{2.19}$$

I tried a number of heuristics for choosing the initial λ_i^a and λ_i^d values passed to the `nls()` function. For about half of the probes, `nls()` failed to converge within 1000 iterations, regardless of the initial values chosen. To find out what characteristics of the data make it so difficult to minimize the loss function for certain probes, I decided to examine the loss function surface for a set of simulated probe intensities generated from known rate parameter values. By characterizing the topography of the loss function, I expected also to find a better method for selecting initial values.

2.4.3 Loss functions for simulated probes

In order to simulate intensity values as realistically as possible, I used as inputs the same combinations of concentration and hybridization time covariates as were found in the real data. I generated input rate constants k_i^a and k_i^d as random values from log-uniform distributions, and fixed the scale factor γ at $60,000$. I fed these inputs into equations 2.18 and 2.8 to produce raw intensity values. I noted that the average sample variance of log intensities for replicate probes on the same arrays was about 0.04; therefore, to simulate random measurement errors, I added noise distributed as $N(0, 0.04)$ to the raw log intensities.

With the simulated intensities as input, I then plotted contours of the loss function W given by equation 2.17 for each simulated probe over a range of k_i^a and k_i^d parameters. All probes from both real and simulated data had loss function surfaces with similar shapes; examples for two simulated probes are shown in Figure 2.7. The surface is characterized by a deep valley, which runs parallel to the k_i^d axis for small values of k_i^d, and then bends to follow a path with unit slope for larger k_i^d values. The bottom of the valley is nearly flat in regions far from the bend, and then dips to its minimum value near the bend. The bend occurs near the true (input) values of k_i^a and k_i^d, which are shown as green triangles in the figure.

To understand the topography of the loss function surface, we need to look at the limiting behavior of equation 2.18, which can be rewritten:

$$\theta_{ij} = k_i^a c_j \tau_{ij} (1 - e^{-t_j/\tau_{ij}}) \tag{2.20}$$

with $\tau_{ij} = 1/(k_i^a c_j + k_i^d)$. When $k_i^d << k_i^a c_j$ for all concentrations c_j represented in the dataset, the relaxation time $\tau_{ij} \approx 1/(k_i^a c_j)$. As $k_i^d \to 0$, the bound fraction on array j converges to a maximum value $\theta_{ij}^{(max)} = 1 - e^{-k_i^a c_j t_j}$, which depends on k_i^a only. Thus, when $k_i^d << k_i^a c_j$ for all arrays, reducing k_i^d further has little effect on the predicted probe intensities, and thus, on the value of the loss function.

At the opposite extreme, where $k_i^d >> k_i^a c_j$ for all arrays, the relaxation time is dominated by the effect of the desorption rate, i.e. $\tau_{ij} \approx 1/k_i^d$. In this limit, $\theta_{ij} \approx \frac{k_i^a c_j}{k_i^d}(1 - e^{-k_i^d t_j})$. In the further limit where $k_i^d >> 1/t_j$ for all arrays, the exponential vanishes and the bound fraction further simplifies to $\theta_{ij} \approx k_i^a c_j / k_i^d$, so that

$$\log \theta_{ij} \approx \log k_i^a + \log c_j - \log k_i^d \tag{2.21}$$

Thus, along a line of unit slope for which $\log k_i^a - \log k_i^d$ is constant, $\log \theta_{ij}$ and array j's contribution to the loss function are also constant. This behavior produces the segment of the valley having unit slope.

The characteristic shape of the loss function surface suggests a better method for selecting initial parameter values for the nonlinear least squares optimization procedure. The idea is to find initial k_i^a and k_i^d values near the bend in the valley. I first initialized the scale factor γ to $1.25 y^{max}$, where y^{max} was the maximum intensity observed across all arrays. Next, I chose an initial value for k_i^a, by fixing $k_i^d = 0$ and minimizing W with respect to k_i^a. This k_i^a value corresponds to the bottom of the part of the valley that runs parallel to the k_i^d axis. Finally, I choose a value for k_i^d that was consistent with the intensity at the longest hybridization time ($t_{max} = 16$ hours), given the initial k_i^a value and the assumption that most probes are near their equilibrium intensity by this time. Solving equations 2.18 and 2.8 for k_i^d gives $k_i^{d(init)} = c_m k_i^{a(init)} (\gamma / y_{im} - 1)$, where m is the index of the array with the longest hybridization time.

Using these initial values, I fitted γ, k_i^a and k_i^d values to the simulated data using `nls()`, and plotted the input and fitted rate constants against one another. The fitted value for

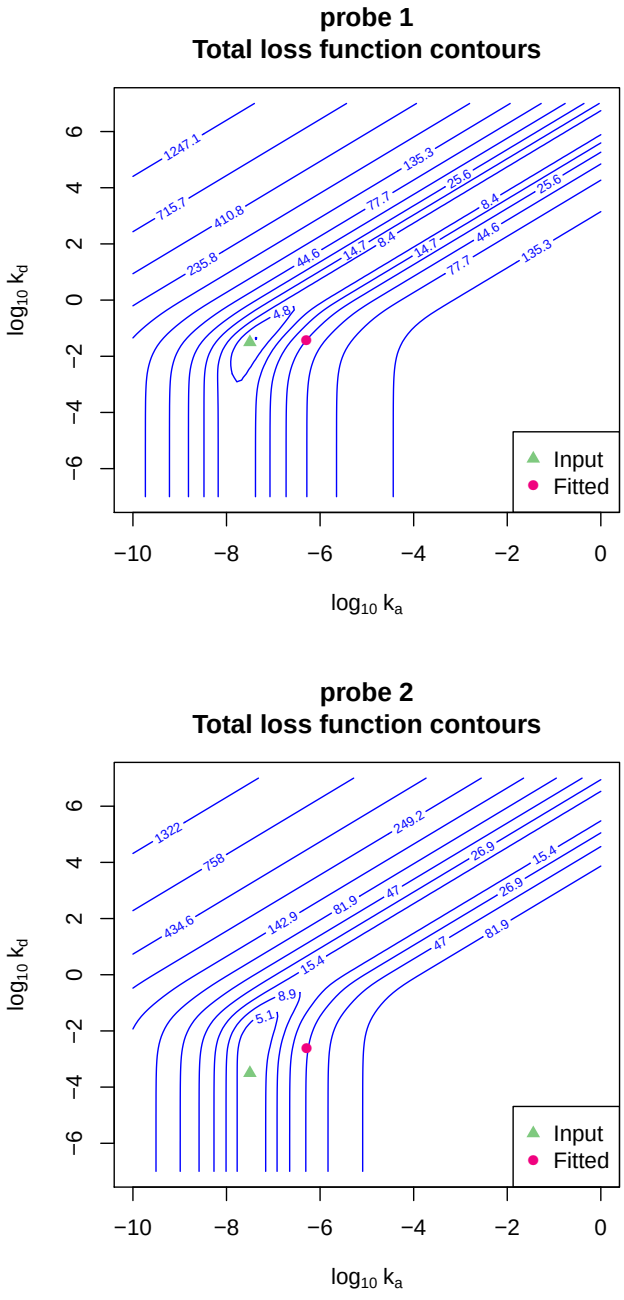

Figure 2.7: Loss W for simulated data for two probes having same k_i^a and different k_i^d values, as function of k_i^a and k_i^d.

γ was 80,216, somewhat higher than the input value 60,000. The results for the other parameters are shown in Figure 2.8. In each of the panels, both input and fitted variables are plotted on log scales, and an identity line is drawn to facilitate comparisons. In the upper left panel, we see that the fitting procedure does an excellent job of recovering the input k_i^a values, except that they are scaled downward by a factor that reflects the overestimation of γ. At upper right, we see that the fitted values of k_i^d are close to the originals when k_i^d is large, but deviate widely for small values; in particular, many k_i^d values are underestimated by one or two orders of magnitude. These deviations result in incorrect estimates for the affinity constants K_{ij} and relaxation times τ_{ij} derived from the rate parameters, as shown in the bottom two panels of figure 2.8.

To clarify why `nls()` fails to converge for small values of the true k_i^d, I plotted the loss W as a function of k_i^d, along a line passing through the fitted value of k_i^a for two simulated probes, as shown in Figure 2.9. Data for the first probe was generated using a large k_i^d value ($10^{-1.5}$), while the second probe had a small k_i^d ($10^{-3.5}$). Since the minimum value of the loss for the second probe is not much different from its limit as $k_i^d \to 0$, iterative procedures for finding the minimum tend to become trapped at small k_i^d values, where the gradient of the loss function is nearly zero. This is the proximate cause for our difficulty in fitting k_i^d for the second probe.

To see why the loss function behaves differently for probes 1 and 2, I plotted the contributions to the loss from individual arrays for the two simulated probes. For both probes, much of the loss contribution at the true (input) parameter values comes from the arrays with the longest hybridization times. Examples for the array with a 16 hour hybridization are shown in Figure 2.10. The lines drawn over the contour plots show the boundaries between various regions of interest. The most important difference between simulated probes 1 and 2 is in the relaxation times for the input k_i^a and k_i^d values, at the concentrations used in the dataset. For probe 1 these are all in the neighborhood of 31 minutes, so that the simulated dataset includes hybridization times $t_j >> \tau_{ij}$ as well as times below τ_{ij}. For probe 2, τ_{ij} ranges from 1581 to 2874 minutes for the concentrations seen in the dataset. Since the longest hybridization time represented in the data is 960 minutes, most of the data points represent values far from equilibrium, with $t_j << \tau_{ij}$.

2.4.4 Penalized least squares approach to parameter fitting

To deal with the problems in fitting the rate parameters of the hybridization kinetics model, I used a simple penalized least squares procedure (abbreviated "pLS", not to be confused with partial least squares, or PLS).

The first step in the procedure is to attempt to fit the log transformed parameters $\phi = \log \gamma$, $\lambda_i^a = \log k_i^a$ and $\lambda_i^d = \log k_i^d$ using `nls()` , as before. When applied to the real *E. faecalis* dataset, `nls()` converged for 351 of the 491 probes. I then used the fitted λ_i^a and λ_i^d values for these probes to construct a bivariate normal "prior" for a penalized likelihood fit. I estimated the mean vector μ and covariance matrix Σ using the `CovSest()` function in the R package `rrcov`, which has various functions for robust covariance estimation.

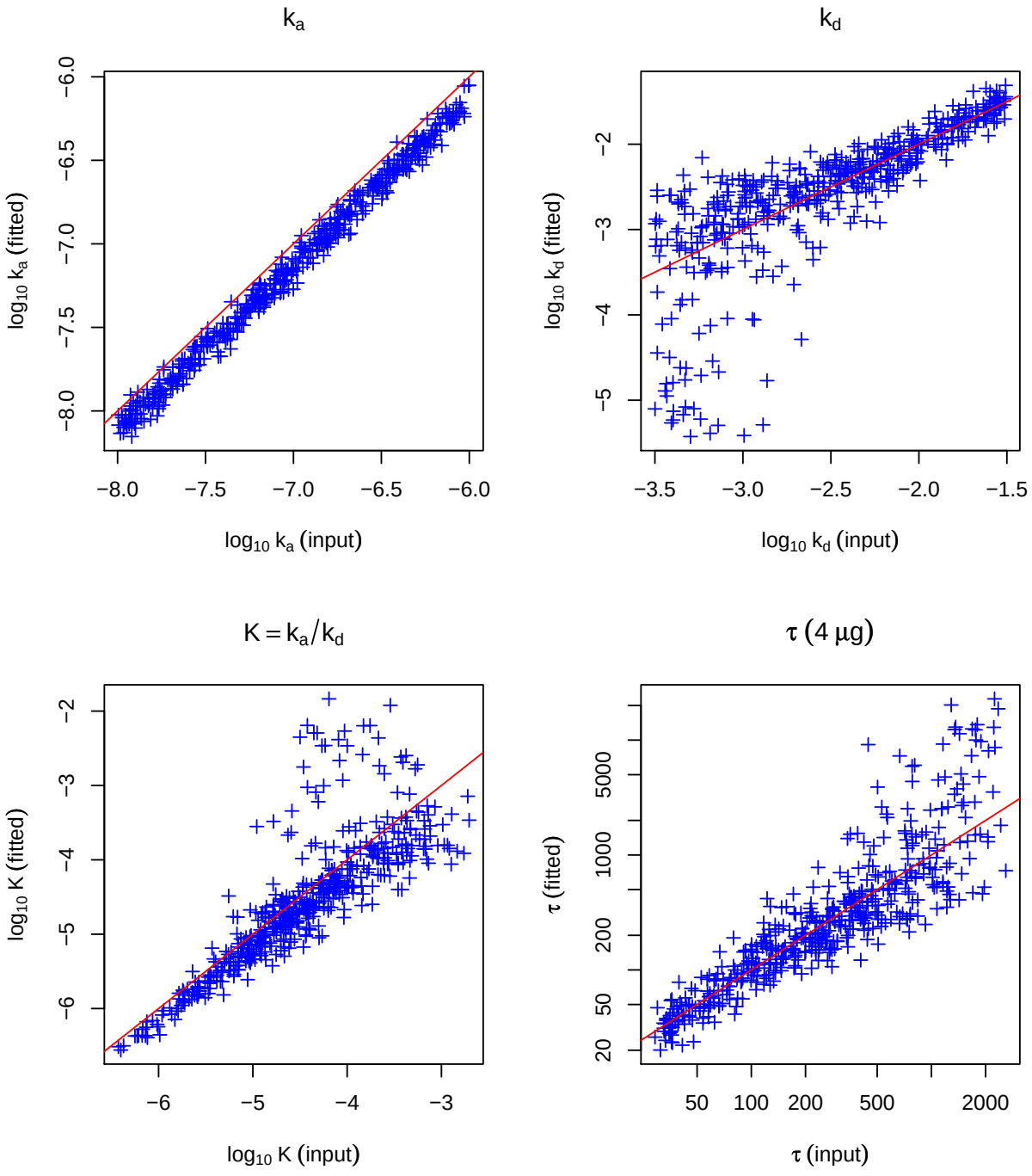

Figure 2.8: Rate constants k_i^a and derived variables (affinity constants and relaxation times) fitted to simulated intensity data, compared to the input values used to generate the data

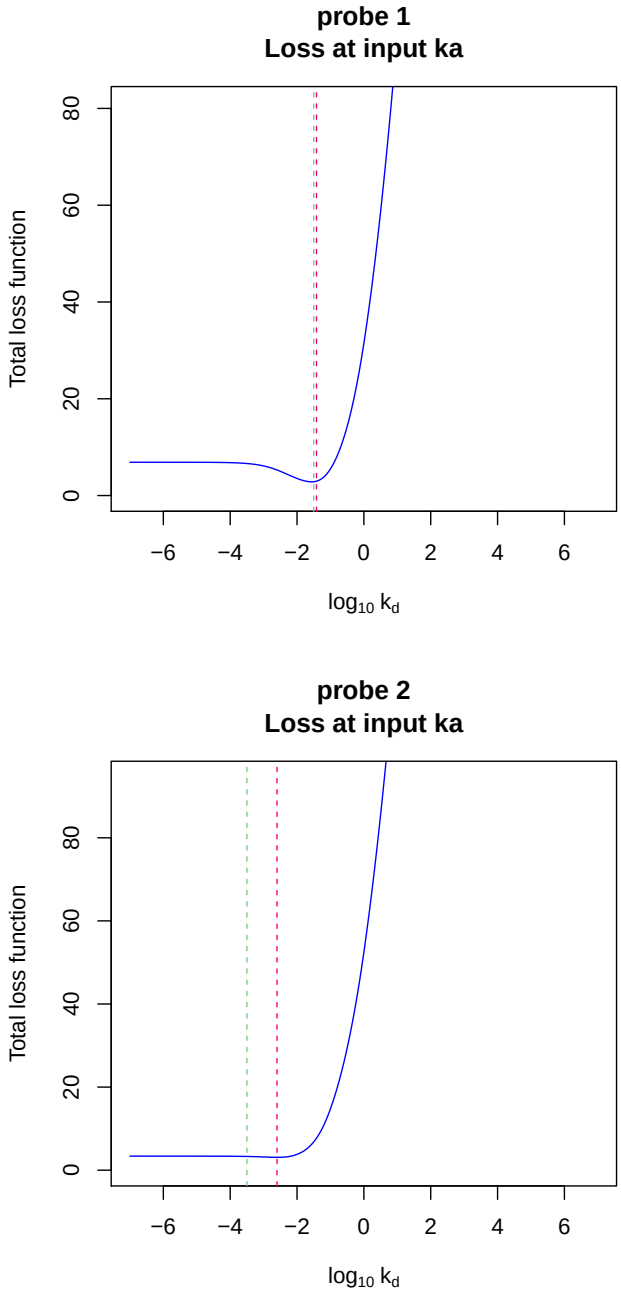

Figure 2.9: Loss as function of k_i^d at input k_i^a, for two simulated probes; input and fitted k_i^d values are plotted as vertical lines.

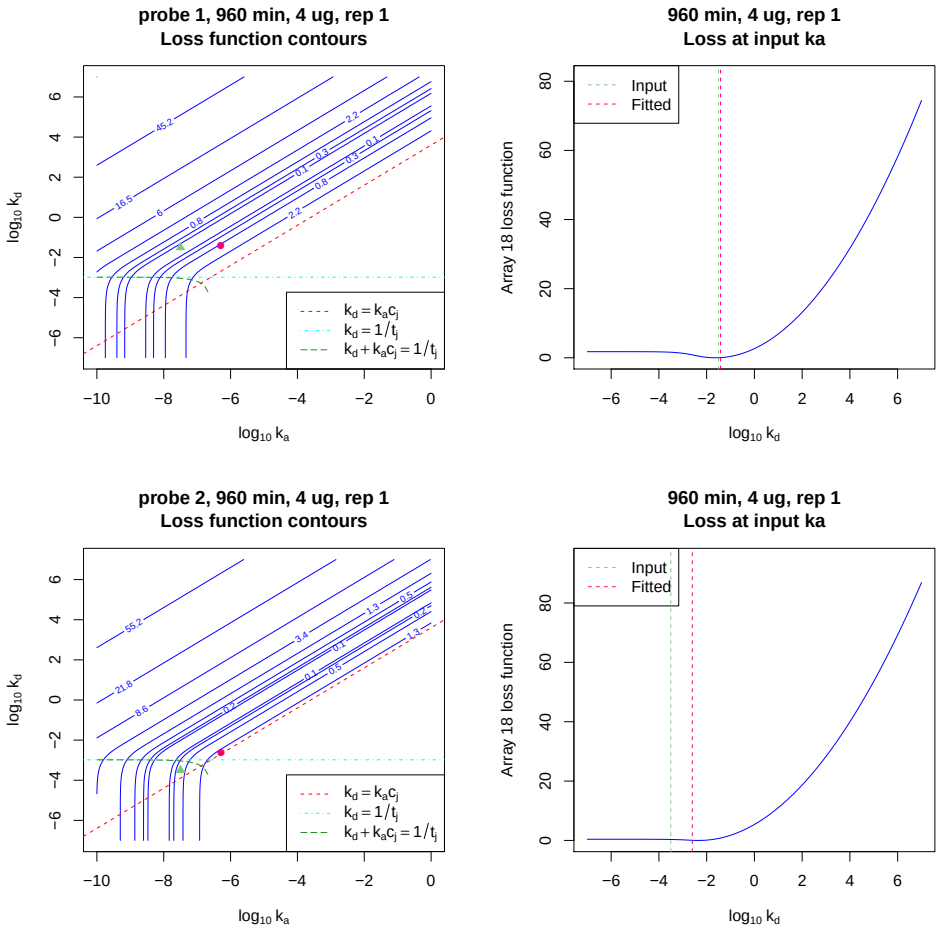

Figure 2.10: Loss function contributions from one array for two simulated probes, as contours for combinations of k_i^a and k_i^d, and as function of k_i^d for the input k_i^a.

With this "prior" in hand, the penalized likelihood is

$$L(\gamma, \lambda, \sigma, \Sigma) \propto \prod_{i=1}^{n} \exp\left[-\frac{1}{2\sigma_i^2}\sum_{j,r}(\log y_{ijr} - \log\gamma - \log\theta_{ij})^2 - \frac{1}{2}(\lambda_i - \mu)^T\Sigma^{-1}(\lambda_i - \mu)\right] \tag{2.22}$$

where $\lambda_i = (\lambda_i^a, \lambda_i^d)^T$. I used one of two methods to estimate the per-probe variance σ_i^2 in the above equation. For probes where `nls()` converged successfully, I estimated σ_i^2 from the residual sum of squares of the `nls()` fit:

$$\widehat{\sigma_i^2} = \frac{1}{n_i - 2}\sum_{j,r}(\log y_{ijr} - \log\gamma - \log\theta(\hat{k_i^a}, \hat{k_i^d}, c_j, t_j))^2 \tag{2.23}$$

When `nls()` failed to converge, I used the median of the $\widehat{\sigma_i^2}$ estimates over the probes where

it did converge.

Since there are no cross terms in the penalized log likelihood, we can maximize it by separately minimizing the term for each probe. I did so using the conjugate gradient method implemented in the R `optim()` function. As before, I set γ initially to the value fitted by `nls()`; after fitting the λ_i^a and λ_i^d parameters, I computed a new maximum likelihood estimate for γ with the other parameters fixed.

The results from fitting the penalized model against real data are shown in Figure 2.11. The penalized fit values are plotted in different colors, depending on whether the corresponding `nls()` fit converged. For comparison, the original parameters fit by `nls()` are shown as green squares. The penalized method successfully fit rate parameters for all 491 probes. Not surprisingly, penalization shrinks many of the fitted values toward their mean.

We see that the $\log k_i^a$ and $\log k_i^d$ values are highly correlated ($r = 0.87$ for the values fit by `nls()` , and 0.93 for the penalized fit). At first blush this might suggest that the k_i^a and k_i^d parameters aren't separately identifiable. However, this distribution is consistent with a limited range of values for the affinity constant $K_i = k_i^a/k_i^d$, as we'll see later.

To test the performance of the pLS fitting procedure against data from known parameters, I generated simulated intensity data for a set of 500 probes, with k_i^a and k_i^d values sampled from a bivariate normal distribution. The parameters of the distribution were the same $\mu_a, \mu_d, \sigma_a, \sigma_d$ and ρ values obtained from the robust covariance fit to the k_i^a and k_i^d values from `nls()` , except that I doubled σ_a and σ_d to get a broader range of rate constants. I generated intensities according to the model of Equation 2.18, using the same combinations of concentration and time covariates as in the real data, and the same number of replicates for each probe. The noise values ϵ_{ijr} were sampled from a $N(0, \sigma^2)$ distribution with $\sigma = 0.53$, which was the average residual standard error from the `nls()` fits to the real data.

I followed the same procedure to analyze the simulated data as I had with the real data. In this case, while `nls()` converged for only 241 of the 500 simulated probes, the pLS method converged for all of them. Figures 2.12 and 2.13 show the fitted k_i^a and k_i^d values plotted against the input values used to generate the data. As before, blue and red points indicate probes for which `nls()` converged or did not. When the change in the fitted value exceeds 0.1, the value fit by `nls()` is marked with a green square, and a cyan arrow shows its difference from the value fit by pLS. A red dashed line indicates the identity mapping.

We see that pLS does an excellent job of recovering the original $\log k_i^a$ values, even when `nls()` fails to converge. When `nls()` does converge, pLS produces nearly the same value as the `nls()` fit. As we saw before, the k_i^d parameters are harder to fit with our dataset, because the experiment design is unbalanced. Nevertheless, pLS fits k_i^d values reasonably close to the inputs for all the probes where `nls()` failed, and (in most cases) improves the fit for the ones for which `nls()` converged.

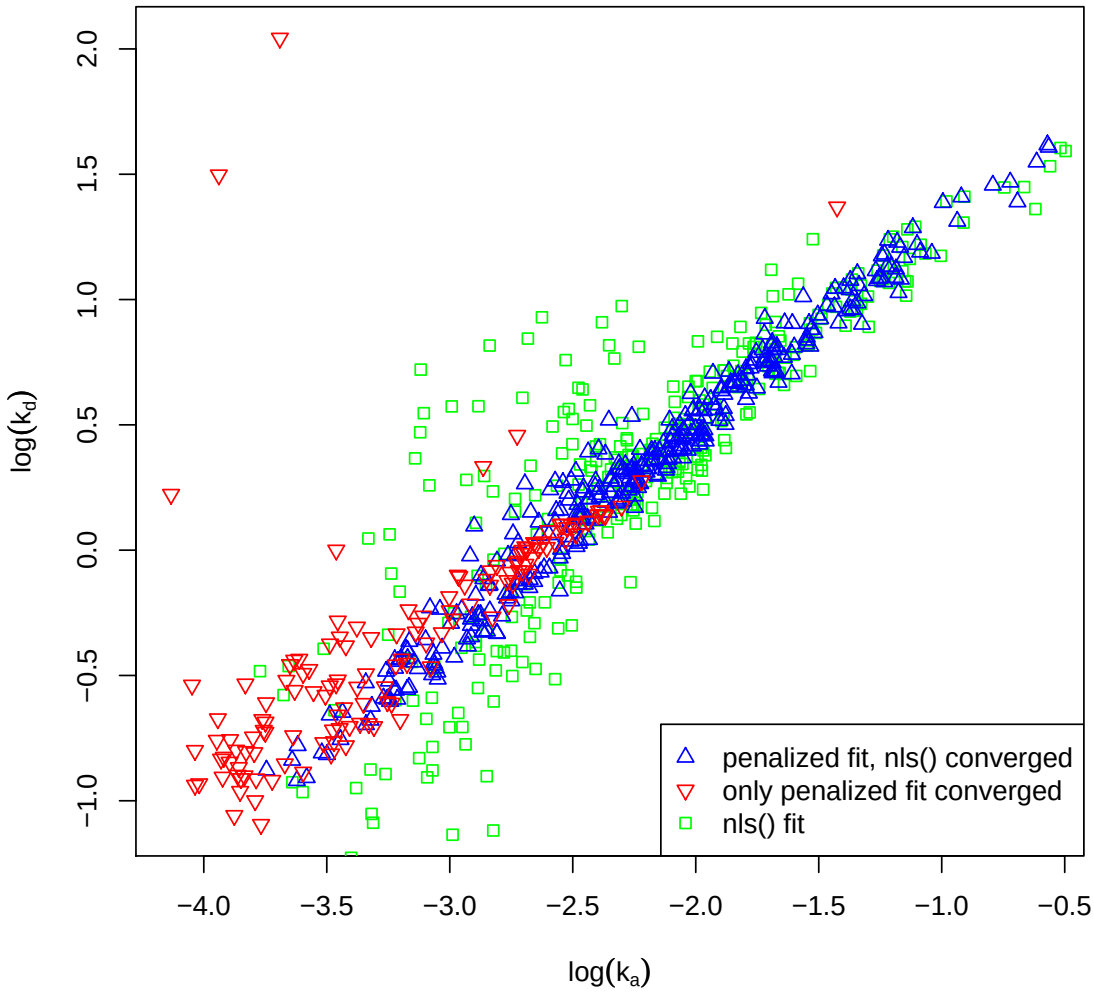

Figure 2.11: Scatter plot of $\log k_i^a$ and $\log k_i^d$ values, fitted by penalized least squares to real data

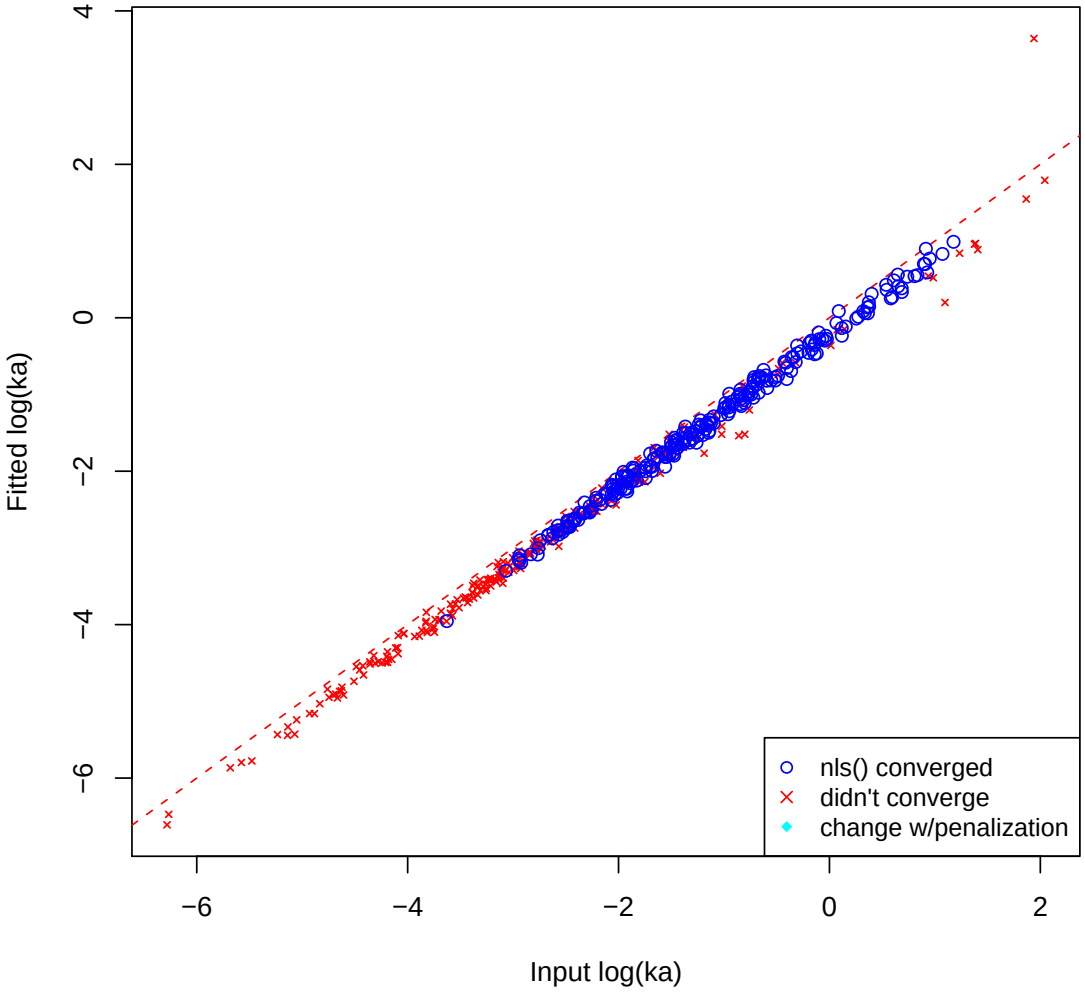

Figure 2.12: Penalized fit values for $\log k_i^a$ vs simulated inputs

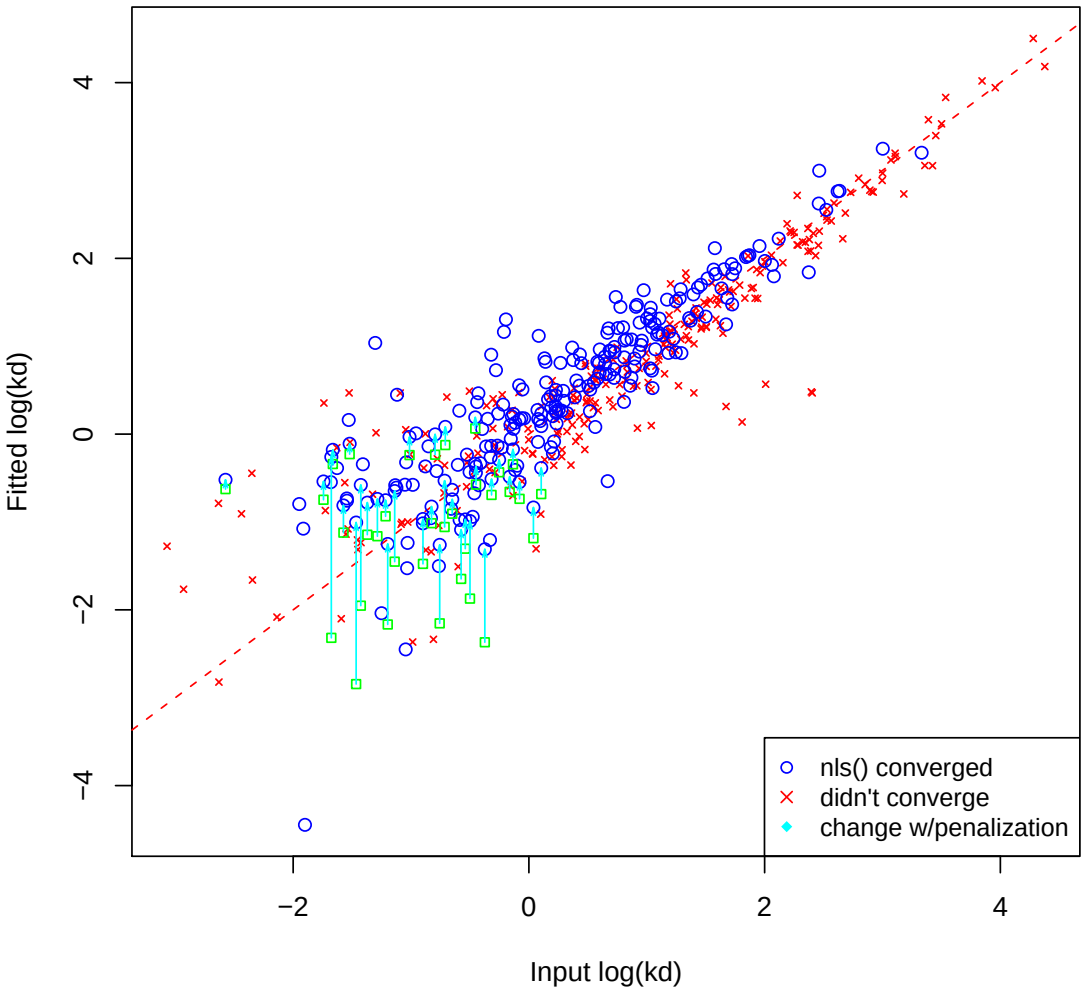

Figure 2.13: Penalized fit values for $\log k_i^d$ vs simulated inputs

2.4.5 Implications of kinetics results for use of equilibrium models

The original goal motivating my study of kinetics was to determine whether the relaxation times for typical probe-target hybridization reactions are short enough to justify the use of equilibrium models of hybridization. From Equation 2.18, we can see that, during array hybridization, a probe's intensity increases over time according to a factor $\phi_{ij}(t) = (1 - \exp(-t/\tau_{ij}))$, which approaches 1 at equilibrium. Here $\tau_{ij} = 1/(k_i^a c_j + k_i^d)$ is a characteristic time, which decreases with increasing DNA concentration, representing the time at which the intensity reaches $(1-e^{-1}) = 63\%$ of its equilibrium level. Figure 2.14 shows two different ways of looking at the distribution of τ_{ij} values, for a typical experiment in which 4 μg of DNA is hybridized to the array. The left panel shows a density curve, based on the τ_{ij} values computed from the fitted k_i^a and k_i^d parameters. The vertical dashed lines on this plot mark the hybridization times used for this dataset. The median τ_{ij} is about 5.4 hours, and nearly all probes (485/491) have characteristic times below the 17 hour hybridization time typically used in our lab.

The right hand panel of Figure 2.14 shows density curves for the predicted equilibrium fraction $\phi_{ij}(t)$, with 4 μg DNA, at each hybridization time. Although most probes are well below their equilibrium intensities at 5 minutes or 1 hour, after 16 hours over 80% of probes are above 80% of their equilibrium levels, and no probes are below half their equilibrium intensities. Therefore, predictions based on equilibrium models should result in acceptable results for most probes, for the hybridization times typically used in our lab.

I calculated equilibrium affinity constants $K_i = k_i^a/k_i^d$ from the rate constants estimated with pLS, and plotted their distribution, as shown in Figure 2.15. We see that the range of affinities is quite narrow, spanning a factor of 3 between the lowest and highest values (excepting two outliers). This explains the apparent correlation between the estimates of k_i^a and k_i^d that I noted in Figure 2.11. The narrow range of affinities is not surprising for this dataset, because all the probes examined here were designed to be perfect matches to parts of the target genome, with lengths varied in order to achieve roughly equal melting temperatures. In the next section, I will estimate affinities directly for a more diverse set of probe/target combinations.

2.5 Equilibrium models of microarray hybridization

As mentioned earlier, affinity constants for pairs of DNA strands hybridizing in solution depend on free energies, which can be predicted using programs such as UNAFold [Markham 08]. These programs are based on frameworks called "nearest-neighbor" (NN) models. NN models are based on experiments showing that the stability of the bond between nucleotides depends not only on the pair of nucleotides forming the bond, but also on the adjacent pairs of nucleotides (the nearest neighbors). The dependency is due to stacking interactions between the neighboring base pairs. In this section, I will discuss the construction of NN models

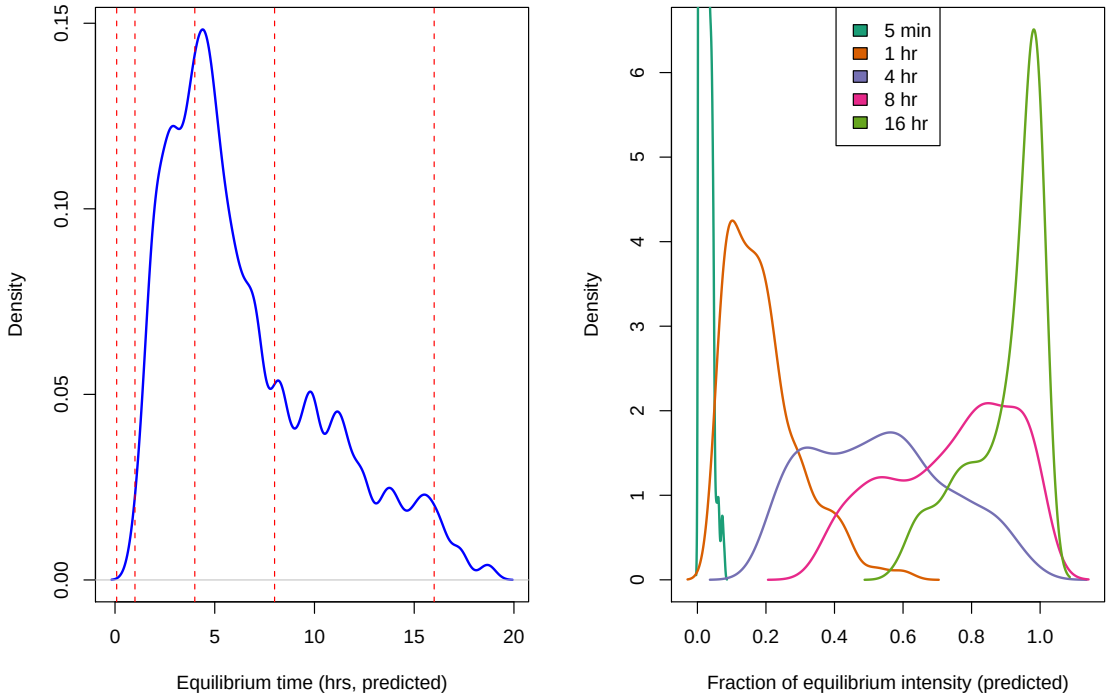

Figure 2.14: Left: Distribution of τ_{ij} computed from rate constants fit by pLS, assuming 4 μg DNA in sample. Right: Distributions of equilibrium intensity fractions with 4 μg DNA, at each time point.

and the problems encountered in applying them to microarrays. I will also describe how I used data sets generated in our laboratory to fit the parameters for a position-dependent NN model, that can predict free energies for microarray probes and targets.

2.5.1 Parameterization of nearest-neighbor free energies

Using data from an extensive set of DNA melting experiments, SantaLucia *et al.* found that the free energy associated with a DNA duplex could be parameterized as a sum of contributions from its component dimer pairs [SantaLucia 04]. For example, the hybridization free energy of the trimer pair GGC/CCG contains contributions from the dimer pairs GG/CC (-1.8 kcal/mol) and GC/CG (-2.2 kcal/mol), for a total of -4.0 kcal/mol. Note that the pair GC/CG contributes more than the pair GG/CC, even though they involve the same number of G-C bonds. Thus, estimates based on dimer pair contributions are much more accurate than values based solely on the GC content.

Programs like UNAFold perform a typical NN model free energy calculation by extracting

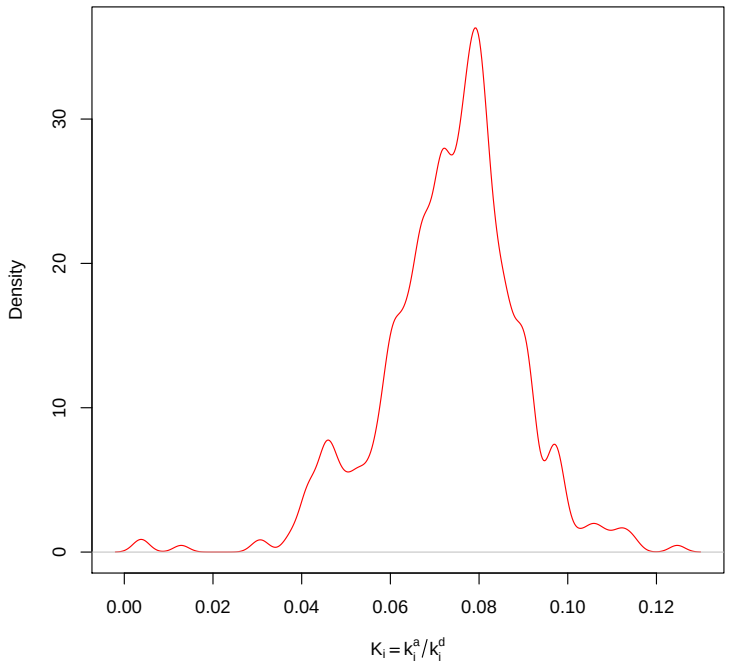

Figure 2.15: Distribution of affinities fit by penalized least squares

dimer pairs d_i from each position of the aligned probe and target sequences. For the dimer pairs in which at least one of the two pairs of aligned bases is a perfect match, it adds the corresponding free energy contributions g_d tabulated in [SantaLucia 04]. Next, the program identifies stretches of two or more mismatched bases in the alignment, which generate internal loops in the hybridized duplex. For each of these, the software adds a corresponding ΔG penalty term, also tabulated in [SantaLucia 04], according to the length of the loop (defined as one plus the number of adjacent mismatched bases). Finally, it adds a constant initiation term for each duplex.

Figure 2.16 shows the terms that are summed to calculate the free energy for an example probe-target duplex. It also shows the reverse complement of the target sequence (labeled "(RC)") to facilitate comparison with the probe sequence. The bases with mismatches and the corresponding dimer pair terms are highlighted in a different color.

To summarize, the parameters of a nearest-neighbor model are the free energy contributions for each unique dimer pair having one or zero mismatched bases; the loop penalty terms for each length of internal loop to be considered; and the initial offset parameter. The contributions for a dimer pair and its reverse complement are assumed to be identical. As a result, there are 10 unique g_d parameters for perfect match dimer pairs, and 48 g_d parameters for dimer pairs with single mismatches. If we allow for internal loops with lengths between 3

```
Probe: 5′-GTGATTGATACGTTTG-3′
Target:3′-CACTAGCTTCGGAAAC-5′
(RC):  GTGATCGTAGCGTTTG
```

probe dimer	target dimer	match type	delta-delta G
		(initiation)	1.97
GT	GT	PM	-1.45
TG	TG	PM	-1.46
GA	GA	PM	-1.31
AT	AT	PM	-0.87
TT	TC	MM	0.34
TG	CG	MM	-0.47
GA	GA	PM	-1.31
AT	AA	MM	0.69
TA	AG	Internal loop	3.20
AC	GC	MM	0.47
CG	CC	MM	-0.11
GT	CT	MM	-0.13
TT	TT	PM	-0.99
		Total:	-1.45 kcal/mol

Figure 2.16: Example nearest-neighbor model free energy calculation for a short probe-target DNA duplex

and 10, and include the initial offset term, this basic "position-independent nearest-neighbor" (PINN) model has 67 parameters in all.

The UNAFold program uses this PINN model, together with the tabulated g_d parameters, to compute a first-order estimate of the free energy for a probe-target DNA duplex. It then refines its estimate by determining the most stable secondary structures of the duplex and accounting for their effects on the free energy. This refinement is usually not required in the context of microarrays, because the secondary structure is constrained by the anchoring of the probe oligo to the array surface and steric hindrance by neighboring oligos.

2.5.2 Discrepancies between solution-phase and microarray free energies

As I indicated in section 2.1.1, solution-phase free energy predictions are about a factor of 10 more negative than those inferred from microarray experiments. Part of the discrepancy can be resolved by fitting new values for the dimer pair free energy contributions g_d, to account for the differences in configurational entropy and diffusion kinetics that are unique to microarray probes. This will be the topic of section 2.5.6 below.

Another source of variation is a relative lack of interaction between the target DNA and the tethered 3' end of the probe oligo; more generally, the contribution of a dimer pair to the probe affinity depends on its position within the probe. We explored this effect by designing an array with perfect match (PM) probes for four bacterial genomes. For selected

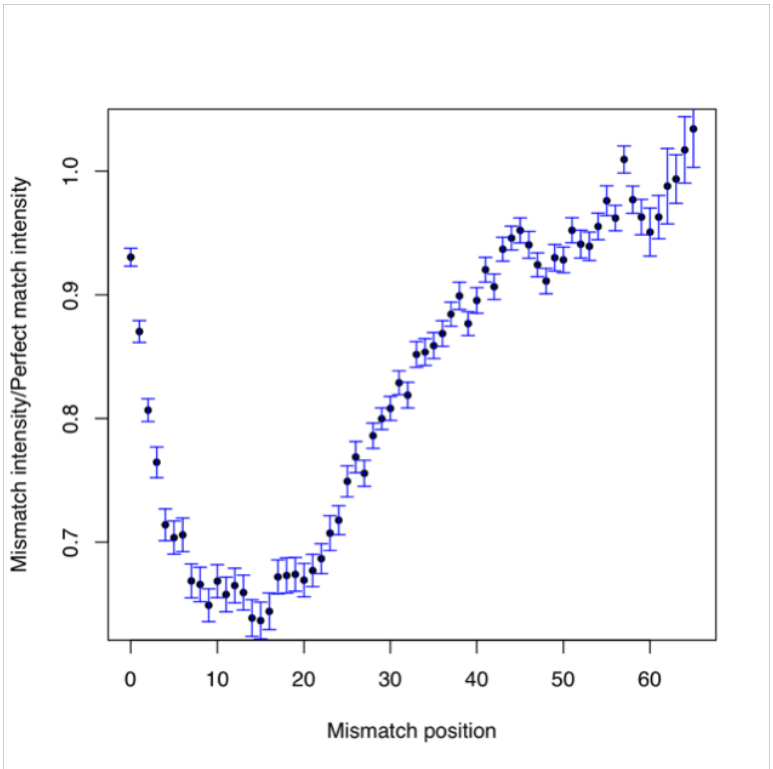

Figure 2.17: Effect of single mismatches on probe intensities as a function of position

probes, a set of mismatch (MM) probes was also included, with mismatched bases placed at different positions along the length of the probe. After hybridizing samples from each of the four target bacteria to the array, I compared intensities between each MM probe and its corresponding PM probe. The results of the comparison are plotted in Figure 2.17. It shows that mismatches have the strongest effect on probe intensity when they occur 15 bases, or one fourth of the way, from the free end of the probe, and have little or no effect when they occur near the tethered end of the probe.

These data suggest that, to construct a predictive model for free energies of microarray probes, we should assign different weights to the contributions from dimer pairs according to their position along the length of the probe. This idea was originally suggested in [Zhang 03], and the corresponding model is called a position-dependent nearest-neighbor (PDNN) model. Methods for fitting the position effect in a PDNN model are described in section 2.5.7.

2.5.3 Tiling array dataset for fitting free energy parameters

To fit parameters for the free energy contributions and position effects, I used a dataset which was part of an experiment to identify single-nucleotide polymorphisms (SNPs) associated with antibiotic resistance in bacteria. The data came from two sets of tiling arrays, in which

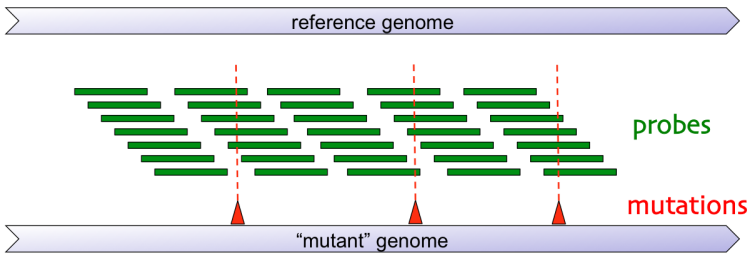

Figure 2.18: Layout of tiling array probes relative to target genomes

probes were selected as 32-to-40 base pair segments of a reference genome sequence; either *Francisella tularensis* strain LVS, or *Bacillus anthracis* strain Sterne. The segments were chosen so that they overlapped by about 1/7 of their length, as shown in Figure 2.18; as a result, each position in the reference genome was covered by between 3 and 7 probes.

To calibrate the arrays, we performed two-color hybridizations in which we labeled the reference strain with a red or green fluorophore and another "mutant" strain of the same species with a green or red fluorophore. We used mutant strains for which we had high-quality genome sequences, so that we knew the positions of hundreds or thousands of isolated single-base mutations relative to the reference sequence. Since each mutation position was overlapped by multiple probes, I was able to separately assess the effects of the bases involved in the mutation and the effect of the position within the probe. The five calibration arrays yielded a total of 177,816 pairs of intensities from probes overlapping known mutations.

2.5.4 The three-state Langmuir model

To fit the free energy parameters to a two-color microarray dataset, we must use a version of the Langmuir model that can handle the two different sets of labeled targets. Under this model, a probe oligo can be in one of three states: unbound, bound to one target, or bound to the other target.

Recall that, for probe i on array a, I model the probe intensity for channel (dye) j as

$$y_{iaj} = b_{aj} + \gamma_{aj}\theta_{iaj}$$

where b_{aj} is the background, γ_{aj} is an array- and channel-specific scale factor, and θ_{iaj} is the fraction of oligos in the probe bound to a target labeled with dye j. For the sake of brevity, I'll use the index j or the combined index a, j henceforth to indicate the target labeled with dye j on array a; the meaning should be clear from the context. The bound fraction θ_{iaj} is given by the modified Langmuir equation,

$$\theta_{iaj} = \frac{y_{iaj} - b_{aj}}{\gamma_{aj}} = \frac{K_{ij}c_{aj}}{1 + K_{i1}c_{a1} + K_{i2}c_{a2}} \tag{2.24}$$

where c_{aj} is the *molar* concentration of target j and K_{ij} is the affinity constant. K_{ij} can be expressed in terms of the free energy for probe i binding target j, ΔG_{iaj}, along with a

constant parameter K_0:

$$K_{ij} = K_0 e^{-\Delta G_{iaj}/RT} \tag{2.25}$$

2.5.5 Applying the Langmuir model to log ratio data

In one set of experiments with whole genome tiling arrays, the two targets hybridized to each array are a reference strain and a "mutant" strain of the same species, both with known genome sequences. On each array, one target was labeled with Cy3 and the other with Cy5. The probes are overlapping subsequences of the reference genome, so that each has a perfect match somewhere within the genome sequence, and each position in the genome is covered by between 2 and 7 probes. The mutant genomes each contain between 400 and 8000 isolated SNPs relative to the reference genome. Thus, there are two classes of probes for each array: those with a perfect match in both genomes, and those overlapping a single mismatch in the mutant genome. We will call these PM and MM probes, respectively. (Note that these terms have different meanings than for Affymetrix arrays, in which "PM" and "MM" denote different probes for the same subsequence. Here, both PM and MM probes have perfect matches in the reference genome.)

Let $M_{ia} = \log(y_{ia1} - b_{a1})/(y_{ia2} - b_{a2})$ be the usual log ratio of background-corrected intensities, where indices 1 and 2 refer to the mutant and reference targets, respectively. Under the 3-state Langmuir model, the log ratio reduces to a nice linear expression:

$$\begin{aligned}
M_{ia} &= \log \frac{\gamma_{a1}\theta_{ia1}}{\gamma_{a2}\theta_{ia2}} \\
&= \log \frac{\gamma_{a1}}{\gamma_{a2}} + \log \frac{K_{i1}c_{a1}}{K_{i2}c_{a2}} \\
&= \log \frac{\gamma_{a1}}{\gamma_{a2}} + \log \frac{c_{a1}}{c_{a2}} + \log \frac{e^{-\Delta G_{ia1}/RT}}{e^{-dgit/RT}} \\
&= \log \frac{\gamma_{a1}}{\gamma_{a2}} + \log \frac{c_{a1}}{c_{a2}} + \frac{1}{RT}(\Delta G_{ia2} - \Delta G_{ia1})
\end{aligned}$$

In these experiments, the two targets were always different strains of the same species, with nearly identical genome sizes, applied at the same mass concentration. Thus, the molar concentrations were almost identical, so the $\log(c_{a1}/c_{a2})$ term can be dropped, leaving us with the following expression for the log intensity ratio:

$$M_{ia} = \log \frac{\gamma_{a1}}{\gamma_{a2}} + \frac{1}{RT}(\Delta G_{ia2} - \Delta G_{ia1}) \tag{2.26}$$

2.5.6 Fitting free energy contributions under the position-independent nearest-neighbor model

As I showed with an earlier, smaller data set, the effect on the affinity of a mismatch between the probe and target sequences depends on the position of the mismatch within the probe. At some point we need to estimate this position effect, but we have other parameters to estimate first. To make things simpler, we'll fit a position-independent model using only data points in which the mismatch falls within the middle 17 bases of the probe. Our past results suggest that the position effect is nearly constant within this range.

Under a position-independent nearest-neighbor (PINN) model of probe affinities, the free energy for probe i hybridizing to target j is a sum of contributions from the set of dimer pairs $D(i,j)$ in the alignment of the probe and target:

$$\Delta G_{iaj} = \sum_{d \in D(i,j)} g_d$$

For a MM probe, the difference in free energies between the mutant and reference targets results from replacing two perfect match dimer pairs with two mismatch pairs (unless the mismatch falls at the beginning or end of the probe; I exclude such cases from the subsequent analysis):

$$\Delta\Delta G_{ia} = \Delta G_{ia1} - \Delta G_{ia2} = g_{mm_{ia1}} + g_{mm_{ia2}} - g_{pm_{ia1}} - g_{pm_{ia2}} \tag{2.27}$$

where mm_{iak} and pm_{iak} denote the MM dimer pairs created and the PM pairs destroyed by the presence of the mismatch. Combining equations 2.26 and 2.27, I model the log intensity ratio by

$$M_{ia} = \log \frac{\gamma_{a1}}{\gamma_{a2}} - \frac{1}{RT}\left(g_{mm_{ia1}} + g_{mm_{ia2}} - g_{pm_{ia1}} - g_{pm_{ia2}}\right) + \epsilon_{ia} \tag{2.28}$$

with an error term $\epsilon_{ia} \sim N(0, \sigma^2)$.

I fit the parameters of the model in two stages. First, I use the log ratio data for PM probes, for which $\Delta\Delta G_{ia} = 0$, to estimate the ratio of the scale factors in the two channels of each array:

$$\widehat{\log \frac{\gamma_{a1}}{\gamma_{a2}}} = \frac{1}{n_{PM(a)}} \sum_{i \in PM(a)} M_{ia} \tag{2.29}$$

The PM probe data can also be used to estimate the noise parameters σ_a^2:

$$\widehat{\sigma_a^2} = \frac{1}{n_{PM(a)} - 1} \sum_{i \in PM(a)} \left(M_{ia} - \widehat{\log \frac{\gamma_{a1}}{\gamma_{a2}}}\right)^2 \tag{2.30}$$

Secondly, I write equation 2.28 in matrix form and solve the resulting linear model. Let $\mu_{ia} = M_{ia} - \log \frac{\gamma_{a1}}{\gamma_{a2}}$, and let μ be the vector of μ_{ia} values for n_{MM} (MM probe, array) tuples. Let X be an $n_{MM} \times 58$ matrix, whose columns correspond to the 58 possible dimer pairs; X_{kd} is the number of dimer pairs of type d added (if positive) or removed (if not) in the

(MM probe, array) tuple represented by the combined index k. Finally, let g be the vector of free energy contributions for the 58 dimer pairs. Then our model becomes

$$\mu = Xg + \epsilon \tag{2.31}$$

We plug in our estimates of $\widehat{\log \frac{\gamma_{a1}}{\gamma_{a2}}}$ to estimate μ. Since X is known, we can solve equation 2.31 by ordinary least squares to determine the dimer pair free energy parameters g_d.

There is only one problem with the above strategy, which is that X is rank deficient. It is not obvious how a matrix with 166,000 columns and 58 rows could be so. The reason this happens is that X contains only 192 unique row vectors, corresponding to the 64 possible base triplets centered at each SNP, combined with 3 possibilities for the mismatch base. Furthermore, the 192 row vectors can all be generated by linear combinations of 51 basis vectors. Therefore, X has rank 51, though it has 58 columns, and equation 2.31 has an infinite number of solutions.

I tried two approaches to select a single solution for g. One was QR decomposition with pivoting, using the R `qr.coef()` function. For an $(n \times p)$ design matrix X of rank r, `qr.coef` simply zeros the last $p-r$ coefficients; I arranged the columns of the X matrix so that these last $p-r$ dimer pairs were the ones for which SantaLucia's g_d estimates were closest to zero.

The other approach was to choose the minimum norm solution, i.e. the one that minimizes $\parallel g \parallel^2$ while also minimizing $\parallel X\mu - g \parallel^2$. This can be found by computing the singular value decomposition $X = UDV^T$, with U and V orthonormal and $D = diag(d_1, d_2, \ldots d_r, 0, 0, \ldots 0)$. If W is the matrix consisting of the first r columns of U, and $\Lambda = diag(d_1, d_2, \ldots d_r)$, it can be shown [Golub 89] that the minimum norm solution is $g_{min} = V\Lambda^{-1}W^T\mu$.

Figures 2.19 and 2.20 compare the solutions resulting from these two approaches. Although the parameter estimates are similar for most dimer pairs, we see significant differences for a few of them, including the perfect match pairs GT/GT and GC/GC. We can obtain more definitive estimates by fitting the nonlinear model (equation (2.24)) to the raw log intensity data (rather than log ratios), using either the QR or the minimum norm solution as initial values. Before we proceed to that step, however, we need to deal with the position effect.

2.5.7 Fitting the position effect

In Zhang's position-dependent nearest-neighbor (PDNN) model [Zhang 03], the free energy of probe i binding to target j on array a is a weighted sum of the dimer pair contributions at each position:

$$\Delta G_{iaj} = \sum_{l=1}^{n_i} w_l g_{d(l,i,j)} \tag{2.32}$$

where n_i is the number of dimers in probe i, $d(l, i, j)$ is the dimer pair at position l and w_l is the weight factor. In this model, probes are assumed to be 25-mers, and a separate w_l parameter is fitted for each of the 24 positions.

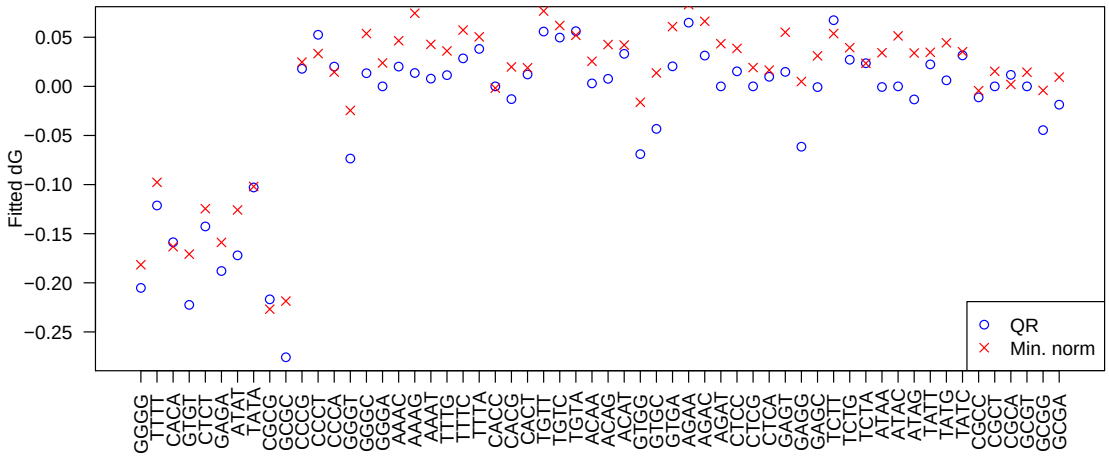

Figure 2.19: Comparison of fitted g_d free energy parameters by dimer pair, using QR decomposition with pivoting vs minimum norm solution

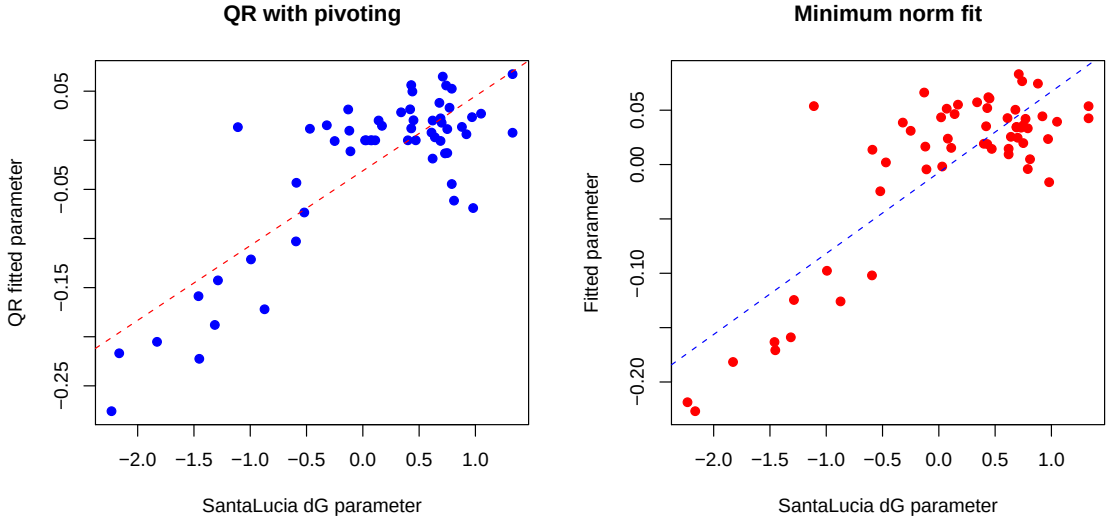

Figure 2.20: Comparison of g_d free energy parameters fitted using QR and minimum norm strategies to SantaLucia solution-phase parameters

This approach does not work well for the long, variable-length oligos used on our NimbleGen arrays, since it requires too many parameters to be fitted. Instead, I decided to fit weighted sums of cubic B-spline basis functions, evaluated at the fractional positions $t_l = l/L_i$, where L_i is the length of the probe. Each basis function $f_k(t)$ is associated with a knot position $\kappa_k, k = 1, \ldots \nu$, with $0 \leq \kappa_k \leq 1$. The number of knots ν can be selected as the minimum value needed to get a reasonable fit. This greatly reduces the number of parameters in the model. The general form for the position effect is then a function

$$\phi(t) = \sum_{k=1}^{\nu} w_k f_k(t) \tag{2.33}$$

I considered two methods for incorporating the position effect into my models. The simpler approach is to add $\phi(t)$ to the expression for the response variable, and fit the weights w_k to the residuals from fitting the position-independent model. For log ratio data, the only position value that is important is t_{ia}, the location of the SNP within probe i on array a. We add the position effect by modifying equation (2.31):

$$M - \log \frac{\gamma_1}{\gamma_2} - Xg = Fw + \epsilon \tag{2.34}$$

where F is an $n \times \nu$ B-spline basis matrix with $F_{iak} = f_k(t_i a)$, w is a vector of ν weight values to be fitted, ia is a combined index for the probe and array together, γ_1 and γ_2 are vectors containing the appropriate γ_{aj} values for the mutant and reference strain arrays in each row, and the error term $\epsilon \sim N(0, \sigma_\epsilon^2)$. To fit the weights, I equate the left side of equation (2.34) to the vector of residuals R from the position-independent fit, and use ridge regression to find w that minimizes a penalized spline objective function:

$$\psi(w, \alpha) = \frac{1}{\sigma_\epsilon^2} \parallel R \parallel^2 + \frac{\alpha}{\sigma_\epsilon^2} \parallel w \parallel^2 \tag{2.35}$$

In equation (2.35), α is a penalty parameter which we adjust to avoid overfitting, using a generalized cross-validation (GCV) method [Ruppert 03]. I implemented a Demmler-Reinsch orthogonalization algorithm described by Ruppert et al. which automatically and rapidly tests many candidate values to find the optimum α and the corresponding weights. Figure 2.21 shows the fitted position effect $\phi(t)$ for a penalized spline function with 13 knots and the optimum penalty parameter $\alpha = 225$. Although an additive position effect improves the fit to the log intensity ratio data for MM probes, it makes more intuitive sense to apply a *multiplicative* effect to the free energy contributions from each position in the probe. Otherwise, the model predicts a position-dependent change in intensity for PM probes, for which the change in free energy $\Delta\Delta G = 0$. A multiplicative position effect function is also closer in spirit to the PDNN model proposed by Zhang. We can construct such a function by fitting splines to the ratio of the free energy change inferred from log intensity ratios to the free energy changes predicted by the PINN model. We rewrite equation (2.28)

$$M_{ia} = \log \frac{\gamma_{a1}}{\gamma_{a2}} - \frac{1}{RT} \Delta\Delta G_{ia} \phi(t_i a) + \epsilon_{ia} \tag{2.36}$$

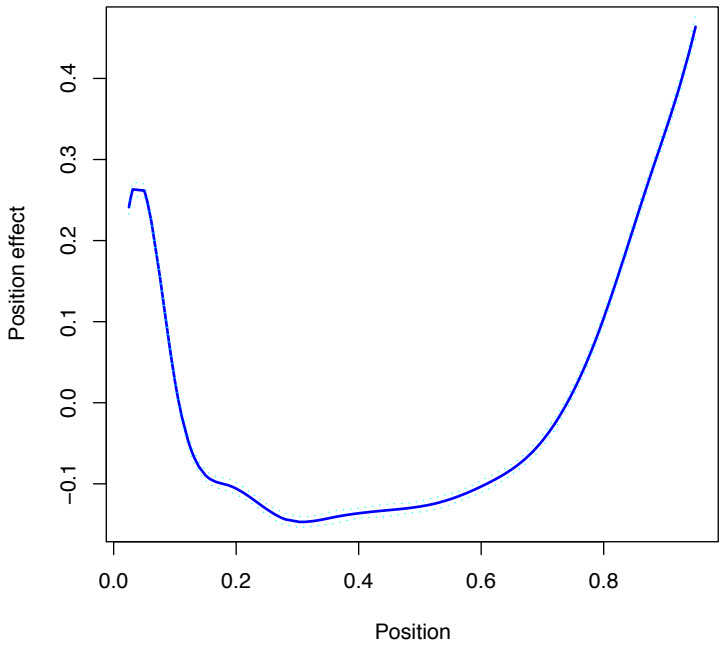

Figure 2.21: Fitted additive position effect function for log ratio data, using a B-spline basis with 13 knots

to get

$$\phi(t_{ia}) = -RT\frac{M_{ia} - \log\frac{\gamma_{a1}}{\gamma_{a2}}}{\Delta\Delta G_{ia}} + \epsilon' \tag{2.37}$$

I fit a penalized spline to the right hand site of equation (2.37), plugging in the observed M_{ia} values and the fitted or predicted values for γ_{a1}, γ_{a2}, and $\Delta\Delta G_{ia}$ from the PINN analysis. The results (with 13 knots) are shown in Figures 2.22 and 2.23. Figure 2.22 shows the fitted function together with the input data; Figure 2.23 shows the function by itself, on a different y-axis scale. Although the input data is extremely noisy, there are more than enough points to yield an estimate for the "average" position effect. One may question the value of including an effect that is so small compared to the noise in the data. The value will become clearer when we proceed to the next step, which is to predict probe intensities (rather than log ratios).

2.5.8 Fitting free energy parameters and predicting intensities under the position-dependent nearest-neighbor model

We now have initial estimates for all the parameters needed to predict intensities under the PDNN model: background and scale factors for each channel on each array, spline weights for the position effect, and the 58 dimer pair free energy parameters g_d. Since the g_d parameters were estimated from a rank deficient system of equations and without accounting for the

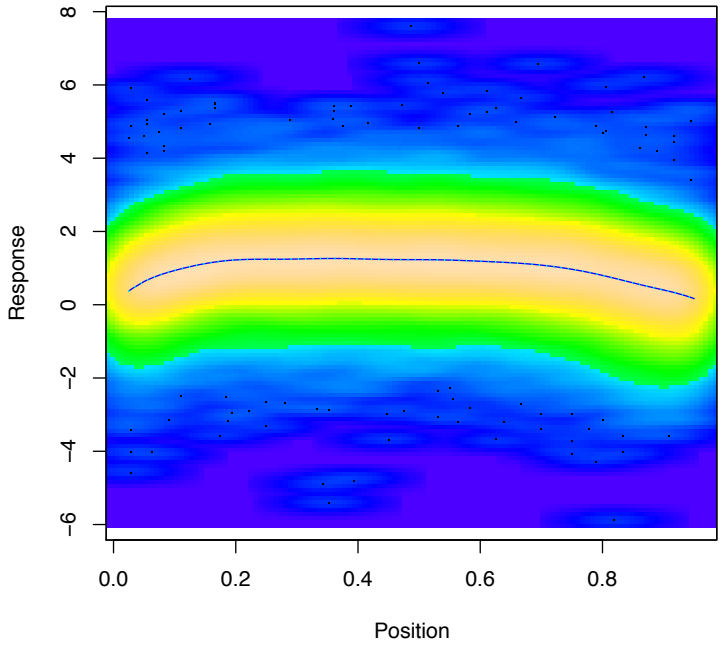

Figure 2.22: Smoothed scatterplot of ratio of inferred to predicted free energy changes from log ratio data, and penalized spline fit for multiplicative position effect with 13 knots

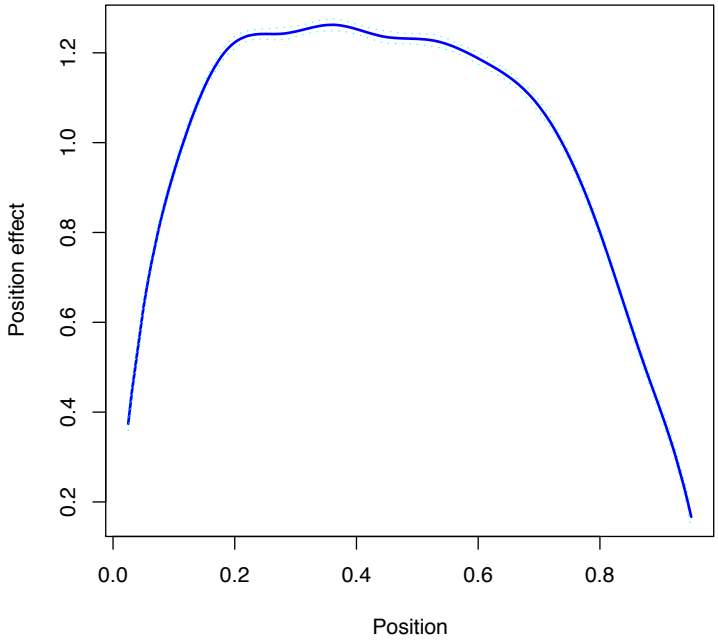

Figure 2.23: Fitted multiplicative position effect function for free energy changes, as in Figure 2.22, scaled to show variation with position

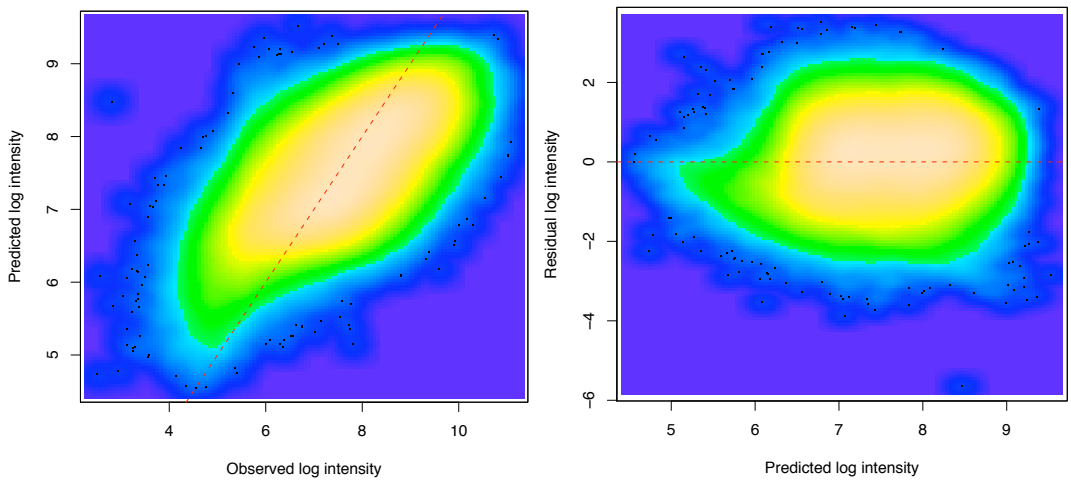

Figure 2.24: Predicted log intensities plotted against observed values, and residuals vs predicted intensities, based on initial ΔG parameters estimated from log ratio data

position effect, I expected that the estimates could be improved further by refitting the full PDNN model to the log intensities, rather than log intensity ratios between reference and mutant targets. Before doing so, I plotted the intensities and residuals predicted from the existing parameters against the observed data, as shown in Figure 2.24, to provide a baseline for later comparison.

Because the PDNN model for intensities is nonlinear, and much more complex than the model for log ratios, I only refitted the ΔG contributions g_d for the 10 perfect match dimer pairs, rather than all 58 g_d parameters. I ran `optim()`with the conjugate gradient method for 200 iterations, and plotted the RSS against the iteration count to assess convergence, as shown in Figure 2.25. After the first 150 iterations, there was very little additional improvement in the residuals. Figure 2.26 compares the refitted PM dimer g_d parameters to their initial values. The overall effect of refitting is to decrease g_d for dimers containing G's or C's, and to increase it for those with A's and T's only.

Figure 2.27 shows plots of predicted vs observed log intensities and residuals vs predicted values, based on the refitted parameters. The main difference from the predicted intensities shown in Figure 2.24 is that the range of predicted values is broader under the new parameters, and thus closer to that of the observed values.

2.5.9 How much variance remains unexplained?

To get a sense of whether the complexities of the PDNN model were worth the effort involved to fit it, I compared fitted vs observed log intensities from the tiling array dataset using a succession of models, and computed the residual variances, as shown in Table 2.3. For this particular dataset, the PDNN model explains about half of the total variance; it is a modest

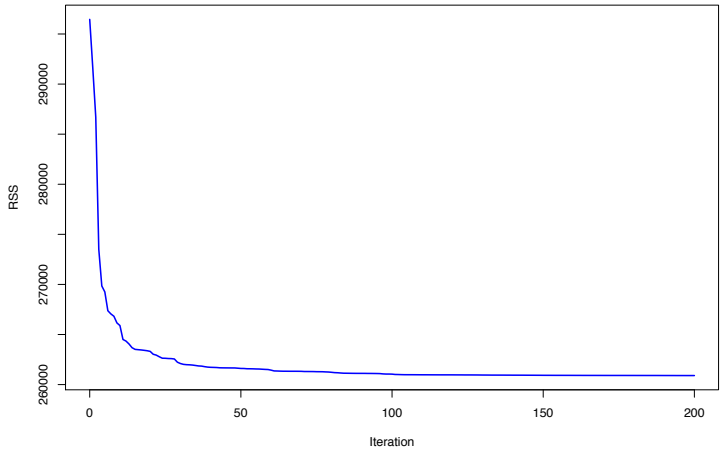

Figure 2.25: Residual sum of squares as a function of iteration count, for fitting perfect match dimer pair free energies to log intensity data.

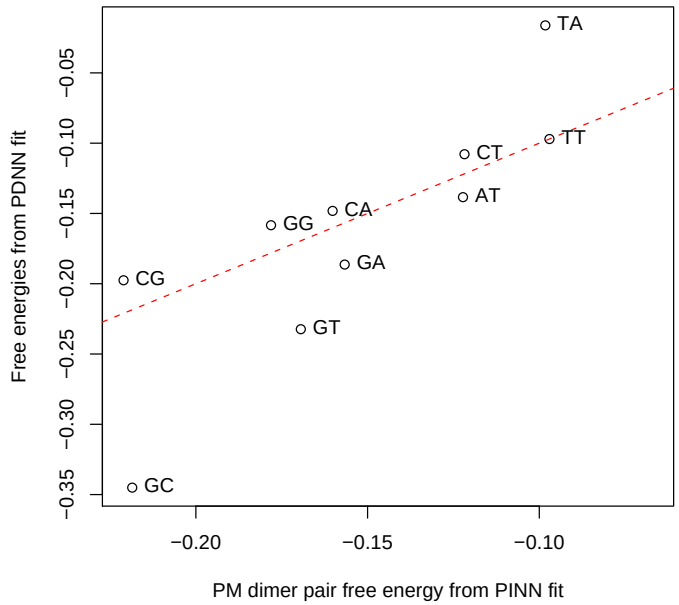

Figure 2.26: Refitted ΔG parameters for PM dimer pairs, plotted against initial values estimated from log ratios

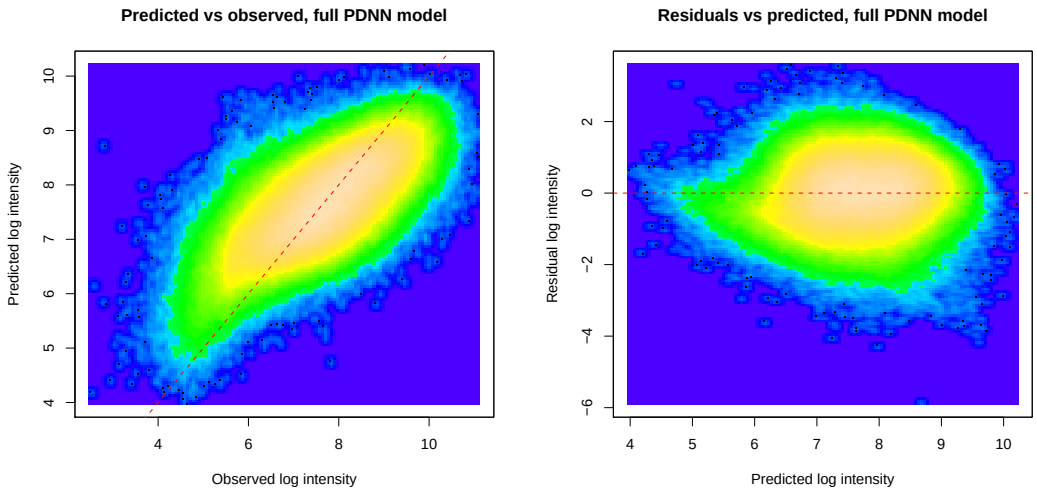

Figure 2.27: Predicted vs observed values, and residuals vs predicted values, of log intensities calculated using ΔG parameters fitted to the full PDNN model.

improvement over the position-independent model, and a substantial improvement over just fitting a mean log intensity for each channel on each array.

	Model	DF	RSS	Variance	StdError
1	None - intercept only	1	487638.07	1.38	1.17
2	Mean effect for array and channel only	10	373691.21	1.06	1.03
3	Position-independent NN model	69	265125.67	0.75	0.87
4	Position-dependent NN model	82	244676.45	0.69	0.83

Table 2.3: Residual sum of squares, variance and SE from the tiling array data, using a succession of models of increasing complexity

The question then arises, how much more of the variance could be explained by an even better predictive model? One way to get a sense of the limits is to compare intensities from the two channels on each two-color array, for probes that are perfect matches to the genomes of both samples, and compute residual variances. In this case, we expect differences because of the dye effect, but this should appear as a constant factor for each array, and thus as an offset in the log intensities. Figure 2.28 shows the logged background-corrected intensities from the two channels plotted against one another, for 4 of the 5 arrays. I fitted a line with unit slope and variable intercept to these data, and computed residuals and the corresponding variances. The results for each array are shown in Table 2.4.

A more realistic estimate for the minimum variance one can achieve can be obtained by comparing intensities for the same probe on two replicate arrays hybridized to the same

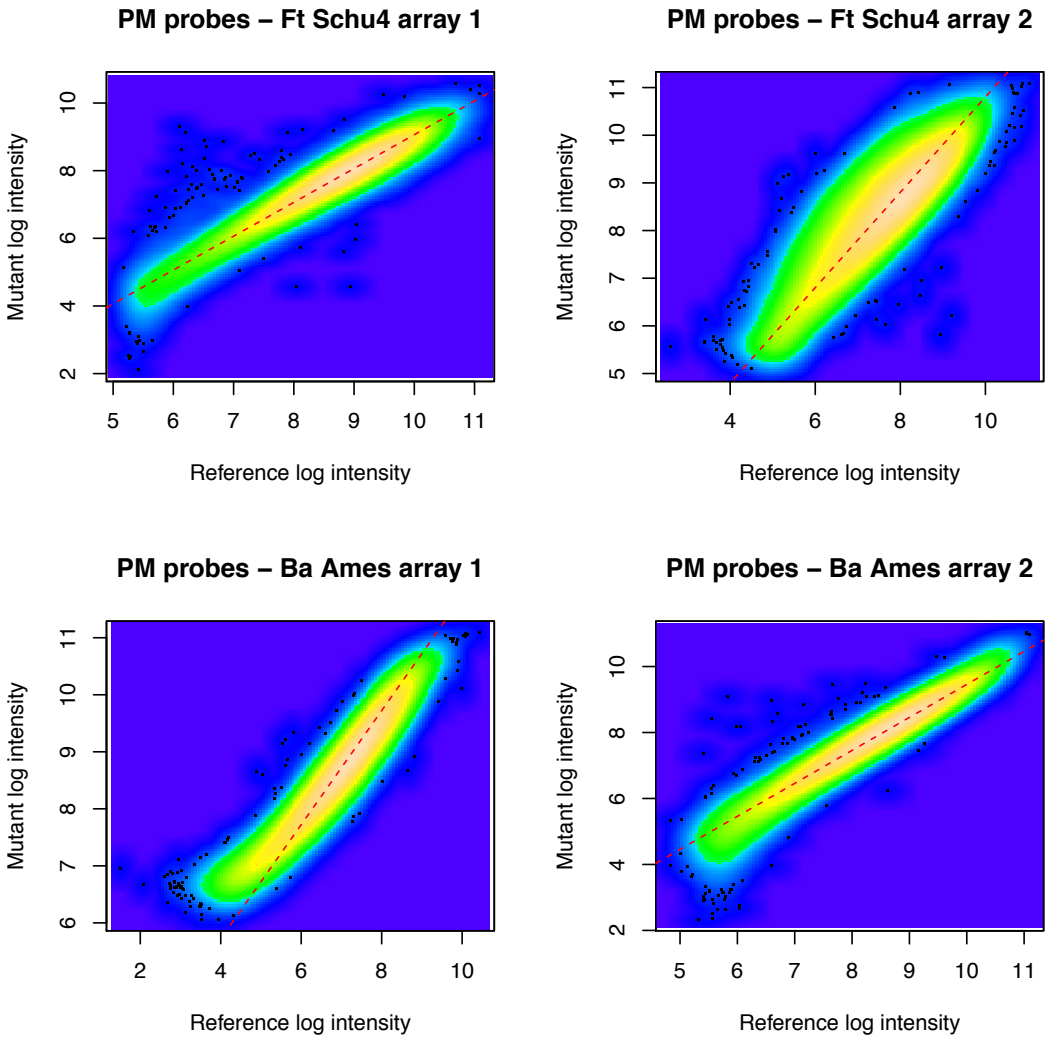

Figure 2.28: Smoothed scatterplots of background corrected intensities from two channels on each tiling array

	Array ID	Variance	StdErr	N	Scale
1	Schu1	0.06	0.25	320172	0.39
2	Schu2	0.22	0.47	320172	2.23
3	U112	0.12	0.34	226563	0.60
4	Ames1	0.09	0.31	355036	5.59
5	Ames2	0.05	0.22	355036	0.58

Table 2.4: Within-array variances for perfect match probes against both strains on array

strain. The replicates are dye-swapped, so I expected to see scale differences due to both dye and array effects. Again, these are fitted to a line with unit slope and variable intercept, as shown in Figure 2.29 and Table 2.5.

We see that the random variations in intensity for the same probe with the same sample between two arrays can be substantial, amounting to as much as half of the residual variance seen with the PDNN model. From this I conclude that, although it may be possible to do better with a more complex predictive model, there are unknown sources of variation in the intensity measurements that cannot be explained by a model based on probe and target sequences. In spite of these limitations, the PDNN model should have enough predictive value to be useful for solving the inverse problem, of inferring target concentrations in a sample, given a set of microarray probe intensities. This problem will be the subject of the next chapter.

	Strain	Variance	StdErr	N	Scale
1	Ft LVS	0.33	0.58	320172	0.35
2	Ft Schu4	0.14	0.38	320172	1.98
3	Ba Sterne	0.23	0.48	355036	3.79
4	Ba Ames	0.25	0.50	355036	0.39

Table 2.5: Between-array variances for perfect match probes

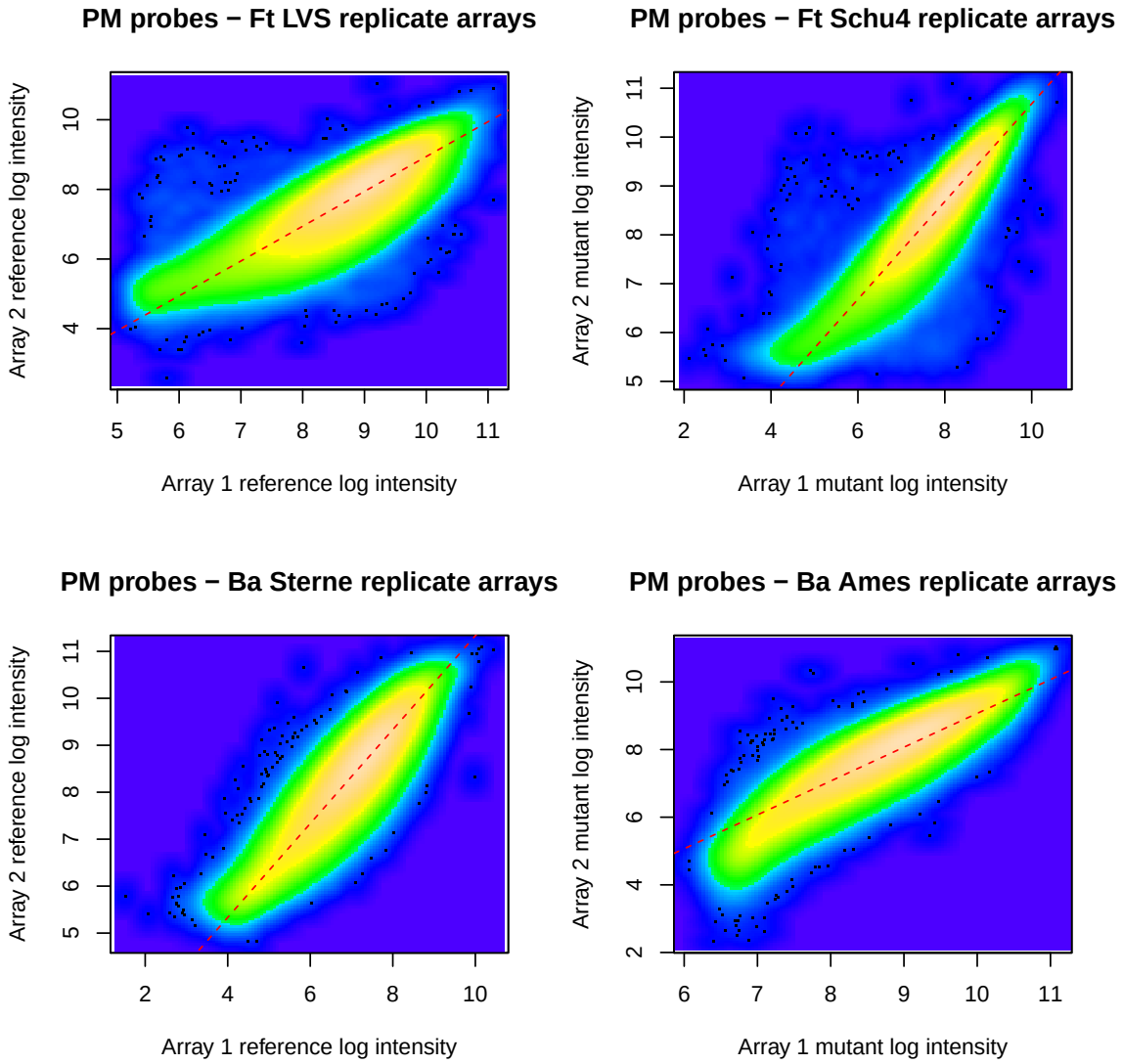

Figure 2.29: Smooth scatterplot of background corrected intensities from same sample strain in replicate pairs of tiling arrays

Chapter 3

Target identification and quantitation from microbial detection array data

3.1 Introduction

The PDNN model developed in the previous chapter allows us to estimate hybridization free energies and probe affinities from alignments of probe and target sequences. Using the affinities, together with the Langmuir equation, we can predict the intensity of a probe matching part of a target sequence, when a known concentration of target DNA is hybridized to an array. In this chapter, I'll use the predictive model to solve an inverse problem: given the probe intensities from a sample of unknown composition, determine the targets present and their concentrations. I'll refer to this as the "identification/quantitation" or "I/Q" problem.

A straightforward approach to this problem is to define a density function for each probe's observed intensity, with parameters dependent on target concentrations; construct a likelihood function based on these densities, and maximize it with respect to the concentrations. Before attempting this, I needed a way to estimate the array- and scan-specific scale factor γ_a relating the fraction of bound oligos in a feature to its intensity. This was best accomplished using intensities from a set of positive control probes, complementary to DNA spiked in at a known concentration. To make this work, I had to have accurate affinity estimates for the positive control probes. Section 3.2 details the process by which I estimated these affinities and the algorithm for inferring scale factors.

Next, I had to develop a procedure to solve a more restricted quantitation problem: Assuming the sample hybridized to the array contains target DNA from one or more known species with dissimilar genome sequences, estimate the set of target concentrations that best explains the observed probe intensities. This is simpler than the more general I/Q problem, because each probe is assumed to be capable of binding at most one target. Section 3.3 describes the quantitation algorithm, the design of an experiment to test it, and the results of the experiment.

Once these problems were solved, I was able to address the general identification/quantitation problem, using a latent variable mixture model together with an EM algorithm to maximize the complete data likelihood with respect to the target concentrations. In section 3.4 I discuss the latent variable model, derive the equations to be solved at each step of the EM algorithm, and describe its performance against a test dataset.

3.2 Estimation of scale factors using spike-ins and positive control probes

Consider a series of arrays to which we hybridize various concentrations c_a of DNA from the same target organism. We model the observed intensity y_{iaj}^{obs} of replicate feature j with probe sequence i on array a by combining the Langmuir equation 2.1 with an array-specific scale factor, and including both additive (background) and multiplicative noise components:

$$y_{iaj}^{obs} = b_{iaj} + \gamma_a \frac{c_a K_i}{1 + c_a K_i} e^{\epsilon_{iaj}} \tag{3.1}$$

Here K_i is the affinity of probe sequence i for the single known target, which we estimate either from direct measurements, or using the free energy ΔG_i predicted by our PDNN model and equation 2.2. The multiplicative noise factor is modeled as $e^{\epsilon_{iaj}}$, where ϵ_{iaj} is normally distributed with zero mean. In the remainder of this chapter, we'll assume that the intensities are background-corrected, defining $y_{iaj} = y_{iaj}^{obs} - b_{iaj}$. Our ultimate task is to infer the concentrations c_a, given the intensities y_{iaj}.

The first problem to solve is estimating the scale factor γ_a for the array, which in the absence of multiplicative noise would be the background corrected intensity for a chemically saturated feature. The scale factor varies between array experiments because of differences in scanner gain settings, efficiency of target labeling, degradation of fluorophores, and other uncontrolled factors. To estimate the scale factor, we need intensities for a set of reference probes for which the fraction of bound oligos is known or can be calculated.

3.2.1 Positive control probe design

To facilitate scale factor estimation, I designed a set of 572 positive control probes that were included in version 5 of the LLMDA. The probes are based on the genome sequence of the hyperthermophilic bacterium *Thermotoga maritima*. By spiking *Thermotoga* DNA into the hybridization mixture at a known concentration, we can use the intensities of the positive control probes, along with their known affinities, to estimate the scale factor γ_a.

Thermotoga DNA is commonly used as a reference control in nucleic acid assays because it naturally occurs only in deep sea hydrothermal vents, and is phylogenetically distant from most bacteria found in more temperate environments (such as the human body). Therefore, it is unlikely to appear in the types of samples we analyze with the LLMDA, and probes

designed against it are not expected to cross-hybridize with bacteria that do occur in these samples.

The positive control probes include both perfect match (PM) oligos and mismatch (MM) sequences derived by replacing one or more PM probe nucleotides with their complements. The mismatch probes are included to broaden the range of affinities spanned by the positive control probes. Mismatches are spaced at intervals of 5 base positions to minimize the unwinding of the probe-target DNA duplexes. Successive mismatches were added in 5' to 3' order, starting from the untethered end of the probe. Six replicates of each positive control sequence are included on the array, enabling estimation of within-array noise.

3.2.2 Affinity estimation for positive control probes: experiment design and simulations

In order to compute the probe affinities from the free energies ΔG_i estimated using the PDNN model, we must know the value K_0, such that $K_i = K_0 e^{-\Delta G_i/RT}$ is an affinity constant in inverse concentration units (e.g., inverse picomolar, pM^{-1}). According to our physical hybridization model, K_0 should be constant for all arrays with a given surface oligonucleotide density. Unfortunately, the estimate obtained from the tiling array experiments described in section 2.5.3, $4.1 \times 10^{-8} pM^{-1}$, differs from the K_0 values obtained from other data, such as the short hybridization time experiments described in section 2.4.1.

It turns out that, with the tiling array dataset, the parameters K_0 and γ_a for each array are not identifiable. Since these arrays were almost all run using the same target concentrations; multiplying K_0 by some factor and dividing all the γ_a by the same factor has little effect on the predicted intensities.

To resolve this issue, I designed an experiment to measure the affinities directly for the positive control probes on the LLMDA version 5 array. In this experiment, two replicate arrays were hybridized to *Thermotoga* DNA at each of four concentrations: 1, 4, 16 and 32 pM. The goal was to fit scale factors γ_a and affinity constants K_i to the model described by equation 3.1, given the measured intensities and the known DNA concentrations c_a on these arrays.

To make the model more tractable, I log-transformed both sides of equation 3.1 after taking background correction into account, yielding for the intensity y_{iaj} of replicate j of probe i on array a:

$$\log y_{iaj} = \log \gamma_a + \log c_a + \log K_i - \log(1 + c_a K_i) + \epsilon_{iaj} \tag{3.2}$$

where ϵ_{iaj} is a Gaussian noise term. Although this model is partially linear (because of the term $\log \gamma_a$), it is still challenging to fit, especially with noisy data. I eventually succeeded by using the weighted nonlinear least squares (NLS) algorithm implemented in the R `nls()` function to fit the parameters $\log K_i$ and $\log \gamma_a$ simultaneously. I weighted observations on probe i by a factor $1/s_i$, where s_i was the sample standard deviation of the intensities of the 6 replicates of probe i. I also excluded probes with more than 9 mismatches from the datasets used for fitting, since these probes had intensities in the same range as the background noise.

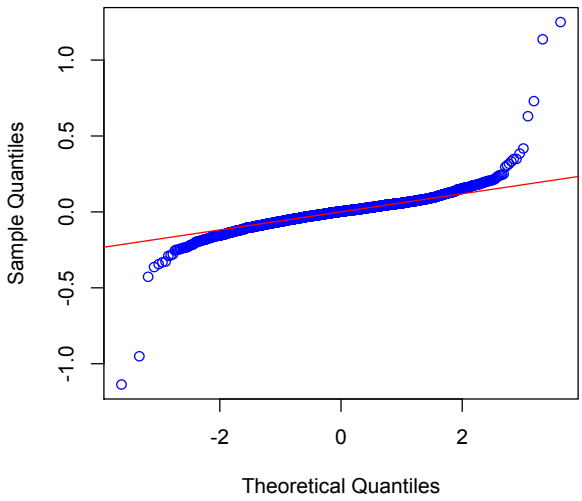

Figure 3.1: Normal quantile-quantile plot of replicate probe log intensity deviations on a typical LLMDA version 5 array, showing heavy tails of distribution

I tested the weighted NLS procedure with a simulated data set, to see how well it recovered the affinities and scale factors used to generate the data. I generated simulated intensities for the *Thermotoga* probes based on the model in equation 3.1, using the same DNA concentrations for each simulated array as in the real experiment, and affinities based on the free energies predicted by the PDNN model. Simulating the additive and multiplicative noise components presented some challenges, as sampling them from normal or log-normal distributions produced intensities lacking the heavy tail behavior seen in the real data (as shown in Figure 3.1). I obtained more "realistic" data by sampling background contributions b_{iaj} from the empirical distribution of negative control probe intensities on one of the real arrays. To simulate multiplicative noise, I computed deviations d_{ij} of the log intensities for the n_{reps} replicate features from the means for each positive control probe sequence, $d_{ij} = \log y_{ij} - \sum_k \log y_{ik}/n_{reps}$, sampled values ϵ_{iaj} from the empirical distribution of the deviations, and multiplied the simulated intensity by $e^{\epsilon_{iaj}}$.

Figure 3.2 shows the fitted affinities plotted against the input affinity values used to generate the data, on a log scale. Figure 3.3 plots the fitted scale factors γ_a against the simulation inputs. Except for the affinities at the upper extreme of the range, the weighted NLS procedure did an excellent job of recovering the input values.

To make sure that the good performance of weighted NLS was reproducible, I created 20 simulated datasets, generating probe affinities $K_i = K_0 e^{-\Delta G_i/RT}$ with the same probe free energies as for the earlier dataset, but varying inputs for the factor K_0. I also varied the input scale factors γ_a, and sampled the additive and multiplicative noise contributions independently for each dataset. I fitted affinities and scale factors to each dataset using weighted NLS as before. For most (17/20) of the datasets, weighted NLS did a good job of recovering the input values, yielding fitted affinities with median values above 80% of the

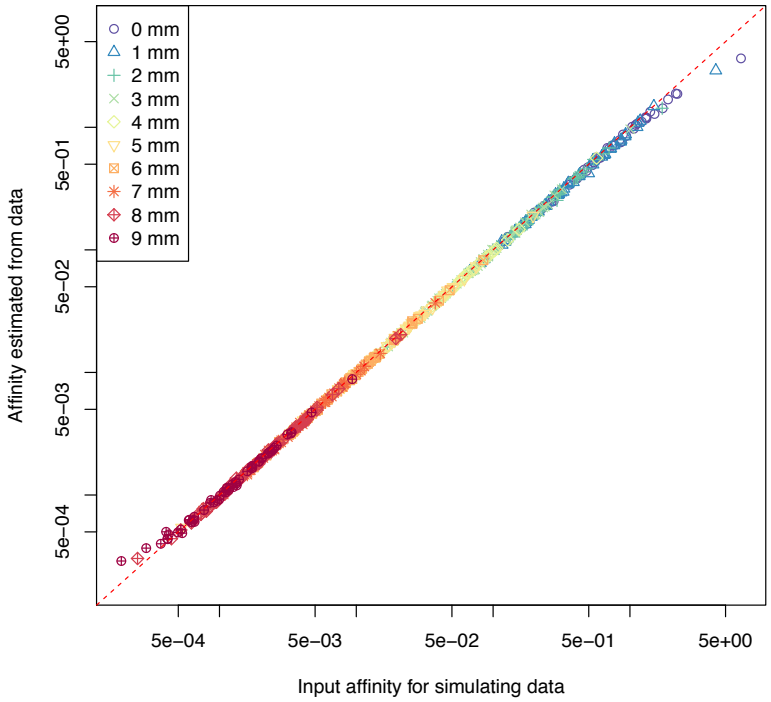

Figure 3.2: Affinities fitted from simulated data vs input affinities used to generate the data, plotted on log scale. Plot symbol shape and color indicates the number of mismatches between the probe and target sequences.

input values. The top panel of Figure 3.4 shows the ratios of the fitted values to the inputs for one such dataset. The fitting procedure performed substantially worse on the three other datasets, such as the one shown in the lower panel of Figure 3.4, although the median ratio of fitted to input affinities was still above 50%.

When I compared the different datasets, I saw that the ones for which weighted NLS didn't work well were those with the smallest values of the input factor K_0, and thus lower input affinity values. As a crude measure of the performance of the fitting procedure, I computed the median ratio of fitted to input affinities for each dataset and plotted it against the input K_0 value, as shown in Figure 3.5. The ratio falls off sharply for input $\log K_0$ values below -16, and is close to 1 for $\log K_0$ greater than -15.5. These low fitted affinity values are coupled with overly high estimates for the scale factors γ_a. This makes some intuitive sense; when the actual affinities are too low, the probe intensities do not approach the limit imposed by chemical saturation (i.e., γ_a), so the dataset provides little information about the actual saturation level.

The problem is illustrated by Figure 3.6, which compares fitted vs (simulated) observed intensities, and observed intensity vs fitted free energy, for the same two datasets used to generate Figure 3.4. The dataset shown in the upper panel has higher overall input affinities, so that the intensities level off at the lower free energy (higher affinity) values. With the

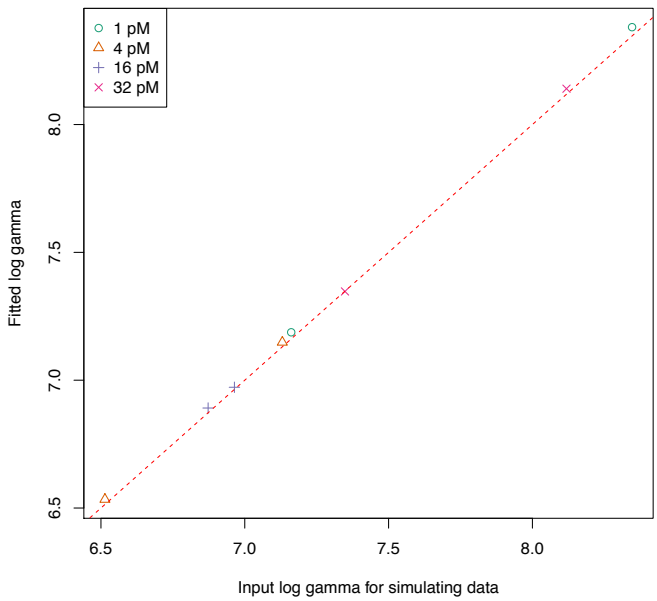

Figure 3.3: Scale factors fitted from simulated data vs input scale factors used to generate data

lower average affinities used for the lower panel, the curvature of the fitted intensity curve is much smaller, leading to much more variance in the γ_a estimate, and larger errors in the affinity estimates.

The results from simulated spike-in experiment data suggest that, when we estimate affinities using real data, the accuracy of our estimates will depend on the *true* value of K_0. We may get a sense of whether the true affinities are higher or lower by examining plots of observed intensities vs free energy, and noting whether the intensities level off as the free energy becomes more negative.

3.2.3 Affinity estimation for positive control probes: results

After using simulated data to explore the behavior of the affinity fitting algorithm, I was ready to look at the data from the real *Thermotoga* spike-in experiment. Each of the arrays from this experiment was scanned multiple times using a 2 μm resolution Roche/NimbleGen MS-200 scanner. On the MS-200 scanner, unlike the older Axon 4000B, the PMT gain setting is expressed as a percentage. One set of scans was run using the autogain setting, resulting in gain percentages between 300% and 500%; it produced images with many saturated features. A second set was run at the 100% gain setting. Although no features were fully saturated in these scans, subsequent analysis showed that the feature intensities for the higher concentrations were strongly affected by saturated pixels. I only discovered this two months after the arrays were originally hybridized and scanned; luckily, our technician had

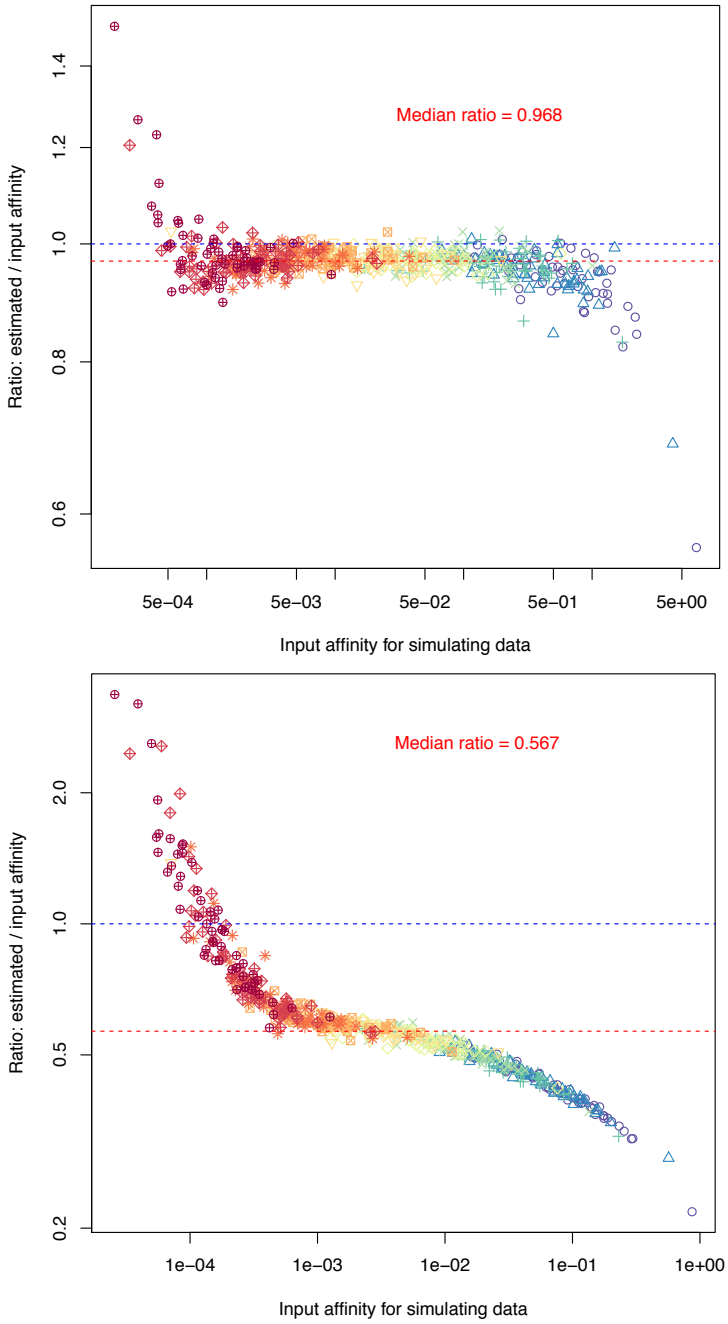

Figure 3.4: Ratios of affinities fitted from simulated data to input affinities, as a function of the input affinities, for two simulated datasets

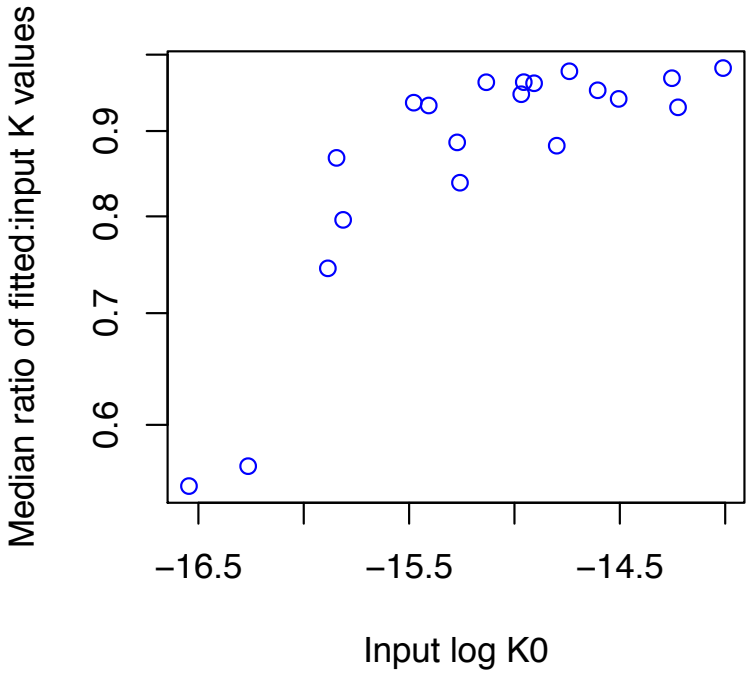

Figure 3.5: Median ratio of fitted to input affinities from simulated data, as a function of the input K_0 value, for 20 simulated datasets

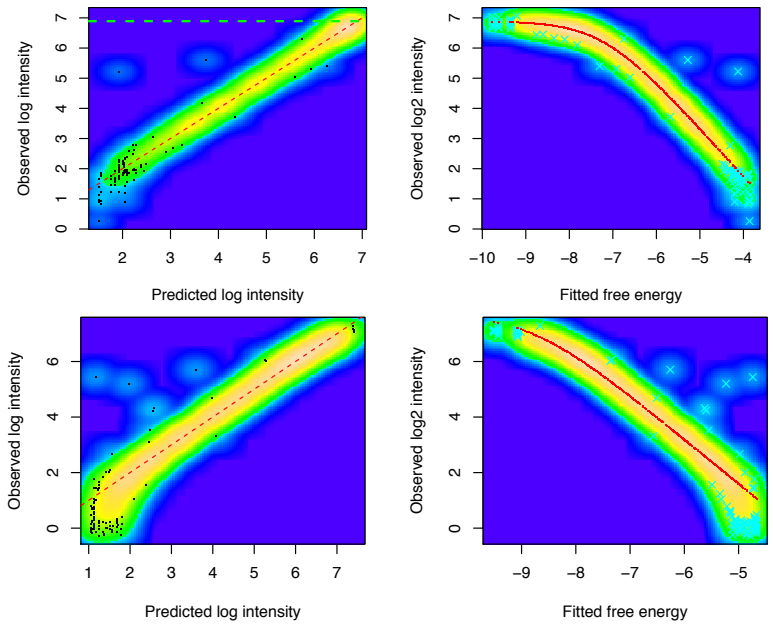

Figure 3.6: Fitted vs observed intensities at concentration 16 pM for two simulated datasets

preserved the arrays. Our technician then ran two additional scans of each array: one repeated at 100% gain, to compare with the original scan to test whether the arrays had degraded during storage; and another at 10% gain, to remove any chance of optical saturation effects. Overall, intensities for the second scan at 100% gain were only slightly lower than the corresponding scan done two months earlier, so I decided degradation was not a serious problem. As expected, there were very few saturated pixels in the scans done at 10% gain.

I used the weighted NLS procedure to fit affinities to the intensities from the arrays scanned at 10% gain. In order to get `nls()` to converge, I had to use the subset of the data containing only probes with 7 or fewer mismatches to the target sequence. The upper panel of Figure 3.7 shows the fitted $\log K_i$ values plotted against $\Delta G_i/RT$, where ΔG_i is the free energy *predicted* by the PDNN model. In the lower panel are plotted the differences $\log K_i - \Delta G_i/RT$ as a function of $\log K_i$. The horizontal dashed line drawn in the lower panel indicates the median difference, which I use as an estimate of $\log K_0$. The diagonal line in the upper panel has unit slope and intercept $\log K_0$. The median $\log K_i - \Delta G_i/RT$ value is -16.44, implying an estimate for K_0 of $e^{-16.44} = 7.25 \times 10^{-8}$, about twice the value estimated from the tiling array experiments. Unfortunately, this K_0 estimate does fall in the range in which simulations predict that the scale factors tend to be overestimated and the affinities underestimated, at least for lower values of the spike-in concentration. To obtain better estimates, we plan to perform additional experiments using higher spike-in concentrations that would result in stronger saturation effects at the higher affinities.

Examining Figure 3.7, we see there are also systematic errors in the ΔG_i estimates which depend on the number of mismatches. For probes with one to four mismatches, as the free energy increases, the measured affinities fall less rapidly than expected; for probes with more than five mismatches, the affinities fall more rapidly. The true dependence of the affinities on the mismatch count is shown for a representative sampling of probes in Figure 3.8. While the PDNN model predicts that the log affinity should decrease approximately linearly with the number of mismatches, the observed affinities drop off more steeply beginning with the fifth or sixth mismatch. Since successive mismatches are added at uniform five base intervals along the length of the probe, the observed nonlinearity may result when the probe-target duplex does not contain a sufficiently long stretch of perfect matches to stabilize it. These nonlocal structural effects are not accounted for in simple nearest-neighbor hybridization models.

It does appear, however, that with the fitted affinities and scale factor parameters, the Langmuir model fits the real data as well as it does simulated data. Figure 3.9 shows the observed intensities plotted against the predicted intensities and free energies, for one of the arrays at concentration 16 pM. At this concentration, the observed and fitted intensities both exhibit saturation effects for the probes with the most negative free energies, leveling off rather than continuing to increase linearly. The close fit of the observed intensities to the predicted intensity curve suggests that the fitted affinities for the positive control probes are sufficiently accurate to allow us to estimate the array-specific scale factors, provided *Thermotoga* DNA is spiked in at a known concentration.

The procedure to estimate the array-specific scale factors can be summarized as follows:

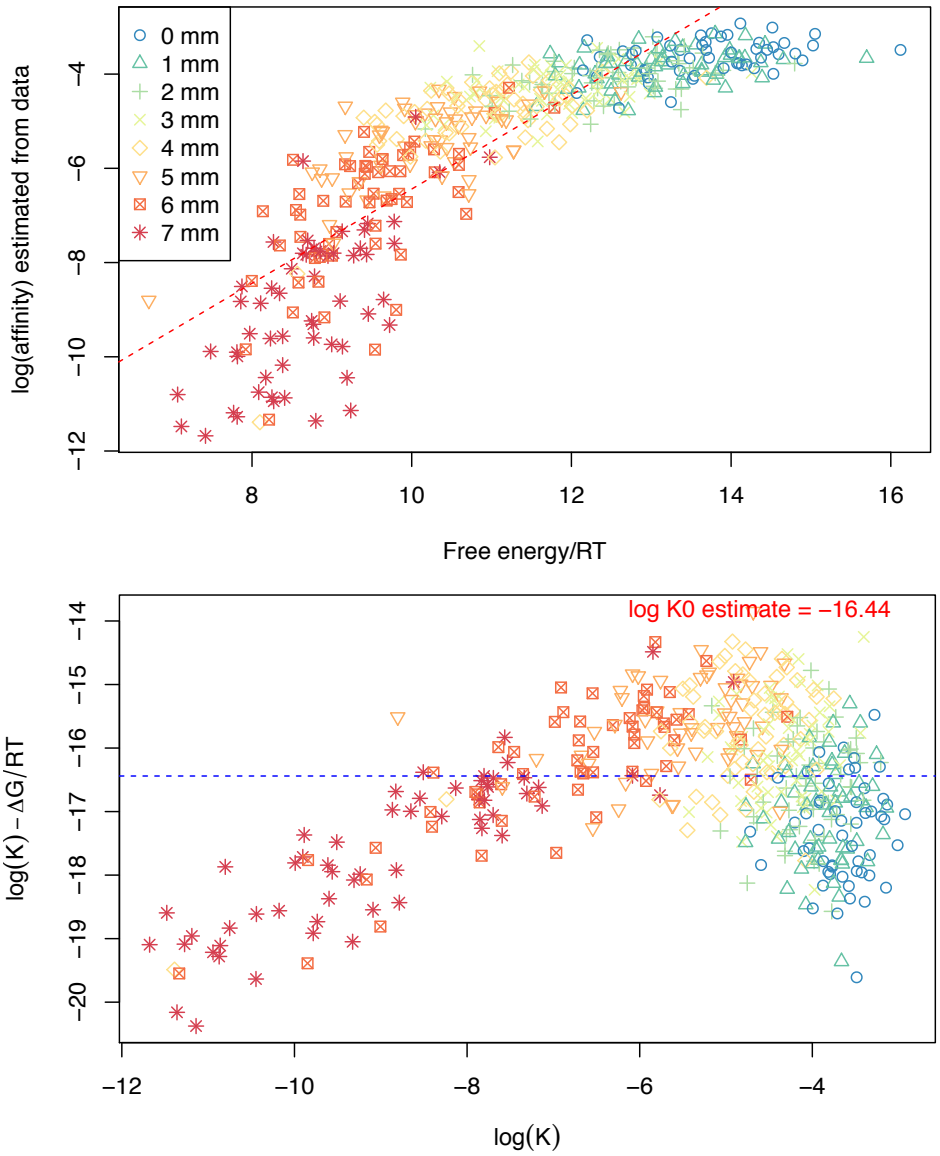

Figure 3.7: Top: Log affinities for *Thermotoga* probes fitted to real data, as a function of predicted free energy. Bottom: Differences between log affinities fitted to real data and predicted free energy / RT, as a function of fitted log affinity. Plot symbol and color indicates the number of mismatches between the probe and target sequences.

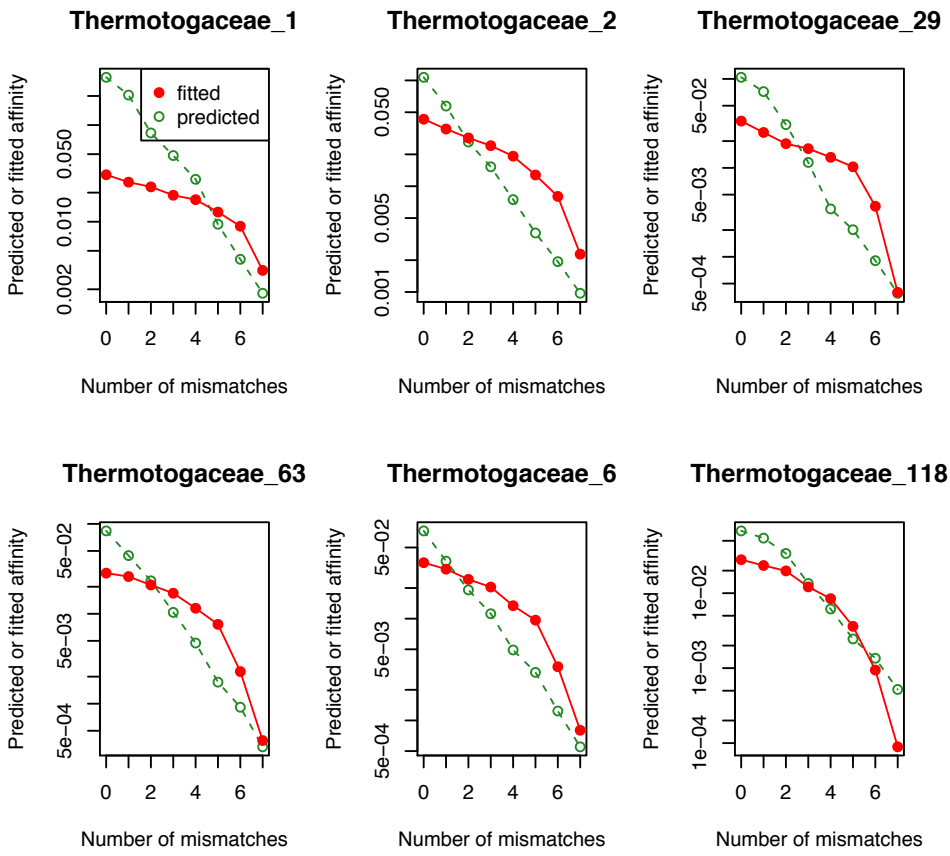

Figure 3.8: Fitted and predicted affinities for six perfect match *Thermotoga* probes and the first seven mismatch probes derived from them, as a function of the number of mismatches added.

- Background correct the observed probe intensities, producing corrected intensities y_{iak} for replicate k of probe i on array a.
- Given the corrected intensities for the spike-in probes, the known concentrations c_a of spiked-in *Thermotoga* DNA, and the measured affinities K_i of the *Thermotoga* probes, use weighted least squares to solve the log-transformed Langmuir model (Equation 3.2) for the array scale factor γ_a.

3.3 Quantitation of targets known to be present in a sample

My next step toward addressing the general identification/quantitation problem was to devise an algorithm for a much simpler quantitation task. To simplify the problem, we assume that

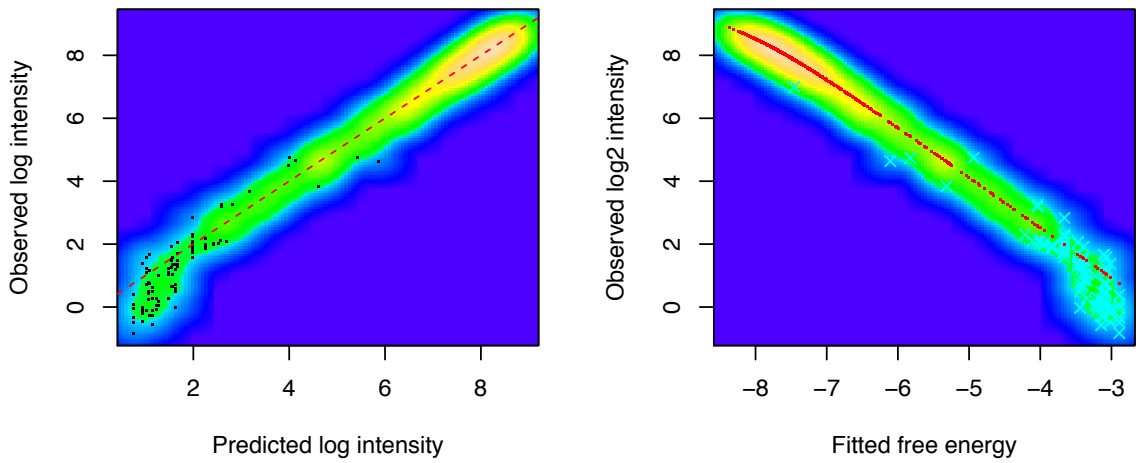

Figure 3.9: Observed log intensities for positive control probes vs predicted log intensities and free energies, for an example array with *Thermotoga* DNA spiked in at 16 pM concentration. Red points in the right hand panel represent the predicted log intensity values.

the targets present in the sample are known beforehand or have already been identified, and that the genomes of these targets are sufficiently dissimilar to one another that each probe on the array can only bind to one target. We are given an estimate of the scale factor γ_a for array a, calculated by spiking the sample with *Thermotoga* DNA at a known concentration and using the procedure outlined in the previous section. Given the above information, we are asked to estimate the set of DNA concentrations c_{aj} for each target j that best explains the observed probe intensities on the array.

To address this quantitation task, we can solve Equation 3.2 for the concentrations c_{aj}, this time treating the scale factor γ_a as a known quantity. Since we haven't performed an exhaustive set of dilution series experiments to estimate affinities for every probe/target pair of potential interest, as we did for the *Thermotoga* probes, we will have to use the affinities K_{ij} predicted by PDNN for probe i binding to the target j. We then use weighted nonlinear least squares to estimate the concentrations c_{aj}, using the R `nls()` function. `nls()` requires initial values for the parameters c_{aj}, which we calculate as:

$$c_{aj}^{(init)} = \text{median}\left(\frac{y_{iak}}{K_{ij}(\gamma_a - y_{iak})}\right) \tag{3.3}$$

where the median is computed over all i, k such that $y_{iak} < \gamma_a$.

One concern I had with this plan was the imprecision of the PDNN affinity estimates. While I have accurate estimates for the spike-in probe affinities, we saw in section 3.2.3 that they had both random and systematic deviations from the affinities predicted by the PDNN model. Therefore, I wanted to know how the quantitation algorithm performs in the presence of these errors. I addressed this question using simulated data, as described below.

3.3.1 Testing the quantitation algorithm: experiment design

To test the quantitation algorithm, I designed an experiment in which a series of 12 arrays were hybridized to mixtures of DNA from 5 different organisms at different concentrations, following a Latin square layout, along with *Thermotoga* DNA at a fixed concentration . The organisms are *Bacillus anthracis*, *Francisella tularensis*, *Burkholderia thailandensis*, human adenovirus B type 7, and vaccinia virus strain Lister, abbreviated henceforth as Ba, Ft, Bt, Av and Vv, respectively. I chose these species because the percent GC content of their genomes spans a wide range, from 31% (Ba) to 68% (Bt), and because we had stocks of DNA for them in our lab. The current version of the LLMDA has large numbers of probes for each of these species, ranging from 75 (Ft) to 251 (Ba).

The experiment layout is shown in Table 3.1. Two replicate arrays were hybridized for each sample mixture. Each set of replicates included a sample containing spiked-in *Thermotoga* DNA only.

I ran simulations to determine the optimal concentration of *Thermotoga* DNA to spike into each sample. The concentration affects the accuracy of the scale factor estimates; one wants to choose a concentration large enough that the highest affinity probe intensities are affected by chemical saturation, but not so high that all probes are saturated. The scale factor estimate, in turn, affects the estimates of the target DNA concentrations. I ran simulations for several combinations of input values of the log scale factor ($\log \gamma_a$) and spike-in concentration. Spike-in concentrations ranged by powers of 2 from 4 to 512 pM. To avoid overplotting of the results, I added zero mean Gaussian noise to the input scale factor and target concentration values before generating simulated data. Each simulated data set contained 225 arrays in all.

Sample	Concentration (pM)				
number	Ba	Bt	Ft	Av	Vv
1	0	0	0	0	0
2	0	2	8	32	128
3	2	8	32	128	0
4	8	32	128	0	2
5	32	128	0	2	8
6	128	0	2	8	32

Table 3.1: Layout for quantitation test experiment, showing concentrations of five known targets in each sample. Each sample was run on two replicate arrays.

Figure 3.10 shows the scale factors γ_a fitted to the simulated data plotted against the input γ_a used to generate the data, using different plot colors and symbols to indicate the spike-in DNA concentration. We see that, for concentrations below 8 pM, there were larger deviations from the input values. Figure 3.11 shows the fitted concentrations plotted against the input values, for 4 of the 5 targets. Again, the deviations were greatest below 8 pM.

Table 3.2 shows the RMS deviations of the fitted log concentration values from the input values, as a function of the spike-in concentration. The smallest average deviation was obtained with a simulated spike-in concentration of 8 pM.

Spike-in concentration	RMS deviation of fitted concentrations from inputs	
	With input affinities	With noisy affinities
2	2.59	3.91
4	1.47	3.15
8	0.95	2.94
16	1.00	2.89
32	1.17	3.10
64	1.21	3.23
128	1.41	3.29
256	1.47	3.45
512	1.46	3.60

Table 3.2: Root mean square deviation of concentrations fitted to 25 simulated array data sets generated using 9 different concentrations of spiked in *Thermotoga* DNA. Concentrations were fitted using either the input affinities or with noise added to the log affinities, to simulate the effect of errors in the affinity estimates.

To test the effect of errors in the predicted affinities, I generated additional data sets in which I added zero mean Gaussian noise with SD 1.0 to the logs of the affinities used to generate the intensities, while continuing to use the unmodified PDNN affinities to fit the concentrations. The resulting fits are shown for 4 of the targets in Figure 3.12, and the corresponding RMS deviations are shown in the third column of Table 3.2. The fits were surprisingly robust to errors in the affinities. The spike-in concentration yielding the most accurate target concentration estimates, in the presence of these errors, was 16 pM; therefore I chose this concentration for the spike-ins in the actual experiment.

3.3.2 Testing the quantitation algorithm: results

The six samples for the Latin square test experiment were prepared and hybridized to duplicate subarrays of a single 12-plex LLMDA version 5 array. DNA from each target was labeled, quantitated using a Qubit fluorometer, and then diluted to the required concentration for each sample. After hybridizing for 40 hours, the array was scanned on our Roche MS-200 scanner at three gain settings: autogain, 100%, and 50%. I examined the intensity distributions for the probes matching the six targets used in the samples, and determined that the highest intensities were affected by optical saturation, even at 50% gain. After the array was rescanned at 20% gain, I found that saturated pixels were largely absent.

I applied the quantitation algorithm to the intensity data from the scans at 20% gain to obtain a concentration for each of the five targets. The results for all 12 arrays are

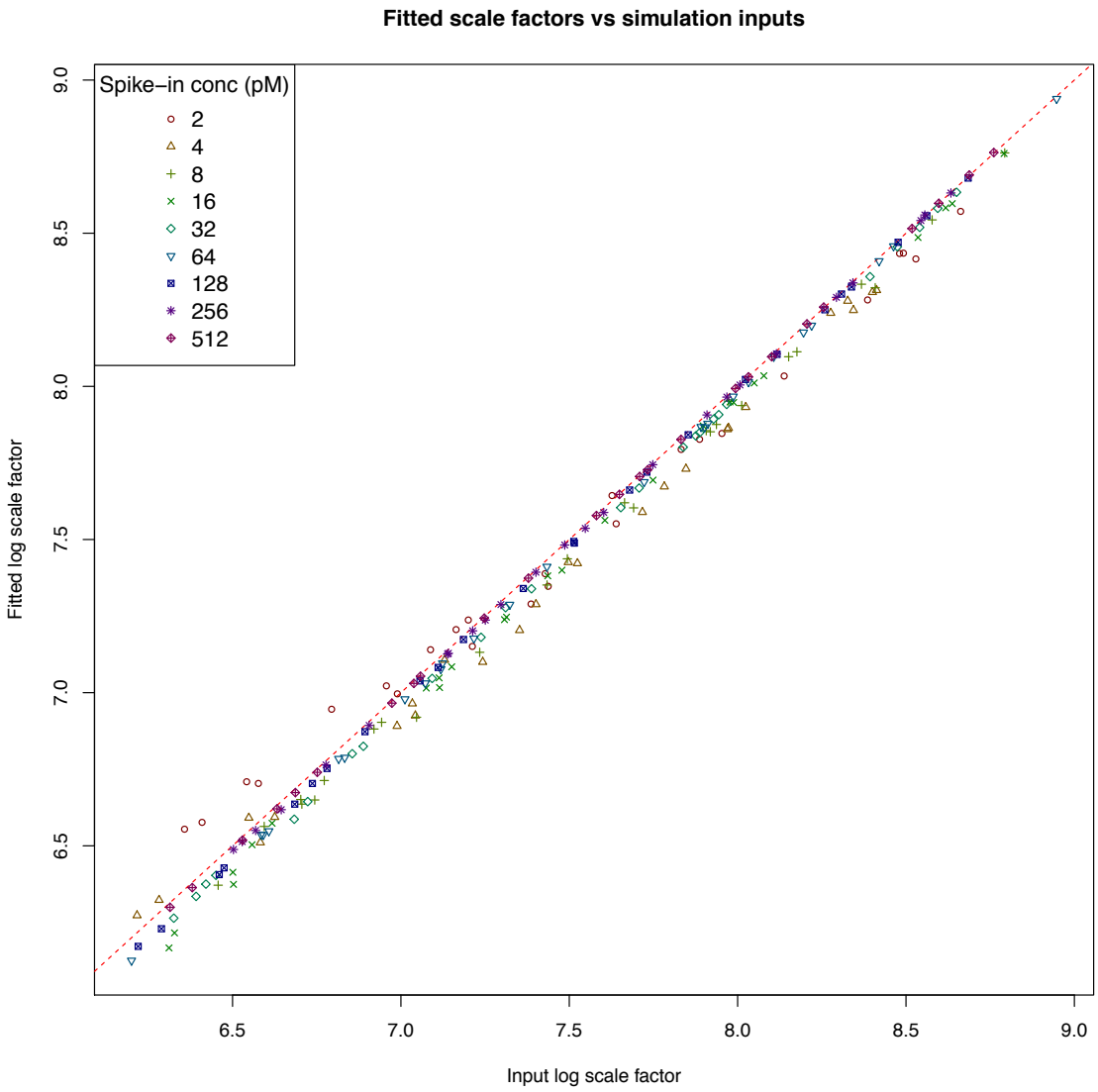

Figure 3.10: Scale factors fitted from simulated data vs input scale factors used to generate data. Plot symbols indicate the concentration of *Thermotoga* DNA spiked into the sample to use as a calibration reference.

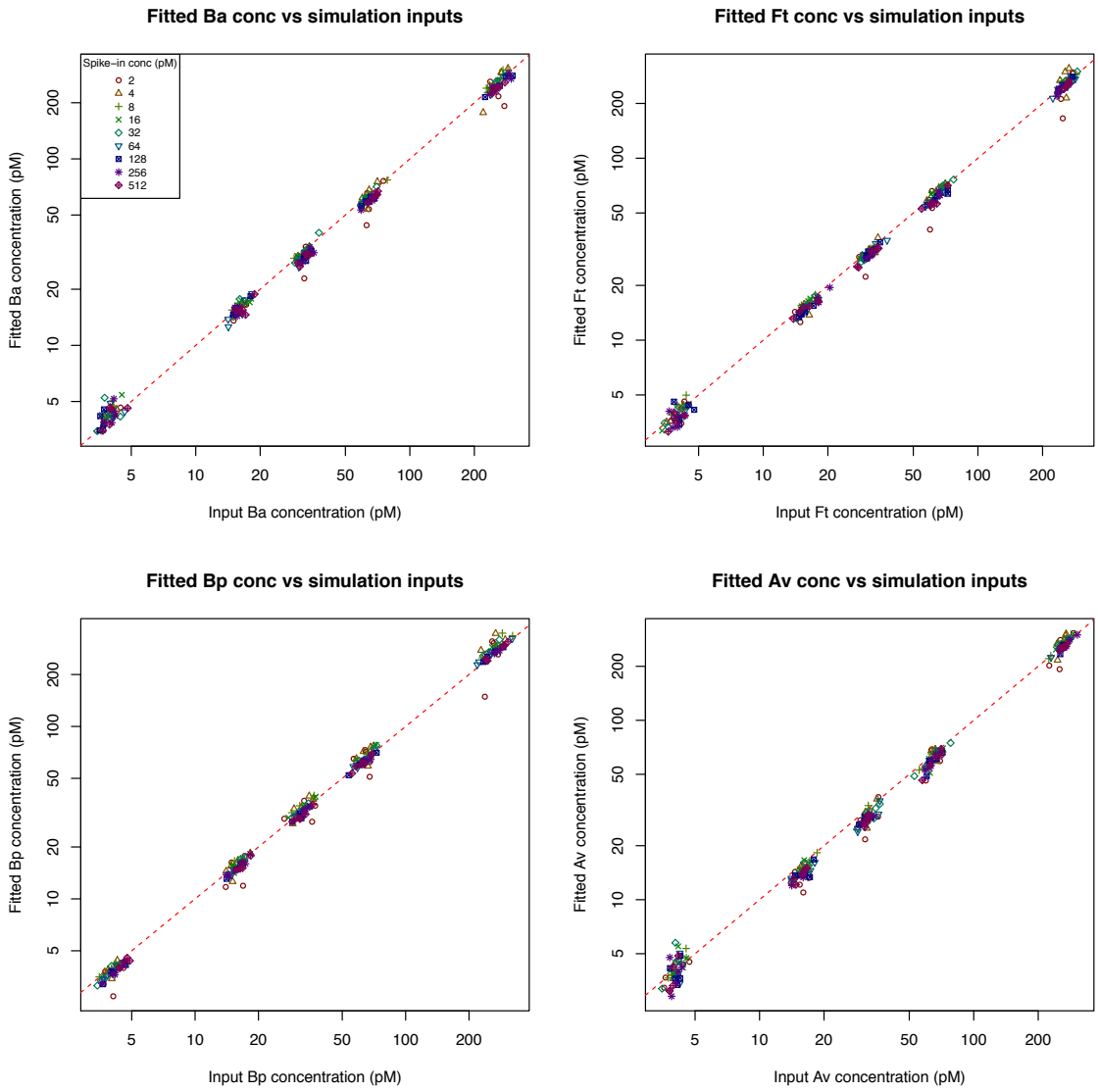

Figure 3.11: Target concentrations fitted from simulated data vs input concentrations used to generate data

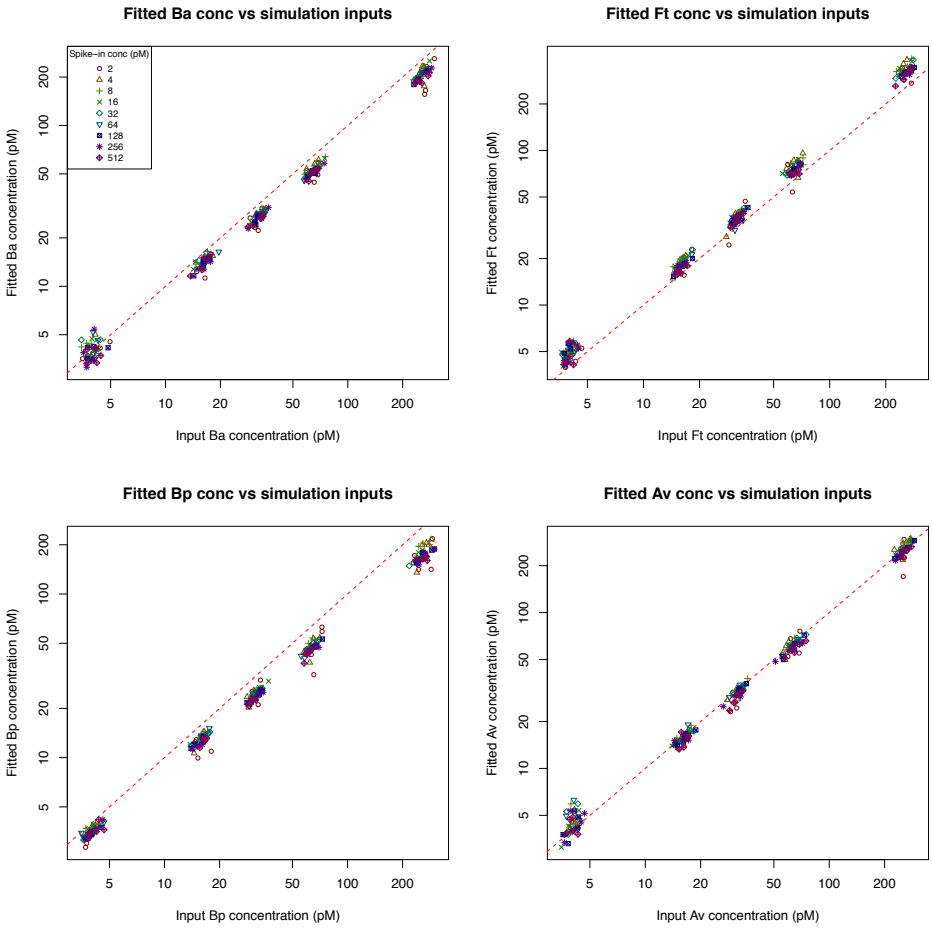

Figure 3.12: Target concentrations fitted from simulated data vs input concentrations used to generate data, when Gaussian noise (with SD 1.0) was added to the log affinities used for the simulation

summarized in Figure 3.13, which shows log-log plots of the fitted vs actual concentrations. For targets Ba, Ft, Bt and Av, there is reasonably good correspondence between the fitted and actual concentrations, although the fitted values increase more slowly that the actual ones; thus smaller concentrations are overestimated and larger ones are underestimated. The results from replicate arrays are remarkably close, suggesting that the discrepancies originate from systematic errors rather than noise in the array hybridization and scanning processes. For vaccinia virus, the concentrations are underestimated except on the arrays where the true concentration was 2 pM. Since for all other targets the fitted concentration increases monotonically with the true concentration, the anomaly at 2 pM is better explained by an error in the dilution procedure than by issues with the fitting algorithm. The low estimates for the higher Vv concentrations could have resulted either from overestimating

the affinities for the Vv probes, or from degradation of our Vv DNA stock by repeated thawing. Degradation of the DNA could affect its binding to Vv specific probes, without affecting the fluorometric concentration measurements, since the latter are not sequence specific.

3.4 Solving the general identification/quantitation problem

As we saw in the preceding section, our quantitation procedure produces fairly accurate target concentration estimates when we know that the sample is a mixture of targets belonging to a small set, matched by nonoverlapping sets of probes. However, in most real-life applications of the LLMDA, samples must be tested against a much wider range of targets, since the targets actually present are not known beforehand. The most recent version of the LLMDA (v5) contains probes matching over 67,000 target genome segments from multiple strains of nearly 6,000 microbial species. Many probes are expected to bind to DNA from different strains of the same species, and even different species within the same family. If we were to apply the simple quantitation algorithm to array data from an unknown sample, fitting concentrations for the full set of potential targets, it would erroneously assign nonzero concentrations to a wide range of closely related targets, only one of which would likely be present.

Therefore, solving the general identification/quantitation problem requires an algorithm that can resolve the ambiguity resulting from probes with multiple possible targets. To address this ambiguity I developed a latent variable model for the general I/Q problem and an expectation-maximization algorithm for solving it, termed MIQ (for microbial identification and quantitation, and pronounced "mike"). I tested the MIQ algorithm using the same Latin square array data set described earlier.

3.4.1 Latent variable model for unknown sample data

When a probe i may potentially bind to one of m different targets, the total fraction of bound oligos in the associated feature is described by an extended version of the Langmuir equation, which is duplicated here as equation 3.4:

$$\theta_i = \frac{\sum_{j=1}^{m} c_j K_{ij}}{1 + \sum_{j=1}^{m} c_j K_{ij}} \tag{3.4}$$

As usual, the intensity y_{ik} observed for replicate feature k of probe i is modeled as a product of the bound fraction, an array-specific scale factor γ and by a multiplicative noise factor $e^{\epsilon_{ik}}$, with ϵ_{ik} following a $N(0, \sigma^2)$ distribution. (Since we are only analyzing data from one array at a time, I've dropped the array index a from the variables in this discussion.) The

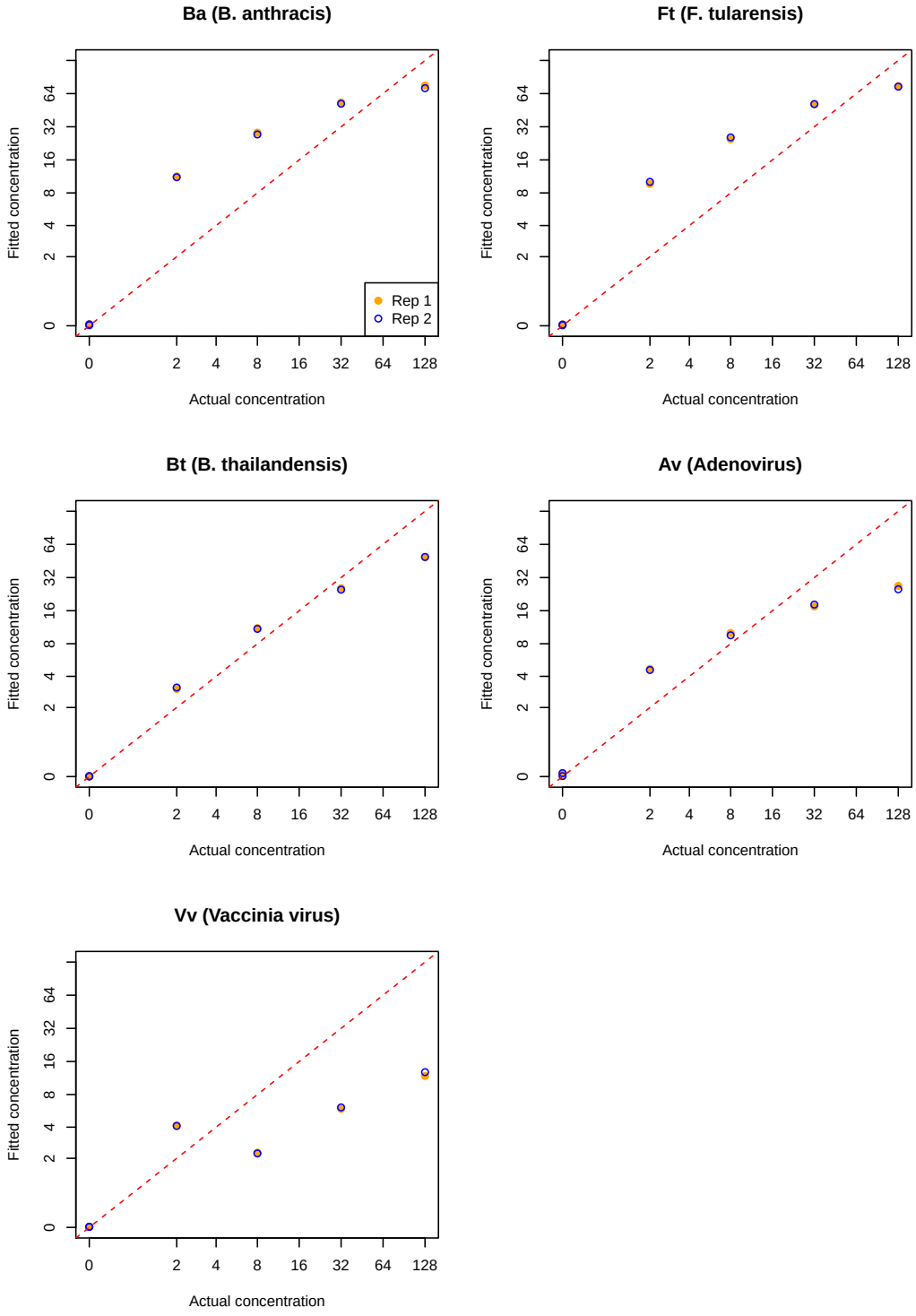

Figure 3.13: Concentrations fitted to intensities from 12 arrays from Latin square experiment vs actual concentrations of each target on the respective arrays. Values from replicate arrays are plotted with different colors and symbols.

resulting log likelihood takes the form 3.5:

$$\log L(c;y) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{k=1}^{r}\left[\log y_{ik} - \log\gamma - \log\frac{\sum_{j=1}^{m} c_j K_{ij}}{1+\sum_{j=1}^{m} c_j K_{ij}}\right] \tag{3.5}$$

with n being the number of probe sequences and r the number of replicate features per probe. Ideally, we would like to maximize the likelihood with respect to the target concentrations $c = (c_1, \ldots, c_m)$. Although the actual sums for probe i can be restricted to the few hundred targets j for which $K_{ij} \neq 0$, rather than the full set of 67,000 targets in our database, maximizing the likelihood directly still presents computational challenges, because of the sum over targets inside the logarithm.

Latent variable models have been applied to similar problems in the analysis of transcript quantitation using high-throughput cDNA sequence ("RNA-Seq") data [Taub 09], [Roberts 12]. Using these as inspiration, I decided to develop a similar approach for target identification and quantitation using detection array data. The simplifying assumption in my approach is that each probe only binds to (at most) one target. This assumption can be justified by observing that samples rarely contain multiple closely related strains of the same species, and that probes usually have much greater affinity for one species than another; thus, in a sample containing a mixture of species, one strain of one species will dominate the binding to a particular probe.

Under the latent data model, the complete data consist of the observed intensities y_{ik} and a set of unobserved index variables $t_i \in \{1, 2, \ldots, m, m+1\}$ indicating which target probe i is bound to. The special value $t_i = m+1$ indicates that the probe is not bound to any target. We assume that the t_i are multinomially distributed with probabilities $\phi_{ij} = \mathbb{P}[t_i = j]$ satisfying $\sum_{j=1}^{m+1}\phi_{ij} = 1$, and that the y_{ik} and t_i are independent. The parameters ϕ_{ij} are analogous to the mixing proportions in a mixture model.

The complete data likelihood under this model can be written as a function of the parameters c and ϕ:

$$L(c,\phi;y,t) = \prod_{i,k}\phi_{i,t_i}\exp\left[-\frac{1}{2\sigma^2}\left(\log y_{ik} - \log\gamma - \log\frac{K_{i,t_i}c_{t_i}}{1+K_{i,t_i}c_{t_i}}\right)^2\right] \tag{3.6}$$

If we define $u_{ik} = \log y_{ik} - \log\gamma$ and $\mu_{ij} = \log\frac{K_{ij}c_j}{1+K_{ij}c_j}$, we can express the complete data log likelihood as:

$$\begin{aligned} l(c,\phi;u,t) &= \sum_{i,k}\left[\log\phi_{i,t_i} - \frac{1}{2\sigma^2}(u_{ik}-\mu_{i,t_i})^2\right] \\ &= \sum_{i,k}\left[\log\phi_{i,t_i} - \frac{\mu_{i,t_i}^2}{2\sigma^2} + \frac{u_{ik}\mu_{i,t_i}}{\sigma^2} - \frac{u_{ik}^2}{\sigma^2}\right] \end{aligned}$$

Let $\delta(i,j)$ be the function equal to one when $i = j$ and zero otherwise, and rewrite the complete data log likelihood to make its dependence on the t_i more explicit:

$$l(c,\phi;u,t) = \sum_{i,k}\left\{-\frac{u_{ik}^2}{\sigma^2} + \sum_{j=1}^{m}\delta(j,t_i)\left[\log\phi_{ij} - \frac{\mu_{ij}^2}{2\sigma^2} + \frac{u_{ik}\mu_{ij}}{\sigma^2}\right]\right\} \tag{3.7}$$

We can now apply the EM algorithm to maximize the complete data log likelihood with respect to the parameters c and ϕ. We first choose initial values for the parameters. To initialize the concentrations c_j, the procedure is similar to the one used in the quantitation algorithm for known targets (equation 3.3), except that the median of the ratios $\frac{y_{ik}}{K_{ij}(\gamma - y_{ik})}$ is only computed over probes i with nonzero affinity K_{ij} for target j. The parameters ϕ_{ij} are set initially to $1/(1+|H_i|)$ for targets with nonzero K_{ij} and the special "unbound" target, where H_i is the set of targets with $K_{ij} \neq 0$ for probe i.

In the E step, we need to define the conditional expectation of the complete data log likelihood under the current parameter values ϕ and c, given the observed data u. At iteration q, this is:

$$\begin{aligned}
Q(c,\phi;c^{(q)},\phi^{(q)}) &= \mathbb{E}_{\phi^{(q)},c^{(q)}}\left[l(c,\phi;u,t)\right] \\
&= \sum_{i,k}\left\{-\frac{u_{ik}^2}{\sigma^2} + \sum_{j\in H_i}\mathbb{E}_{\phi^{(q)},c^{(q)}}\left[\delta(j,t_i)|u\right]\left[\log\phi_{ij} - \frac{\mu_{ij}^2}{2\sigma^2} + \frac{u_{ik}\mu_{ij}}{\sigma^2}\right]\right\} \\
&= \sum_{i,k}\left\{-\frac{u_{ik}^2}{\sigma^2} + \sum_{j\in H_i}\rho_{ij}\left[\log\phi_{ij} - \frac{\mu_{ij}^2}{2\sigma^2} + \frac{u_{ik}\mu_{ij}}{\sigma^2}\right]\right\}
\end{aligned} \tag{3.8}$$

where ρ_{ij} is the "responsibility" of target j for the intensity of probe i:

$$\begin{aligned}
\rho_{ij} &= \mathbb{E}_{\phi^{(q)},c^{(q)}}\left[\delta(j,t_i)|u\right] \\
&= \mathbb{P}_{\phi^{(q)},c^{(q)}}\left[t_i = j|U_i = u_i\right] \\
&= \frac{\mathbb{P}_{\phi^{(q)},c^{(q)}}\left[t_i = j, U_{i1} = u_{i1},\ldots,U_{ir} = u_{ir}\right]}{\mathbb{P}_{\phi^{(q)},c^{(q)}}\left[U_{i1} = u_{i1},\ldots,U_{ir} = u_{ir}\right]} \\
&= \frac{\phi_{ij}^{(q)}\exp\left[-\frac{1}{2\sigma^2}\sum_{k=1}^{r}(u_{ik} - \mu_{ij}^{(q)})^2\right]}{\sum_{l\in H_i}\phi_{il}^{(q)}\exp\left[-\frac{1}{2\sigma^2}\sum_{k=1}^{r}(u_{ik} - \mu_{il}^{(q)})^2\right]}
\end{aligned} \tag{3.9}$$

In the M step, we maximize $Q(c,\phi;c^{(q)},\phi^{(q)})$ with respect to c and ϕ, subject to the constraint $\sum_j \phi_{ij} = 1$. Using Lagrange multipliers and setting derivatives equal to zero, we get:

$$\begin{aligned}
0 &= \frac{\partial}{\partial\phi_{ij}}\left[Q(c,\phi;c^{(q)},\phi^{(q)}) - \lambda_i\phi_{ij}\right] \\
&= \frac{\rho_{ij}}{\phi_{ij}} - \lambda_i \\
\Rightarrow \phi_{ij} &= \rho_{ij}/\lambda_i
\end{aligned}$$

The constraint $\sum_j \phi_{ij} = 1$ then implies $\lambda_i = 1$, so that $\hat{\phi_{ij}} = \rho_{ij}$ is the value that maximizes the Q function. As for the concentrations, we have

$$\begin{aligned}
0 &= \frac{\partial}{\partial c_j} Q(c, \phi; c^{(q)}, \phi^{(q)}) \\
&= \sum_i \frac{\partial Q(c, \phi; c^{(q)}, \phi^{(q)})}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial c_j} \\
&= \sum_{i,k} \rho_{ij} \frac{u_{ik} - \mu_{ij}}{\sigma^2} \frac{\partial}{\partial c_j} \left[\log K_{ij} c_j - \log(1 + K_{ij} c_j)\right] \\
&= \sum_{i,k} \rho_{ij} \frac{u_{ik} - \mu_{ij}}{\sigma^2} K_{ij} \left[\frac{1}{K_{ij} c_j} - \frac{1}{1 + K_{ij} c_j}\right] \\
&= \frac{1}{\sigma^2 c_j} \sum_{i,k} \frac{\rho_{ij}}{1 + K_{ij} c_j} \left[u_{ik} - \log \frac{K_{ij} c_j}{1 + K_{ij} c_j}\right]
\end{aligned}$$

Thus, the Q function is maximized by $\hat{c_j}$ that satisfies Equation 3.10:

$$\sum_{i,k} \frac{\rho_{ij}}{1 + K_{ij} \hat{c_j}} \left[u_{ik} - \log \frac{K_{ij} \hat{c_j}}{1 + K_{ij} \hat{c_j}}\right] = 0 \tag{3.10}$$

This equation can be solved numerically using a variety of techniques, such as the safeguarded polynomial interpolation algorithm implemented in the R `uniroot()` function.

I implemented the EM algorithm as an R function. The code alternates the E- and M-steps described above and evaluates the Q function after each iteration. It terminates when the relative increase in the Q function value falls below a specified threshold, or after a prespecified number of iterations.

3.4.2 Testing the identification/quantitation algorithm

I tested the MIQ algorithm against the same array dataset used earlier to test the quantitation procedure. I analyzed each sample under the assumption that its constituent targets were a subset of a particular set of candidates. Because testing against our full candidate set of 67,000 targets would have been computationally infeasible for my R function implementation, I used a restricted set of 364 candidate targets that included the eight target sequences actually present in the array samples, plus up to 20 "decoy" targets randomly selected from each of the five taxonomic families represented among the sample targets ("present families") and 12 additional viral and bacterial families not represented in the samples ("unrepresented families"). I fit concentrations c_j and mixing proportions ϕ_{ij} to the intensities from each array, terminating the EM loop when the increase in the Q function value between two iterations fell below 0.1% of the function value. For most arrays, this convergence criterion was reached after about 40 EM iterations.

Figure 3.14 shows the fitted concentrations for the true targets and the decoy targets in the same families as the true targets, plotted against the true target concentrations. Fitted values are plotted with different symbols to distinguish the true targets from the decoys, and between the true target fits for the two replicate arrays at each set of concentrations. Separate concentrations were estimated for the two chromosomes of Bt and for the chromosome and two plasmids of Ba, which are represented by distinct target sequences. Since the average copy numbers can vary between the different plasmids and chromosomes in a bacterial genome, and plasmids in particular can be present in multiple copies, variation among the estimated concentrations for these genome elements is expected.

The targets with the highest fitted concentrations in each of the five present families are listed in Table 3.3, together with the actual target concentrations, for one of the two arrays run against each sample. If we interpret these as predictions of the targets most likely to be present, then all of the actual targets are correctly predicted to at least the species level, and in some cases, to the subspecies or strain level. For the Burkholderiaceae, the highest concentrations were consistently assigned to the two chromosomes of the correct strain *B. thailandensis* E264. Among the targets in family Adenoviridae, the correct target had the largest fitted concentration for four of eight arrays on which it was present; another strain of the same species (human adenovirus B) had the largest concentration on the other four. Among the Francisellaceae, several decoys had fitted concentrations larger than that of the true target. All of these were strains of the same subspecies as that of the true target, *F. tularensis* subspecies holarctica. In general, the MIQ algorithm fitted concentrations for the true targets that were close to those obtained by the quantitation procedure.

As currently implemented, the MIQ algorithm fits nonzero but (usually) small concentrations for targets that are absent from the sample, even when they share little sequence similarity with the targets that are present - as is generally true when the targets belong to different families. To get a sense of the probability that a target is actually present given its estimated concentration value, I plotted the distributions of the fitted $\log_2$ concentrations for targets in the 13 unrepresented families. The distributions over all arrays for five example families are shown in Figure 3.15, along with the combined distribution for all unrepresented families.

The range of fitted concentrations varies widely between families; although the vast majority of values fall well below 1 pM, there are certain families in which a handful of targets are assigned values greater than 1 pM. The overall distribution is somewhat long-tailed; although the 95^{th} percentile of all fitted values for unrepresented families is 0.9 pM, there is one target (hepatitis delta virus strain dTk38) with a fitted concentration on one array over 3000 pM. Most of these high fitted values are probably due to cross-hybridization between the probes for the target and some component of the sample.

Hepatitis delta virus (HDV) is an unusual case; its genome contains many stretches of long G homopolymers, which are difficult to avoid in the design of probes against this virus. Probes with G homopolymers of length 5 or greater often bind nonspecifically to many targets, a behavior observed by other researchers, but unfortunately not well understood [Upton 08, Langdon 09]. If we define a "binding rate" for a probe as the fraction of all

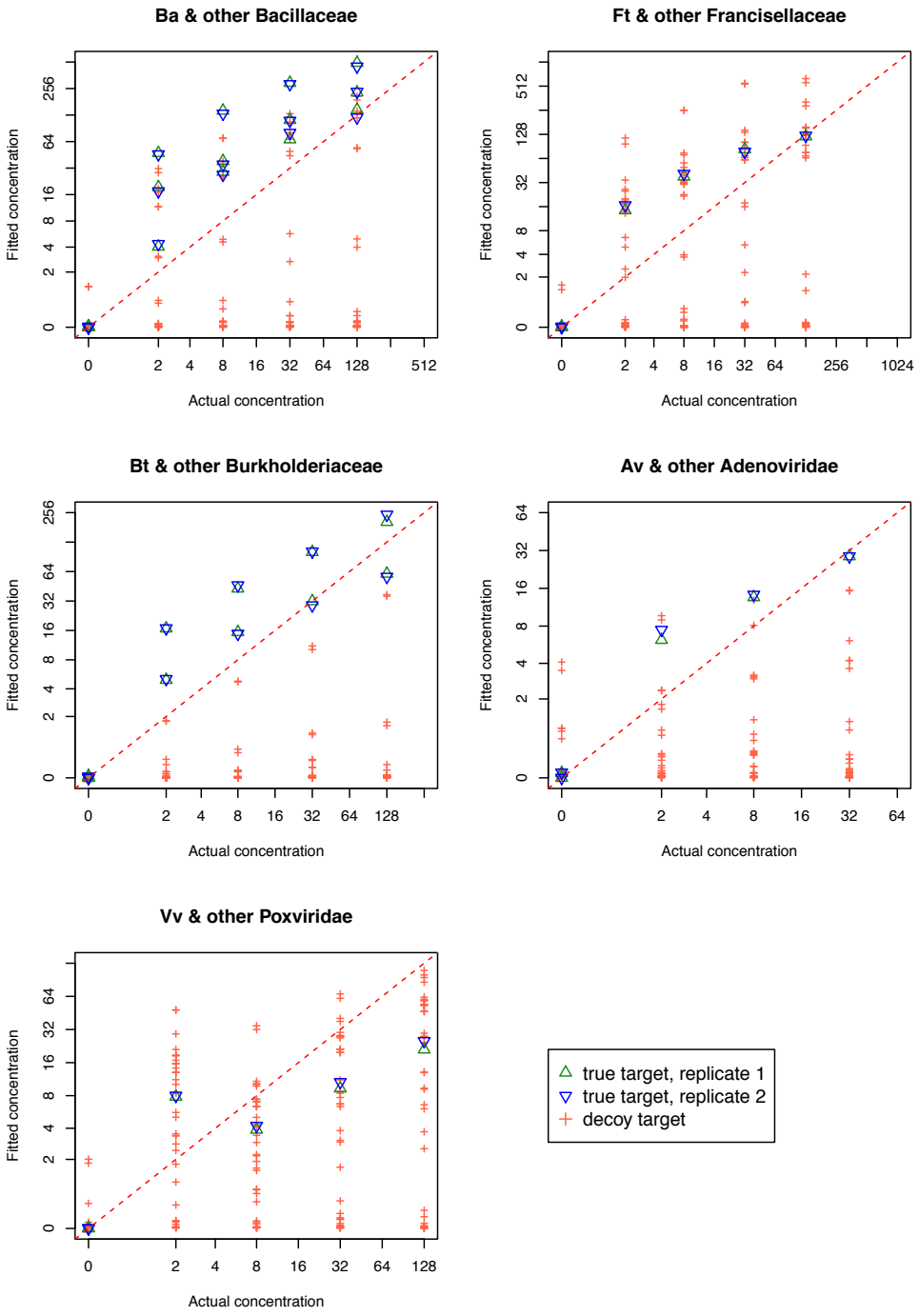

Figure 3.14: Fitted concentrations for candidate targets from Latin square experiment data, plotted against actual concentrations for the true targets that were present from each family. Separate concentrations were fitted for the genome elements of Ba (chromosome and 2 plasmids) and Bt (two chromosomes).

Family/Target	**Actual** or fitted concentration (pM)				
	2	3	4	5	6
Adenoviridae					
Human adenovirus B type 7	**32.00**	**128.00**	**0.00**	**2.00**	**8.00**
Human adenovirus B type 7	28.78	57.02	0.09	6.19	13.60
Human adenovirus B type 11a	15.28	16.63	0.11	8.96	12.88
Simian adenovirus 46	4.22	8.03	0.03	2.38	2.98
Bacillaceae					
B. anthracis str. Ames	**0.00**	**2.00**	**8.00**	**32.00**	**128.00**
B. anthracis plasmid pXO1	0.01	47.40	143.42	296.55	500.60
B. anthracis plasmid pXO2	0.03	4.04	29.00	112.69	230.51
B. anthracis str. A0174	0.04	28.42	69.63	132.85	209.12
B. anthracis str. Ames chr	0.01	19.44	38.41	67.76	146.79
B. anthracis str. Sterne	0.01	18.61	38.02	68.10	141.28
B. anthracis str. A0442	0.01	11.65	25.95	44.25	54.39
Burkholderiaceae					
B. thailandensis E264	**2.00**	**8.00**	**32.00**	**128.00**	**0.00**
B. thailandensis E264 chr 1	16.77	42.89	101.32	205.45	0.04
B. thailandensis E264 chr 2	4.95	15.39	31.80	60.79	0.01
B. mallei NCTC 10247 chr 1	1.79	4.81	11.10	35.81	0.03
Francisellaceae					
F. tularensis holarctica str. LVS	**8.00**	**32.00**	**128.00**	**0.00**	**2.00**
F. tularensis holarctica FSC200	255.37	545.23	635.49	1.30	115.10
F. tularensis holarctica KO971026	70.97	136.25	289.72	0.05	26.37
F. tularensis holarctica OSU18	49.44	101.65	157.87	0.04	19.43
F. tularensis holarctica LVS	38.05	83.24	120.76	0.02	14.43
F. tularensis novicida HHS	21.68	17.84	126.12	0.07	4.95
Poxviridae					
Vaccinia virus str. Lister	**128.00**	**0.00**	**2.00**	**8.00**	**32.00**
Vaccinia virus str. Acambis	100.23	0.04	48.26	34.60	61.45
Variola virus str. UK 1947	86.09	0.01	29.21	10.81	40.42
Variola virus str. Congo 9	60.17	0.02	16.81	7.39	30.13
Vaccinia virus str. MVA-572	53.05	0.02	18.54	10.21	27.63
Vaccinia virus str. Lister	21.06	0.01	7.82	3.88	9.31

Table 3.3: Actual (bold) and fitted concentrations for top targets in each bacterial or viral family represented in Latin square dataset, for one replicate array hybridized to each of samples 2 through 6.

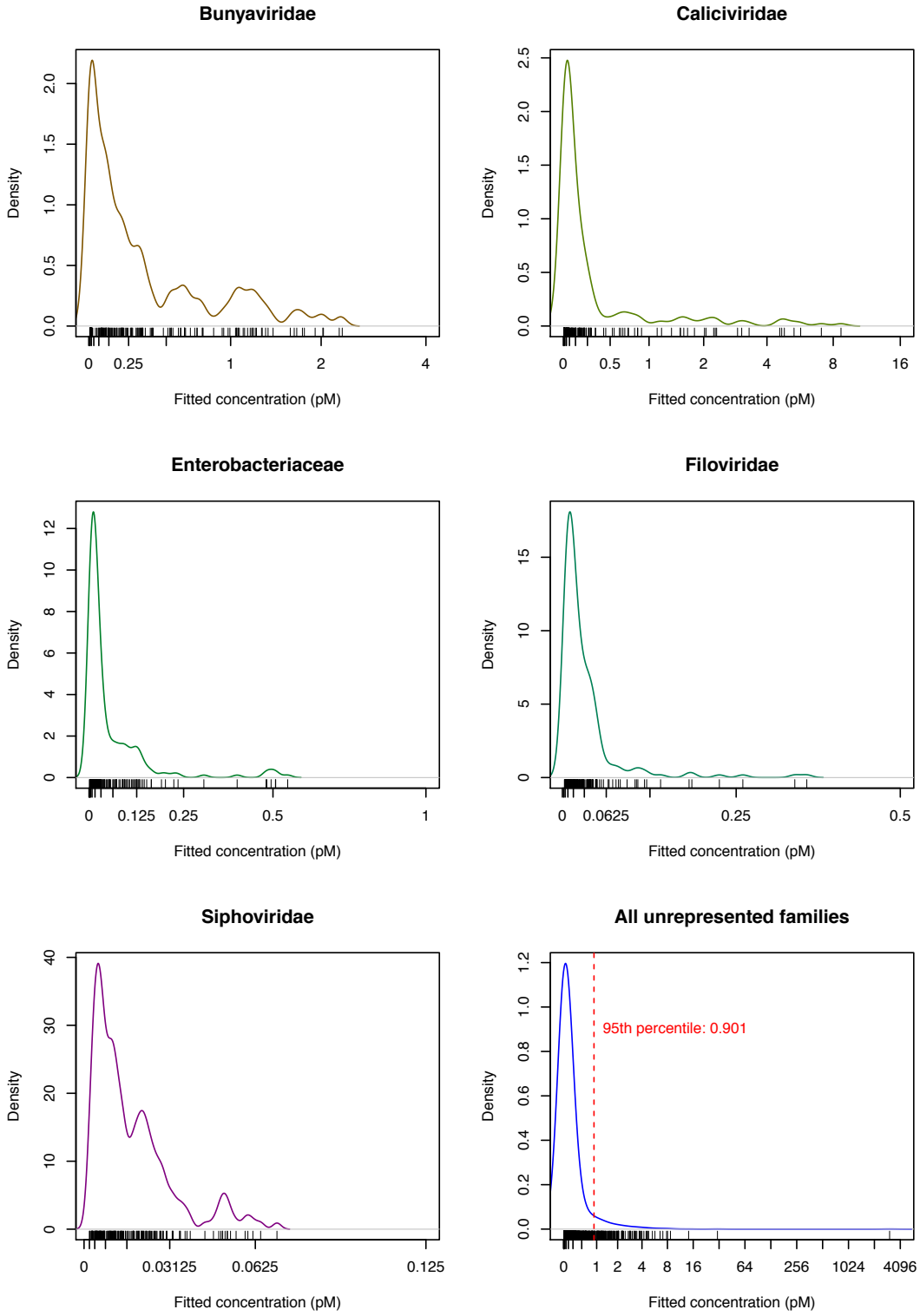

Figure 3.15: Kernel density plots of fitted $\log_2$ concentration distributions for decoy targets in selected families not represented in the Latin square experiment samples; plus overall distribution for all 13 unrepresented families. Axis labels indicate untransformed concentrations; scale varies between plots.

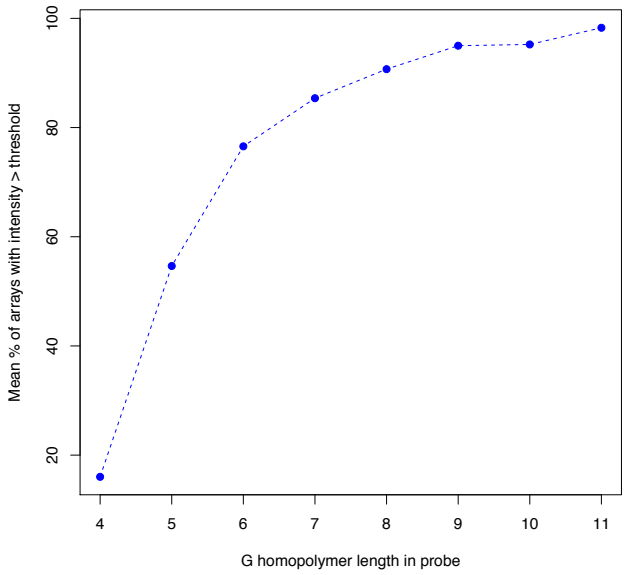

Figure 3.16: Mean binding rates over 581 LLMDA version 5 arrays for probes containing G homopolymers, as a function of polymer length.

array experiments in which the probe intensity is above an array-specific threshold (the 99^{th} percentile of the negative control probe intensities for the array), we see a strong association between the maximum G polymer length and the binding rate, as shown in Figure 3.16. The plot shows average binding rates across the first 581 LLMDA version 5 experiments, which involved a diverse variety of samples. Thus, many probes that were designed against HDV end up binding nonspecifically with high affinity, leading to frequent false detection of HDV.

3.4.3 Issues with strain-level target identification

In my formulation of the latent data I/Q model, I made the assumption that a probe matching multiple related species would have much greater affinity for one target than for another, so that one target would account for most of the binding to a particular probe. When the decoy targets in the candidate set all vary in sequence from the true targets at the genome locations covered by probes, the MIQ algorithm does a good job of assigning large mixture weights to the correct targets. When the candidate set contains several targets that are matched with nearly equal affinities by the probes against the true target, the algorithm tends to distribute the weights equally among the targets similar to the true target. The result is that the fitted model contains nonzero concentration terms for multiple targets, making it difficult to identify one target as being uniquely present.

Figure 3.18 illustrates these two cases. The bar plots show the fitted values of the mixing parameters ϕ_{ij} for one array, for the probes against targets Bt (chromosome 2), Ft and Ba.

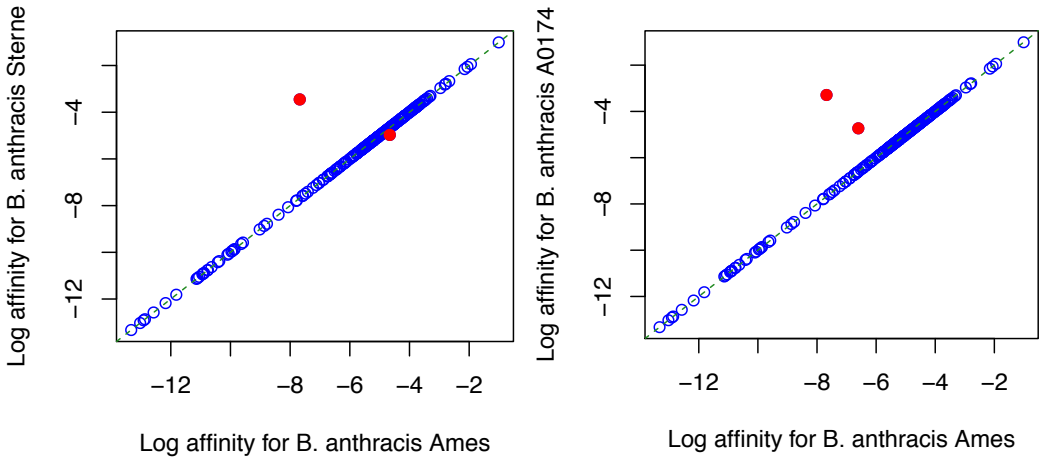

Figure 3.17: Comparison of affinities K_{ij} for probes matching *B. anthracis* str. Ames (target Ba) and decoy strains Sterne and A0174.

The bars indicate the ϕ_{ij} values for each probe, with segments colored by target; the dark blue segment indicates the proportion assigned to the correct target. Bars are ordered from left to right in decreasing order of probe intensity. In general, a much larger fraction of the mixture weights are assigned to the correct target for the Bt probes than for Ft or Ba. For Ft, the mixture weights for most probes are divided fairly evenly between the true target (*F. tularensis* LVS) and a number of other strains in the same subspecies of *F. tularensis*. The probes for Ba follow a similar pattern as for Ft, with equal weights for multiple strains of *B. anthracis*.

When I compared the affinities of the Ba probes for the correct target (*B. anthracis* strain Ames) to those for two other targets in the same species that were assigned high concentrations on the arrays where Ba was present (strains Sterne and A0174), I found they were nearly identical for 250 of 251 probes matching Sterne, and 243 of 245 probes matching A0174. (Figure 3.17). These similarities are not seen between Bt and the decoy targets in family Burkholderiaceae. The differences between these families reflect their different representation biases in the candidate target set. *B. anthracis* and *F. tularensis* are both species of special interest for biodefense, for which many isolates have been sequenced. They are also species with relatively little genomic diversity among strains. *B. thailandensis* is not pathogenic to humans and is therefore not a select agent for biodefense (although it is closely related to *B. pseudomallei*, which is). Few strains of *B. thailandensis* have been sequenced, and only one strain in addition to the true target strain E264 was selected for the candidate target set. The Burkholderiaceae are in general much more genetically diverse than *B. anthracis* or *F. tularensis*, so that the alternative MSMB43 strain of *B. thailandensis* was easily distinguished from strain E264.

Thus, when the sample on the array contains targets such as Ba and Ft, with many similar candidate strains, the maximum likelihood solution to the model given by Equation 3.6 is

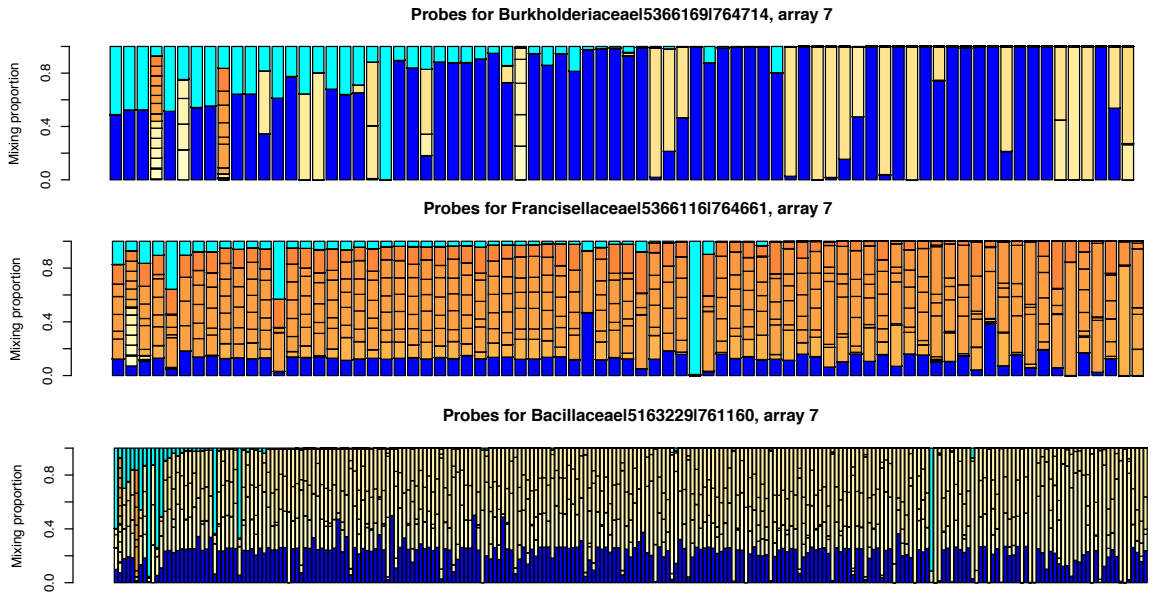

Figure 3.18: Fitted mixing proportions ϕ_{ij} for probes against targets Bt (top) Ft (middle), and Ba (bottom) on array 7 in the Latin square data set. Each bar represents the ϕ_{ij} values for one probe, with segments colored by target: dark blue for the correct target, cyan for the "unbound" contribution, and shades between yellow and red for decoy targets in the same family as the true target.

frequently a mixture of related targets at nonzero concentrations. The mixing parameters ϕ_{ij} provide extra degrees of freedom for fitting concentrations to the probe intensity data. Unfortunately, this means that the "best" solutions can have concentrations for the decoy targets, especially those with overall lower probe affinities, greater than the fitted and actual concentrations for the true target. In the absence of mixture weights, as in the scenario addressed by the quantitation algorithm in Section 3.3, assigning high concentrations to low affinity targets results in deviations between the predicted and observed intensities, particularly at the high end. The mixing parameters reduce these discrepancies by giving the fitting algorithm a means of downweighting them. This is a possible explanation for the large proportions assigned to the special "unbound" target for the brightest probes (the light blue segments seen on the left sides of the barplots in Figure 3.18. Although these solutions fit the data nicely, and give us species-level information about the likely sample composition, we know that, in most real-life samples, only one of the related strains is likely to be present. To get strain-level target identification, we need to change the model and/or the fitting procedure to favor sparser solutions.

3.4.4 Improving the MIQ algorithm

I considered a number of approaches to improve the strain-level identification performance of the MIQ algorithm. The most obvious, and the easiest to implement, was to add an L_1 penalty to the Q function (the expectation of the complete data likelihood) to be maximized in the M step, giving preference to solutions with smaller concentrations and fewer nonzero concentration terms. With the L_1 penalty, equation 3.8 for the Q function becomes:

$$Q(c, \phi; c^{(q)}, \phi^{(q)}) = \sum_{i,k} \left\{ -\frac{u_{ik}^2}{\sigma^2} + \sum_{j \in H_i} \rho_{ij} \left[\log \phi_{ij} - \frac{\mu_{ij}^2}{2\sigma^2} + \frac{u_{ik}\mu_{ij}}{\sigma^2} \right] \right\} - \alpha \sum_{j=1}^{m} |c_j| \qquad (3.11)$$

where α is a tuning parameter that controls the strength of the penalization.

I modified the EM algorithm code to maximize the penalized Q function with respect to the concentrations c_j and mixture weights ϕ_{ij}, as before, under the constraint $c_j \geq 0 \; \forall j$. This is straightforward, except that the code has to deal with the possibility that Q is maximized at a boundary where some of the c_j are zero. To find the optimal value of the α parameter, I performed several fits with values of α ranging from 0 to 0.05.

The effect of adding L_1 penalties of various strengths on the fitted concentrations for an example array is shown in Figure 3.19. The L_1 penalization strategy was partially successful in improving the fit to the model. For Ft, it shrank the concentrations assigned to the decoy strains of *F. tularensis* subspecies holarctica, while leaving the value for the correct LVS strain relatively unchanged; although the decoy concentrations were not reduced to zero at the largest value of α, increasing it further might do so. At $\alpha = 0.05$, the LVS strain was already the top hit.

For Ba, the concentrations most affected by the L_1 penalty were for the two plasmids pX01 and pX02; at $\alpha = 0.05$ their fitted values were much closer to the correct Ba concentration than without penalization. The fitted values for the Ba (*B. anthracis* strain Ames) chromosome and the three decoy strains Sterne, A0174 and A0442 were relatively unaffected by penalization. As we saw in Figure 3.17, nearly all of the Ba probes have identical affinities for the Ames and Sterne strains; not surprisingly, the fitted concentrations were also nearly identical, so that the Sterne concentration curve is hidden below the curve for Ames in Figure 3.19.

One problem with the L_1 penalization scheme (and perhaps any scheme for improving sparsity) is that it gives preference to strains represented by draft genome assemblies, in which all genome elements are grouped into one target sequence, over strains in which the chromosomes and plasmids are treated as separate targets. Targets A0174 and A0442 are both draft genomes which incorporate pX01 and pX02 plasmid sequences together with the chromosome sequence. In this case, the L_1 penalty is smaller for a solution that assigns concentration $3x$ to strain A0174 than for one that assigns concentration $x + \epsilon$ (with $\epsilon > 0$) to each of the separate chromosome and plasmid targets. This is most likely the reason why the fitted concentrations for the Ba plasmids decrease with increasing α, while the concentrations for the draft genome strains stay approximately the same. The only solution

to this problem is to modify the candidate target database so that draft assemblies and finished genomes are treated the same: either by grouping genome elements for finished genomes, or splitting the contigs in draft genomes into separate target sequences for each chromosome and plasmid.

There are other potential ways to modify the identification/quantitation algorithm to favor sparse sets of targets, and thus reduce ambiguity in identifying the microbial strains that are present. One approach is to add an L_∞ penalty to the Q function, i.e. a negative term proportional to the number of targets with nonzero concentrations. Another is to change the support of the latent index variable t_i to exclude the case $t_i = m + 1$, indicating that the probe is not bound to any target. This would remove the ability of the MIQ algorithm to assign large mixture weights for the "unbound" target to high affinity probes whose observed intensities deviate from the values predicted at some target concentration, and thereby diminish their negative effect on the Q function value. Finally, we could replace the EM iterations with a greedy likelihood maximization procedure, which would make repeated scans through the candidate set to construct an "identified target list". In each scan, we would use the quantitation algorithm of Section 3.3 to fit a concentration for each target in succession. We would then compute the resulting change in the log likelihood (Equation 3.5) given the observed intensities, the fitted concentration, and the previously identified targets, and add to the identified list the target yielding the greatest increase, or terminate the scans if adding further targets would decrease the likelihood. This greedy algorithm could be combined with the latent data model underlying the MIQ algorithm, with the restriction that nonzero values for the mixture weights ϕ_{ij} could only be assigned to sequences in the identified target list. All of these approaches will be investigated as I continue development of the MIQ algorithm.

3.5 Conclusions and future directions

In this dissertation, I have demonstrated methods of using detection microarray data to accurately identify the microbial species present in a sample and produce quantitative estimates of their abundances. Since my colleagues at Lawrence Livermore and elsewhere continue to develop and find new applications for these microarrays, further refinements of my identification and quantitation methods will be needed to support these applications. In the preceding section, I proposed some changes to the MIQ algorithm to facilitate target identification at the strain level. Additional improvements in sensitivity, specificity, and quantitation accuracy will require that we resolve some more global issues. I expect to address some of these in the not too distant future, but some of them will have to be solved by my colleagues or by outside entities.

One of the most basic requirements for any microbial detection assay, whether based on microarrays, PCR, or DNA sequencing, is a well-curated reference sequence database that includes metadata linking the elements (chromosomes, plasmids, and viral segments) of microbial genomes, and linking genomes to taxonomy. A major effort is currently underway

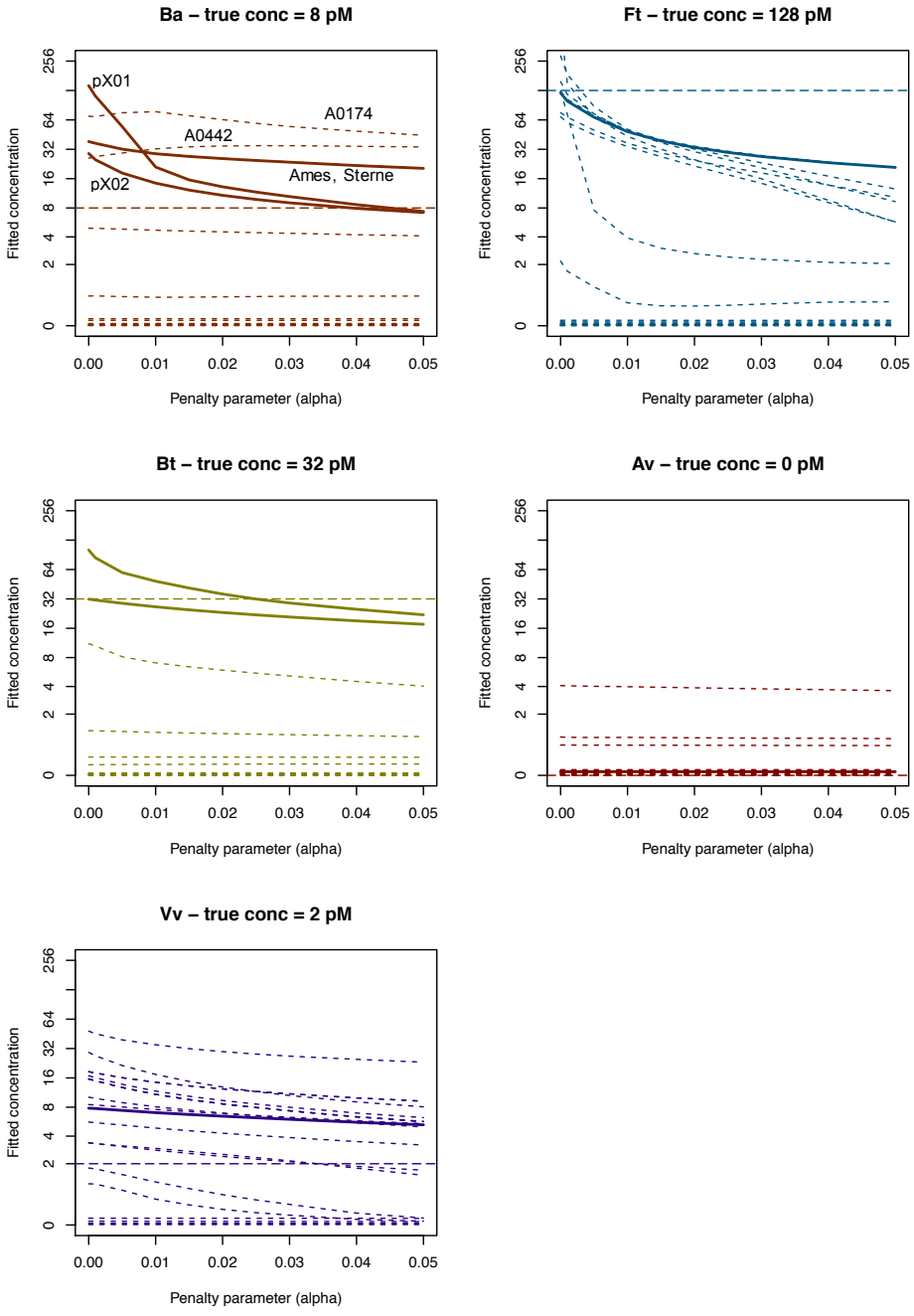

Figure 3.19: L_1 penalized likelihood fits of candidate target concentrations to intensities on array 7 in Latin square data set, plotted against the penalty parameter α. Correct targets are indicated by solid lines, decoys by dashed lines. A horizontal dashed line indicates the actual concentration of the true target.

at the National Center for Biotechnology Information (NCBI), with input from the FDA, to create such a database. This compendium of high quality genome sequences will greatly enhance our ability to design microarray probes and predict what targets they will bind to.

Another key ingredient for measuring concentrations of microbial DNA is a set of accurate affinities for the relevant probe-target pairs. In Chapter 2, I developed a scheme for predicting hybridization free energies and affinities from probe and target sequences using a position-dependent nearest-neighbor (PDNN) model. I fit the parameters in this model using data from tiling array experiments that were designed for an entirely different purpose, namely, validating arrays for SNP discovery. Ideally, the PDNN parameters would be fit to data from a carefully designed experiment that included a wide range of probe-target mismatch configurations, with a large number of probes for each target genome sequence, and with target genomes spanning a wide range of percent GC contents.

I later compared the affinities predicted by the PDNN model to affinities that were directly measured by hybridizing *Thermotoga* DNA to arrays at known concentrations. I observed systematic deviations between the two sets of affinities, which seemed to depend on the number and placement of mismatches between the probe and target sequences. It would be useful to perform similar comparisons for additional sets of targets that were hybridized at known concentrations, to further elucidate the issues with the PDNN model predictions.

In section 3.4.2, I noted that probes containing long G homopolymers frequently bound nonspecifically to the majority of the samples we tested. The same is true for probes containing other types of low complexity sequences, such as tandem repeats. The LLMDA contains several thousand other probes lacking low complexity sequences that have similar nonspecific binding behavior. We refer to these as "sticky" or "promiscuous" probes. There is a growing literature, reviewed in [Harrison 13], in which microarray researchers have identified probe sequence patterns that correlate with promiscuous binding and other anomalous behaviors, and attempted to explain them by physico-chemical models. This body of work needs to be incorporated into a predictive model of probe-target hybridization, to augment the PDNN model.

In conclusion, I believe that microbial detection microarrays have a promising future for a variety of public health, environmental monitoring, energy, food and drug safety, and medical applications. Robust analysis algorithms that can produce reliable, easily interpretable, actionable information from their extremely complex signals will be essential for their success. The work I've presented here is an important step toward making these algorithms possible, and thus toward the evolution of microarrays from research tools into commonplace devices for diagnosis and surveillance.

Bibliography

[Allred 10] Adam F Allred, Guang Wu, Tuya Wulan, Kael F Fischer, Michael R Holbrook, Robert B Tesh & David Wang. *VIPR: A probabilistic algorithm for analysis of microbial detection microarrays*. BMC Bioinformatics, vol. 11, page 384, 2010.

[Altschul 90] S F Altschul, W Gish, W Miller, E W Myers & D J Lipman. *Basic local alignment search tool*. Journal of Molecular Biology, vol. 215, no. 3, pages 403–410, October 1990.

[Bailey 98] T L Bailey & M Gribskov. *Combining evidence using p-values: application to sequence homology searches*. Bioinformatics, vol. 14, no. 1, pages 48–54, 1998.

[Bej 90] A K Bej, M H Mahbubani, R Miller, J L DiCesare, L Haff & R M Atlas. *Multiplex PCR amplification and immobilized capture probes for detection of bacterial pathogens and indicators in water*. Molecular and Cellular Probes, vol. 4, no. 5, pages 353–365, October 1990.

[Belosludtsev 04] Yuri Y Belosludtsev, Dawn Bowerman, Ryan Weil, Nishanth Marthandan, Robert Balog, Kevin Luebke, Jonathan Lawson, Stephen A Johnston, C Rick Lyons, Kevin Obrien, Harold R Garner & Thomas F Powdrill. *Organism identification using a genome sequence-independent universal microarray probe set*. BioTechniques, vol. 37, no. 4, pages 654–8, 660, October 2004.

[Bengtsson 04] Henrik Bengtsson, Göran Jönsson & Johan Vallon-Christersson. *Calibration and assessment of channel-specific biases in microarray data with extended dynamical range*. BMC Bioinformatics, vol. 5, page 177, November 2004.

[Bengtsson 08] Henrik Bengtsson, J Bullard, K Simpson & K Hansen. *aroma - An R Object-oriented Microarray Analysis environment*. Department of Statistics, UC Berkeley Technical Report, no. 745, 2008.

[Bolstad 03] B M Bolstad, R A Irizarry, M Astrand & T P Speed. *A comparison of normalization methods for high density oligonucleotide array data based on variance and bias.* Bioinformatics, vol. 19, no. 2, pages 185–193, January 2003.

[Bolstad 04] Benjamin M Bolstad. *Low-level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization.* PhD thesis, UC Berkeley, May 2004.

[Burden 04] Conrad J Burden, Yvonne E Pittelkow & Susan R Wilson. *Statistical analysis of adsorption models for oligonucleotide microarrays.* Statistical Applications in Genetics and Molecular Biology, vol. 3, page Article35, 2004.

[Chen 11] Eunice C Chen, Steve A Miller, Joseph L DeRisi & Charles Y Chiu. *Using a pan-viral microarray assay (Virochip) to screen clinical samples for viral pathogens.* Journal of Visualized Experiments : JoVE, no. 50, 2011.

[Cohen 50] AC Cohen. *Estimating the mean and variance of normal populations from singly truncated and doubly truncated samples.* The Annals of Mathematical Statistics, vol. 21, no. 4, pages 557–557569, 1950.

[Dai 02] Hongyue Dai, Michael Meyer, Sergey Stepaniants, Michael Ziman & Roland Stoughton. *Use of hybridization kinetics for differentiating specific from non-specific binding to oligonucleotide microarrays.* Nucleic Acids Research, vol. 30, no. 16, page e86, August 2002.

[Dodd 04] Lori E Dodd, Edward L Korn, Lisa M McShane, G V R Chandramouli & Eric Y Chuang. *Correcting log ratios for signal saturation in cDNA microarrays.* Bioinformatics, vol. 20, no. 16, pages 2685–2693, November 2004.

[Erlandsson 11] Lena Erlandsson, Maiken W Rosenstierne, Kevin McLoughlin, Crystal Jaing & Anders Fomsgaard. *The microbial detection array combined with random Phi29-amplification used as a diagnostic tool for virus detection in clinical samples.* PLoS One, vol. 6, no. 8, page e22631, 2011.

[Gardner 10] Shea N Gardner, Crystal J Jaing, Kevin S McLoughlin & Tom R Slezak. *A microbial detection array (MDA) for viral and bacterial detection.* BMC Genomics, vol. 11, page 668, 2010.

[Gibas 10] Cynthia J Gibas. *The Biophysics of DNA Microarrays.* In Robert Splinter, editeur, Handbook of Physics in Medicine and Biology, page 42. CRC Press, 2010.

[Golub 89] G H Golub & C F Van Loan. Matrix Computations. Johns Hopkins University Press, 2nd edition, 1989.

[Hamamatsu 06] Hamamatsu. Photomultiplier Tubes: Basics and Applications. Hamamatsu Photonics, 3rd edition, 2006.

[Harrison 13] Andrew Harrison, Hans Binder, Arnaud Buhot, Conrad J Burden, Enrico Carlon, Cynthia Gibas, Lara J Gamble, Avraham Halperin, Jef Hooyberghs, David P Kreil, Rastislav Levicky, Peter A Noble, Albrecht Ott, B Montgomery Pettitt, Diethard Tautz & Alexander E Pozhitkov. *Physicochemical foundations underpinning microarray and next-generation sequencing experiments.* Nucleic Acids Research, vol. 41, no. 5, pages 2779–2796, March 2013.

[Held 03] G A Held, G Grinstein & Y Tu. *Modeling of DNA microarray data by using physical properties of hybridization.* Proceedings of the National Academy of Sciences of the United States of America, vol. 100, no. 13, pages 7575–7580, June 2003.

[Hooyberghs 09] J Hooyberghs, P Van Hummelen & E Carlon. *The effects of mismatches on hybridization in DNA microarrays: determination of nearest neighbor parameters.* Nucleic Acids Research, vol. 37, no. 7, page e53, April 2009.

[Hooyberghs 10] J Hooyberghs, M Baiesi, A Ferrantini & E Carlon. *Breakdown of thermodynamic equilibrium for DNA hybridization in microarrays.* Physical Review E, Statistical, Nonlinear, and Soft Matter Physics, vol. 81, no. 1 Pt 1, page 012901, 2010.

[Hughes 01] T R Hughes, M Mao, A R Jones, J Burchard, M J Marton, K W Shannon, S M Lefkowitz, M Ziman, J M Schelter, M R Meyer, S Kobayashi, C Davis, H Dai, Y D He, S B Stephaniants, G Cavet, W L Walker, A West, E Coffey, D D Shoemaker, R Stoughton, A P Blanchard, S H Friend & P S Linsley. *Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer.* Nature Biotechnology, vol. 19, no. 4, pages 342–347, April 2001.

[Irizarry 03] Rafael A Irizarry, Benjamin M Bolstad, Francois Collin, Leslie M Cope, Bridget Hobbs & Terence P Speed. *Summaries of Affymetrix GeneChip probe level data.* Nucleic Acids Research, vol. 31, no. 4, page e15, February 2003.

[Jaing 08] Crystal Jaing, Shea Gardner, Kevin McLoughlin, Nisha Mulakken, Michelle Alegria-Hartman, Phillip Banda, Peter Williams, Pauline Gu, Mark Wagner, Chitra Manohar & Tom Slezak. *A functional gene array*

for detection of bacterial virulence elements. PLoS One, vol. 3, no. 5, page e2163, 2008.

[Kooperberg 02] Charles Kooperberg, Thomas G Fazzio, Jeffrey J Delrow & Toshio Tsukiyama. *Improved background correction for spotted DNA microarrays.* Journal of Computational Biology, vol. 9, no. 1, pages 55–66, 2002.

[Ksiazek 03] Thomas G Ksiazek, Dean Erdman, Cynthia S Goldsmith, Sherif R Zaki, Teresa Peret, Shannon Emery, Suxiang Tong, Carlo Urbani, James A Comer, Wilina Lim, Pierre E Rollin, Scott F Dowell, Ai-Ee Ling, Charles D Humphrey, Wun-Ju Shieh, Jeannette Guarner, Christopher D Paddock, Paul Rota, Barry Fields, Joseph DeRisi, Jyh-Yuan Yang, Nancy Cox, James M Hughes, James W LeDuc, William J Bellini, Larry J Anderson & SARS Working Group. *A novel coronavirus associated with severe acute respiratory syndrome.* The New England Journal of Medicine, vol. 348, no. 20, pages 1953–1966, May 2003.

[Langdon 09] William B Langdon, Graham J G Upton & Andrew P Harrison. *Probes containing runs of guanines provide insights into the biophysics and bioinformatics of Affymetrix GeneChips.* Briefings in Bioinformatics, vol. 10, no. 3, pages 259–277, May 2009.

[Leski 09] Tomasz A Leski, Baochuan Lin, Anthony P Malanoski, Zheng Wang, Nina C Long, Carolyn E Meador, Brian Barrows, Sofi Ibrahim, Justin P Hardick, Mohamed Aitichou, Joel M Schnur, Clark Tibbetts & David A Stenger. *Testing and validation of high density resequencing microarray for broad range biothreat agents detection.* PLoS One, vol. 4, no. 8, page e6569, 2009.

[Lin 06] Baochuan Lin, Zheng Wang, Gary J Vora, Jennifer A Thornton, Joel M Schnur, Dzung C Thach, Kate M Blaney, Adam G Ligler, Anthony P Malanoski, Jose Santiago, Elizabeth A Walter, Brian K Agan, David Metzgar, Donald Seto, Luke T Daum, Russell Kruzelock, Robb K Rowley, Eric H Hanson, Clark Tibbetts & David A Stenger. *Broad-spectrum respiratory tract pathogen identification using resequencing DNA microarrays.* Genome Research, vol. 16, no. 4, pages 527–535, April 2006.

[Malanoski 06] Anthony P Malanoski, Baochuan Lin, Zheng Wang, Joel M Schnur & David A Stenger. *Automated identification of multiple micro-organisms from resequencing DNA microarrays.* Nucleic Acids Research, vol. 34, no. 18, pages 5300–5311, 2006.

[Markham 08] Nicholas R Markham & Michael Zuker. *UNAFold: software for nucleic acid folding and hybridization.* Methods in Molecular Biology, vol. 453, pages 3–31, 2008.

[Mullis 86] K Mullis, F Faloona, S Scharf, R Saiki, G Horn & H Erlich. *Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction.* Cold Spring Harbor Symposia on Quantitative Biology, vol. 51 Pt 1, pages 263–273, 1986.

[Palacios 07] Gustavo Palacios, Phenix-Lan Quan, Omar J Jabado, Sean Conlan, David L Hirschberg, Yang Liu, Junhui Zhai, Neil Renwick, Jeffrey Hui, Hedi Hegyi, Allen Grolla, James E Strong, Jonathan S Towner, Thomas W Geisbert, Peter B Jahrling, Cornelia Büchen-Osmond, Heinz Ellerbrok, Maria Paz Sanchez-Seco, Yves Lussier, Pierre Formenty, M Stuart T Nichol, Heinz Feldmann, Thomas Briese & W Ian Lipkin. *Panmicrobial oligonucleotide array for diagnosis of infectious diseases.* Emerging Infectious Diseases, vol. 13, no. 1, pages 73–81, 2007.

[Pease 94] A C Pease, D Solas, E J Sullivan, M T Cronin, C P Holmes & S P Fodor. *Light-generated oligonucleotide arrays for rapid DNA sequence analysis.* Proceedings of the National Academy of Sciences of the United States of America, vol. 91, no. 11, pages 5022–5026, May 1994.

[Quan 07] Phenix-Lan Quan, Gustavo Palacios, Omar J Jabado, Sean Conlan, David L Hirschberg, Francisco Pozo, Philippa J M Jack, Daniel Cisterna, Neil Renwick, Jeffrey Hui, Andrew Drysdale, Rachel Amos-Ritchie, Elsa Baumeister, Vilma Savy, Kelly M Lager, Jürgen A Richt, David B Boyle, Adolfo García-Sastre, Inmaculada Casas, Pilar Perez-Breña, Thomas Briese & W Ian Lipkin. *Detection of respiratory viruses and subtype identification of influenza A viruses by GreeneChipResp oligonucleotide microarray.* Journal of Clinical Microbiology, vol. 45, no. 8, pages 2359–2364, August 2007.

[R D 11] R Development Core Team. *R: A Language and Environment for Statistical Computing*, April 2011.

[Ratushna 05] Vladyslava G Ratushna, Jennifer W Weller & Cynthia J Gibas. *Secondary structure in the target as a confounding factor in synthetic oligomer microarray design.* BMC Genomics, vol. 6, no. 1, page 31, 2005.

[Reed 03] William J Reed & Murray Jorgensen. *The Double Pareto-Lognormal Distribution - A New Parametric Model for Size Distributions.* Communications in Statistics - Theory & Methods, vol. 33, pages 1733–1753, 2003.

[Rehrauer 08] Hubert Rehrauer, Susan Schönmann, Leo Eberl & Ralph Schlapbach. *PhyloDetect: a likelihood-based strategy for detecting microorganisms with diagnostic microarrays.* Bioinformatics, vol. 24, no. 16, pages i83–9, August 2008.

[Ritchie 07] Matthew E Ritchie, Jeremy Silver, Alicia Oshlack, Melissa Holmes, Dileepa Diyagama, Andrew Holloway & Gordon K Smyth. *A comparison of background correction methods for two-colour microarrays.* Bioinformatics, vol. 23, no. 20, pages 2700–2707, October 2007.

[Roberts 12] Adam Roberts & Lior Pachter. *Streaming fragment assignment for real-time analysis of sequencing experiments.* Nature Methods, vol. 10, no. 1, pages 71–73, January 2012.

[Ruppert 03] D Ruppert, M P Wand & R J Carroll. Semiparametric Regression. Cambridge, 2003.

[Sanger 77] F Sanger, S Nicklen & A R Coulson. *DNA sequencing with chain-terminating inhibitors.* Proceedings of the National Academy of Sciences of the United States of America, vol. 74, no. 12, pages 5463–5467, December 1977.

[SantaLucia 04] John SantaLucia & Donald Hicks. *The thermodynamics of DNA structural motifs.* Annual Review of Biophysics and Biomolecular Structure, vol. 33, pages 415–440, 2004.

[Sartor 04] Maureen Sartor, Jennifer Schwanekamp, Danielle Halbleib, Ismail Mohamed, Saikumar Karyala, Mario Medvedovic & Craig R Tomlinson. *Microarray results improve significantly as hybridization approaches equilibrium.* BioTechniques, vol. 36, no. 5, pages 790–796, May 2004.

[Schena 95] M Schena, D Shalon, R W Davis & P O Brown. *Quantitative monitoring of gene expression patterns with a complementary DNA microarray.* Science, vol. 270, no. 5235, pages 467–470, October 1995.

[Silver 09] J D Silver, M E Ritchie & G K Smyth. *Microarray background correction: maximum likelihood estimation for the normal-exponential convolution.* Biostatistics, vol. 10, no. 2, pages 352–363, August 2009.

[Smyth 03] Gordon K Smyth, Yee Hwa Yang & Terry Speed. *Statistical issues in cDNA microarray data analysis.* Methods in Molecular Biology, vol. 224, pages 111–136, 2003.

[Taub 09] Margaret Taub. *Analysis of high-throughput biological data: some statistical problems in RNA-seq and mouse genotyping.* PhD thesis, UC Berkeley, December 2009.

[Upton 08] Graham Jg Upton, William B Langdon & Andrew P Harrison. *G-spots cause incorrect expression measurement in Affymetrix microarrays.* BMC Genomics, vol. 9, page 613, 2008.

[Urisman 05] Anatoly Urisman, Kael F Fischer, Charles Y Chiu, Amy L Kistler, Shoshannah Beck, David Wang & Joseph L DeRisi. *E-Predict: a computational strategy for species identification based on observed DNA microarray hybridization patterns.* Genome Biology, vol. 6, no. 9, page R78, 2005.

[Vandenvelde 90] C Vandenvelde, M Verstraete & D Van Beers. *Fast Multiplex polymerase chain reaction on boiled clinical samples for rapid viral diagnosis.* Journal of Virological Methods, vol. 30, no. 2, pages 215–227, November 1990.

[Victoria 10] Joseph G Victoria, Chunlin Wang, Morris S Jones, Crystal Jaing, Kevin McLoughlin, Shea Gardner & Eric L Delwart. *Viral nucleic acids in live-attenuated vaccines: detection of minority variants and an adventitious virus.* Journal of Virology, vol. 84, no. 12, pages 6033–6040, June 2010.

[Wang 02] David Wang, Laurent Coscoy, Maxine Zylberberg, Pedro C Avila, Homer A Boushey, Don Ganem & Joseph L DeRisi. *Microarray-based detection and genotyping of viral pathogens.* Proceedings of the National Academy of Sciences of the United States of America, vol. 99, no. 24, pages 15687–15692, November 2002.

[Wang 03] David Wang, Anatoly Urisman, Yu-Tsueng Liu, Michael Springer, Thomas G Ksiazek, Dean D Erdman, Elaine R Mardis, Matthew Hickenbotham, Vincent Magrini, James Eldred, J Phillipe Latreille, Richard K Wilson, Don Ganem & Joseph L DeRisi. *Viral discovery and sequence recovery using DNA microarrays.* PLoS Biology, vol. 1, no. 2, page E2, November 2003.

[Watson 07] Michael Watson, Juliet Dukes, Abu-Bakr Abu-Median, Donald P King & Paul Britton. *DetectiV: visualization, normalization and significance testing for pathogen-detection microarray data.* Genome Biology, vol. 8, no. 9, page R190, 2007.

[Zhang 03] Li Zhang, Michael F Miles & Kenneth D Aldape. *A model of molecular interactions on short oligonucleotide microarrays.* Nature Biotechnology, vol. 21, no. 7, pages 818–821, July 2003.