# Resequencing Data of 20 *Arabidopsis* Ecotypes

Georg Zeller

Diploma thesis in Bioinformatics
Universität Tübingen

supervised by
Richard Clark, PhD[⋆], Dr. Gunnar Rätsch[‡],
Prof. Daniel Huson[♯] and Prof. Detlef Weigel[⋆]

November 15, 2005

[⋆] Max Planck Institute for Developmental Biology, Tübingen
[‡] Friedrich Miescher Laboratory of the Max Planck Society, Tübingen
[♯] Center for Bioinformatics Tübingen (ZBIT), University of Tübingen

# Abstract

*This diploma thesis describes work on a chip resequencing project of 20 ecotypes belonging to the plant model species* Arabidopsis thaliana*, and these ecotypes are accessions from natural populations. Chip resequencing primarily aims at identifying single nucleotide polymorphisms (SNPs), the most abundant class of naturally occurring sequence variation. For resequencing, DNA microarrays are employed on which a genome-wide tiling of 25-mer probes is spotted. These probes are designed complementary to an* a priori *reference genome sequence. For each interrogated site probes with any of the four possible nucleotides in the middle are represented so that a nucleotide substitution in the interrogated genome will generally lead to a hybridization signal that is strongest for the corresponding non-reference probe at a SNP position.*

*The huge data set resulting from the resequencing of 20 genomes of ∼125 Mb has been stored in a MySQL database and a viewer has been implemented in Java for graphical display of resequencing data recovered directly from the database.*

*Part of this thesis is a basic characterization of the resequencing data. Intensity and specificity of hybridization exhibit a large degree of variability, the difference in intensity being more than 10-fold in extreme cases. Examinations revealed that this variability is in part caused by experimental factors, and in part determined by sequence properties of the probe. High AT content and self-complementarity, favoring hairpin formation, negatively affect hybridization, whereas probes with high-complexity sequences, measured by sequence entropy, hybridize better on average.*

*In order to estimate the potential of a given probe for cross-hybridization to multiple DNA sequence tracts in the genome, a systematic search for repeated 25-mers in the reference genome has been conducted. The result suggests that more than 90 % false SNP calls in the reference ecotype, Col-0, are caused by cross-hybridization found with this search method. The error rates for SNP calls in other ecotypes can be improved with a filter based on 25-mer matches.*

*Finally, an algorithm has been developed for the prediction of large deletions from resequencing data. It is a comparative loss-of-signal approach that identifies regions where the target ecotype exhibits strongly reduced hybridization signal relative to the reference. More than 700 deletions larger than 200 bp have been predicted for the ecotype Ler-1 some of which are accurate estimates of deletions known from dideoxy sequencing. The main obstacles for deletion calling are regions which are repetitive or produce an ambiguous hybridization signal from the reference. This leads to uncertainties about start and end points of putative deletions. As the set of known large deletions in Ler-1 is incomplete, it is difficult to assess the specificity of our deletion calling heuristic. Indirect evaluations suggest that among the predictions the number of true deletions is higher than the number of false positives. A better assessment will be possible when some regions containing putative deletions have been sequenced.*

## Acknowledgement

Many thanks go to Christian Klug, Tobias Dezulian and Tobias Klöpper for their help and support in the early stage, especially with the computation of repeated $k$-mers and VMATCH.

I wish to thank Stephan Ossowski for his advices and discussions on MySQL databases and many other bioinformatics topics, and Norman Warthmann for sharing his insights into various biological phenomena. Special thanks to Gabriele Schweikert for discussions on machine learning and particularly for reading and commenting on a draft of this diploma thesis.

I am very grateful to Gunnar Rätsch for his valuable ideas and advices which will also keep me busy in the future. Particular thanks go to Richard Clark for his guidance, support, discussions and encouragement throughout. Moreover, I am very grateful for his review of draft chapters and the suggestions he made to significantly improve this work.

Finally, I would like to thank Daniel Huson and Detlef Weigel who made it possible for me to work on this outstanding resequencing project.

## Statement of Originality

I hereby declare that the contents of this diploma thesis are my own original work and to the best of my knowledge it contains no material previously published or written by another person except where otherwise stated. Other sources are acknowledged in the text and a bibliography is appended.

Tübingen, November 15, 2005

# Contents

# 1 Introduction

## 1.1 *Arabidopsis thaliana* as a model for genetics

*Arabidopsis thaliana*, Thale Cress, is a small flowering plant. It is a member of the mustard family (Brassicaceae or Cruciferae) to which agricultural plants such as cabbage, cauliflower, radish and rapeseed also belong. It is an annual weed that procreates predominantly through self-pollination.

*Arabidopsis* has become a model organism of choice for molecular plant biology for several reasons. It is small enough to be grown in petri dishes and later in small pots as mature plants reach about 20 cm in height. It can grow quickly, so that the whole life-cycle is completed in 6 weeks. Because it is self-pollinating, accessions from natural populations already exhibit a small degree of heterozygosity and inbred lines are easily produced. More than 1000 accessions from natural populations or ecotypes[1] have been collected world-wide. As with other model organisms, a broad research community has produced an extensive collection of mutants. Genetic protocols for transformation mediated by *Agrobacterium* or transposon mutagenesis have been established [21], [16].

A major reason activating genetic research using *Arabidopsis* came from the finding that the genome is small and enriched for coding sequences and thus sequencing is cost-effective, especially in comparison to many crop plants whose genomes are large and polyploid (i.e. the result of genome duplications and fusions). The *Arabidopsis* genome comprises about 125 million euchromatic bases organized into five chromosomes. The sequencing project for the genome of the *Arabidopsis* accession Columbia (Col-0) was finished in 2000 and resulted in a high-quality sequence with a small number of gaps and an estimated error rate of approximately 1:10 000 [13].

## 1.2 Natural variation and genetic polymorphism

Once the genome sequences of several higher organisms were published, questions about naturally occurring polymorphisms in individuals and populations have been a focus of genome research in these organisms. In humans this line of research is driven by the discovery that in many cases segregating polymorphisms are the genetic cause underlying susceptibility to diseases. But natural variation is of much more fundamental interest to biology and some biological questions to be asked are mentioned in the following.

*Arabidopsis* is well suited for studies of natural variation as ecotypes collected from different places around the world exhibit striking morphological differences although they all belong to the same species and it is difficult to infer an unambiguous phylogeny between ecotypes.

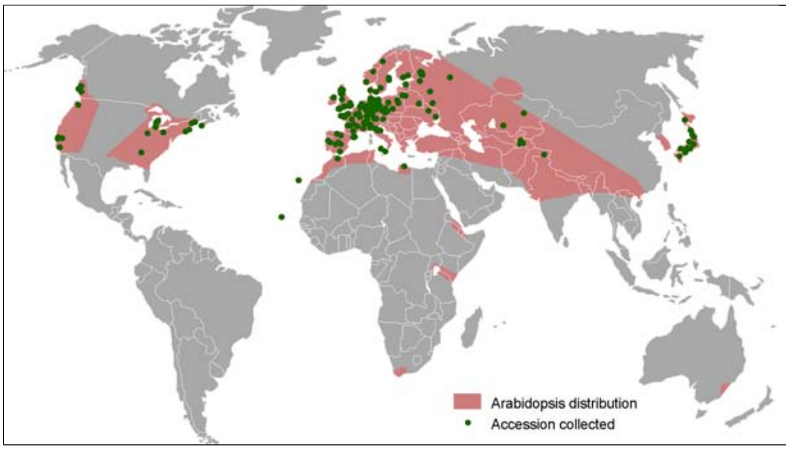

**Figure 1-1**
*Arabidopsis* distribution and places where accessions have been collected. (Taken from natural-eu.org)

[1] In the following we mostly use the term "ecotype" even if "accession" may be more appropriate from an ecological point of view.

Beyond morphology, variation is observed in many other traits such as flowering time in early or late summer and connected to that a life-cycle as either winter or summer-annual. Other variable traits include growth rate, resistance factors for pathogens such as bacteria and for herbivores such as insects. Natural variation is also observed in biochemical traits such as enzyme activity, DNA methylation and gene expression level. Thus, *Arabidopsis*, with its world-wide distribution and colonization of ecologically diverse habitats, is well suited for studies of natural variation between ecotypes [16]. Figure 1-1 illustrates the *Arabidopsis* geographical distribution and places from which accessions where collected[2]. For the project to which this diploma thesis refers, 20 ecotypes have been studied: one reference accession Col-0 is compared to ecotypes from Canada (Van-0), Cape Verde Islands (Cvi-0), Czech Republic (Bor-4 and Br-0), England (Nfa-8), Estland (Est-1), Finland (Tamm-2), Germany (Bay-0 and Got-7), Ireland (Bur-0), Japan (Tsu-1), Poland (Ler-1), Portugal (C24 and Fei-0), Spain (Ts-1), Sweden (Lov-5), Tajikistan (Shahdara, in the following referred to as Sha) and the United States (RRS-7 and RRS-10).

As genetic research in general aims at finding genes causal to a certain observed phenotype, the question, how phenotypic variation is caused by variation on the DNA level, lies at the heart of genetics. However, it can be difficult to pinpoint the genes and alleles which are causal to variation in a specific trait. Variation in complex traits is not caused by only one or two major genetic differences with a Mendelian segregation pattern. Instead, allelic differences in many loci contribute to a certain phenotype and none of these contributions is necessarily large. Identification of these loci and their alleles in the genome can be accomplished with statistical methods such as quantitative trait locus (QTL) mapping. The identification of a QTL has to be confirmed by several experiments in conjunction. A promising approach to identify candidate loci contributing to a quantitative trait is also by association mapping [26].

Association mapping requires genome-wide knowledge of variable sites in high density and an efficient and reliable method to genotype individuals at many sites (markers). Once these data are available, one identifies sequence variants that are statistically correlated with certain phenotypic variants in the trait of interest. This technique is especially promising in *Arabidopsis*, because the distance, in which adjacent markers in the genome are correlated, is generally small, i.e. linkage equilibrium decays rapidly (within 25-50 kb) [23], [26]. Thus, association mapping should be possible with relatively high resolution with respect to marker density.

Discovering sequence variation in high density on a genome scale has motivated the resequencing of the 20 *Arabidopsis* ecotypes. The use of such a genome-wide collection of sequence variation is, however, not limited to association mapping. Key questions of evolutionary genetics and population genetics can be addressed using such a data set. These data will als reveal amino acid changes in proteins, loss of certain genes due to deletions or premature STOP codons (nonsense mutations) or changes in regulatory sequences that may affect gene expression [5].

Being given a genome-wide collection of sequence polymorphisms will facilitate future research in evolutionary biology. It might become possible to address key questions such as whether genetic changes are of adaptive nature, i.e. the result of natural selection, or whether they are evolutionarily neutral, i.e. purely random events. If the neutrality hypothesis can be rejected one aims at identifying specific regimes of selection: Is there evidence that mutations made adaptation to local environment possible and have such mutations been selected for? Or conversely, has selection conserved a certain region and have deleterious mutations been purified before they reached higher allele frequencies? Or has allelic diversity been maintained by balancing selection [5]?

On the basis of genome-wide data on sequence polymorphism future research will aim at inferring the population history and the geographic distribution of genomic diversity, including events such as recent extensions or population bottle necks, as well as geographic routes on which colonization of habitats occurred. New insights will be gained into the correlation of genetic variation with

[2] The image was originally published at http://www.dpw.wau.nl/natural/general/distribution.htm

other properties of the genome such as gene density, repeat density or recombination frequency [5], [23].

## 1.3 Whole-genome tiling arrays for resequencing

Before the resequencing technique is detailed, the term "sequence variation" should be rendered more precisely. The most abundant class of mutations are *s*ingle *n*ucleotide *p*olymorphisms, termed SNPs. Further, there are insertions and deletions of a few to many nucleotides in a specific ecotype, defined with respect to the reference sequence of Col-0 (together referred to as indels). A special class of indels are variations in the repeat number of a small sequence motif; such length polymorphisms are commonly seen at mini- and micro-satellite repeats. Not all of these polymorphisms can be detected with resequencing arrays which aim primarily to identify SNPs. The identification of large deletions is possible, while insertions and length polymorphisms are more difficult to detect.

Chip resequencing is a microarray technology that allows sequence variation to be detected in a massively-parallel and thus cost-effective way. Microarrays are based on the specificity with which fluorescently labelled, denatured nucleic acid sequences bind to a complementary nucleic acid probe which is immobilized on a glass surface. After washing away unspecifically bound sequences, the fluorescence intensity at this spot is measured and one can tell whether the labelled nucleic acids that were put on the chip contained a sequence which is complementary to this probe. By spotting many different probes with high density on an array, many sequences can be interrogated in parallel. Microarrays have been widely used to identify genes that are differentially expressed under certain conditions by hybridization of cellular RNA.

Among a variety of genomic analyses beyond gene expression profiling, DNA arrays can be used for polymorphism discovery and genotyping and even genome resequencing. For these tasks small probes are used, commonly 25 nucleotides in length. The density with which oligonucleotide probes are spotted on such arrays has made it possible to interrogate entire genomes with a single (or a few) hybridization experiment(s) [22].

For the detection of sequence variation, probes are designed complementary to a previously known reference genome sequence. The hybridization of labelled genomic DNA from closely related individuals or strains of the same species reveals polymorphisms relative to the reference where the hybridization intensity is significantly different from a control experiment in which labelled genomic DNA from the reference is hybridized to the array.

The design can be further improved for the discovery and genotyping of SNPs by additional probes that are complementary to polymorphic sequences. In order to interrogate all possible single nucleotide substitutions at a given site, four 25-mer probes are designed which are complementary to 12 bases upstream and downstream of that site but each having another of the four possible nucleotides at position 13. At evolutionarily conserved sites the strongest hybridization signal is expected to come from the probe that perfectly matches the reference genome. If the site is a SNP in the interrogated genome, there will still be a complementary probe and the hybridization intensity is expected to be strongest at this matching non-reference probe. In any case there will be one perfect match (PM) probe and 3 probes with a mismatch in the middle (MM probes) [10], [22].

With a similar approach probes can be designed for the detection of small insertion or deletion polymorphisms, shown in figure 1-2. To interrogate all possible deletions up to length $l$ starting at a specific site, $l$ deletion probes are needed. One needs exponentially many (more precisely $\sum_{i=1}^{l} 4^i$) probes to interrogate insertions up to length $l$ with perfectly complementary probes [10].

To be able to interrogate every single base in a genome, overlapping probes with single nucleotide overhangs are needed. The design of such whole-genome tiling arrays for resequencing is illustrated in figure 1-3. As both strands of the genome are interrogated for substitution of every possible nucleotide in the design discussed here (insertion or deletion probes are not used), $8N$ probes are required for a genome of length $N$ [22].

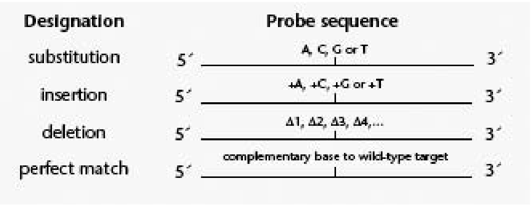

**Figure 1-2**
Probe design for variation detection arrays. Probes interrogating insertions longer than 1 bp would have to contain all possible inserted sequences, which makes this kind of variation detection practically infeasible. (Taken from [10].)

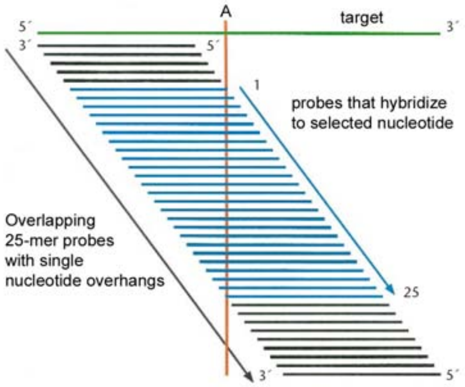

**Figure 1-3**
Probe design for tiling arrays with overlapping 25-mers with single nucleotide overhangs. Each nucleotide in the target DNA contributes to hybridization to 25 perfect match probes for one strand. (Taken from [10].)

For the resequencing of the *Arabidopsis* ecotypes the finished sequence of Col-0 was used for the array design, which only includes substitution probes. Resequencing the 125 Mb genome of *Arabidopsis* was done with approximately 1 billion probes distributed over 5 production wafers. Hence, 20 ecotypes could be resequenced with only 100 hybridization experiments. The actual hybridization experiments of 20 ecotypes including the reference Col-0 were conducted at Perlegen Sciences presumably with protocols similar to those described in [12].

The design of a similar Affymetrix variation detection array is shown in figure 1-4 (spot size and pixel masks might slightly differ from the ones used by Perlegen Sciences). For each of the positions shown, a clear difference in intensity between the PM and the MM probes is observed.

## Detecting SNPs with resequencing arrays

Elaborate approaches to SNP calling and genotyping from resequencing array data have been published [7], [27]. Using previously known SNPs, a third approach based on support vector machines has been developed in the course of this project. But instead of going into details, at this point we only sketch how SNP signatures can be detected in principle.

First of all, a fraction of the sites queried with a tiling array produces a signal too weak or too noisy to make a base call. With Affymetrix variation detection arrays the fraction of sites that could not be called was reported to be 20 % [7], but for the *Arabidopsis* chip resequencing this number is certainly higher. Depending on the ecotype, Perlegen's call rates for SNPs are estimated to be between 19 % and 26 %[3].

At sites where the hybridization intensity of the PM probe exceeds the intensities of the MM probes far enough to make a reliable base call, a SNP can be detected by a combination of two observations. The first signal is obviously a base call that deviates from the reference sequence. Thus one detects simultaneously gain-of-signal of a non-reference probe and loss-of-signal of the reference probe. However among such sites there will be a high number of false positives. To improve accuracy, one also integrates loss-of-signal of adjacent probes in the tiling and thereby exploit the fact that in the probes only the nucleotide at position 13 varies. Consequently, none of the probes around a SNP site in the tiling will hybridize without a mismatch. This is instantly understood with figure 1-3. As mismatches near the ends of a 25-mer duplex have a smaller influence on its stability, one only needs to consider 1 to 5 sites upstream and downstream of the putative SNP to get to a basic SNP calling algorithm that gives acceptable results.

[3] Using known SNPs in dideoxy sequencing data from Magnus Nordborg's lab, see chapter 2

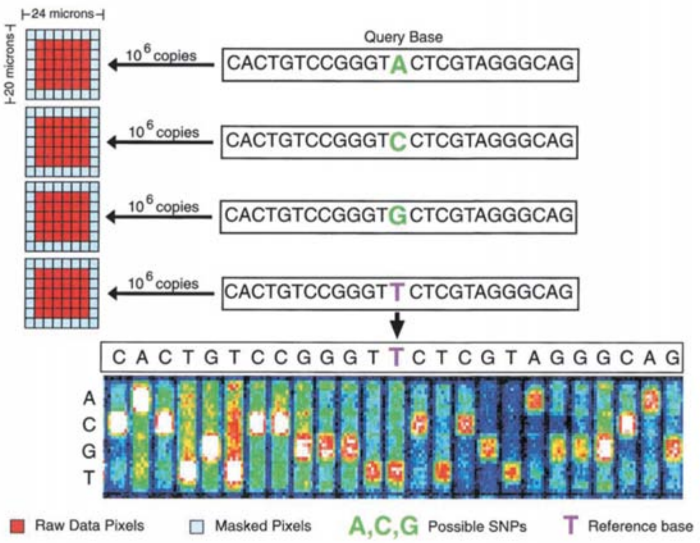
|-24 microns-|
|-20 microns-|
Query Base
10⁶ copies
CACTGTCCGGGTACTCGTAGGGCAG
10⁶ copies
CACTGTCCGGGTCCTCGTAGGGCAG
10⁶ copies
CACTGTCCGGGTGCTCGTAGGGCAG
10⁶ copies
CACTGTCCGGGTTCTCGTAGGGCAG
C A C T G T C C G G G T T C T C G T A G G G C A G
A
C
G
T
Raw Data Pixels
Masked Pixels
A,C,G Possible SNPs
T Reference base

**Figure 1-4** The design of an Affymetrix variation detection array. Each site is interrogated with four different probes per strand and each spot contains about $10^6$ copies of one probe. When the hybridization signal is detected from the chip by a scanner, the outermost pixels are masked. The lower panel shows hybridization intensities from 25 probe quartets which interrogate adjacent sites. (Taken from [7].)

## 1.4 Contributions of this diploma thesis

At this point the content of this diploma thesis and its contributions are summarized.

- The resequencing data are introduced in chapter 2. Some basic analyses show how properties of the probe sequence affect its hybridization characteristics. Similar analyses have been described. However, to our knowledge the dependency of intensity on sequence entropy has not been published before and the finding of GC content being a major determinant of hybridization properties appears to be specific to the resequencing of *Arabidopsis*. Some of these properties are incorporated in SNP calling with support vector machines (SVMs) that has been part of the resequencing project, but will not be explained in detail in this diploma thesis.

- A tool for the visualization of resequencing data together with related information is presented.

- An algorithm for a systematic, genome-wide $k$-mer analysis, described in chapter 3, resulted in an estimate of potential cross-hybridization for every single probe. To our knowledge such an analysis has not been described before in publications on resequencing.

- How the results of the $k$-mer analysis are applied to improve the quality of Perlegen's SNP calls is discussed in chapter 4 as well as their importance for the discovery of deletions. Additionally, they are an important input to SVMs for SNP calling that is currently ongoing.

- A novel method for the detection of large deletions using resequencing data is presented in chapter 5. As our knowledge of large deletions is incomplete, the specificity of the deletion calling algorithm could not be assessed directly, but indirect evaluations of its results are discussed.

In summary, the main contributions of this diploma thesis are a novel deletion calling algorithm and a genome-wide analysis of repeated $k$-mers which is an important prerequisite especially for deletion calling, but also useful either to filter given SNP calls for spurious calls caused by cross-hybridization or as an input for novel SNP calling methods with increased sensitivity and specificity.

# 2 The resequencing data

This chapter gives an overview over the data we received from Perlegen Sciences containing the results of the resequencing experiments. Basic analyses of some properties of these data are described as well as sources of data from other experiments to which the resequencing data are compared. The chapter is concluded with a summary of the database we used to store these data sets and a viewer developed to visualize the resequencing data.

## 2.1 The raw data

The raw hybridization data collected from the tiling arrays were delivered by Perlegen in files formatted like other sequence trace data. The format of these ZTR-files was introduced and described in [3]. They basically contain trace information compressed effectively to facilitate storing and transferring. They can be uncompressed to plain text files with routines contained in a library, io_lib, freely available with the Staden package[4].

From the uncompressed files, hybridization intensities and quality scores can be parsed easily. Only the mean hybridization intensity for every probe is specified, values are in the range between 1 and 4100. Not to be given a standard deviation might not be a big drawback—in [27] it is demonstrated that the standard deviation can be accurately estimated from the mean and thus does not add much information once the mean intensity is known.

Furthermore, for every probe quartet, the data files contain the base call which corresponds to the maximum intensity together with a quality score that is based on the estimated error probability of the base call. Such a quality score $q$ is obtained by a transformation of the error probability as

$$q = -10 \times \log_{10}(p)$$

where $p$ is the estimated error probability [8]. Hence, the higher $q$, the higher the accuracy, while the log transform leads to higher resolution for critical error probabilities close to zero.

These quality scores are calibrated to reflect real error estimates as closely as possible. The calibration method Perlegen Sciences uses (see [12], supplementary material, S7) is a variation of the quality score algorithm in Phred [8]. The values of $q$ are in the range between 3 and 32 (some values 5, 6, 27, 31 are not present which is due to the calibration technique.) This range corresponds to accuracies of a base call from a probe quartet between 50 % (quality score 3) and 99.94 % (quality score 32).

## 2.2 Some properties and simple analyses of the raw data

In the following some sequence properties are described that have an influence on the hybridization intensity of a probe. For these analyses the complete resequencing data from the reverse strand of chromosome 2 are used. The restriction to one of the five chromosomes makes the analyses feasible with a reasonable amount of memory. At the same time, biases compared to the genome-wide distributions are expected to be relatively small.

### Nucleotide content

The largest contribution to hybridization stability and specificity comes from the innermost nucleotides in the 25-mer duplex [20], [18]. The dependance of the intensity of the PM probe (which we also call “primary intensity”) on the nucleotide at position 13 are shown in figure 2-1. It is similar to a histogram plot, but it shows relative occurrences, i.e. occurrences are normalized with the total number of sites with the same nucleotide, to account for the extremely unequal nucleotide frequencies in the *Arabidopsis* genome—*Arabidopsis* has a low GC content, between 33.4 and 35.5 % depending on the chromosome [13]. Instead of a bar plot which is commonly used for histograms, a line is drawn in order to display all four histograms together. Since the number of sites with a primary intensity greater than 300 decreases rapidly the higher the primary intensity is, relative occurrences (y-values) have been log transformed.

[4] at http://staden.sourceforge.net/

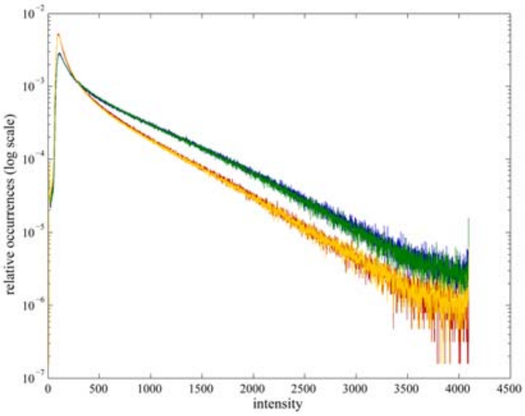

**Figure 2-1**
Histograms of primary intensity partitioned according to the nucleotide at position 13 in the probe. Red – A at position 13, blue – C, green – G, yellow – T. The x-axis shows intensity values, while y-values are log-transformed relative occurrences.

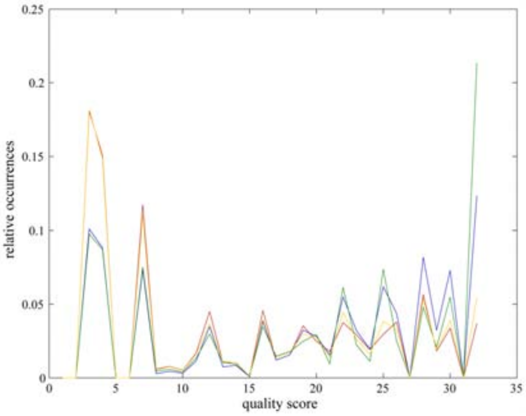

**Figure 2-2**
Histograms of quality score partitioned according to the nucleotide at position 13 in the probe. Red line – A at position 13 in the probe, blue – C, green – G, yellow – T. X-values correspond to quality scores and y-values to the relative number of sites with a given quality score and a given nucleotide at position 13 in the PM probe.

This histogram shows that hybridization intensities are in general higher if the middle nucleotide is a C or a G. This is not completely unexpected as G-C base pairs contribute more to duplex stability than A-T base pairs because of an additional hydrogen bond between G and C.

The histograms oscillate stronger in the high intensity domain, which is likely to be a stochastic effect as the number of sites (sample size) decreases. A saturation effect at an intensity value of 4100 can also be seen in these histograms. Note that close to the saturated intensities G/C probes are also more abundant than A/T probes.

A similar pattern is observed in the quality score histograms, shown in figure 2-2. The lines look ragged since some quality scores are missing or underrepresented. This is due to the calibration algorithm for quality scores (described in detail in [8], [12]). Nevertheless a trend similar to the one observed for intensities can be seen here. Low quality scores are more abundant in the class of A and T probes relative to G and C probes and the opposite is seen for high quality scores.

Taking effects of nucleotide composition one step further, we investigated the dependance on the nucleotide content of the whole 25-mer and its impact on hybridization characteristics. Figure 2-3 shows mean primary intensity and mean quality score as a function of the occurrences of a given nucleotide in the PM probe.

Again the graphs for C and G content look similar and complementary to those for A and T content. This observation is fundamentally different from those made in [7] where intensity is found to be negatively correlated with the number of purines (A and in particular G) in the 25-mer probe. A negative effect of G-rich probes causing problems in base-calling has also been reported [27]. This might be the reason for generally lower primary intensities for G content compared to that for C content, also the shape of the mean quality score graph for C content is slightly broader than the one for G content.

An interesting difference between the graphs for primary intensity and quality scores can be seen when the tails of the A/T histograms are compared. The increase in intensity with a high number of As or Ts is not reflected in quality scores which, instead, show an almost monotonic decrease towards high numbers of As or Ts. This suggests that high intensities of

AT-rich probes are most likely caused by unspecific cross-hybridization to a genomic background that is itself very AT-rich. This kind of cross-hybridization also effects the MM probes and thereby probably reduces the specificity of the PM probe within the probe quartet which in turn results in a lower quality score.

From the observed correlation between G and C nucleotide content opposed to A and T content it appeared natural to pool these nucleotides and conduct the same analysis for GC content.

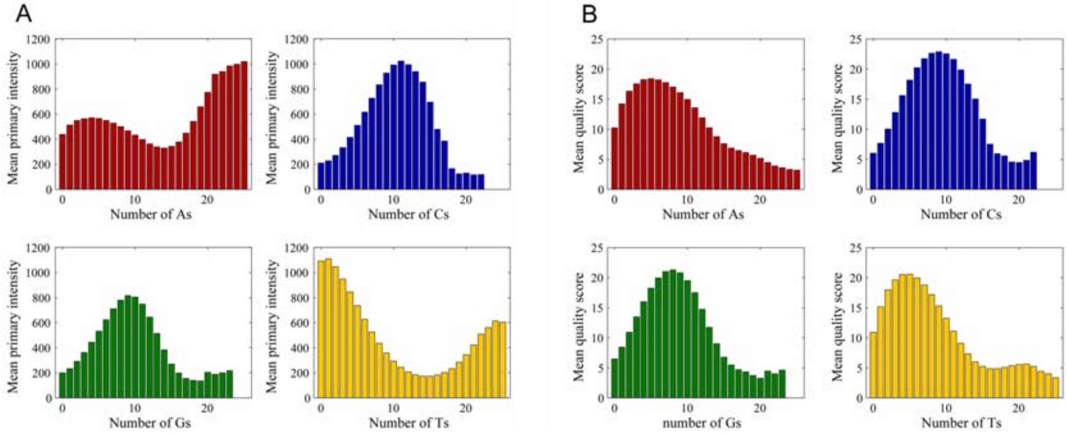

**Figure 2-3**
Mean primary intensity and mean quality score as a function of the nucleotide content of the probe. Red bars – number of As in the 25-mer probe, blue – Cs, green – Gs, yellow – Ts.
**(A)** Mean primary intensity dependance on nucleotide content.
**(B)** Mean quality score dependance on nucleotide content.

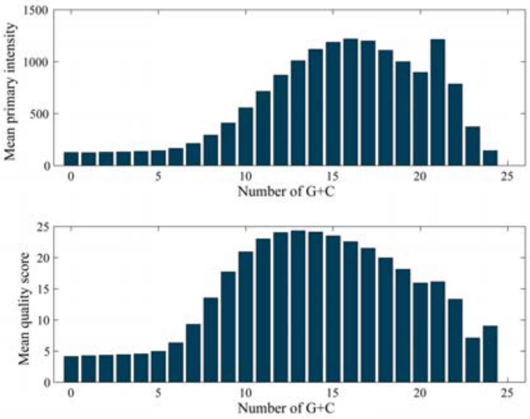

**Figure 2-4**
Mean primary intensity (upper panel) and mean quality score (lower panel) as a function of the GC content of the probe. This is the result of pooling the above dependencies for C and G content.

Primary intensity dependance on GC content is illustrated in figure 2-4. In comparison to the separate analysis for each base this plot reveals that from very AT-rich probes (with 5 or fewer Cs and Gs) the hybridization signal is generally weak. This effect is also reflected in low quality scores suggesting that most of these sites show a weak and noisy signal. Hybridization intensity appears to be strongest if approximately two thirds of the probe consist of Cs and Gs. As GC content increases, quality scores tend to decrease earlier than primary intensities. The reason for this might be decreasing specificity of GC-rich probes under the given hybridization and washing conditions. It might also reflect that very G-rich probes can form stable secondary structures hindering hybridization to complementary genomic DNA. The outliers on the right end could be a stochastic effect as the sample of probes with very high GC content is relatively small.

In conclusion, the effects of nucleotide composition on hybridization properties are pronounced. The strong influence of GC content that has not been reported for previous experiments is likely due to the very AT-rich genome of *Arabidopsis* so that extremely unequal nucleotide frequencies might have a bigger impact than observed previously for genomes with less biased nucleotide composition.

One should keep these nucleotide dependencies in mind when analyzing SNP calls from resequencing data. Systematic biases towards certain nucleotide substitutions among SNP calls from these resequencing data should not be unexpected, especially because of the fact that the middle nucleotide by itself already introduces a strong bias in hybridization intensity and quality score.

### Sequence entropy

An investigation of probes with low hybridization specificity (in a experiment of comparably smaller scale) was done in [15] with the result that regions were found to be overrepresented in which runs of single nucleotides occurred, sometimes interleaved with a second nucleotide, for instance TTTTCCTTTT or ACCCCAAAAA. Often these homopolymers do not have a simple repeat structure such as $(AT)_n$, but can be well described with a measure of sequence complexity such as sequence entropy. Sequence entropy is higher the more heterogenous nucleotide composition is, hence sequence tracts consisting of a single nucleotide have low sequence entropy (0 indeed). Entropy and other measures of sequence complexity have been described in [24]. The entropy $E$ of a sequence over an alphabet of size $K$ (for DNA $K = 4$) in a window of length $w$ is defined as

$$E = \sum_{i=1}^{K} \frac{n_i}{w} \log_K \left(\frac{n_i}{w}\right)$$

where $n_i$ is the number of occurrences of symbol $i$ (or nucleotide in case of DNA) in the window. In this context the length of the probe is chosen as window length, $w = 25$.

How mean primary intensity and mean quality score depend on sequence entropy is depicted in figure 2-5. The error bars indicate 1 standard deviation. Quality scores in the lower panel exhibit a clear dependance on sequence entropy. For probes whose entropy is close to 1 a significantly higher mean quality score is observed.

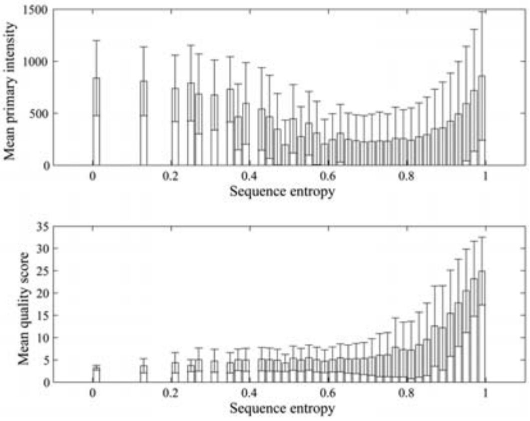

**Figure 2-5**
Mean primary intensity (upper panel) and mean quality score (lower panel) as a function of the sequence entropy of the probe. Error bars indicate 1 standard deviation.

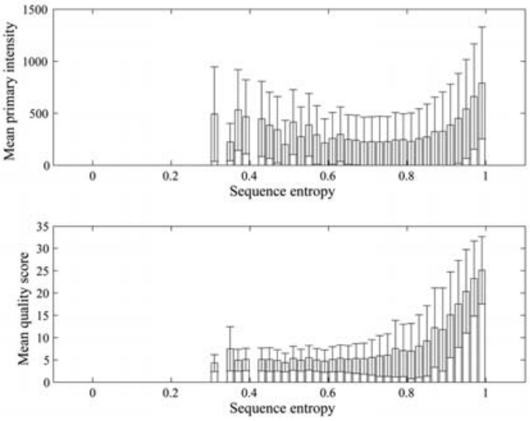

**Figure 2-6**
Mean primary intensity (upper panel) and mean quality score (lower panel) as a function of the sequence entropy of the probe. Repetitive sites have been excluded. Error bars indicate 1 standard deviation.

The dependance of primary intensity on sequence entropy is less obvious than that of quality scores. One could hypothesize that this is due to a strong correlation between low entropy sequences and repeats, since low entropy indicates the abundance of a single nucleotide which is A or T in most cases because of low genomic GC content. Indeed, if the same plot is made after repetitive 25-mers have been filtered out[5], it looks different (see figure 2-6). The tail for low entropies disappears as it does not pass the filter and the dependance of primary intensities can be explained more parsimoniously.

### Simple sequence repeats

We also investigated the extent to which simple sequence repeats (SSR), also known as microsatellites, have an effect on hybridization. Here we are not concerned with the problem of cross-hybridization to DNA from other genome locations. This problem has to be addressed with a systematic, genome-wide $k$-mer analysis, wich is explained in chapter 3. We are only interested in local effects of repeat structures like $(AT)_n$. Such simple sequence repeats are common throughout the genome, thus indeed many of them coincide with 25-mers that are not unique in the genome sequence. Nevertheless, these SSRs often contain point mutations or differ in the number of tandem units (i.e. they differ in $n$) which can result in unique 25-mers. It is known that polymerases have high error rates at SSRs due to a process called replication slippage which causes length polymorphisms. As the amplification of genomic DNA for the hybridization experiments is done by polymerases, the question is whether SSRs have a noticeable effect on hybridization intensities and quality scores. Because of the high level of polymorphism in SSRs, a result based on the data for Col-0 may not be generally applicable to other ecotypes.

To this end SPUTNIK[6], a widely used program which detects SSRs with a unit length up to 5, was slightly modified. SSRs are detected by first searching for perfect tandem repeats with 2 copies and unit length 2-5. Upon detection an extension in both directions is triggered. The extension is done with dynamic programming to align repeat units allowing for substitutions and indels. The location and score of each SSR is reported.

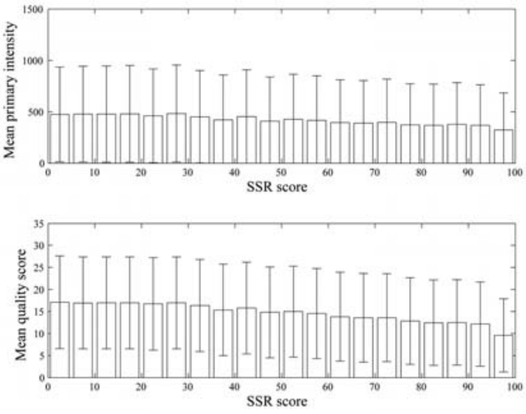

**Figure 2-7**
Mean primary intensity (upper panel) and mean quality score (lower panel) as a function of repeat scores. Error bars indicate 1 standard deviation. For the rightmost bar the mean was taken over all probes with a repeat score greater or equal to 95.

In order to make the graph shown in figure 2-7, at each site the maximum repeat score was taken (for simple sequence repeats with a low degree of conservation, there are sometimes alternatives to assign unit length and unit number which results in different scores). These maximum values were summed over all 25 positions of the probe and plotted against mean primary intensities and mean quality scores, respectively.

As we are not interested in the effect of cross-hybridization which is of course a problem at many sites with simple sequence repeats, all probes that are not unique in the whole genome have been excluded from this analysis[7]. There is only a loose negative correlation between probes with high repeat scores and primary intensities as well as quality scores, suggesting that SSRs only have a minor effect on hybridization properties.

### Self-complementarity

In [15] it is also discussed that self-complementarity of the probe could affect its ability to hybridize to the interrogated genomic DNA. By self-complementarity we mean the

[5] all positions where there is an exact, inexact or short $k$-mer match, see chapter 3 for details.

[6] The source code is freely available, see http://espressosoftware.com/pages/sputnik.jsp

[7] More specifically, all positions where a $k$-mer match was found have been filtered out; for details see chapter 3.

tendency of an oligo to form a hairpin structure. Theoretically, one could compute the Gibbs free energy for the secondary structure of every 25-mer on the array, and for some probes this is shown in [15], but to my knowledge programs like Mfold [28] are not capable of processing literally hundreds of millions of DNA oligonucleotides. Such computations are hardly possible using algorithms with a complexity of $O(n^2)$ or higher with current computing power.

Instead of accurate free energy calculations, simple approximations are used. One alternative is to simply count the number of complementary bases in a possible hairpin structure without bulges. We used the maximum number of consecutive base-pairs in such a gapless self-alignment as a measure of self-complementarity. The hairpin turn is required to be at least 3 nucleotides long. (The same heuristic is also part of the algorithm for the prediction of hybridization intensity in [20]. A similar hairpin score is used in [27] where additionally a single mismatch is allowed.)

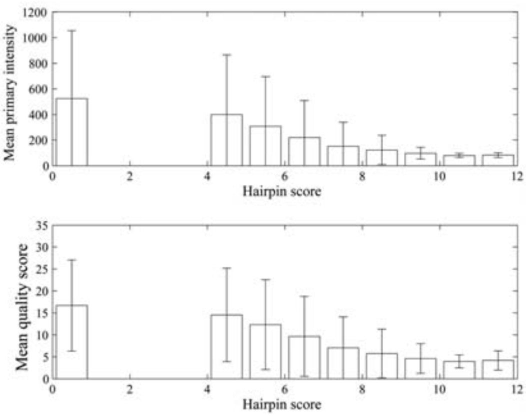

**Figure 2-8**
Mean primary intensity (upper panel) and mean quality score (lower panel) as a function of hairpin score. Error bars indicate 1 standard deviation. Scores below 4 have been set to 0, therefore the missing bars.

Mean primary intensity as well as mean quality score as functions of the hairpin score are shown in figure 2-8. Hairpin scores below 4 have been set to 0, the maximum score in a 25-mer for the perfect hairpin with a loop of 3 bases and 11 base pairs in the stem is obviously 11. The graphs suggest that hairpin formation indeed occurs on the arrays as probes with high hairpin score exhibit reduced intensities and quality scores consistently and with small deviation (error bars).

## 2.3 Intensity variability

In the previous section it was analyzed how sequence properties of the probe affect its hybridization intensity. It has been shown that from the probe sequence and its properties a great deal of the intensity variance can be explained. Consequently, it has been attempted to predict the intensity from the probe sequence (provided that it specifically hybridizes to only one region of labelled genomic DNA) and these predictions can be very accurate, even if the set of training data is small [11], [20], [27].

However, it is known that generally for microarray experiments it can be difficult to reproduce the absolute scale of intensities as subtle differences in experimental conditions can have drastic effects on hybridization intensity. An example is shown in figure 2-11. In order to correct for this kind of variance between experiments, usually technical replicates are made which then allow to normalize intensities between replicates. Technical replicates also allow to correct for spatial artifacts like smudges on a particular array (see [4] for an example). But because of the sheer size of the resequencing tiling arrays (approximately 1 billion probes for each of 20 ecotypes) and the cost of these array experiments, no technical replicates were made, which would allow to apply a standard normalization technique.

In the following three kinds of variability are discussed. First, variability between corresponding sites on identical arrays to which genomic DNA of different ecotypes has been hybridized. This kind of variability is also called “inter-sample variability”. Second, different sites on the array, but within the same ecotype are compared to obtain a measure of intra-sample variability. Third, as there is not only one, but five wafers for each ecotype, each of these wafers containing many chips, one can compare the intensities between different wafers and chips.

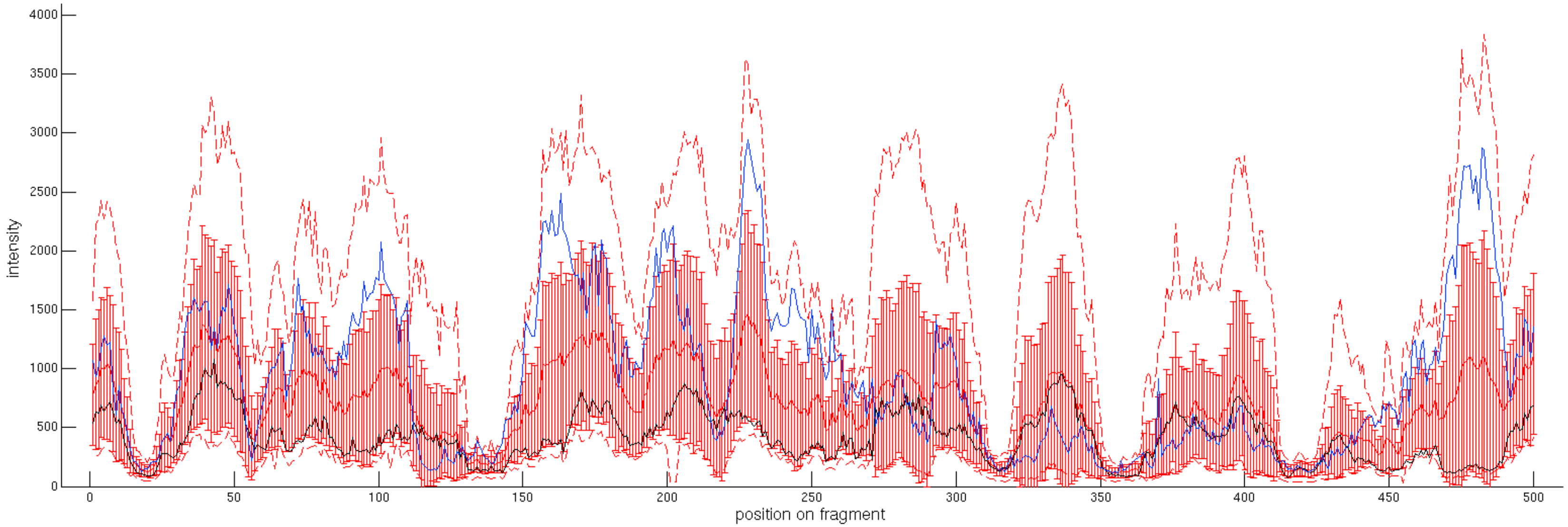

**Figure 2-9**
Intensity variability between ecotypes on a conserved fragment of 500 bp that does not contain any SNPs or indel polymorphisms (all probes on the same chip). Average primary intensity over all ecotypes (except for Van-0, see section 2-4) is drawn as red line, one standard deviation is indicated by error bars. Minimum and maximum intensity are shown as dashed lines. For comparisons the primary intensities of two ecotypes, Col-0 in black and Ler-1 in blue, are also included.
The primary intensities of Ler-1 are at most sites higher than those of Col-0, but not everywhere, which indicates that intensities cannot just be globally scaled to reduce this kind of variability. This plot not only shows high variability between ecotypes, but also between different sites in horizontal direction. The pattern of valleys with low primary intensity between mountains with high primary intensity appears to be consistent between ecotypes as the minimum and maximum lines are. Particularly the deep valleys are common to all ecotypes and variance is comparably low there.

### Intensity variability between ecotypes

Comparisons of intensity values of different ecotypes are only meaningful if for every ecotype the interrogated genomic DNA perfectly matches the PM probe. Therefore, comparing intensities over a longer range of contiguous sites is only possible for genomic regions which are perfectly conserved between ecotypes. A region that is known to be identical from dideoxy sequencing[8] was used to create figure 2-9.

Two kinds of variability are observed. In horizontal direction primary intensity shapes a pattern of mountains and valleys. Such patterns have been observed before (e.g. in [20]) and can be explained by different probe sequences having very different hybridization properties. Particularly probes in the deep valleys consistently produce low hybridization signal. There the signal can be too weak to reliably call bases and consequently SNPs or deletions are hard to detect at these sites.

However, different probe characteristics cannot account for the observed variability of primary intensities in vertical direction, i.e. across ecotypes. When comparing the primary intensities of two ecotypes (such as Col-0 and Ler-1 in the figure), it is not even the case that one of them is consistently higher than the other. This kind of variability clearly limits the accuracy of any prediction of hybridization intensities based on the probe sequence, as five-fold difference and more is commonly observed between different ecotypes.

### Intensity variability of identical probes

To show that the variability of hybridization intensities of the same probe is not an ecotype-specific effect, the intensities for one ecotype from probes with the same sequence that occur multiple times in the *Arabidopsis* genome are compared. Figure 2-10 show three examples, each of which is the histogram of primary intensities of 30 identical probes. Such a probe set is perfectly complementary to 30 genomic sites, still generally the hybridization does not appear to be saturated. Instead of a usual bar plot, histograms are drawn as lines facilitating the depiction of overlapping ones.

The three histograms shown suggest that the histogram width, which reflects the degree of variability, depends on the mean intensity. One might find the width broad considering that these are perfect replicates. The difference between the histograms of forward and reverse strand is large only in one case (shown in green), which suggests that the variability between probe quartets of the same site might depend on mean primary intensity as well.

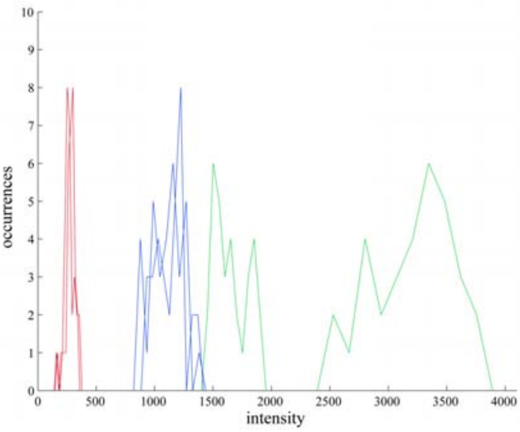

**Figure 2-10**
Intensity variability at sites with identical probes. Histograms of primary intensity (as line plots) for three sets (each in its own color) of identical probes are shown for the forward and reverse strand. Each probe set consists of 30 identical probes with 30 genomic target sites.

### Intensity variability across wafers and chips

A drastic example of intensity variability across wafers and in a somewhat smaller scale also across chips on the same wafer can be seen in a genome-wide plot of average intensity in Bur-0 shown in figure 2-11. Intensities have been smoothed with a moving average under a window of length 15 000 bp.

The finding that intensities across wafers can vary drastically can in principle explain why the intensity variance of identical probes for the same ecotype spotted on different wafers is high. Intensity variability across wafers can also account for high variance of intensities between ecotypes, as each ecotype is hybridized to the wafers in an independent experiment.

[8] see section 2.4 on known polymorphisms

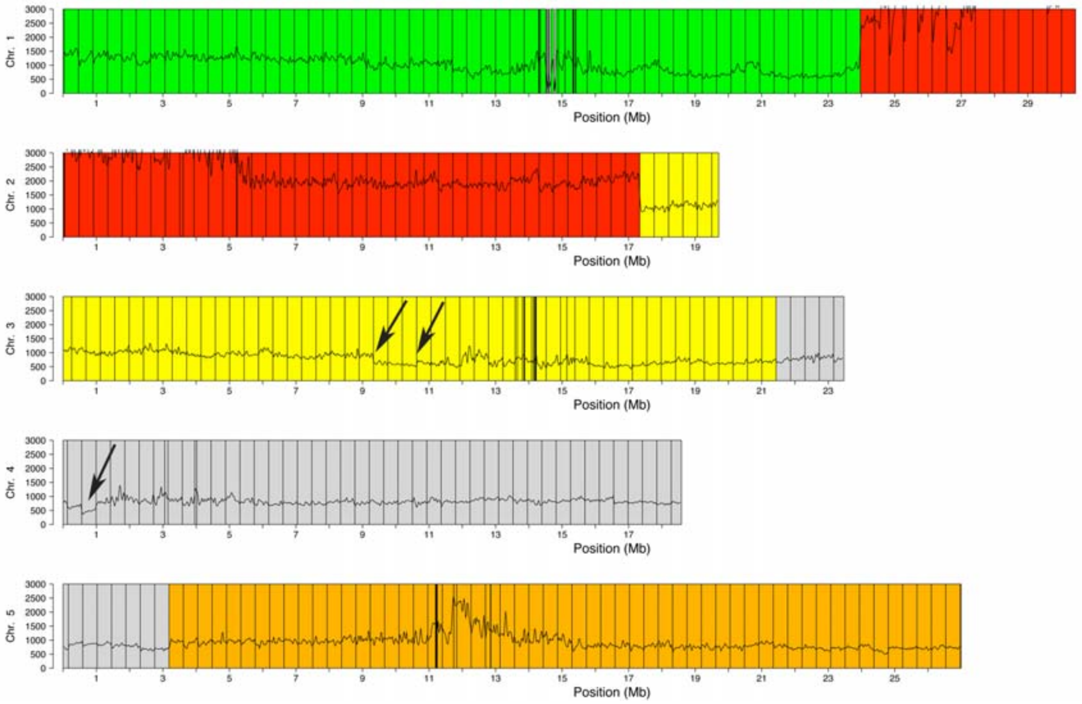

**Figure 2-11**
Genome-wide intensity variability across wafers and chips for Bur-0. Mean intensities were averaged using a window of length 15 000 bp. The five wafers are shaded in different colors, chip boundaries are indicated by vertical lines. Note the discontinuities between some wafers, especially at the boundaries of the red one. Some instances of more subtle discontinuities between chips on the same wafer are marked by arrows. (This plot was made with an R script by Richard Clark.)

## 2.4 Known polymorphisms as control and training data

The resequencing of 20 *Arabidopsis* ecotypes has not been the first effort to discover genetic variation between these ecotypes. Magnus Nordborg's group has collected dideoxy sequences for 96 individuals of *Arabidopsis* including the 20 ecotypes that were resequenced [23]. These sequenced fragments are designed such that they cover the genome relatively uniformly. Between 586 164 and 629 715 bp have been sequenced per ecotype in the release used here[9] which comprises approximately 0.5 % of the whole genome (654 491 bases are reported for Col-0). The number of SNPs discovered in these fragments varies between 2243 and 3904 per ecotype, and the number of indel polymorphisms between 326 and 617 per ecotype.

Unfortunately, the Van-0 accession used for sequencing appeared to be problematic and therefore this ecotype was excluded from the analyses published in [23]. For this reason we usually exclude Van-0 as when we compare the resequencing data to the sequenced fragments. (However, there is no indication that the Van-0 resequencing did not work properly, only comparisons are not meaningful.)

These data have proven very useful for comparisons to the resequencing data. Because PCR-based dideoxy sequencing very reliably uncovers SNPs and small indels, i.e. virtually without false positives or false negatives, it can be used to evaluate the performance of SNP calling methods. For the purpose of assessing the false-positive rate of SNP calling it is crucial to have a data set for comparison which has very low error-rates and in this regard the dideoxy sequences are ideal.

Many other comparative analyses are pos-

[9] see website at http://walnut.usc.edu/2010/

sible having the sequenced fragments, for example estimation of SNP density and the average distance between SNPs, which is important because the resolution at which SNPs can be called from the chip resequencing data is limited. The very low error rate in the dideoxy sequences further allows to estimate biases in SNP calls from the resequencing arrays, for instance a bias towards coding regions of the genome or the dependance of call rates on the SNP base or the nucleotide content of the $k$-mer around it.

Moreover, the fact that a considerable fraction of polymorphisms was known facilitated the development of new methods to detect polymorphisms. Support vector machines for SNP calling, which belong to the field of supervised learning algorithms, were developed successfully only because of the existence of a training and test data set with known labels[10].

The indel polymorphisms found in these sequenced fragments also allowed to test the basic idea of a deletion calling algorithm in regions where a few deletions are known. An indirect evaluation of deletion calling results is also possible with these fragments; details are described in chapter 5.

Another large data set of dideoxy sequences was useful for comparisons to the resequencing data. It consists of a collection of reads from shotgun sequencing of the genome of Ler, an ecotype which has been used in a large number of molecular genetic studies. This sequencing project has been undertaken by Cereon Genomics, a subsidiary of Monsanto. The reads resulted in contigs which covered approximately 70 % of the Col-0 genome on which they were assembled. From this assembly a list of sequence polymorphisms was derived containing more than 700 large indels (at leats 100 bp long) in Ler relative to Col-0. These large deletions allowed evaluation of the performance of the deletion calling method we developed.

## 2.5 A database for the resequencing project

The sheer mass of resequencing data as well as the need to compare data from different sources made it necessary to store all this information in a database and organize it in a way that facilitates relating data from different sources. This database was realized in MySQL. The table design was mainly created by Stephan Ossowski, Richard Clark, and myself.

### Storing the resequencing data

The information contained in the ZTR files was parsed and inserted into 20 tables, one for each ecotype. An entry in such a table corresponds to a single site in the genome. As such it contains an id, which can be thought of as a genomic address with the function to allow efficient combination (joins) of information across tables. An entry further comprises information about chromosome and position, 4 intensity peaks for the probe quartet which queried the forward strand and another 4 intensities for the reverse strand (A_A_peak, A_C_peak,... to Z_G_peak, Z_T_peak), and two quality scores (A_qscore and Z_qscore), one for each probe quartet and two characters indicating the call from the respective probe quartet (A_call and Z_call). The minimum space requirement for a single row is 29 Bytes $(4+1+4+(4\times 2\times 2)+(2\times 1)+(2\times 1)$ Bytes).

As there are approximately 120 million sites per genome for each of 20 ecotypes, a minimalist, ecotype-specific table needs more than 3.2 GB of hard disk memory. Together with an index which makes query speed in such huge tables acceptable, one needs over 5 GB for each ecotype. Together, all ecotype tables with minimum information account for more than 100 GB of disk space.

Our tables contain also reference information, as well as additional indices for improved look-up speed of chromosome and position. The final table size is approximately 10 GB of disk space.

### Storing reference data

An additional table contains the reference sequence, one base per row. However, as comparisons between the reference sequence and resequencing data are very frequently made, the reference base is also added to each entry in the ecotype tables in order to avoid trivial, but time-consuming joins over these huge tables.

Information on the annotation of the genome with gene structures is stored in further tables

[10] Sites where it is known whether or not there is a SNP and between which nucleotides.

(gene annotation parsed by Stephan Ossowski). This is needed to determine whether a SNP causes a change in the amino acid sequence of a protein. There are other, similar questions that can only be answered with genome annotation at hand.

Another set of tables contains information on whether the 25-mer probe at a given site is unique in the genome or whether it occurs multiple times. This allows us to filter the genome for repetitive regions which potentially cause cross-hybridization and consequently interfere with analyses. How this kind of information is computed and stored in the database is described in chapters 3 and 4.

### Other data sets

The most important other source of information that was used for comparison to the resequencing data is the collection of sequenced fragments from Magnus Nordborg's lab. This data was also stored in MySQL tables (the latest releases is contained in the tables mn_bases_v2, mn_snps_v2, mn_insertions_v2 and mn_deletions_v2) which makes it possible to link this collection of known polymorphisms to the resequencing data for various kinds of comparisons that can be conveniently formulated using SQL[11] queries.

In summary, the MySQL database allows us to store and access the huge amount of resequencing data as well as to integrate other data sources. The data can be directly accessed by the user through SQL queries, but more importantly, other programs can query and modify the data through interfaces to the database. Over the course of this project such programs have been written in Perl, Python, Java, Matlab and R.

One of these programs is a viewer that visualizes the data sets just described and additionally evaluates properties of the interrogated DNA sequence that affect hybridization. It was written with the purpose of facilitating visual inspections of genomic regions of interest.

## 2.6 ReseqView – a viewer for resequencing data

In order to develop methods that extract interesting information from the resequencing data such as SNPs or deletions, it is helpful to visualize and inspect the data to get an intuitive assessment of how SNP or deletion features appear.

Chromatograms (or trace files), the output of dideoxy sequencers, can be visualized using viewers such as Trev [2], the trace editor and viewer that is part of the Staden package. As the resequencing data was originally saved in ZTR files, developed for such chromatograms, Trev can be used to display intensities and quality scores from the resequencing arrays.

Using Trev for our project was not very convenient, however, for several reasons. There are several hundred ZTR files per ecotype and if one wants to look at a certain region of the genome, it can be cumbersome to find the right file. This data is stored and organized in the database, but for visualization with Trev it would have to be written to files first. As Trev was not designed to visualize other kinds of data, comparisons between the resequencing data and for example sequenced fragments from the Nordborg lab is difficult using only this program.

Consequently I developed a tool, ReseqView, to visualize not only the resequencing data, but also sequenced fragments and deletions from the Ler sequencing project. Additionally, the reference sequence and some of its properties, most importantly repetitive 25-mers, can be displayed in alignment with intensities and quality scores. As all these data are stored in the MySQL database, ReseqView is understood as a graphical front-end. The direct interaction with the database allows users to conveniently load and visualize any genomic region from any ecotype in just a few seconds.

The graphical user interface is realized with the Swing libraries of Java 1.5. It consists of two components, a display window and a command editor. Apart from scrollbars the display window is completely passive without any buttons or menus for user interaction. Instead, the program is controlled by commands that are entered into the command editor which behaves similar to a Unix shell. The decision to use text commands instead of buttons and menus was motivated by the possibility to easily realize a batch mode in which text com-

[11] Structured Query Language, the standard language for interaction with databases

mands are read from a file and processed exactly as if they had been entered into the command editor. That allows to visualize many different regions with the same display settings without having to adjust buttons or rulers each time a new region is loaded. Instead, one just changes the load command inserting new genome coordinates in a preexisting batch file, or alternatively types the load command into the command editor and inserts the commands for display configuration into the command editor via cut-and-paste. As the syntax of the commands is trivial, it is very easy to write scripts that automatically generate batch files for several ecotypes or genomic regions of interest, for example if one wants to inspect multiple traces at a given genetic locus.

A command to create image files with screen dumps is also provided and a couple of such images are contained in this thesis. (Such images can also be created in the batch mode even without opening the display window). The key display features are illustrated by several screen dumps shown in figure 2-12 and 2-13. Most of them show only a few features. It is possible, though, to combine all of them or any subset in a customized display.

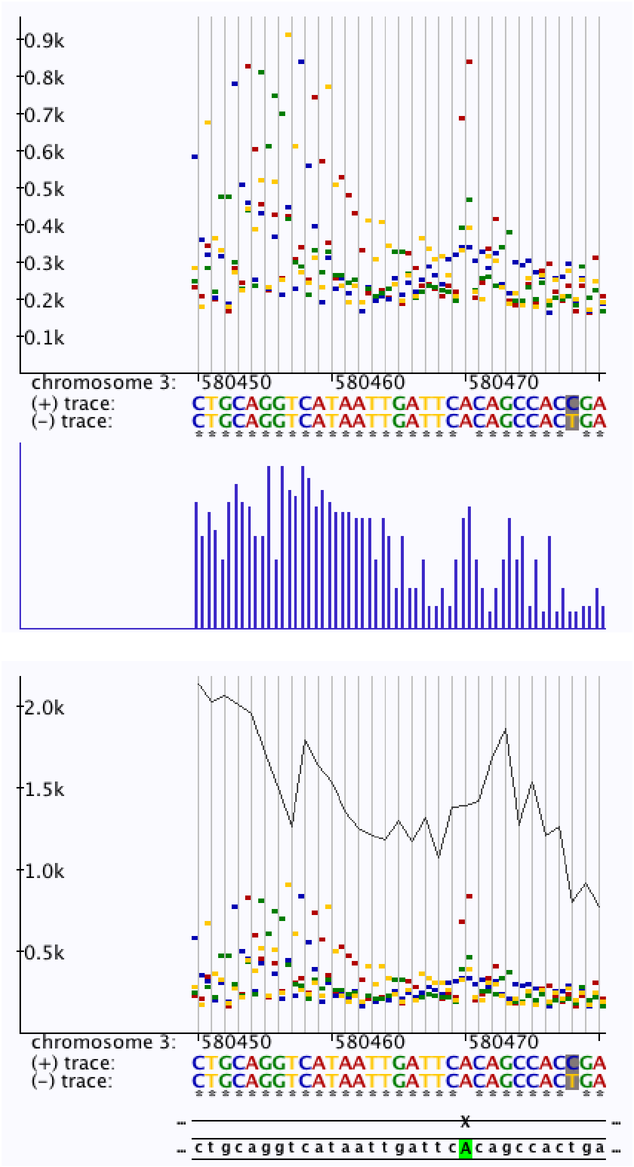
0.9k
0.8k
0.7k
0.6k
0.5k
0.4k
0.3k
0.2k
0.1k
chromosome 3:
580450
580460
580470
(+) trace:
(−) trace:
CTGCAGGTCATAATTGATTCACAGCCACCGA
CTGCAGGTCATAATTGATTCACAGCCACTGA
2.0k
1.5k
1.0k
0.5k
chromosome 3:
580450
580460
580470
(+) trace:
(−) trace:
CTGCAGGTCATAATTGATTCACAGCCACCGA
CTGCAGGTCATAATTGATTCACAGCCACTGA
x
ctgcaggtcataattgattcAcagccactga

**Figure 2-12**
**(A)** A basic visualization of trace data in Lov-5. Intensity values are displayed as small colored squares. Values for the forward strand are drawn left of the grey vertical lines, those for the reverse strand right of it. The base calls corresponding to the maximum peaks are written below the intensity values in corresponding colors. If calls between strands disagree, they are shaded in grey. Asterisks indicate sites where both calls match the reference. In the lower panel quality scores for forward and reverse strand are shown as blue bars.

**(B)** A SNP in Lov-5 confirmed by dideoxy sequencing. In this screen dump the same region as in (A) is shown, but part of a sequenced fragment is displayed instead of quality scores (framed lower-case letters). The SNP in the dideoxy sequence is shaded in light-green. The intensity from both probe quartets is also maximum for the A probe (red squares). A depression of intensity values can be seen a few bases upstream and downstream of the SNP. For comparisons to the reference, Col-0, its maximum intensities are shown as a black line in the upper panel.

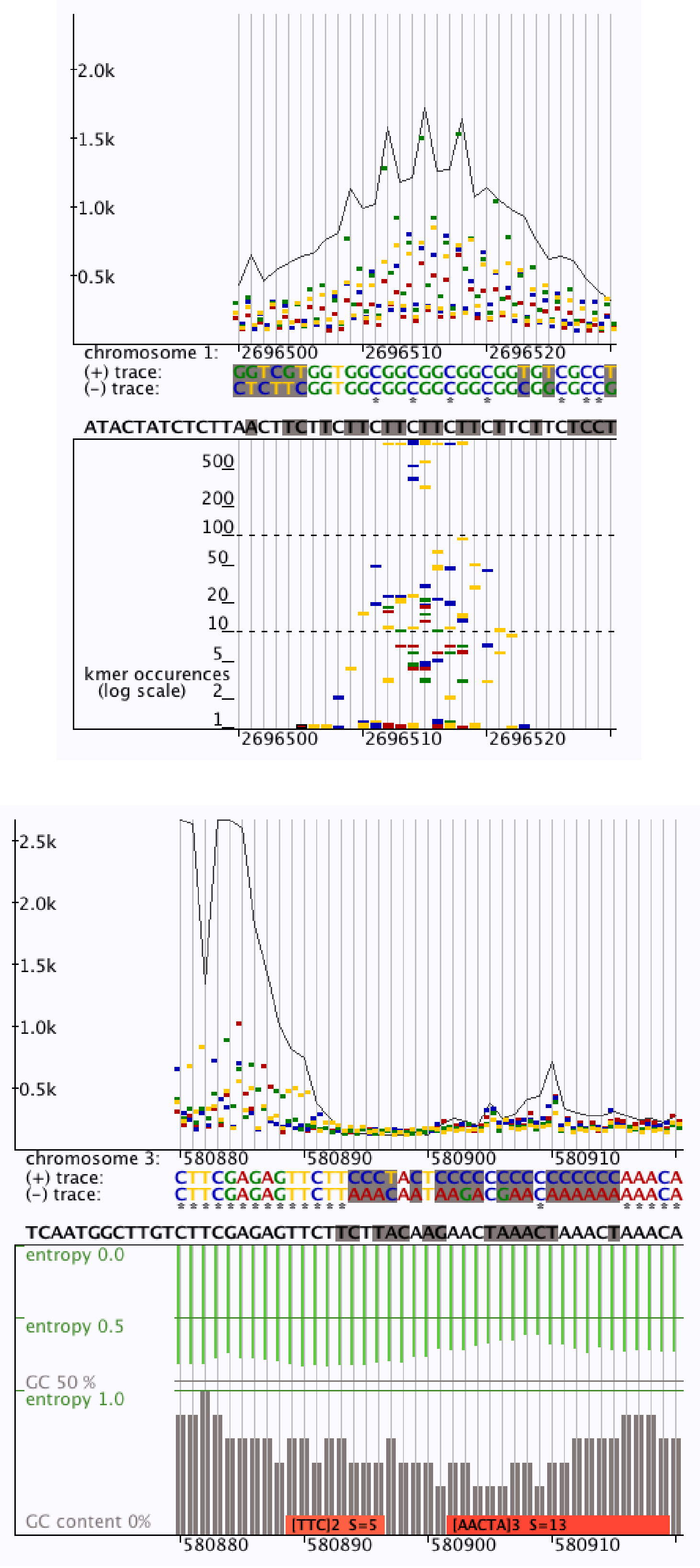

**Figure 2-13**
**(A)** A visualization of a repetitive region in Bur-0. Below the calls, the reference sequence is shown in black letters. Those shaded in grey indicate positions where calls in Col-0 deviate from the reference. Colored bars below the reference sequence denote how often (in a logarithmic scale) a region occurs in the genome that is complementary to a probe querying this site. The colors of these bars indicate to which probe they bind, for example blue bars lead to higher hybridzation signal of the probes for C calls. the thickness of the bars corresponds to different match types, similar to different degrees of conservation between the match partners. Details are explained in chapter 3.

**(B)** A visualization of a simple sequence repeat in Lov-5. Several properties of the reference sequence are plotted. Simple sequence repeats are drawn as horizontal bars in redish colors, the darker, the higher the repeat score (which is also written on the bar). The height of the grey bars in the lower panel corresponds to the GC content of the probes at that site. Sequence entropy is depicted by green bars drawn inversely from the top of the panel.
A possible explanation for very low intensities in the right half of this display is very low GC content in addition to the simple sequence repeat. Note that not only in Lov-5 but also in Col-0 maximum intensity calls deviate from the reference sequence (letters shaded in grey).

# 3 K-mer analysis of the *Arabidopsis* genome

This chapter addresses the problem of non-unique genomic sequences which potentially interfere with genome-wide resequencing through cross-hybridization.

Although the tiling array experiments are designed to detect sequence polymorphisms such as single base substitutions, this is not a technique to directly compare corresponding genome sites. In other words, no matter from which genome location a 25-mer sequence comes, it can hybridize to any spot on the chip on which complementary oligos are attached.

Given that a 25-mer occurs somewhere else in the genome, but with a different nucleotide in the middle, in theory that can have the same effect as if the same oligomer was hit by a point mutation that changed its middle base. In both cases labelled genomic DNA will be detected at a spot that indicates a non-reference nucleotide. Thus, if a 25-mer is not unique throughout the genome, it has the potential to interfere with SNP detection.[12]

If one wants to detect deletions, regions that display only a weak hybridization signal will be considered candidates. However, if the deleted sequence is not unique, genomic DNA from another locus will produce a similar hybridization signal as if there was no deletion at all.

To circumvent problems like the ones described here, caused by $k$-mers that are not unique, one needs to have knowledge of all $k$-mers that occur multiple times in the genome.

If genomic DNA was generated by a random process, the chance to observe a certain $k$-mer sequence would be $0.25^k$ (in an overly simplified model assuming equal nucleotide frequencies). For $k = 25$ that is about $10^{-15}$. Even in a random genome of 120 million base pairs (like the one of *Arabidopsis*) repeated $k$-mers would be expected to be rare events, not likely to have a major effect on the detection of SNPs or deletions.

However, after several whole genome sequencing projects have been finished, it is known that genomes are highly repetitive and that there are many different processes that generate a wide distribution of repeat sizes. The degree of conservation or decay of duplicated sequences also varies, primarily depending on the time that elapsed since the duplication event.

It is a widely accepted explanation for the genome size of higher organisms that large scale duplications have occurred frequently during genome evolution. This is even more pronounced in the plant kingdom where many genomes—including that of *Arabidopsis*—have been shaped by genome duplication and polyploidy [13]. This further emphasizes the importance of being able to correct for multiply occurring $k$-mers when analyzing the *Arabidopsis* resequencing data.

## 3.1 Repeated k-mers

The $k$-mer analysis[13] aims at identifying $k$-mers that have the potential to cross-hybridize to probes on the tiling-array that correspond to other genomic positions. From that it follows that not only exact duplicates of a $k$-mer sequence are to be found, but also $k$-mers that mismatch in the middle. As there are probes for all four middle nucleotides on the chip, these will also form a perfect duplex. It is also immediately evident that, in addition to the forward genome strand, also the reverse strand has to be considered, as DNA from both strands is hybridized to the resequencing arrays.

Although the resequencing chip technique is based on the ability to detect single-base mismatches, 25-mers are long enough to form more or less stable duplexes even if they do not match perfectly. This has been known from PCR primers and there are actually laboratory techniques such as site-directed mutagenesis, which take advantage of the possibility to use a primer with a mismatch to its binding site. The melting temperature $T_m$ of imperfectly matched oligonucleotide duplexes

[12] The important difference between these two cases is that the original reference oligomer is still present in the first one, but not in the second. That means that a true SNP will also have an effect on the intensities around the polymorphic site that a repeated 25-mer will not have.

[13] In the following we will use the term $k$-mer almost simultaneously to 25-mer to indicate that this analysis is not restricted to 25-mers in principle, even though in practice we used $k = 25$ as this is the length of the probes on the arrays. In any case we assume $k$ to be odd.

have been studied experimentally and the results have led to *in silico* prediction methods. Formulas to accurately calculate $\triangle T_m$ can be quite complex, particularly so if several mismatches are considered. Hence we decided to rather identify matches that are candidates for binding to a probe with a relatively small difference in $\triangle T_m$ compared to a perfectly matched $k$-mer.

Figure 3-1 shows the dependency of $\triangle T_m$ (the change in oligo-duplex stability) on mismatch position within a 20-mer. This clearly illustrates that mismatches at the end of an oligo cannot be treated the same way as internal mismatches. Note the large variance, which is mostly due to dependency on the mismatching two nucleotides. We decided, however, not to model this nucleotide dependance, as there is no simple pattern that could be easily incorporated into a matching algorithm. The dependance on mismatch position is similar also for two mismatches, which can be seen in figure 3-2 where $\triangle T_m$ dependance on combinations of two mismatches is displayed. Note again that even two mismatches at the very ends of a 20-mer duplex do not decrease its ability to form a stable duplex to an extent that a single internal mismatch does.

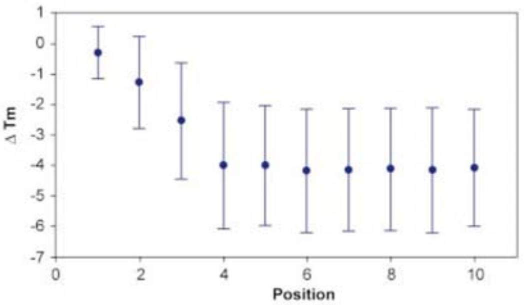

**Figure 3-1**
Calculated $\triangle T_m$ dependency on mismatch position in 20-mers. Average and standard deviation over all nucleotide exchanges are shown. Taken from [18].

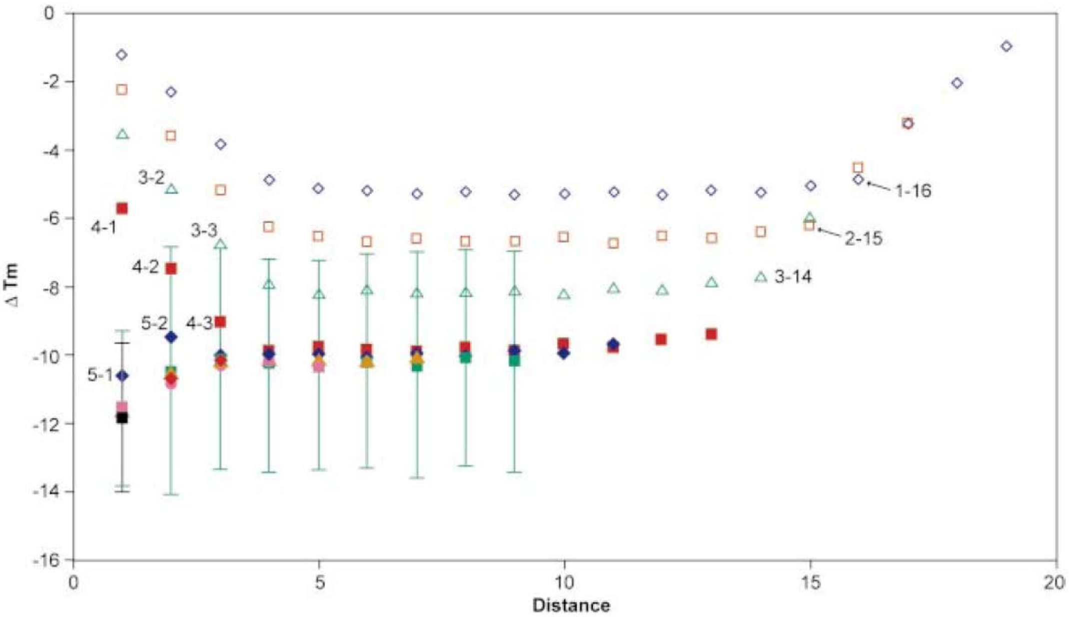

**Figure 3-2**
Calculated $\triangle T_m$ dependency on two mismatch positions in a 20-mer duplex. On the y-axis (negative) $\triangle T_m$ is shown. The first mismatch position is indicated by point shapes: position 1 – open blue diamonds, position 2 – open red squares, position 3 – open green triangles, position 4 – filled red squares, position 5 – filled blue diamonds, positions 6-10 remaining shapes. On the x-axis distance between mismatch positions in the 20-mer is indicated. For more details see original publication [18].

In conclusion we made the decision to analyze the genome for three types of matches: matches with only one mismatch in the middle, termed *exact* $k$-mer matches, matches with an additional internal mismatch, termed *inexact* $k$-mer matches, and those that have several mismatches which only affected the outermost positions of the $k$-mer, termed *short* $k$-mer matches.

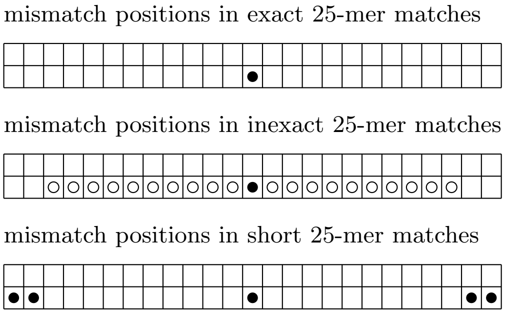
mismatch positions in exact 25-mer matches
mismatch positions in inexact 25-mer matches
mismatch positions in short 25-mer matches

**Figure 3-3**
Positions at which mismatches are tolerated in the three $k$-mer match types. At any number of positions marked by bullets, mismatches are tolerated. At only one position of those marked by a circle an (additional) mismatch is tolerated.

**Exact $k$-mer matches:**

By exact $k$-mer matches of position $p$ we mean all $k$-mers contained in a genome sequence $S$ that exactly match the oligos at one of the 8 spots on the wafer corresponding to position $p$.

This includes any direct occurrence of the sequence $S[p-\frac{(k-1)}{2}, p+\frac{(k-1)}{2}]$ (including the last position) in $S$, but also its reverse complement $\overline{S}[p-\frac{(k-1)}{2}, p+\frac{(k-1)}{2}]$.

Furthermore, matches with a different middle base are also considered to be "exact" here. So, only the first 12 and the last 12 bases of a pair of 25-mers have to agree. Formally we are interested in pairs of positions $p$ and $p'$ where

$$S[p-i] = S[p'-i] \text{ and } S[p+i] = S[p'+i]$$

for all $i = 1, .., \frac{(k-1)}{2}$
Matches between the (+)-strand and the (−)-strand are also of interest:

$$S[p-i] = \overline{S}[p'+i] \text{ and } S[p+i] = \overline{S}[p'-i]$$

for all $i = 1, .., \frac{(k-1)}{2}$.
The Watson-Crick complement (the $3' - 5'$ sequence of the (−)-strand) of the bases in $S$ is denoted by $\overline{S}$. Note that reverting is done by flipping the indices.

**Inexact $k$-mer matches:**

This class of matches contains a second mismatch in addition to the middle one. We restrict this mismatch position so that two positions at either end of the oligo-duplex have to be the same. Thus, effectively mismatches are tolerated at $(k-1)/2-2$ positions upstream and downstream of the middle one. This definition may seem unnecessarily complicated, but by these constraints an overlap between inexact and short $k$-mer matches (defined hereafter) is avoided.

Formally we want to find a pair of positions $p$ and $p'$ where the Hamming distance[14] between matching $k$-mers is exactly 1, excluding 2 positions at either end and not considering a possible middle mismatch:

$$\sum_{i=1}^{(k-1)/2\,-2} (S[p-i] \otimes S[p'-i])$$

$$+ \sum_{i=1}^{(k-1)/2-2} (S[p+i] \otimes S[p'+i]) = 1,$$

or in case of a match to the (−)-strand

$$\sum_{i=1}^{(k-1)/2\,-2} (S[p-i] \otimes \overline{S}[p'+i])$$

$$+ \sum_{i=1}^{(k-1)/2\,-2} (S[p+i] \otimes \overline{S}[p'-i]) = 1,$$

where $(S[i] \otimes S[j]) = \begin{cases} 1 & \text{iff } S[i] \neq S[j] \\ 0 & \text{otherwise} \end{cases}$.

All inexact matches have to meet the constraints for conserved ends:

$$S[p-i] = S[p'-i] \text{ and } S[p+i] = S[p'+i]$$

for all $i = \frac{(k-1)}{2} - 1, \frac{(k-1)}{2}$ or in case of a match between different strands

$$S[p-i] = \overline{S}[p'+i] \text{ and } S[p+i] = \overline{S}[p'-i]$$

for all $i = \frac{(k-1)}{2} - 1, \frac{(k-1)}{2}$.

[14] i.e. the number of mismatches

**Short $k$-mer matches:**

Short $k$-mer matches are defined to be matches that have mismatches at the ends. Any positive number of mismatches at the first two and the last two positions is tolerated while all other bases (except the middle) have to be the same:

$$S[p-i] = S[p'-i] \text{ and } S[p+i] = S[p'+i]$$

for all $i = 1, .., \frac{(k-1)}{2} - 2$.
To avoid an overlap with exact matches we require that there is at least one mismatch among the end positions:

$$\sum_j (S[p+j] \otimes S[p'+j]) \geq 1$$

where $j \in \{\pm(\frac{(k-1)}{2} - 1),\ \pm\frac{(k-1)}{2}\}$.
Similarly matches on different strands:

$$S[p-i] = \overline{S}[p'+i] \text{ and } S[p+i] = \overline{S}[p'-i]$$

for all $i = 1, .., \frac{(k-1)}{2} - 2$.
and additionally

$$\sum_j (S[p+j] \otimes \overline{S}[p'-j]) \geq 1$$

where $j \in \{\pm(\frac{(k-1)}{2} - 1),\ \pm\frac{(k-1)}{2}\}$.

## 3.2 A usual string matching problem?

Finding all exact, inexact and short $k$-mer matches in a genome is a string matching task that is not unusual in bioinformatics. As string comparison and matching lies at the heart of bioinformatics (or computer science in general) there are many good approaches and solutions to problems like this. However, the $k$-mer analysis also has some particularities:

- The problem is actually to find (imperfect) repeats—direct and palindromic ones.
- Only matches of fixed length are of interest here (length $k$ for exact and inexact matches, length $k-4$ for short matches).
- On a whole-genome scale 25 is quite small as match size, and many matches are expected to be found.[15]
- A mismatch in the middle is always tolerated.

These differences to many other pattern matching tasks have several consequences:

Simple exact string matching (using maps or hashing) is not sufficient but most local alignment search tools like BLAST will return hits of a more general type. These would have to be parsed and filtered, which would probably take more time than the actual searches. Consequently, the output of any method should be as easy to parse as possible and redundancies should be avoided. Furthermore, one does not need a very sensitive search method, as all matches are required to have high identity values.

That leads to fast string comparison approaches that use exact seeds even when some mismatches are allowed. Among such algorithms are suffix trees and suffix arrays. Both are similar in that they use a data structure of ordered suffixes of a text against which they compare queries. (When finding repeats, instead of matching a query, the whole data structure is searched for common prefixes). In the former case there is an explicit tree, while in the latter case an array of sorted suffixes has to suffice in order to avoid the memory overhead of a tree. That makes suffix arrays more appropriate in our case, too, as we want to perform a genome-wide analysis. In addition, research on suffix arrays has recently made impressive advances so that enhanced suffix arrays are now at least as fast as suffix tree implementations [1].

In order to find imperfect matches in both approaches exact seeds are computed first, which are then extended in a second step. The number of mismatches limits the length of these seeds (seed lemma) [17] and the smaller the seeds are, the more time consuming the search will be as only a fraction of the seeds can be successfully extended to yield imperfect matches.

[15] When the genome sequence of *A. thaliana* was published, the genome was analyzed for repeats at least 1000 bp long [13].

## 3.3 Analysis of the 120 mega-base genome of *Arabidopsis* Col-0

The goal of the $k$-mer analysis was to report, for every position in the genome of *Arabidopsis*, whether the $k$-mer has an exact, inexact or short match to any other position in the nuclear genome, the mitochondrial genome or the chloroplast genome, and the position of every matching partner. The organellar genomes were included in the analysis because the genomic DNA preparations for chip resequencing were known to be contaminated with sequences from the organelles. So due to their high copy number, $k$-mers from organelles are at least as unfavorable for polymorphism detection as non-unique genomic $k$-mers.

Of course, the $k$-mer analysis only yields a complete and accurate picture for the ecotype Col-0, for which the genome sequence is already known. For all other ecotypes this $k$-mer analysis is somewhat approximate as their genome sequence is not identical with that of Col-0. Some of the duplications that occurred in Col-0 and the corresponding $k$-mer matches are not common to other ecotypes. Or more problematic, a duplication could have taken place in one or more ecotypes that is not observed in the Col-0 genome. Despite these limitations the resequenced ecotypes are related closely enough to make use of the Col-0 $k$-mer data. The actual profits from the analysis are summarized in chapter 4.

The finished sequence of *Arabidopsis* (accession Col-0) has a total length of 119 186 498 bp, the mitochondrial genome comprises 366 924 bp, the chloroplast genome 154 478 bp. After ambiguous bases were removed and the chromosome ends were trimmed[16], there were 239 026 370 $k$-mers from both strands of the genome for which reciprocal matches had to be found and reported.

## 3.4 Description of our K-mer Analyzer

### The basic idea

If one is interested in all $k$-mers which occur more than once in a genome, one can generate a list of all genomic $k$-mers and sort this list lexicographically. After sorting all $k$-mer sequences which are not unique can then be reported in a linear traversal of this list.

As we are interested in pairs of matching positions rather than in the $k$-mer sequences themselves, a list of positions is generated together with the $k$-mer list and permuted in the order of the sorted $k$-mer sequences. Pairs of positions with matching $k$-mers are reported in a straight-forward procedure from these two lists. In the following section we will extend this principle idea in order to compute exact, inexact and short $k$-mer matches on both strands.

### K-mers from both strands of the genome

So far, only $k$-mers from the forward (+) strand are considered and consequently all matches will be (+/+) matches. In fact, the case of $(-/-)$ matches is already covered as every position of a (+/+) match is also one of a $(-/-)$ match and vice versa. But we also want to detect $(+/-)$ matches (which are symmetric to $(-/+)$ matches). This is done by including with every $k$-mer sequence also its reverse complement into the list before it is sorted. Unfortunately this means that the list size, the effort for sorting and the time needed to report matches double as well. To be able to discern between $k$-mers from the two strands, negative positions are used to indicate $k$-mers from the $(-)$-strand.

### Efficiently sorting the $k$-mer list

Whenever one cannot make any assumptions about the input to a sort algorithm, $\Omega(n \log n)$ is a worst-case lower bound for the run-time where $n$ is the length of the list to be sorted [6] (Theorem 8.1, p 167). However, in our case, the length of every list entry is $k$ and the alphabet $\Sigma$ is fairly small: $|\Sigma| = 4$ as we are dealing with DNA sequences from which ambiguous bases (e.g. 'N') have been removed. In this case it is possible to sort in time $O(n\,k\,|\Sigma|)$, which is effectively linear complexity since $k$ and $|\Sigma|$ are bounded.

[16] The first 12 as well as the last 12 nucleotides of a chromosome cannot be in the middle of a 25-mer, so strictly speaking, these positions cannot be resequenced. A similar argument applies for 12 nucleotides upstream and downstream of an ambiguous base in the reference sequence.

We are using a bucket sort-like procedure to sort the $k$-mer list on letter $i$, initially $i = 1$. Assuming a DNA alphabet, afterwards the $k$-mer list $L$ is an ordered partition $L = (L_A, L_C, L_G, L_T)$, where $L_x$ contains all $k$-mers with prefix $x$ of length $i$, so $L$ is now sorted on prefix $x$. Then the sort is recursively carried out on each sub-list $L_x$, now partitioning according to the letter $i{+}1$ which yields $L = (\{L_w | w \prec x\}, L_{xA}, L_{xC}, L_{xG}, L_{xT}, \{L_y | y \succ x\})$ and so forth.[17]

Sorting a (sub-)list $L_x$ of size $m$ on letter $i = (|x| + 1)$ is done by $|\Sigma|$ scans through the list. This is realized by consecutive $c$-scans, one for each letter $c \in \Sigma$ (in their lexicographic order) using two pointers *swp* and *scn*. *swp* is initialized to—and invariantly keeps track of—the first element in $L_x$ for which $d \succ c$, where $d$ is the letter at position $i$ in that $k$-mer. *scn* is initialized to $swp + 1$ and then linearly scans through $L_x$ until it reaches the end of $L_x$. Each time *scn* encounters a $k$-mer having letter $c$ at position $i$, the $k$-mers to which *swp* and *scn* point, are swapped and also *swp* is incremented to the next $k$-mer in $L_x$. This requires $O(|\Sigma|\, m)$ steps.

As the recursion depth of the bucket sort is bounded by $k$ and because *effectively* the whole $k$-mer list is sorted on the $i$th letter exactly once (in $|\Sigma|$ linear scans)—albeit not as a whole in the same method call—the total run-time is $O(n\, k\, |\Sigma|)$.

Note that with a linear runtime this sort procedure is theoretically optimal, although it has some substantial overhead so that it might not be superior to other $O(n \log n)$ sort methods in practice. Note further that there is no extra memory needed in addition to the $k$-mer list which is a crucial feature considering the size of this list.

## Encoding $k$-mer sequences

As already mentioned, the size of the $k$-mer list containing both strands of the whole nuclear genome of *Arabidopsis* plus those from the mitochondrion and the chloroplast genome—239 026 370 $k$-mers in total—motivates a space-efficient encoding of $k$-mers. In Java a (unicode) character is stored in 2 bytes, a single $k$-mer thus takes 50 bytes and the whole list would need more than 11 GB of memory.

DNA sequences (of fixed length and without ambiguous nucleotides) can in principle be stored using only 2 Bits per base—a whole $k$-mer can be encoded as a 64-Bit long ("wasting" 14 Bits). That way the size of the $k$-mer list is reduced to $\frac{8}{50} < 20$ % and is now less than 1.8 GB. Another 900 MB are needed for the list of positions, but the computation is still feasible on a 64-Bit processor with 4 GB RAM.

When nucleotides are encoded in their lexicographic order, $A \equiv 00$, $C \equiv 01$, $G \equiv 10$, $T \equiv 11$, faster arithmetic comparisons can be used instead of string comparison methods. Two $k$-mers can now be compared with a single operation instead of comparing all $k$ pairs of letters. (This kind of comparisons is used when matches are reported.)

Single-letter comparisons are still efficient, provided that nucleotides at the same position are compared (which is always the case in the above algorithm). Before comparing the $i$th position of two $k$-mers, each has to be AND-masked with a long that contains 11 at the two bits encoding the $i$th character and 0..0 elsewhere. This is illustrated by a toy example of 5-mers encoded as 2-Byte words and a comparison of their 3rd character:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1st 5-mer | | | | G | A | G | C | T |
| encoding | 00 | 00 | 00 | 10 | 00 | 10 | 01 | 11 |
| AND-mask | 00 | 00 | 00 | 00 | 00 | $\boxed{11}$ | 00 | 00 |
| 1st masked 5-mer | 00 | 00 | 00 | 00 | 00 | $\boxed{10}$ | 00 | 00 |
| | | | | | | $\prec$ | | |
| 2nd masked 5-mer | 00 | 00 | 00 | 00 | 00 | $\boxed{11}$ | 00 | 00 |
| 3rd char of 2nd 5-mer | | | | | | T | | |

## Dealing with mismatches

The algorithm described so far is only able to find exact matches in the literal sense, it is not even able to detect exact $k$-mer matches as defined above with a potential mismatch in the middle.

The solution of this problem could be thought of as an excision of the nucleotide that is allowed to mismatch. Imagine a list of $(k-1)$-mers consisting of two concatenated $((k{-}1)/2)$-mer "halves" without the nucleotide in the middle instead of the $k$-mer list. Such a

[17] Lexicographic order is denoted by '$\prec$'. $c \prec d$ means that string $c$ is lexicographically smaller than string $d$.

list could be sorted exactly the same way as described above, match sets could be found in the sorted list and reported as before and the result would be matches potentially with a mismatch in the middle.

The excision does not have to occur in reality because we use a sort method that ordered the $k$-mer list on a single position at a time rather than comparing whole $k$-mers. Therefore it suffices to sort the list on $k-1$ positions leaving out the middle one. So surprisingly, finding $k$-mer matches with a mismatch at a fixed position can be done even faster than finding exact matches (in the literal sense).

Clearly this approach generalizes naturally to any number of mismatches, as long as the mismatch positions are all fixed. Thus, finding exact $k$-mer matches and short $k$-mer matches according to the above definitions is straight forward: assuming that $k = 25$, in the first case the $k$-mer list has to be sorted on positions $1-12$ and $14-25$, in the second case on positions $3-12$ and $14-23$.

Considerably more effort has to be made to find inexact $k$-mer matches, since the additional mismatch position is not fixed. In a first step all matches with a mismatch at position 3 (in addition to the middle) are searched which requires to sort on positions $1, 2, 4, 5, .., ((k+1)/2)$ and $((k+1)/2+1), .., k$. Then matches are reported and saved and the sorting is done again, this time leaving out position 4 and inexact matches with possible mismatches at positions 4 and $(k+1)/2$ are saved, and so forth. In total, to detect all inexact 25-mer matches, sorting and reporting matches has to be done 20 times with mismatch position $i = 3, 4, .., 11, 12, 14, 15, .., 22, 23.$[18]

### Reporting matches

To report matching positions $(p, p')$, the sorted $k$-mer list is traversed from begin to end using two pointers, $b$ and $e$, indicating respectively the begin and end of a set of matching positions (Both are initialized to the first list entry). In each step of the list traversal the $k$-mer sequences of list entries $i$ and $(i+1)$ are compared. If both are equal, the $e$ pointer is set to $(i+1)$. If they differ, a set of matching positions can be reported from the position list in the interval $[b, e]$, provided that $e > b$. Then the $b$ pointer is set to $(i+1)$. When $e$ reaches the end of the list, a last set of matches is reported from the list interval $[b, e]$ in case that $e > b$. Upon its detection a match set is converted to reciprocal hits $(p, p')$ and $(p', p)$ for all positions $p$ and $p'$ in the match interval $[b, e]$ (in time proportional to $(e-b+1)^2$).

In the comparison step, mismatch positions have to be taken care of by masking them prior to the actual comparison. This can be done e.g. with an OR-mask containing 11 at the two bits encoding the mismatch positions and 0..0 elsewhere so that mismatch positions are the same in all comparisons regardless of the actual nucleotide.

## 3.5 Discussion of our approach

In principle such a $k$-mer analysis could as well be done with a suffix array implementation as e.g. VMATCH [1] and such a highly optimized tool is probably faster for some of the matching tasks. In practice, however, there are some hurdles.

First, it is highly inefficient to use every $k$-mer from the tiling as a query in the search for matches in the genomic text, since these queries are highly redundant as two neighboring $k$-mers have $(k-1)$ characters in common. On the other hand, if longer, less overlapping queries or a repeat search are used, $k$-mer matches have to be parsed from alignments longer than 25 bp. This is trivial for exact matches, but when it comes to mismatches, a lot of the alignments will not meet the inexact $k$-mer match criterion and subsequently be discarded.

Second, with a suffix array approach one could naively try to find short $k$-mer matches by allowing for 5 mismatches. This is, however, not advisable since in case of 25-mers the seed length decreases to 4 and matching speed is reduced drastically. Instead one could try to find matches of length $(k-4)$ allowing for a single mismatch (in the middle). The disadvantage then is that the overlap with the set of exact and inexact matches will be large. Another task that has to be solved is to extend alignments of size $< k$ in order to obtain short $k$-mer matches.

[18] Note that inexact matches with a mismatch at either of the first two or last two positions are not considered here because these matches are already contained in the set of short $k$-mer matches.

In summary it seemed that the advantage of faster matching was counterbalanced by the time needed to parse $k$-mer hits from the output of VMATCH. This was probably mostly due to reading/writing to hard disk during a parsing step which can be avoided computing $k$-mer matches directly. The decision to implement the $k$-mer analyzer as described here was finally motivated by the impression that correctness was easier to ensure with a direct approach than with processing VMATCH alignments.

This advantage and the gain in speed come however at the cost of memory usage—VMATCH only uses a fraction of the RAM that our $k$-mer analyzer needs. In comparison to suffix arrays our $k$-mer analyzer is special-purpose software that does not generalize well to the solution of similar problems. It is also very limited in the number of mismatches at arbitrary positions in $k$-mer matches. If for example one wants to tolerate 2 mismatches at positions $3, 4, .., 11, 12, 14, 15, .., 22, 23$, it will be necessary to sort the whole $k$-mer list 190 times.

An extension of our $k$-mer analyzer seems possible: Relaxed short $k$-mer matches with larger mismatch regions at either end can be easily computed. However, gapped matches between $k$-mers, modelling small bulges in the oligo-duplexes, are impossible to detect efficiently without the introduction of further data structures. This could be a more serious drawback than it first seems. In resequencing experiments in which also probes for insertion and deletion (indel) detection were used, it has been found that 1 bp indels have a destabilizing effect on the oligomer duplex that is lower than for single base mismatches [15].

In some sense, one can argue, our $k$-mer analyzer is a very rudimentary suffix array approach that exploits that only suffixes of length 25 matter. (Thus one neither needs to keep track of the suffix length nor the length of shared prefixes.) The second simple fact that can be taken advantage of is that we simply need *all* genome-wide matches. (Therefore an inverse position array is not required, one can simply scan the whole sorted $k$-mer list and report all matches in a single pass.)[19]

Simple matching problems like the one presented here often have the nice property that they can be easily parallelized. In our $k$-mer analyzer the 22 sorting steps are completely independent and do not have to run sequentially, they can be computed just as well in parallel.

## 3.6 Results

In total, the $k$-mer analysis resulted in almost 1 billion matches. All reciprocal matches including the chloroplast and the mitochondrial genome as follows:

**Table 3.1 K-mer matches**

| | |
|---|---|
| Exact $k$-mer matches | 333 983 864 |
| Inexact $k$-mer matches | 305 872 698 |
| Short $k$-mer matches | 292 480 337 |

Reciprocal matches $(p, p')$ where $p$ is on the nuclear genome, while the match partner $p'$ may also be on the nuclear genome or on an organellar genome:

**Table 3.2 K-mer matches with one partner in the nuclear genome**

| | |
|---|---|
| Exact $k$-mer matches | 333 577 772 |
| Inexact $k$-mer matches | 305 844 001 |
| Short $k$-mer matches | 292 464 314 |

All statistics on $k$-mer matches in the following are calculated only from this set of matches.

Of course the mere number of matches does not mean that there are no unique $k$-mers in the *Arabidopsis* genome. Although the different match classes are defined such that they do not overlap, a given $k$-mer can have exact, inexact and short matches, hence match positions can overlap. Counting the positions where the $k$-mer is not unique yields:

**Table 3.3 Positions with k-mer matches**

| | |
|---|---|
| Positions with exact $k$-mer matches | 12 970 807 |
| Positions with inexact $k$-mer matches | 14 510 324 |
| Positions with short $k$-mer matches | 7 059 270 |
| Total number of positions with $k$-mer matches | 21 338 048 |

Table 3.3 shows that in fact most of the matches are between a fairly small number of

[19] For details on suffix arrays see [19] [1].

sites and most of the *Arabidopsis* genome was found to be unique in the sense that at such positions no $k$-mer matches are found. The overlap between positions having a certain type of $k$-mer match is depicted by a Venn-diagram in figure 3-4.

Figure 3-5 gives an illustration of the distribution of duplicated $k$-mers over the genome. It was created with a moving average technique. The fraction of match positions in a window of size 100 kb is plotted, so for example a value of 1 means that at all 50 000 positions upstream and at all 50 000 positions downstream there is at least one $k$-mer match. Different match types are drawn in different colors. A clear correlation between $k$-mer matches of different types can be seen as well as a general increase of duplicated segments around the centromeres, which are indicated by grey sketches of the chromosomes below the x-axis. Larger duplications appear as spikes (values $> 0.5$ correspond to well-conserved duplications longer than 50 kb). The baseline below 0.1 is mostly due to smaller duplications, small simple sequence repeats like common $(\mathrm{AT})_n$ repeats and interspersed $k$-mers that occur multiple times in the genome but do not belong to larger repeats.

Figure 3-6 shows from which genome location match partners come in longer non-unique stretches. This graph was created with a similar method used for figure 3-5, but with a smaller window of 10 kb. Matching regions are only shown if at least 50 % of the positions inside the window have matches to a certain chromosome or organellar genome (mitochondrion or chloroplast). The location of the match partner is color-coded. There appear to be no longer high-identity matches to the chloroplast genome, while a 270 kb insertion from the mitochondrial genome on the left arm of chromosome II can be clearly seen (drawn in red and marked by an asterisk). This insertion has previously been found [9] and reported to contain about 75 % of the mitochondrial genome; it is well-conserved with 99 % identity (to the mitochondrial genome of *Arabidopsis* accession C24). That it is almost identical to the original sequence is reflected in the red peak very close to 1.0.

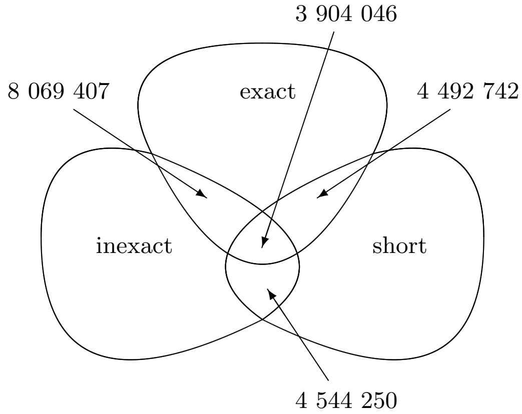

**Figure 3-4**
Intersection between positions with $k$-mer matches. The overlapping areas indicate that these three sets of positions are not disjoint. In the overlaps, numbers are given for the intersections between 2 of these sets or all of them respectively, i.e. at 3 904 046 of the 8 069 407 positions where there is an exact and an inexact $k$-mer match, there is also a short $k$-mer match.

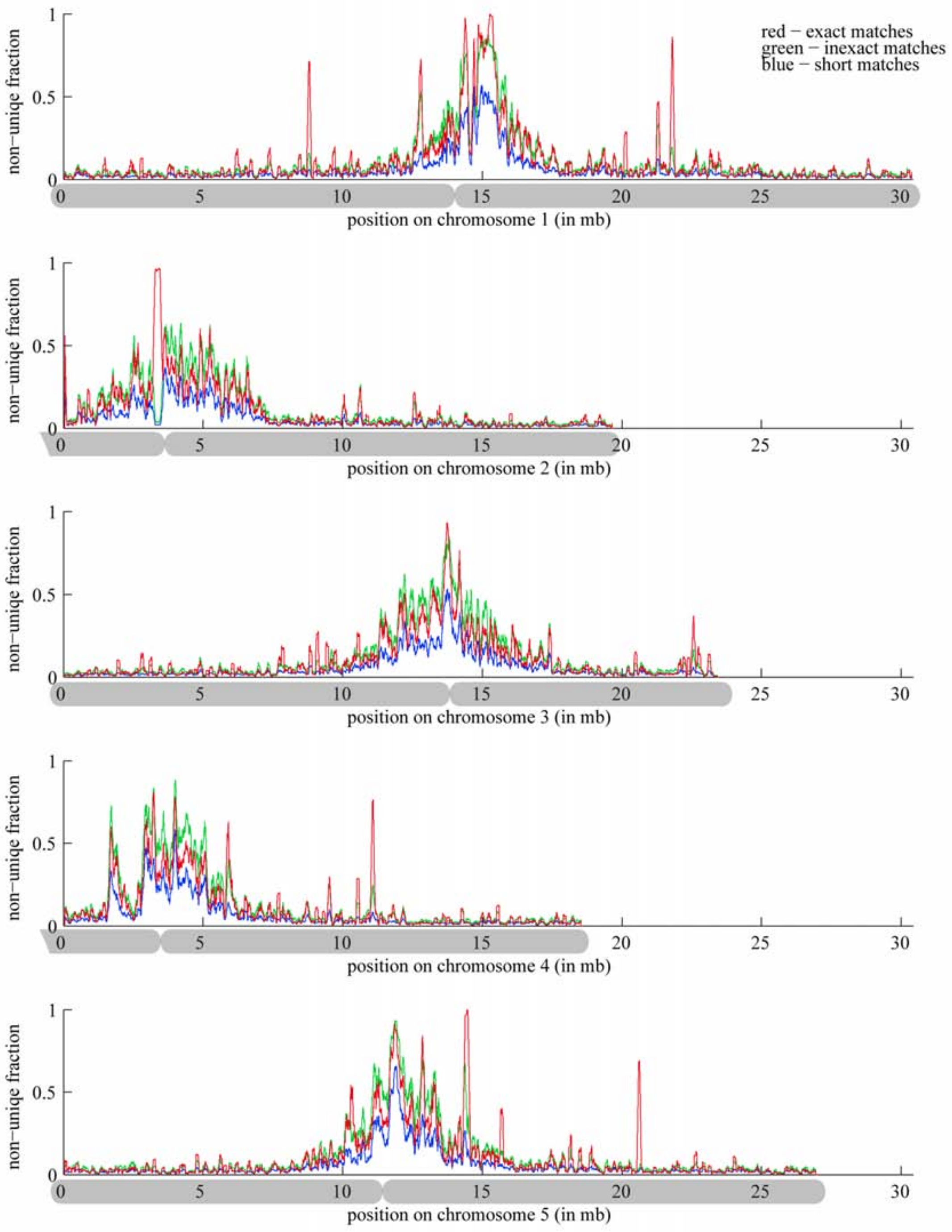

**Figure 3-5**
Distribution of duplicated *k*-mers over the genome. In y-direction the fraction of positions with non-unique *k*-mers in a sliding window of size 100 kb is drawn. Different *k*-mer match types are color-coded. Below the x-axis grey shapes indicate the approximate locations of the centromeres.

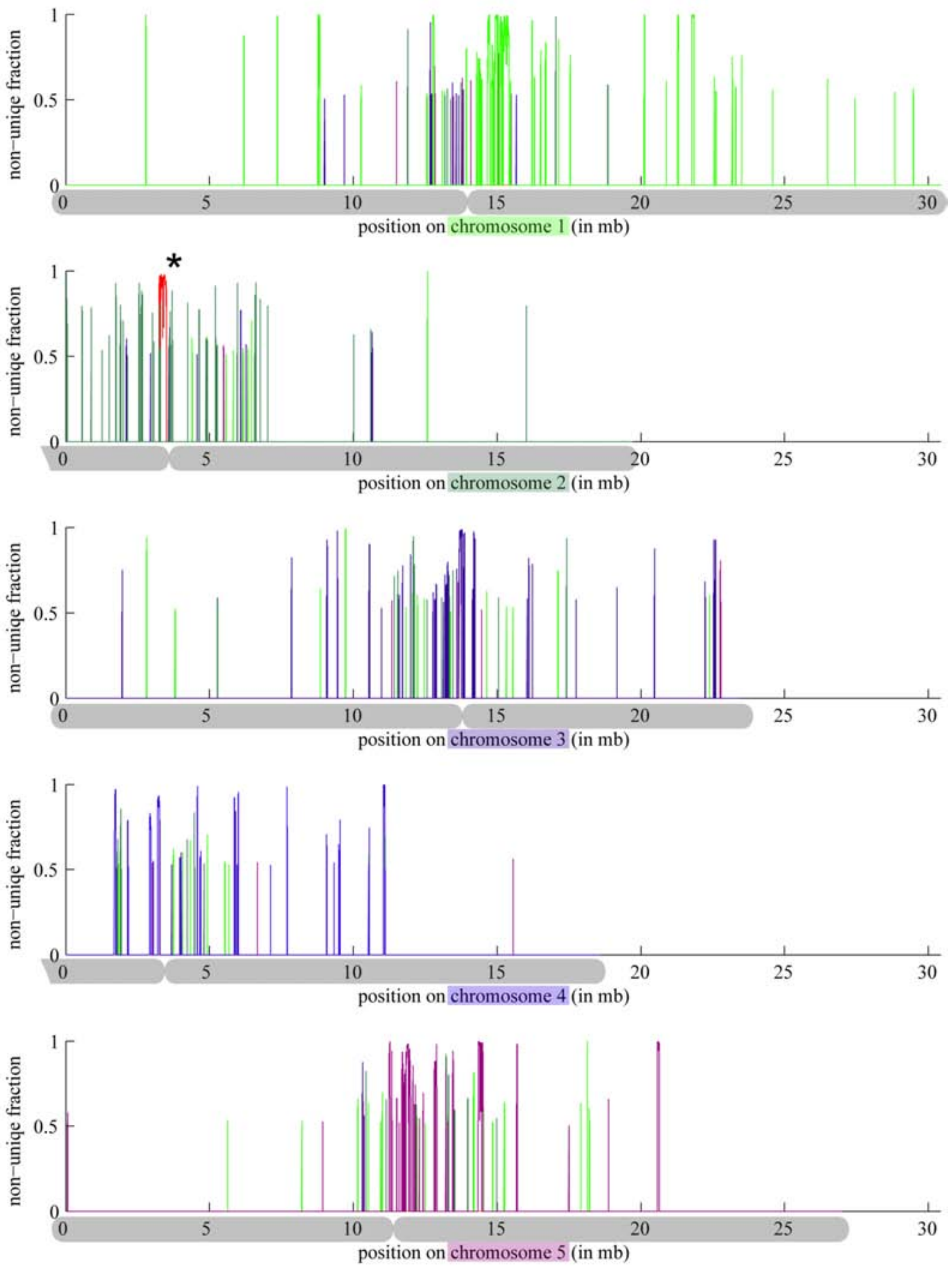

**Figure 3-6**
K-mer matches colored according to the location of the match partner. Spikes indicate positions around which k-mer matches to a certain chromosome occur with a frequency $\geq 0.5$ in a sliding window of size 10 kb (peak height corresponds to frequency, i.e. the relative number of match positions in the window). The location of the match partner is specified by different colors: chromosomes (x-axis labels) are shaded in their own color, matches to the mitochondrial genome are drawn in red. A large insertion of mitochondrial DNA into the left arm of chromosome 2 is marked with an asterisk. The chloroplast genome was also included in this analysis, but does not appear to have an insertion into the nuclear genome long enough to be seen in this plot.

# 4 Application of the k-mer data

In this chapter it is illustrated how the $k$-mer matches, once computed, are applied to improve the performance of SNP calling and deletion calling. Before that it is described how the $k$-mer data is stored and organized in the database, which facilitates the integration with other data sources already contained in the database.

## 4.1 Storing k-mer data in the database

There are three SQL tables for the $k$-mer matches, one for each match type (kmers_exact_matches, kmers_inexact_matches and kmers_short_matches). In addition to the pair of positions, entries in these tables also contain strand information, i.e. whether a match is a (+/+) or a (+/-) match, and which nucleotide it *supports*. As in the $k$-mer analysis mismatches in the middle of the 25-mer are always tolerated, the information, which nucleotide is actually present in position 13 of the match partner, is important as it determines to which of the probes in a quartet at a given site there will be cross-hybridization. In other words, a match that *supports* nucleotide A at a given position potentially increases the intensity recorded for the A probe.

These match tables contain the complete result of the $k$-mer analysis and are thus quite large. In fact information retrieval from these tables is time-consuming particularly when $k$-mer data is needed for longer stretches of the genome. In order to increase the speed of queries to determine which positions correspond to unique $k$-mers and which positions do not, three further tables were derived, in which matches are mapped to genome position, discarding the information *to* which position there is a match. That means that these tables now contain at most one entry per position—the number of rows is actually reduced more than 20-fold—and thus support much faster query speed. As at a given position there can be matches that support different base calls, the derived tables contain one column for each of the four possible supports and these support columns simply contain the number of respective supporting matches. Therefore these tables are called “count” tables (kmers_exact_counts, kmers_inexact_counts and kmers_short_counts).

## 4.2 Using k-mer data to improve SNP-calls

As briefly mentioned when motivating the $k$-mer analysis, duplicated $k$-mers have the potential to interfere with SNP calling in two ways. One possibility is that a real SNP is suppressed by an excess of $k$-mers supporting the reference. The SNP then goes undetected and the reference allele would be assigned as the genotype at that site. But basically reference supporters can decrease sensitivity, which is not as dramatic as reduced specificity.

Reduced specificity or a higher number of false positive SNP calls can be caused by cross-hybridizing effects that look like SNPs. But only a minority of $k$-mer matches can give rise to such effects namely those matches that support a base call that differs from the reference.

In a strict sense this statement can only be made if all $k$-mer matches are exactly known for the ecotype of interest. However for the $k$-mer analysis, only the reference genome is available, thus the result is only approximately correct for other ecotypes. For these ecotypes another scenario can be thought of: Imagine a $k$-mer that occurs in high copy-number in the genome because it belongs to a transposon. In one ecotype this $k$-mer could have been hit by a point mutation in the middle *before* it was duplicated many times. Now all these mutated copies could in principle interfere with SNP detection, while in our analysis they would look like reference supporters. Although this seems possible and some transposable elements are known to have evolved very fast and recently in higher organisms, this scenario would be expected to be rare as all the ecotypes are quite closely related and do not have a long evolutionary history on their own. This implies that it is unlikely that they have developed fundamentally distinct compositions of transposon families. Therefore the introduction of a concept of *dominating* $k$-mer matches appears helpful to characterize matches with non-reference support.

### Dominating k-mer matches

A position is said to have *dominating* $k$-mer matches, simply if for any non-reference base the number of supporting matches is greater or equal to the number of matches supporting the reference base at this position.

The following table summarizes the number of positions with dominating $k$-mer matches (on the nuclear genome only):

**Table 4.1 Dominating k-mer matches**

| | |
|---|---|
| Positions with dominating exact $k$-mer matches | 544 241 |
| Positions with dominating inexact $k$-mer matches | 1 229 912 |
| Positions with dominating short $k$-mer matches | 873 801 |
| Total number of positions with dominating $k$-mer matches | 2 158 458 |

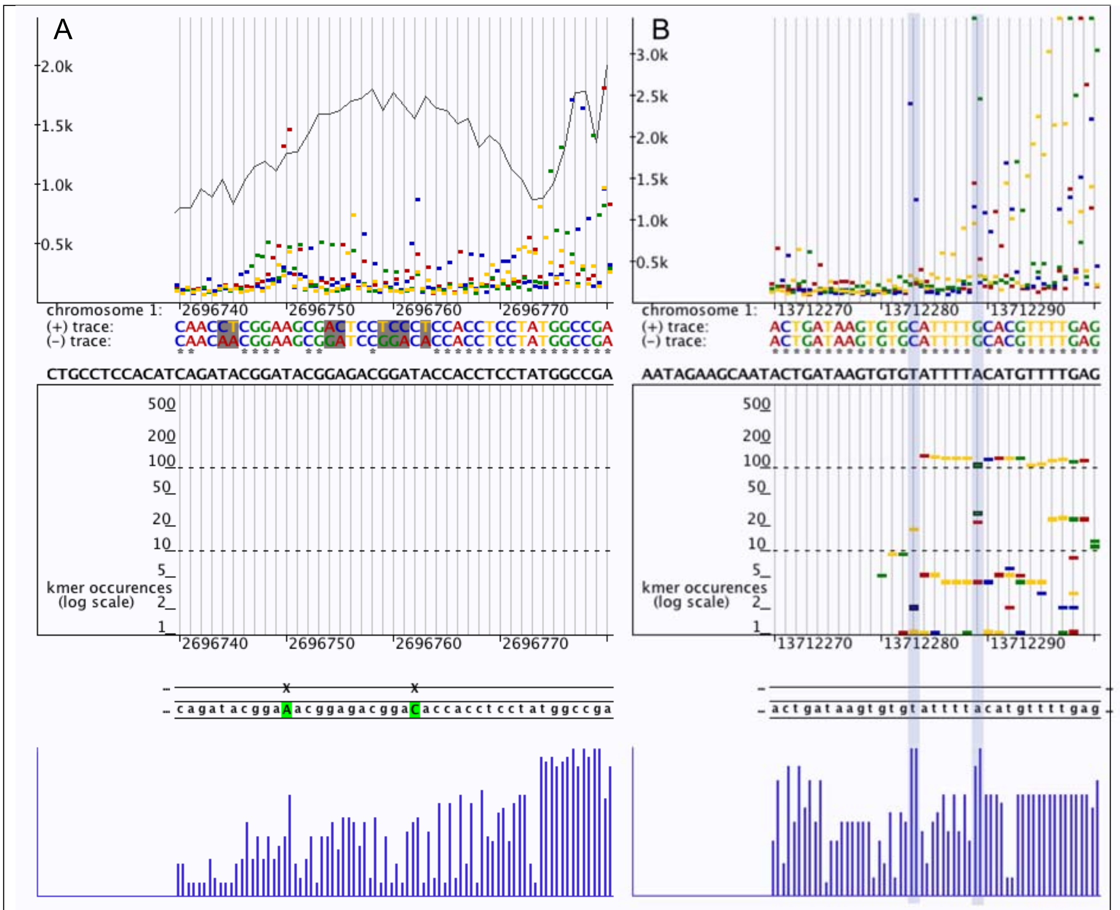

**Figure 4-1**
**(A)** Two SNPs in Bur-0 confirmed by Magnus Nordborg's data in a region of unique sequence. Intensities are visualized in the upper panel as colored squares (the maximum intensity of the Col-0 reference is drawn as black line). The middle panel shows that there are no $k$-mer matches in this region (compare to B). The dideoxy sequence is visualized below that, SNPs are highlighted in light-green. The lower panel shows quality scores for both strands as blue bars.
**(B)** A region of non-unique sequence for which the dideoxy sequences of Bur-0 do not contain any SNP. In the middle panel $k$-mer matches are plotted as small bars in the color of the nucleotide they support. The height at which they are drawn depends on the number of matches at this site (logarithmic scale). The match type is indicated by the thickness: thick – exact matches, medium – inexact matches, thin – short matches. Dominating matches are framed in black. Two positions with dominating $k$-mer matches are shaded in blue. Note that calls of both strands deviate from the reference at these positions despite very high quality scores.

Figure 4-1 illustrates the importance to integrate $k$-mer match data into a SNP calling algorithm (or at least filter the result with $k$-mer match data). It shows examples of real SNPs, confirmed by dideoxy sequencing done at Magnus Nordborg's lab [23][20], and of some sites that are not polymorphic, but where dominating $k$-mers cause a signal which is hard to distinguish from a real SNP signal.

The right panel of figure 4-1 also contains a site (position 13 712 292, just 2 bp downstream of the right site shaded in blue) that shows a limitation of our concept of dominating $k$-mer matches. Although the number of matches supporting T is greater than the number of those supporting C, the intensity of the C-probe is higher, most likely because C-G base pairing adds so much to the stability of an AT-rich oligo duplex that it even exceeds the signal from A-T pairing at higher concentration. Consequently, with an accurate model of intensity dependency on $k$-mer sequence and copy number one could develop a much more sophisticated concept of dominating $k$-mer matches which could probably more accurately predict false positive SNP calls caused by cross-hybridization.

### K-mer matches partially explain Col-0 failures

As mentioned in the discussion of the resequencing data, a significant fraction of the queried positions failed in the control resequencing experiment of the reference ecotype Col-0. By "failure" in this context we mean a position where at least on one strand intensity is maximum for a non-reference probe. (These positions were also labelled "bad" and collected in the SQL table bad_position.) By this definition 24 716 041 positions, which is about 20 % of all queried sites, failed in the Col-0 reference resequencing. The intersection of these and all positions with $k$-mer matches contains 4 447 806 sites, the intersection with positions with dominating $k$-mer matches contains 1 194 004 sites. These intersections may seem insignificant and it is clear that $k$-mer matches and cross-hybridization cannot account for all challenges one faces with the resequencing technique. It might as well indicate that there is potential to increase sensitivity and specificity of the $k$-mer analysis in order to detect more potentially cross-hybridizing sequences, but also to develop better criteria for the extent to which different mismatches affect hybridization properties.

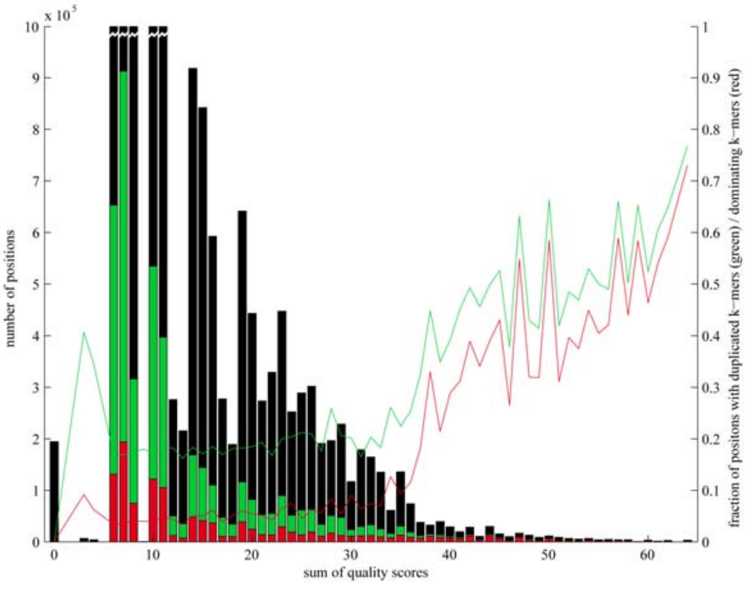

**Figure 4-2**
Histogram of failed sites in the Col-0 control experiment. On the x-axis the sum of forward and reverse quality scores is drawn. The bar plot shows the number of positions (total number in black) with $k$-mer matches (green) and those with dominating matches (red). Relative fractions are also drawn as lines in the respective colors (y-axis on the right). They show that the higher the quality score the more failed sites are probably caused by cross-hybridization from (dominating) $k$-mer matches.

[20] see also section 2.4

Accepting that some sites fail in the resequencing experiments, one can of course focus on those failed sites that will most likely mislead SNP calling by increasing the number of false positive calls. Without making too many assumptions about the actual calling method, it is obvious that failed sites with higher quality scores pose more serious problems than low quality sites, as the latter will probably not be called at all, but labelled ‘N’.

Figure 4-2 illustrates which fraction of failed sites in Col-0 coincides with positions of duplicated $k$-mers. Note that the higher the quality score the larger the relative intersection with positions with $k$-mer matches. It is also worth noting that this correlation is only slightly worse for dominating $k$-mer matches than for all matches, although at only $\sim$ 10 % of all positions with $k$-mer matches there are also dominating $k$-mer matches (2 158 458 of 21 338 048). Note further that this is most likely not a stochastic effect, since even the small bins with quality sum scores $\geq 50$ contain more than 300 samples each, most of them even more than 1000. In conclusion, filtering for positions with $k$-mer matches has the potential to substantially reduce the false discovery rates of SNP detection.

### K-mer matches and Perlegen's SNP calls

Together with the raw data from the resequencing experiments performed at Perlegen Sciences we also received a list of high confidence SNP calls (the calling algorithm is described in the supplementary material of [12]). Using Magnus Nordborg's data from dideoxy sequencing we could verify the outstanding specificity of these SNP calls, especially in coding regions of the genome. However, at some SNP sites the allele called in Col-0 differed from the reference sequence. To my knowledge the calling algorithm does not take any long-range information about the reference sequence into account, i.e. there is no possibility to correct for cross-hybridizing $k$-mers. Therefore it seems obvious to compare those false positive Col-0 SNPs to the lists of positions with (dominating) $k$-mer matches; the result is summarized in table 4.2.

These data suggest that a filter for the detection of problematic sites in Col-0 which is based on dominating $k$-mer matches has almost the same sensitivity as a filter based on all $k$-mer matches, but the specificity is much higher in case that dominating $k$-mer matches are used.

**Table 4.2 K-mer matches and SNP calls in the reference Col-0**

| | |
|---|---|
| SNP calls in Col-0 | 3 471 |
| Col-0 SNPs coinciding with exact $k$-mer matches | 2 717 |
| Col-0 SNPs coinciding with inexact $k$-mer matches | 1 436 |
| Col-0 SNPs coinciding with short $k$-mer matches | 662 |
| Col-0 SNPs coinciding with any $k$-mer match | 3 287<br>(94.6 %) |
| Col-0 SNPs coinciding with dominating exact matches | 2 585 |
| Col-0 SNPs coinciding with dominating inexact matches | 1 129 |
| Col-0 SNPs coinciding with dominating short matches | 566 |
| Col-0 SNPs coinciding with any dominating match | 3 196<br>(92.1 %) |

These numbers further emphasize the importance of analyzing genome wide $k$-mer repeats in order to be able to make predictions where cross-hybridization is likely to occur. This result also indirectly verifies the correctness of the $k$-mer analyzer, as there are 4 874 positions with dominating $k$-mer matches of which 3 196 overlap with SNP calls in Col-0 while the total number of bases sequenced in Col-0 is 654 491.

The distribution of false positive Col-0 SNPs over the genome and the spatial correlation to $k$-mer matches (of any type) or to dominating $k$-mer matches is illustrated by figure 4-3 and by figure 4-4 respectively.

Since we cannot modify Perlegen's SNP calling algorithm, the only option is to filter the results with the $k$-mer data and exclude all SNPs at non-unique $k$-mers in order to decrease the number of false positives which are due to cross-hybridization. The problem with such filters is that they will also filter out many true SNPs. An estimate for these filter properties can be made on those parts of the genome where dideoxy sequences are available. The results are summarized in table 4.3. By false positive SNP calls we mean all positions at which there is a resequencing SNP call, that is reported as reference by dideoxy sequencing; additionally all SNP positions at which the genotype (other than ‘N’) of any ecotype differs between the resequencing and the dideoxy data.

**Table 4.3 K-mer matches and SNP calls evaluated on dideoxy sequencing data**

| | |
|---|---|
| Total number of bases in dideoxy sequences | 686 924 |
| Total number of SNPs in dideoxy sequences | 12 946 |
| Number of false positive SNP calls in dideoxy sequences | 143 |
| Positions with exact $k$-mer matches coinciding with true SNPs | 414 |
| Positions with exact $k$-mer matches coinciding with false positives | 25 |
| Positions with exact $k$-mer matches in dideoxy sequences | 14 588 |
| Positions with inexact $k$-mer matches coinciding with true SNPs | 615 |
| Positions with inexact $k$-mer matches coinciding with false positives | 11 |
| Positions with inexact $k$-mer matches in dideoxy sequences | 21 645 |
| Positions with short $k$-mer matches coinciding with true SNPs | 298 |
| Positions with short $k$-mer matches coinciding with false positives | 6 |
| Positions with short $k$-mer matches in dideoxy sequences | 9 851 |
| Positions with any $k$-mer match coinciding with true SNPs | 987 |
| Positions with any $k$-mer match coinciding with false positives | 34 |
| Positions with any $k$-mer match in dideoxy sequences | 37 491 |

Table 4.3 shows that the ratio between filtered false positive SNPs and lost true SNPs is 1:17 for exact matches, 1:56 for inexact matches and 1:50 for short matches, respectively. This means that exact $k$-mer matches are a much more specific filter than inexact or short matches. Moreover, filtering with inexact matches would exclude the highest number of positions in general as well as positions with true SNP. Thus, combining only exact and short $k$-mer matches into one filter appears to be a good alternative with a filter ratio of 1:22 (false positive SNPs to true SNPs). We preferred this one over the combination of all $k$-mer matches to filter Perlegen’s SNP calls.

The evaluation of this combined filter is summarized as follows:

**Table 4.4 Evaluation of exact and short matches at resequencing SNP calls**

| | |
|---|---|
| Positions with exact and short matches coinciding with true SNPs | 606 |
| Positions with exact and short matches coinciding with false positives | 28 |
| Positions with exact and short matches in dideoxy sequences | 21794 |

The performance of dominating $k$-mers as a filter for SNP calling is shown in the following table:

**Table 4.5 Evaluation of dominating matches at resequencing SNP calls**

| | |
|---|---|
| Positions with exact dominating matches coinciding with true SNPs | 110 |
| Positions with exact dominating matches coinciding with false positives | 8 |
| Positions with exact dominating matches in dideoxy sequences | 865 |
| Positions with inexact dominating matches coinciding with true SNPs | 226 |
| Positions with inexact dominating matches coinciding with false positives | 7 |
| Positions with inexact dominating matches in dideoxy sequences | 2 719 |
| Positions with short dominating matches coinciding with true SNPs | 117 |
| Positions with short dominating matches coinciding with false positives | 1 |
| Positions with short dominating matches in dideoxy sequences | 1 917 |
| Positions with any dominating match coinciding with true SNPs | 363 |
| Positions with any dominating match coinciding with false positives | 14 |
| Positions with any dominating match in dideoxy sequences | 5 070 |

A comparison between all $k$-mer matches and dominating $k$-mer matches as filters shows that dominating $k$-mers have less power to reduce the number of false positives, but also filter out fewer real SNPs. Except for short matches, dominating $k$-mer matches are still more specific. Consequently, dominating $k$-mer matches can be used as a minimalist filter. As such they are well suited for pre-filtering the input data for a support vector machine which attempts to classify the data into conserved versus SNP positions. Such a learning machine can be given the complete set of $k$-mer matches as features so that it might be able to autonomously learn something about the implication of certain types of $k$-mer matches to SNP calling.

When filtering Perlegen's SNP calls, a filter ratio of 1:22 still distinctly improves the quality of the SNP calls, because generally the error rate is lower; depending on the ecotype, the ratio of false positive SNP calls to true positives is between 1:21 and 1:72 (average 1:36)[21].

## 4.3 K-mer data as a prerequisite for deletion-calling

How to call deletions is discussed in chapter 5, so at this point a calling algorithm is not described in detail. Instead, after describing the fundamental signal that any deletion calling algorithm will have to detect, I will focus on the implication of non-unique $k$-mers in deleted regions.

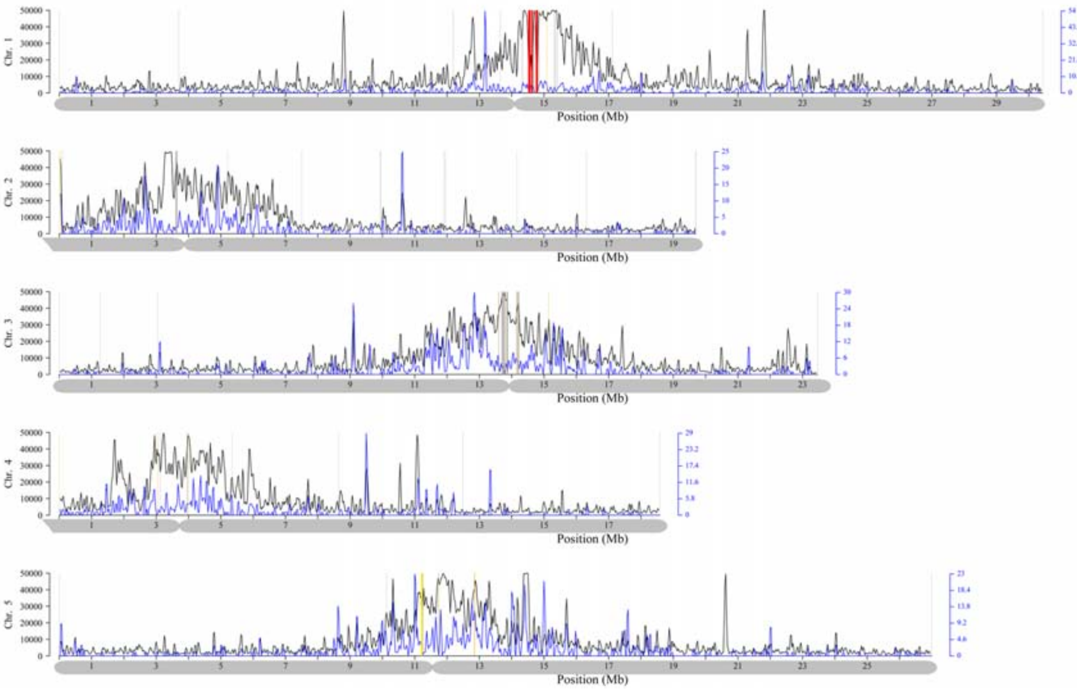

**Figure 4-3**
Distribution of duplicated $k$-mers together with false positive SNP calls in Col-0 over the genome. In black the fraction of positions with non-unique $k$-mers in a sliding window of size 50 kb is drawn; in blue the number of false positive SNPs in the same window. Vertical lines are used to indicate sites of missing data (where there is no unambiguous reference sequence). Their size is color-coded: grey – size > 1, yellow – size > 100, orange – size > 1000, red – size > 4000. (This plot was created with an R script written by Richard Clark.)

[21] excluding the ecotype Van-0

Calling deletions is obviously based on detecting the absence of a clear hybridization signal, for in a deleted region there is no sequence that could hybridize to the corresponding probes on the array. As the intensities in regions that are not deleted already exhibit a large variation, one has to have an accurate expectation of the intensities under the null model that a sequence is not deleted. One such null model is obviously the intensities of the reference ecotype Col-0. Hence, intuitively one attempts to call deletions in a target ecotype in regions where the intensities of the target ecotype are significantly smaller than the reference ecotype.

Without knowing the details about the actual calling method, consider a case in which the deletion did not occur in a region of unique sequence. If the $k$-mers within a deletion are also present somewhere else in the genome, cross-hybridization will lead to normal intensities making it difficult to tell whether there is a deletion or not. This is illustrated by figure 4-5. The example shown is even less problematic than cases where a large deletion contains several smaller areas with $k$-mer matches. In such a case one might call several deletion instead of a single one, as the deletion signature is interrupted by high intensities due to cross-hybridization of repeated $k$-mers. Addressing this problem is only possible if one has data on duplicated $k$-mers. Thus the result of the $k$-mer analysis is a fundamental requirement for deletion calling, which is described in depth in the following chapter.

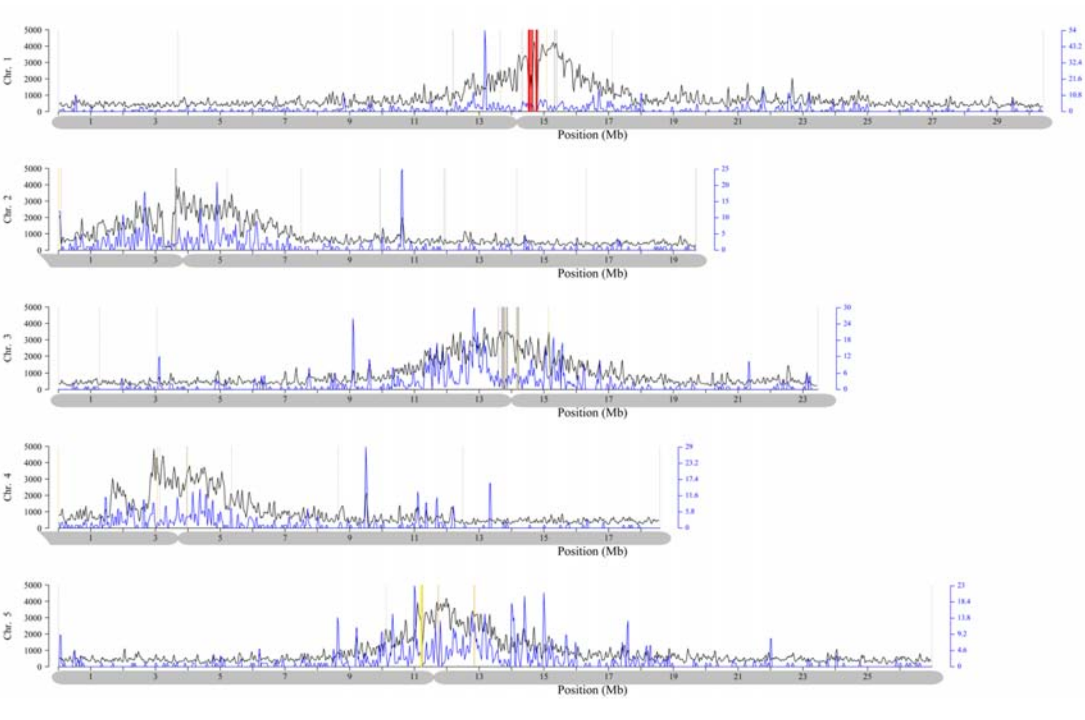

**Figure 4-4**
Distribution of dominating $k$-mer matches together with false positive SNP calls in Col-0 over the genome. In black the fraction of positions with dominating $k$-mer matches in a sliding window of size 50 kb is shown; in blue the number of false positive SNPs in the same window. As in the previous figure vertical lines are used to indicate sites of missing data. (This plot was created with an R script written by Richard Clark.)

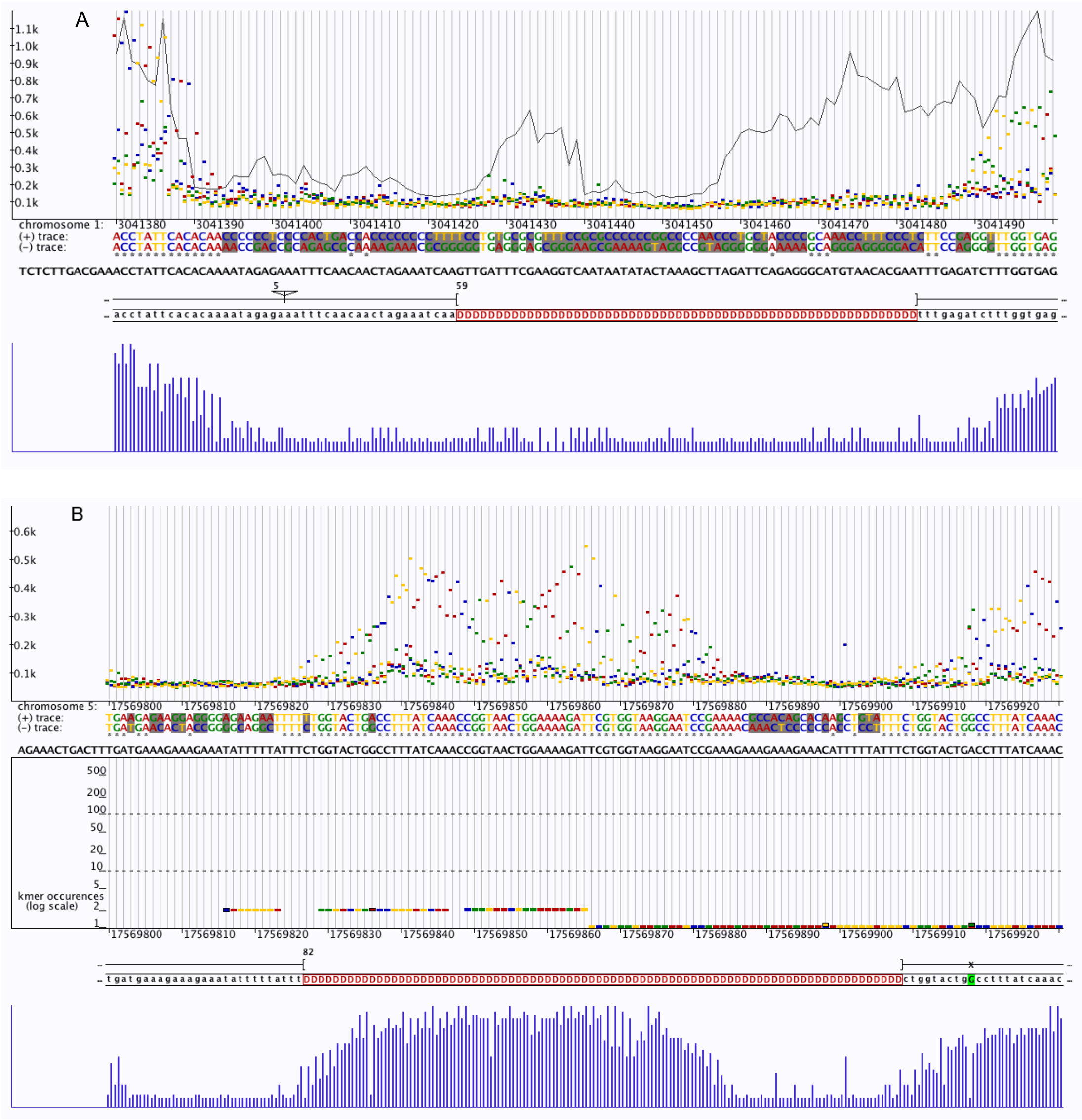

**Figure 4-5**
**(A)** A deletion in a region of unique sequence in Bur-0, i.e. without any *k*-mer match (data not shown). Note that the intensities (colored squares on the upper panel) and quality scores (blue bars) are very low in the deleted region (verified by dideoxy sequences, depicted by red 'D's), but increase quickly downstream of the deletion. About 20 bases upstream there is an insertion (black triangle) that results in weak signals between the two polymorphisms, but upstream of this insertion the transition to normal intensities and quality scores is quite sharp (compare the intensities to the one of the reference Col-0 indicated by the black line).
**(B)** A deletion in a region of non-unique sequence in Ler-1. Virtually the whole deleted region of 82 bp is covered by *k*-mer matches (colored bars in the middle panel). In the 5' half of the deleted region cross-hybridization leads to a signal that agrees almost perfectly with the reference affirmed by high quality scores. The intensities and quality scores in the deletion are even stronger than in the upstream region that is not deleted.

# 5 Calling large deletions

In this chapter it is described how the resequencing data can be used to detect large deletions in the 19 non-reference ecotypes. A brief introduction of the basic concept of deletion calling and the hurdles to be taken by a deletion calling algorithm is followed by an explanation of the heuristic we developed to tackle these problems. The chapter is concluded with an evaluation and discussion of our method.

## 5.1 An elementary concept of deletion calling

The elementary concept of deletion calling is quite obvious: To infer from low hybridization intensities the absence of genomic DNA sequences, i.e. deletions. Deletion calling from microarray data has been approached previously [25], [4]. As the variance of intensity values is usually too large to infer anything meaningful by simple thresholding, comparative approaches are more appropriate (and proposed in both cited publications).

There are several possibilities for comparisons. One is the comparison of the intensity of the perfectly matched probe (PM) to the intensities of mismatched probes (MM) within the same probe quartet. A large difference is expected in case that the interrogated DNA is present. If the interrogated DNA is deleted, this difference is expected to be much smaller [25].

Such a comparison is problematic as there are many sites even in Col-0, in which the ratio between PM and MM intensity is very small, although the interrogated DNA of the reference ecotype is not polymorphic or deleted. Typically such sites are not random, and occur in longer stretches that could easily lead to spurious deletion calls. This is a problem of intra-sample variability.

Another possibility is to compare observed to expected intensities predicted by an algorithm that models hybridization behavior from oligo sequence and concentration. Such algorithms have been proposed [11], [20], [27], but unless these are able to give very accurate predictions, such an approach will have greater difficulties than a comparison between different observations.

A third possibility are inter-sample comparisons between different ecotypes. Naturally, the intensities of the reference ecotype Col-0 are a good null model to which one can compare the intensities of an ecotype in which deletions are to be called. Deletion calling based on this kind of comparisons was described in [4] where expression arrays were used which lack mismatch probes. The problem with this approach is large inter-sample variability or limited reproducibility: Even in non-polymorphic regions corresponding intensities from different ecotypes exhibit a great deal of variability (also discussed in [27]).

A general problem in all cases are spatial differences in intensities, technical artifacts that are sometimes caused by subtle variation in the handling of the arrays during hybridization and washing. Without replicates such spatial artifacts can be hard to corrected for. Thus we had to rely on methods robust enough to handle this kind of variation.

## 5.2 Fundamental difficulties

Assuming that a method can cope with large variance in intensities, still conceptually some problems have to be solved. As mentioned in the section on applications of the $k$-mer data, repeated $k$-mers causing cross-hybridization have a serious potential to interfere with deletion calling. This was already noticed in [25] and it has been recommended to exclude regions with strong likelihood of cross-hybridizing.

If inter-sample comparisons to the reference Col-0 are made, it comes at a cost. Regions where the reference intensities are too weak or too noisy have to be excluded from deletion calling and treated as missing data. As previously described, roughly 20 % of all Col-0 sites failed, i.e. there the maximum intensities do not come from the reference probe. Since only a fraction of them coincides with repeated $k$-mers, a considerable number of additional sites will have to be left aside. Moreover, there are sites where intensities and quality scores are so low that reliable comparisons to them cannot be made either.

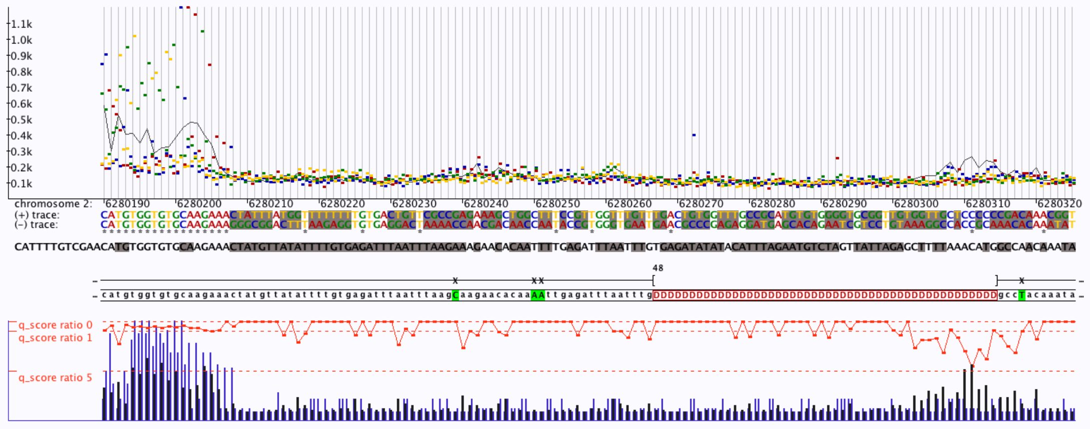

**Figure 5-1**
A 48 bp deletion in the ecotype C24 found by dideoxy sequencing [23]. Deleted bases are marked by red 'D's. Intensities of C24 are plotted as colored squares in the upper panel. They obliterate the maximum intensities of Col-0 (black line) almost everywhere. The probes having maximum intensities in C24 display a random pattern, as (+)-strand and (−)-strand disagree at the majority of sites (positions in the colored sequences that are shaded in grey). Consequently, also quality scores are very low. The reference failed at many sites as well (positions in the black reference sequence that are shaded in grey). Reference mean quality scores are depicted by black bars in the lower panel. The lower panel also shows peach-colored quality ratios (explained in section 5.3 on 'Deletion scores').

A deletion that is practically impossible to detect on that account is shown in figure 5-1. Note that the segment displayed only contains three sites with $k$-mer matches, none of them in the deletion (not shown).

A third type of problematic cases are regions that are very polymorphic, where there are many SNPs within a short distance to one another, often interspersed by small insertions and deletions. This is generally a challenge for SNP calling, because in cases where SNPs are closer than 12 bp to each other, no probe on the array will perfectly match the interrogated DNA sequence in this ecotype. Therefore such regions result in a depression of the intensities in a longer range so that the pattern resembles a deletion. We have not been able to find any detailed discussion of this problem in the literature, probably because in most resequencing experiments SNP density is found to be low enough that this phenomenon does not have a major impact [27] (With an average distance between adjacent SNPs of 1871 bp in human this is obviously a minor issue in [12].) However, Magnus Nordborg's dideoxy sequences [23] draw another picture for the *Arabidopsis* ecotypes. From the sequenced fragments the average SNP distance cannot be computed directly, since these fragments are on average 583 bp long and if a fragment contains fewer than two SNPs the distance to adjacent SNPs cannot be calculated (only a lower bound). For fragments with two or more SNPs the average SNP distance per ecotype varies between 50 and 57 and a considerable fraction of 44-51 % of these SNPs lie within a distance of $\leq$ 12 bp to the nearest neighboring SNP. If indel polymorphisms are also considered, the average distance between adjacent polymorphic features is 45-52 bp and 47-55 % of these features are at a distance of $\leq$ 12 bp to the nearest adjacent polymorphism. Of course this is an underestimation of the true distances as it is only based on more polymorphic fragments. 21-36 % of all fragments contain more than one SNP, 24-39 % contain more than one polymorphism including indels, but the majority of 88-93 % of all SNPs (90-94 % of all polymorphisms) lie on these more polymorphic fragments.

An illustration of this kind of problem is given in figure 5-3. This might seem like an extreme example and clearly is not representative of the majority of sequenced fragments.

However, it is not a clear outlier, but just one of several highly polymorphic fragments found in every ecotype in Magnus Nordborg's data.

Based on these observations we decided not to attempt to call smaller deletions that are more likely to be spurious predictions caused by several polymorphisms close to each other. We expect this problem to be less serious for DNA stretches of several hundred base pairs. A second reason for limiting our search to longer deletions is the necessity to confirm our calls by conventional sequencing experiments. Therefore it appears more useful to predict fewer deletions, but with higher confidence, for which there are resources to sequence the breakpoints.

Unfortunately, the dideoxy sequencing data only comprises a handful of examples of deletions longer than 100 bp (for all ecotypes there are 12 deletions at least 50 bp long of which 6 are actually longer than 100 bp, in Ler-1 there is only a single deletion longer than 50 bp). These are hardly sufficient as a training set for supervised learning approaches or Hidden Markov Models (HMMs). A histogram of deletion length is shown in figure 5-2. Deletions that are common to several ecotypes are counted only once.

For one ecotype, Ler, a low coverage sequencing project has been undertaken by Cereon Genomics, a subsidiary of Monsanto Company. Approximately 2-fold coverage was achieved resulting in roughly 50 000 contigs that covered about 70 % of the genome at the nucleotide level. This project also resulted in a list of SNPs and indel polymorphisms, discovered in Ler [14]. This list contains 747 indels larger than 100 bp, but unlike SNPs, the quality of these indel polymorphism has not been assessed by further sequencing efforts and especially for larger indels the error rate is expected to be considerable. We decided to manually curate these data and to use some of the larger deletions in Ler to develop a deletion calling algorithm. Since we had no data for other ecotypes we focused on Ler-1 (although it is not completely clear whether the sequenced reads are from the same Ler accession whose DNA was used for the resequencing).

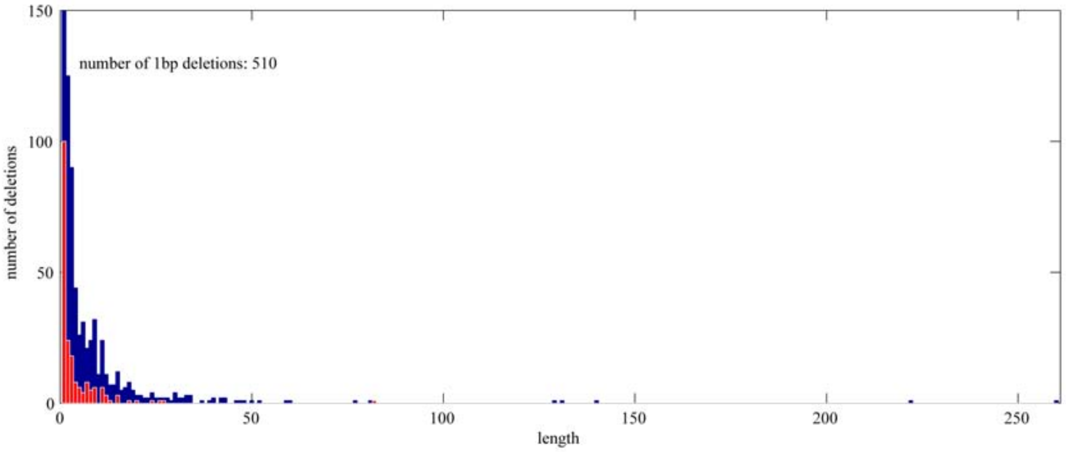

**Figure 5-2**
Histogram of deletion length from dideoxy sequence fragments. Blue bars – deletions found in at least one ecotype (1 bp deletions are truncated) deletions that are common to several ecotypes are counted only once. Red bars – deletions found in Ler-1.

]

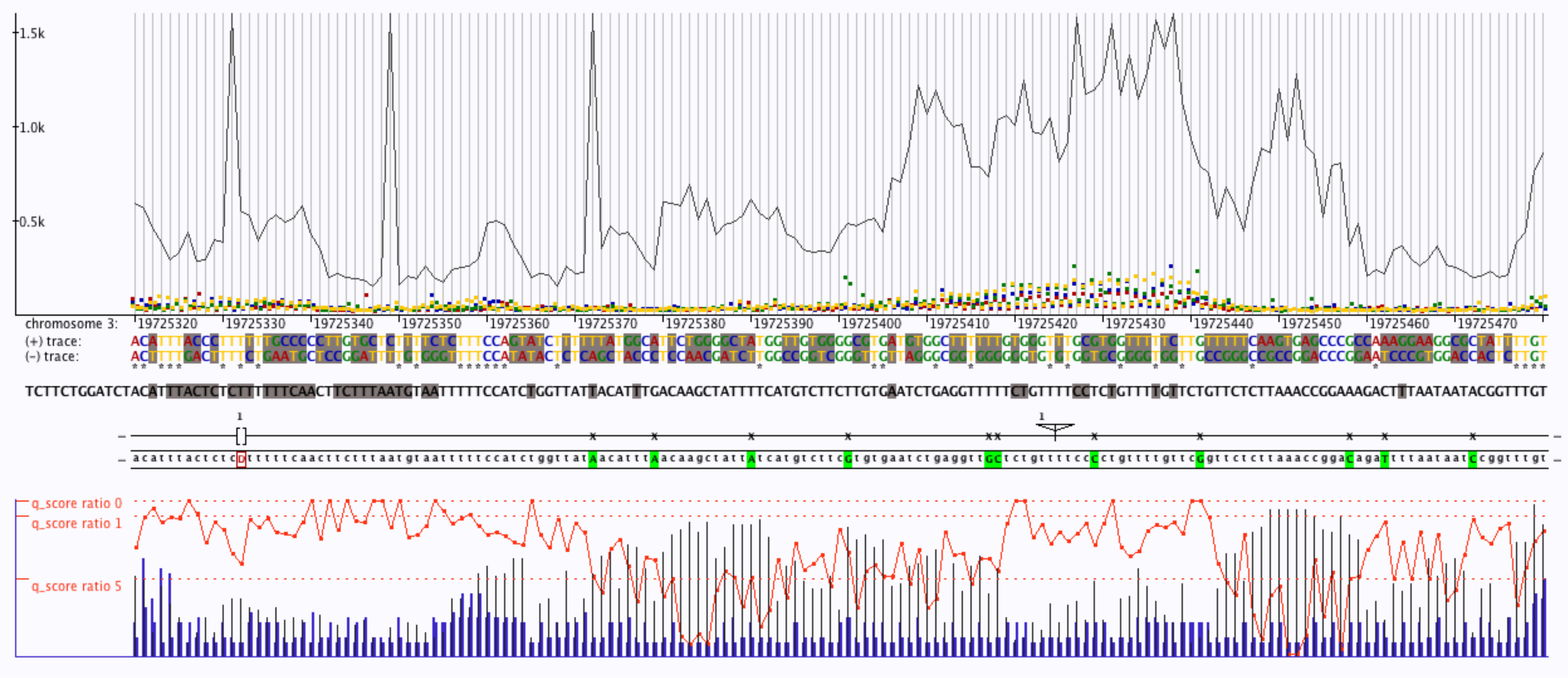

**Figure 5-3**
A region of 160 bp with a high level of sequence polymorphism in the ecotype C24. In the whole region intensities (colored squares) are much lower than in Col-0 (black line in the upper panel), most likely due to 11 SNPs (highlighted in light-green) and 2 short indels (red 'D' and black triangle) which are in most cases close enough to one anther to make the resequencing impossible in this local environment. The agreement between both strands in C24 and to the reference is virtually random. The quality scores for C24 (blue bars) are at minimum almost everywhere while in Col-0 a considerable number of sites displays high quality scores (average quality scores shown as black bars in between the blue bars). (Peach-colored) quality ratios are drawn in the lower panel upside down, so the highest dots correspond to 0 ratios (see axis labels on the left). Only two $k$-mers in this region are repeated (data not shown).

## 5.3 Deletion scores

As elaborated above, intensity values show large variability that complicates deletion calling. However, this variability is reflected in quality scores to a much smaller extent making them more useful for deletion calling than intensities.

We use two types of deletion scores, a quality-ratio score and a mismatch score which are explained in the following. A third measure, perfectly conserved words is also introduced in this section.

### Quality-ratio score

Since we decided to call deletions based on comparisons between the reference Col-0 and a target ecotype, our first score is simply the ratio between the Col-0 quality score and the target quality score. In order to avoid high scores at positions where Col-0 resequencing did not work well, the Col-0 quality score was set to 0 unless the sum of the quality scores of both strands was $> 6$.

We define the quality-ratio score $s_{QR}$ at position $p$ as:

$$n = Col.q^{+}(p) + Col.q^{-}(p)$$

$$s_{QR}(p) = \begin{cases} 0 & \text{if } n \leq 6 \\ \frac{n}{t.q^{+}(p) + t.q^{-}(p)} & \text{otherwise} \end{cases}$$

where $Col.q^{+}(p)$ is the quality score for the forward strand in Col-0 at position $p$ and $q^{-}$ the quality score for the reverse strand, similarly for the target ecotype $t$.

To increase sensitivity and specificity to a reasonable level, information from probes in a local neighborhood is combined. A combination of scores under a window has also been proposed in [25].

To this end we use a sliding window approach. In order to determine whether position $p$ is likely to be deleted, we also consider $(w-1)/2$ positions upstream and downstream of $p$, where $w$ is the length of the window. In this window we compute the median of all $s_{QR}$ values and assign it to position $p$. This median score is denoted by $\hat{s}_{QR}(p)$. Median filtering is preferred over averaging because the median preserves sharp transitions better than the mean (this can be seen for example in image processing quite nicely), making it easier to detect the ends of a deletion.

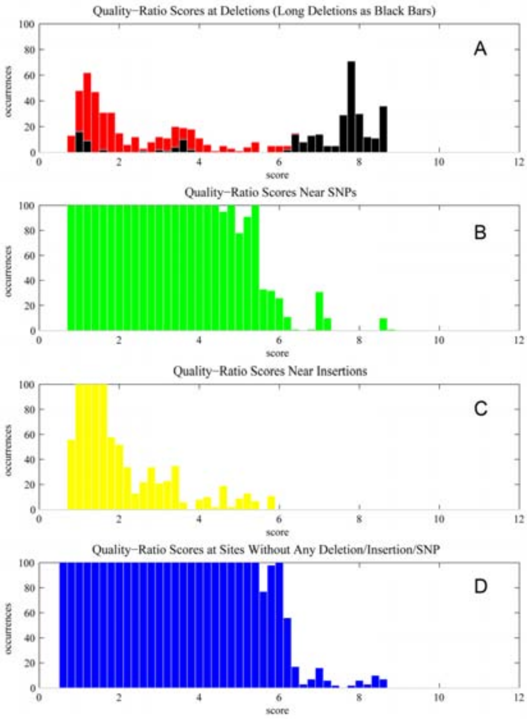

**Figure 5-4**
Truncated histograms of quality-ratio scores in Br-0 in a window of size 101 bp partitioned according to polymorphisms.

(A) scores in deletions (red bars) and in deletions which are at least 25 bp long (black bars).

(B) scores in a 12 bp neighborhood of a SNP.

(C) scores in a 12 bp neighborhood of an insertion.

(D) scores in conserved 25-mers.

As the only longer deletion in Ler-1 in Magnus Nordborg's data lies in a repeat, we evaluated the sliding window method on another ecotype, Br-0. We calculated all median quality-ratio scores on the sequenced fragments using a window of 101 bp. We then partitioned the scores into bins and counted the positions at which a certain score appeared. These positions were also partitioned into ones that are deleted in Br-0 (or are inside a deletion of length $\geq 25$), positions where one or more SNPs are within 12 bp upstream or downstream, positions where there is an insertion in the 12 bp neighborhood, and positions around which the whole 25-mer is conserved between Col-0 and Br-0, respectively. For each of these last four categories a histogram of the quality-ratio scores was made (shown in figure 5-4).

If the same number of sites is sampled from each of the four categories (this can be seen as approximating a probability density function), scores in long deletions are almost disjoint from scores around SNPs, insertions or at conserved sites. But in this context sampling is misleading as deletions are very rare events compared to SNPs and especially compared to conserved sites. Thus, there is some overlap between the tails of the histograms of SNPs and conserved sites on the one hand and the histogram of long deletions on the other hand. Nevertheless these histograms show that longer deletions can be recognized with quality-ratio scores. Detection of deletions shorter than 25 bp, however, is impossible with the kind of scores proposed here.

### Mismatch score

The second score that we use is based on maximum intensities and measures how often the maximum intensity in the target ecotype is observed at another probe than expected from the reference sequence. In order to correct for problematic sites in Col-0 we subtract the number of probe quartets for which in Col-0 the maximum intensity does not come from the PM probe.

The mismatch score $s_{MM}$ at position $p$ is defined as:

$$\begin{aligned} s_{MM}(p) \quad &= \quad (\arg\max t.Q^{+} \otimes S[p]) \\ &\quad + (\arg\ \max t.Q^{-} \otimes \overline{S}[p]) \\ &\quad - (\arg\ \max Col.Q^{+} \otimes S[p]) \\ &\quad - (\arg\ \max Col.Q^{-} \otimes \overline{S}[p]) \end{aligned}$$

where $t.Q^{+}$ denotes the probe quartet of the intensities $I$ from the forward strand: $x \mapsto I_x$, $x \in \{A, C, G, T\}$ in the target ecotype $t$, similarly for the probe quartet $t.Q^{-}$ of the reverse strand and the reference ecotype $Col$.

$S[p]$ denotes the nucleotide at position $p$ in the reference sequence, $\overline{S}[p]$ its Watson-Crick complement and $i \otimes j = 1$ if $i \neq j$, 0 otherwise.

Again, because mismatch scores are not very informative for single positions, we use a window of the same size $w$ as for quality-ratio scores. Position $p$ is then assigned a mismatch score

$$\hat{s}_{MM}(p) = \frac{1}{2w} \sum_{i\,=\,-(w-1)/2}^{(w-1)/2} s_{MM}(p+i).$$

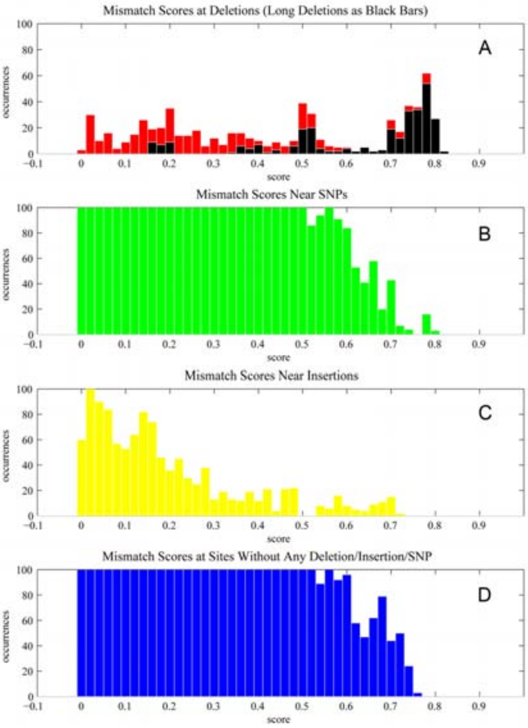

**Figure 5-5**
Truncated histograms of mismatch scores in Br-0 in a window of size 101 bp partitioned according to polymorphisms.

(A) scores in deletions (red bars) and in deletions which are at least 25 bp long (black bars).

(B) scores in a 12 bp neighborhood of a SNP.

(C) scores in a 12 bp neighborhood of an insertion.

(D) scores in conserved 25-mers.

Such a score is considered because in deleted regions one expects random intensity maxima, resulting in high mismatch scores, given that Col-0 worked well. In a simple random model one would expect scores close to $\frac{3}{4}$, assuming equal nucleotide frequencies, independence between strands and no Col-0 mismatches. Scores in conserved regions are expected to be much lower, otherwise resequencing with tiling arrays would be impossible.

A nice property of the mismatch score is its independence of scale, i.e. even in regions where the inter-sample variability of intensities is large, there might be a noticeable effect also on quality scores. However, a mismatch score is expected to be most insensitive to this. The histograms for mismatch scores, computed with the same method as described above are shown in figure 5-5. For mismatch scores the overlap of the histograms of deleted sites to other sites appears to be a little larger in Br-0 than for quality-ratio scores. Nevertheless, when the same comparison is made for other ecotypes, it is observed, that mismatch scores are much more consistent across different ecotypes. The separation of quality-ratio scores of Br-0 is unmatched in other ecotypes, although this can partially be explained by a lack of longer deletions in non-repetitive regions in almost all other ecotypes.

To determine a good threshold for these scores, above which they truly indicate a deletion with high likelihood, the dependance of sensitivity and specificity on such a threshold was evaluated. At this stage we also examined different window sizes and came to the conclusion to use a window of length 101, since with smaller window sizes specificity decreases rapidly, while the performance of windows longer than 101 bp is not better either, most likely because the number of sites in deletions longer than 100 bp is also very limited in our data set. A slight improvement can be seen though, when instead of the median the first quartile of the quality ratios in the window is evaluated. A graph that shows how specificity, i.e. the fraction of true positives among all positively predicted positions, and sensitivity, i.e. the fraction of true positives among all positions in long deletions, depend on score thresholds is shown in figure 5-6. Long deletions in this context are those of at least 25 bp. Here it seems that quality-ratio scores are more specific and sensitive, but this observation cannot be made for all ecotypes. Indeed, mismatch scores are much more consistent in different ecotypes. Therefore a stringent threshold of 0.72 was used for mismatch scores, for quality-ratio scores we used a relaxed threshold of 3.8 in order to avoid very low sensitivity in more problematic ecotypes. As our deletion calling algorithm is based on the conjunction of both scores, specificity can be controlled through the mismatch score threshold.

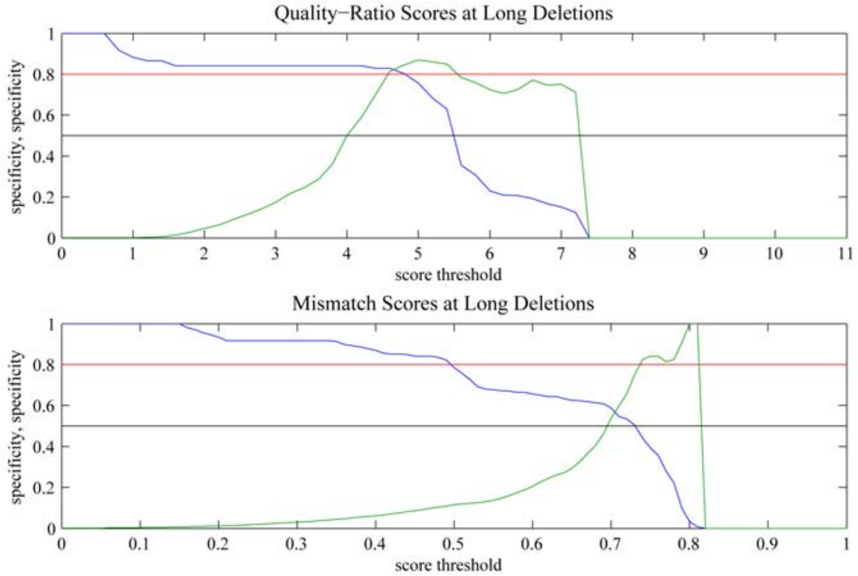

**Figure 5-6**
The dependency of sensitivity (blue line) and specificity (green line) on score thresholds (x-axis) in Br-0. If for example a quality ratio score threshold of 4.6 is used, the detection of sites in long deletions has both sensitivity and specificity above 80 %. However if 80 % specificity is required for mismatch scores too, it is not even possible to detect 50 % of all sites in long deletions.

### Perfectly conserved words

A *perfectly conserved word* of length $l$ is simply $l$ sites *in a row* where for both strands the probe with the maximum intensity matches the reference. For example in figure 5-3 the only perfectly conserved word of length 6 is where the reference sequence as well as both strands read TTTCCA. Long perfectly conserved words are expected to be very rarely found in deleted regions (in non-repetitive DNA), while other regions are expected—and observed—to contain longer perfectly conserved words. This is even true for regions with many SNPs and small indel polymorphisms as they are rarely equally spaced. We only consider words that are at least 3 bp long, for smaller ones are likely to be random events. We do not use perfectly conserved words in the same way as quality-ratio scores and mismatch scores, and in the following only quality-ratio scores and mismatch scores are referred to as deletion scores.

## 5.4 A seed and extend heuristic to call large deletions

Our deletion calling algorithm is based on finding longer stretches where quality-ratio scores and mismatch scores simultaneously exceed the threshold and indicate a large deletion. Such stretches are called seeds. They are processed further in order to define where the deletion most likely begins and ends. A seed and extend approach in this context does not have the purpose to speed up the computation, but reflects that we first detect parts of deletions that can be found even with very stringent criteria and among which only a very small number of false positives is expected. But as these seeds will be underestimations of the actual deletions in most cases, we have to apply more realistic, relaxed criteria to find more likely end points.

Before each step is described in detail, the workflow of our deletion calling is outlined here:

1. The pseudochromosome sequence is filtered to exclude sites with $k$-mer matches and where Col-0 did not work.
2. Seeds are computed on the filtered sequence.
3. Seeds which are close to one another are merged if the region in between is likely to be deleted as well.
4. Deletion boundaries are estimated based on quality-ratio scores, mismatch scores and perfectly conserved words.
5. Overlaps between predicted deletions are resolved.

### Filtering

The necessity to restrict deletion calling to sites of unique $k$-mers has already been motivated. We also exclude all sites where Col-0 did not work sufficiently well to compare to them an other ecotype (all such sites are stored in the SQL table bad_position). After these positions are filtered out, deletion calling is done on this reduced version of the pseudochromosome, before the results are mapped back to the original genome coordinates. Thus if a deletion contains some sites with repeated $k$-mers, we will still be able to call it provided that the majority of deleted sites around them is unique.

### Computing seeds

In order to find seeds, we scan the whole filtered genome with a sliding window procedure. Sites at which both the quality-ratio score and the mismatch score exceed the threshold, are labelled as positives. These positives most often appear in clusters, but in many cases not every single site in such a cluster is positive. Such small gaps of negatives sites are ignored within a positive cluster unless they are longer than 10 bp. Clusters that contain at least 50 positive sites are kept as seeds, smaller ones are put aside. This cutoff is somewhat arbitrary but it turned out that a reasonable number of predicted deletions originated from these seeds.

### Merging seeds

As the thresholds for seed detection are conservative, in many cases larger deletions (> 500 bp) contain more than one seed. So we attempt to merge seeds in the next step. If two adjacent seeds more likely indicate a single deletion than two separate ones, the deletion scores in the gap are expected to be high, at most positions close to or above the score threshold. To decide whether seeds $S_1$ and $S_2$, divided by a gap $G$, are to be merged, the following formula is used

$$\frac{|S_1|\,t + |S_2|\,t}{|G|\,t - \sum\limits_{p \in G} \min(t, \hat{s}(p))} \geq K$$

where $|S_1|$ is the length of $S_1$ (similarly $|S_2|$ and $|G|$), $t$ the score threshold, $\hat{s}(p)$ a deletion score ($\hat{s}_{QR}(p)$ or $\hat{s}_{MM}(p)$) at position $p$ and $K$ a constant to decide whether or not two seeds will be merged. For merging this inequality has to hold for both quality-ratio scores and mismatch scores. This formula is illustrated in figure 5-7 in which the green areas correspond to the numerator, the red area to the denominator in the above formula. We used $K = 2$, which turned out to be rather high (see results for a discussion).

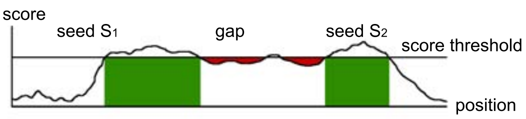

**Figure 5-7**
Illustration of seed merging. Seeds $S_1$ and $S_2$ are merged if the green areas together have at least twice the size of the red area.

### Estimating deletion boundaries

At this point we have got seeds resulting from a conservative estimation. In cases that such seeds are observed to reside inside known deletions, they tend to underestimate the actual length drastically (see results and discussion). This leads to the question how to define the break points (begin and end) of the deletion prediction more accurately. Taking a straightforward approach, one would just scan the areas upstream and downstream from the seed until one reaches high quality scores and few mismatches which indicate that the break point has been passed. Regions with few mismatches and high quality scores result in low deletion scores, consequently one would scan for low scores outside the seed.

However, this is more complicated than it may seem, primarily because of missing information. For example the seed can be next to a region of non-unique $k$-mers (and in some cases these region can be very long, for instance the 270 kb insertion of the mitochondrial genome). In some cases many sites around a seed have low quality and many mismatches in both Col-0 and the target ecotype. Consequently, it might be impossible to find the real breakpoints. Hence, instead of predicting a single break point, an interval is estimated, in which the actual breakpoint is likely to reside.

First of all, we have to define a second pair of lower score thresholds, below which scores indicate that this position is unlikely to be deleted. For quality-ratio scores this threshold is set to 2.5 and for mismatch sores to 0.35, which is roughly half as high as the upper thresholds.

Now a scanning step follows, in which the rightmost position left of the beginning of the seed with deletion scores both below the lower thresholds is found as well as, in an analogous way, the leftmost position right of the deletion end with scores below the lower thresholds. In the following, the part of the predicted deletion corresponding to the seed will also be called “core” whereas the intervals around it will be called “boundaries”. This is illustrated in figure 5-8.

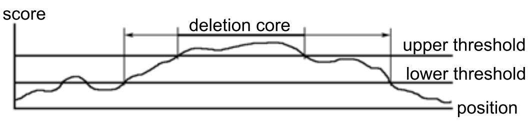

**Figure 5-8**
Illustration of a predicted deletion. It consists of a 5’ boundary where scores are increasing between lower and upper threshold, a core which corresponds to the seed and a 3’ boundary where the score is decreasing between upper and lower threshold.

This first estimate of the boundaries is refined by two further procedures. In a first step, illustrated by cartoons in figure 5-9, the boundary interval is scanned again but with a smaller window. Consider the 5’ boundary: If inside the original boundary interval the leftmost intersection point between the score function and the upper threshold is still right of the rightmost intersection point with the lower threshold, the boundary interval is refined to the two respective intersection points. The 3’ boundary is just a mirror image of the 5’ boundary and is refined with an analogous procedure. This can also enlarge the core while the length of the boundary decreases. If a boundary can be refined, the window size will be reduced again. Refinement is done iteratively until the window size is below 5 or no further refinement is possible. In each step the window is shortened by $\frac{1}{5}$ of its length. Examples for a successful boundary refinement and a case where refinement is not possible are depicted in figure 5-9.

In some cases the refinement procedure results in very sharp boundaries which are accurate estimations of the true boundaries. In other cases, however, the boundaries are very large, often because of missing information.

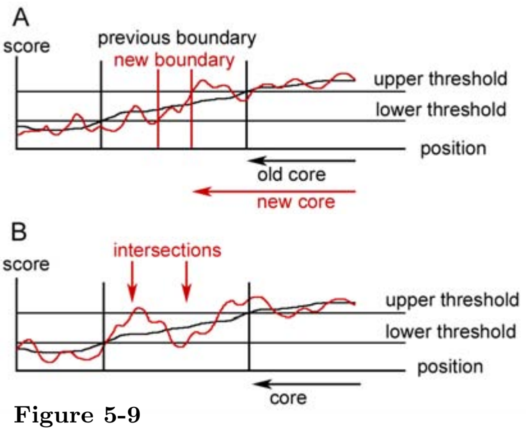

**Figure 5-9**
Cartoon illustrations of the refinement of the 5' boundary. The black line indicates scores computed with the original window size. A reduced window size typically produces a more ragged score line, which is shown in red.
**(A)** An example of successful boundary refinement. Here the leftmost intersection between the red score line and the upper threshold is right of the rightmost intersection with the lower threshold. Note that the deletion core is extended, while the boundary is contracted.
**(B)** In this case refinement is stopped since the intersection points between the red score line and the thresholds are ambiguous. This can be detected considering the order of the rightmost intersection with the lower threshold and the leftmost intersection with the upper threshold. A negative interval size would result from this instance, therefore boundary refinement is terminated.

A second step of boundary refinement is based on perfectly conserved words. The boundary intervals are scanned again, starting from the core. Thereby the core region is extended into the boundaries until a perfectly conserved word of length 6 or longer, several smaller perfectly conserved words close to each other or a region of missing information is encountered. Perfectly conserved words of length $n \in \{3, 4, 5\}$ lead to the termination of core extension if they are within a distance of $n^2$ bp to the next word. If among the 25 next sites 80 % or more are sites of missing information, core extension is also terminated.

After this it is checked whether boundaries contain a long perfectly conserved word. In case that inside a boundary interval a word of length $n \geq 5$ is closer than $n^3$ bp to the end of the boundary, the deletion boundary is shortened such that it does not include this word any longer. If shortening was not necessary, the boundary is extended instead. For this extension the region upstream of the 5' boundary is scanned, increasing the deletion size, until either a perfectly conserved word of length 10 or more is found or more than one word of length $n \geq 5$ within a distance of $n^2$ to each other. Similarly the 3' boundary is extended downstream.

The use of these perfectly conserved words is obviously more or less arbitrary. Nevertheless it is useful to avoid that boundaries contain short regions that produce a clear hybridization signal, but have been blurred under the sliding window. The second purpose of the refinement with perfectly conserved words is to obtain a better ratio of core to boundary size, in other words to obtain more compact deletion predictions where possible.

### Resolving overlaps between predicted deletions

As deletion seeds are processed individually, it sometimes happens that predicted deletions overlap. But as overlapping deletions are biologically impossible and they cannot be verified by sequencing, we attempt to merge such overlapping deletion predictions. Two adjacent predicted deletions are merged, either if their cores overlap or if cores can be merged with the procedure already used to merge seeds. In this final merging step a more stringent constant $K = 5$ is applied to decide whether two deletions are to be merged or not. This appears to be necessary because after boundary estimation deletion cores are larger than the seeds they originated from, so that they tend to fulfill the inequality for merging more often, even if the deletion signal in the region between the cores is weak.

If merging is impossible with this criterion, the overlapping predictions are put aside in order to obtain a set of predicted deletions that is free of contradictions.

## 5.5 Results and evaluation

Results on deletion calls are presented before some indirect methods to evaluate these calls are discussed.

Table 5.1 summarizes the output of our deletion calling algorithm. In addition to the number of predicted deletions, the total number of bases in their cores and boundaries is also given. The last column gives a hint how compact the predictions are. If cores are large compared to the boundaries, this value is high. According to this criterion, deletion calling in Lov-5 produced the most compact predictions.

Concerning the number of predicted deletions, there are two outliers, Cvi-0 and Est-1, whereas for all other ecotypes between 600 and 850 deletions are predicted. The predictions for Cvi-0 are consistent with observations from the sequenced fragments. There it is evident that Cvi-0 is most diverged from Col-0, having 27 % more polymorphisms than the second most polymorphic ecotype.

However, the number of predicted deletions for Est-1 is most likely a technical artifact, as the sequenced fragments show that it has a comparably low number of polymorphisms, in fact the lowest number of deletions, but the difference to Cvi-0 is less than twofold, while for the predictions the difference is more than threefold.

A possibility to evaluate how plausible these predictions are, is a comparison between ecotypes. For simplicity, only deletion cores are compared here. For example 285 of 849 predicted cores in Ler-1 overlap with predicted cores in C24, 18 of them are predicted identically. On the nucleotide level 40 % of the deletion cores in Ler-1 are also predicted to be deleted in C24. A comparison between Ler-1 and Br-0 yields an overlap of 337 Ler-1 deletion cores with Br-0 containing 52 % of all nucleotides in Ler-1 deletion cores. 16 of the predicted deletion cores are identical between Ler-1 and Br-0 .

To a certain extent these values resemble the pattern of SNPs in the sequenced fragments. Ler-1 has 46 % of its SNPs in common with C24 and it has 43 % SNPs in common with Br-0.

**Table 5.1 Predicted deletions**

| ecotype | # deletions | total core length | total boundary length | proportion core/boundary |
|---|---:|---:|---:|---:|
| Bay-0 | 713 | 1 053 867 | 1 198 126 | 0.88 |
| Bor-4 | 601 | 725 557 | 937 013 | 0.77 |
| Br-0 | 758 | 1 065 389 | 1 294 414 | 0.82 |
| Bur-0 | 663 | 847 274 | 1 122 014 | 0.76 |
| C24 | 770 | 884 482 | 1 004 570 | 0.88 |
| Cvi-0 | 1 019 | 1 413 710 | 1 555 356 | 0.91 |
| Est-1 | 320 | 406 850 | 498 031 | 0.82 |
| Fei-0 | 674 | 942 816 | 1 088 730 | 0.87 |
| Got-7 | 610 | 799 245 | 1 066 782 | 0.75 |
| Ler-1 | 849 | 1 192 448 | 1 452 623 | 0.82 |
| Lov-5 | 737 | 1 118 765 | 1 088 989 | 1.03 |
| Nfa-8 | 801 | 1 143 879 | 1 414 009 | 0.81 |
| Rrs-10 | 605 | 818 190 | 995 484 | 0.82 |
| Rrs-7 | 696 | 962 922 | 1 502 504 | 0.64 |
| Sha | 774 | 1 228 239 | 1 508 522 | 0.81 |
| Tamm-2 | 770 | 1 142 890 | 1 299 966 | 0.88 |
| Ts-1 | 763 | 1 073 443 | 1 254 375 | 0.86 |
| Tsu-1 | 628 | 823 264 | 1 044 783 | 0.79 |
| Van-0 | 719 | 1 040 583 | 1 316 483 | 0.79 |

When our deletion calls are compared to the larger deletions in the sequenced fragments of Magnus Nordborg's data, only one large deletion was detected, for which there is a fragment in Br-0 that contains a 260 bp deletion (in fact the largest one in the whole data set). In addition to the correct prediction in Br-0, large deletions in the same region are also predicted for the ecotypes Est-1, Ler-1, Lov-5 and Sha. For these four ecotypes the corresponding fragment could not be amplified and sequenced, while for most of the other ecotypes (except for Bor-4 and Tsu-1) the fragment could be sequenced. This suggests at the deletion predicted in the other ecotypes might be real, which would explain why PCR amplification failed in these ecotypes.

When the boundaries of the deletion predicted in Br-0 are compared to the actual ones, they turn out to be reasonably accurate. The actual length is overestimated by 45 and 17 bp at the 5' and 3' end, respectively. This overestimation is most likely caused by four SNPs around the deletion that also lead to intensity depressions flanking the actual deletion signature.

The other five deletions in the sequenced fragments which are longer than 100 bp are not detected, and in three cases repetitive $k$-mers cover the majority of sites in these deletions. Visual inspection reveals clear deletion signatures in the resequencing data in two cases, suggesting that these deletions could have been found with a more sensitive search.

In order to assess the quality of our deletion predictions, we systematically exploited the fact that there are many fragments in Magnus Nordborg's data that could not be amplified in some of the ecotypes, similar to the example discussed above. Obviously, PCR amplification failure is only a weak indication that (at least one end of) the fragment is deleted. In addition to experimental errors, there are other reasons why PCR amplification can fail, for instance a SNP near the 5' end of the primer binding site. Furthermore, PCR amplification success is most likely negatively correlated with the local density of polymorphisms, as the chance that the priming site is conserved decreases in highly polymorphic regions. But such highly polymorphic regions might also cause spurious deletion calls. Thus the concordance between deletion predictions and missing fragments provides only weak evidence that deletion calls are true positives and it obviously does not allow to evaluate the accuracy of the deletion boundaries. However, if a predicted deletion contains a fragment which could be sequenced, this is a strong indication that the deletion call is a false positive.

In summary, there are 13 predicted deletions overlapping to 13 of the sequenced fragments for the same ecotype (excluding Van-0), one of them is the correct prediction just described. A predicted deletion in Sha overlaps with a single fragment in which only 7 bases could be determined unambiguously. A similar case is observed for a prediction in Nfa-8 where the sequenced fragment contains only 14 unambiguous bases. Nonetheless, after excluding these uncertain ones the remaining 10 cases are most likely false positive predictions. 2 such overlaps are found in Sha, 2 overlaps as well in Bay-0 and 2 in RRS-7. A single overlap is found in each of Cvi-0, Lov-5, RRS-10 and Tsu-1.

The following table summarizes the other cases that a fragment is missing in a given ecotype (excluding Van-0), but present in Col-0 (in the column with the heading "# missing fragments") and how many of the missing fragments overlap to the core of a predicted deletion (in the column with the heading "# overlaps").

**Table 5.2 Deletion predictions evaluated on missing fragments**

| Ecotype | # missing fragments | # overlaps |
|---|---|---|
| Bay-0 | 47 | 10 |
| Bor-4 | 54 | 3 |
| Br-0 | 9 | 60 |
| Bur-0 | 88 | 6 |
| C24 | 61 | 7 |
| Cvi-0 | 68 | 13 |
| Est-1 | 61 | 4 |
| Fei-0 | 81 | 6 |
| Got-7 | 52 | 8 |
| Ler-1 | 63 | 15 |
| Lov-5 | 66 | 11 |
| Nfa-8 | 62 | 9 |
| Rrs-10 | 84 | 9 |
| Rrs-7 | 100 | 8 |
| Sha | 83 | 15 |
| Tamm-2 | 46 | 11 |
| Ts-1 | 66 | 9 |
| Tsu-1 | 65 | 6 |

Note that even though only a small fraction of missing fragments can be explained by our deletion predictions (which is not unexpected considering the limited sensitivity of deletion calling and other reasons for PCR failure), the number of deletion predictions overlapping to Col-0 fragments that are missing in the ecotype with the deletion is about 17 times higher than the number of overlaps between predicted deletions and successfully amplified fragments in the deleted ecotype (as there are 10 present fragments falsifying deletion calls, while 168 deletion calls are found at sites where the sequenced fragment is missing). This indicates that the specificity of our deletion calling algorithm can be expected to be in an acceptable range.

As mentioned earlier in this chapter, there is a large set of reads from the sequencing of Ler undertaken by Cereon. We attempted to map large indel polymorphisms from the list of polymorphisms published by Cereon onto the Col-0 genome sequence with a protocol similar to the one recommended at The Arabidopsis Information Resource (TAIR)[22] where this data set is hosted.

Several of the deletions contained in that list are difficult to map to the Col-0 sequence, for one has to use 20 bp flanking regions given there and try to find their genomic location in Col-0. Unfortunately, many of these 20-mers cannot be unambiguously placed at approximately the right distance and in the right orientation in the genome. The ones for which this is possible though, are compared with our resequencing data. In the light of our data some of them appear to not be present in the resequenced Ler-1 genome. A good way to assess this, is to count the number of perfectly conserved words in the deletions from the Ler sequencing data. For the following evaluation we only consider deletions for which the number of perfectly conserved words of length 10 or longer is smaller than their length divided by 500. (When perfectly conserved words are computed, obviously sites where the $k$-mer is not unique are to be excluded.) This means that in a deletion of 1001 bp we allowed at most two perfectly conserved words of at least 10 bp (which is still unlikely to occur in a real deletion).

The deletions from the Ler sequence are further filtered for massive repeats. All deletions that contain repetitive $k$-mers at more than 50 % of the sites are not included in a comparison to our predicted deletions. This is done because we want to evaluate our deletion prediction method as well as the accuracy of the predicted boundaries in regions of the genome in which one has a realistic chance to make reliable deletion calls. In highly repetitive regions this is not possible as too much information is missing.

It has to be mentioned that the comparison of our predictions to these deletions is not an entirely fair test as a few of these deletions were used to tune our algorithm albeit not in a systematic way.

136 deletions from the Ler sequencing data are at least 100 bp long, contain less than 50 % repetitive $k$-mers and the number of perfect words of length 10 bp or longer is smaller than their length divided by 500. The same criteria are met by almost all predicted deletions (828 of 849).

60 of the predicted deletion cores that meet the above requirements have an overlap of at least 75 % to a deletion from the Ler sequences. (A table containing these overlaps is found in the appendix). This number seems to be small compared to the number of predictions, but because the set of known deletions in the Ler sequences is incomplete, it is not useful to evaluate specificity. Nevertheless it shows that in regions of the genome with a moderate level of repeats, roughly one half of the deletions from the Ler sequences can also be predicted from the resequencing data. This indicates that sensitivity for deletions of 100 bp and more is about 0.5 in such parts of the genome.

The 60 sequenced deletions that are detected, and those 76 that are not, are compared in order to assess systematic biases of our search method. It does not reveal a bias against the relative number of repeated $k$-mers among these deletions (which do not contain more than 50 % repetitive sites after filtering, 12 % on average). However, our predictions are drastically biased towards longer deletions: The average length of all 136 sequenced deletions is 1513, the average length of the detected deletions is 2435, while the undetected ones are

[22] http://arabidopsis.org/Cereon/help.jsp, on www.arabidopsis.org, Oct 25, 2005

on average only 786 bp long. This is neither completely unexpected nor undesired. One can reduce this bias in order to increase the sensitivity for shorter deletions by adjusting the minimal seed length—although one might have to accept more false positives. Histograms of deletion length are shown in figure 5-10.

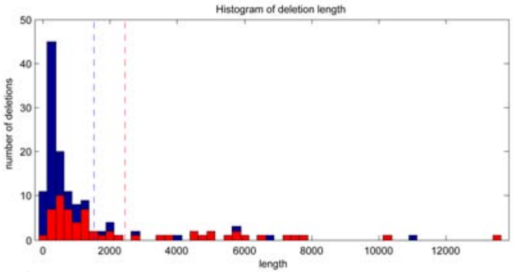

**Figure 5-10**
Histogram of deletion length of all 136 filtered deletions from Ler sequence reads (in blue). In the foreground the histogram of deletion length of those that can be predicted is superimposed (in red). This suggests that our deletion calling method is more sensitive the longer the deletions are.

With a second evaluation it is assessed if we systematically underestimate or overestimate the true deletion length and how accurate our boundary predictions are in general. To be able to address this question one needs a 1:1 relationship between predicted and sequenced deletions. However, in three cases we do not observe this. Instead, these three sequenced deletions contain two (non-overlapping) predictions each. All three sequenced deletions are relatively large, 4747, 6089 and 6623 bp, while the gaps between the predictions are comparably small: 291, 100 and 559 bp, respectively. 4 of 6 predicted outer boundaries are accurate, deviating at most 10 bp from the real breakpoints.

In the remaining set os 1:1 correspondences length deviations $\triangle l$ are computed as the difference between the predicted length and the length of the sequenced deletion. For predicted deletions there are actually two possibilities to measure the length, either between the start and the end points of the boundary or of the core. We will refer to the length deviation of the core as $\triangle l_C$, to the length deviation of the whole deletion including boundaries as $\triangle l_B$. To obtain relative length deviations $\triangle l/L$, we divide $\triangle l$ by the length of the sequenced deletion.

In order to assess the accuracy of our boundary predictions we compared all the predicted deletions that overlapped with deletions from the Ler sequences and computed offsets between the start points and end points of a pair consisting of a sequenced deletion and a deletion prediction. An offset is simply the difference between the position of the predicted and the actual breakpoint. Offsets have negative or positive integer values depending on whether the break point is predicted upstream or downstream of the actual site. An illustration of these offsets is given in figure 5-11. In this ideal example the core is completely contained in the actual deletion and the boundaries extend only little beyond the actual start and end points.

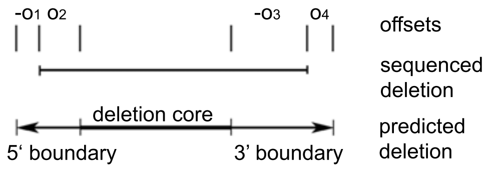

**Figure 5-11**
Offsets between predicted and sequenced deletion start and end points. The offset between predicted 5' boundary and actual start point, $o_1$, is negative in this example, $o_1 < 0$. $o_2$ is the offset between predicted core and actual start point, here $o_2 > 0$. Similarly for the 3' offsets, $o_3$ between predicted core and actual end point, here $o_3 < 0$, and $o_4$ between predicted 3' boundary and actual end point, here $o_4 > 0$.

A complete listing of the compared deletions is found in the appendix, only summary statistics are presented in table 5-3. This summary shows that deletion length including boundaries tends to overestimate the true length while it is still a more accurate estimate of the true deletion length than the length of the predicted core. As expected cores are usually smaller than the true deletions (more than 30 % on average).

For more than half of the deletion predictions the length deviates less than 10 % from the true length.

For 33 of 57 predictions the core is completely contained in the true deletion. With a tolerance of 25 bp on either side more than 90 % of the predicted cores are contained in the true deletion.

Both break points are estimated precisely (within 10 bp of the true breakpoints) in only 7 cases. However, at least one break point is very accurate in 31 cases. For roughly half the predictions, boundary end points are not more than 100 bp off, and in only 11 cases at least one end point deviates more than 500 bp from the true break point (which might still be regarded as a reasonable prediction for deletions longer than 5000 bp).

**Table 5.3 Deletion prediction accuracy**

| | |
|---|---|
| total number of 1:1 overlaps | 57 |
| average $\triangle l_B/L$ | 0.15 |
| average $\triangle l_C/L$ | -0.33 |
| average $\lvert\triangle l_B\rvert/L$ | 0.30 |
| average $\lvert\triangle l_C\rvert/L$ | 0.37 |
| number of predicted deletions with $\lvert\triangle l_B\rvert/L \leq 0.1$ | 30 |
| number of predicted deletions with $\lvert\triangle l_B\rvert/L \leq 0.2$ | 38 |
| number of predicted deletions with $\lvert\triangle l_B\rvert/L \leq 0.5$ | 50 |
| number of predicted cores with $o_2 \geq 0$ and $o_3 \leq 0$ | 33 |
| number of predicted cores with $o_2 \geq -10$ and $o_3 \leq 10$ | 44 |
| number of predicted cores with $o_2 \geq -25$ and $o_3 \leq 25$ | 53 |
| number of predicted deletions with $\lvert o_1\rvert \leq 10$ and $\lvert o_4\rvert \leq 10$ | 7 |
| number of predicted deletions with $\lvert o_1\rvert \leq 50$ and $\lvert o_4\rvert \leq 50$ | 21 |
| number of predicted deletions with $\lvert o_1\rvert \leq 100$ and $\lvert o_4\rvert \leq 100$ | 28 |
| number of predicted deletions with $\lvert o_1\rvert \leq 10$ or $\lvert o_4\rvert \leq 10$ | 31 |
| number of predicted deletions with $\lvert o_1\rvert > 500$ or $\lvert o_4\rvert > 500$ | 11 |

In conclusion, the majority—but certainly not all—of the predicted deletions overlapping with those from the Ler sequences appear to be reasonable estimates of true deletions. The comparison shows that in some cases also the boundaries are predicted very accurately, in most cases they are not completely off, while sometimes the deviation from sequenced deletions is large.

This emphasizes the need to confirm these predictions with biological experiments. The breakpoints of a few hundreds of our predictions are going to be sequenced. To do so, primers will be designed approximately 100-300 bp outside the boundary on both sides. If PCR amplification works, the approximate length of the amplified fragment can be determined by gel electrophoresis. If a length polymorphism can be shown by gel electrophoresis and the prediction is not a large overestimation of the real deletion, one can sequence the PCR product from both ends hoping that the traces overlap and thereby reveal the true deletion break points.

## 5.6 Conclusion and future goals

Provided that the number of false positives found by PCR amplification and sequencing is not too high, the deletion calling heuristic introduced here can be seen as a proof of concept demonstrating that at least larger deletions can be called using resequencing data. A drawback of this method is that it is not founded in statistics and it cannot give likelihoods or confidence values for deletion predictions.

Another limitation is its lack of sensitivity for smaller deletions of length 100-300 bp which might still be possible to call with a reasonable false-positive rate using more sophisticated methods. Predicted deletions having very long boundary regions are also unfavorable—for verification as well as for follow-up experiments.

It might be possible to address these problems with a state transition model that could be incorporated in a Hidden Markov Model (HMM)[23]. Such a model should be able to detect intensity depressions characteristic for sites where none of the probes matches the genomic sequence of a certain ecotype exactly. Obviously, such intensity patterns are also found around insertions or in regions with several SNPs closer than 12 bp to one another. This seems like a drawback, but being able to detect such regions in general could actually be an advantage, since SNP calling is very difficult in regions of accumulated polymorphisms and call rates tend to be especially low there. Thus, if one relies on the density of SNP calls,

[23] Deletion calling with HMMs appears to result in good predictions in the case of arrays originally designed for expression analysis (Justin Borevitz, personal communication)

one might get a completely wrong idea about the overall degree of evolutionary conservation in regions where only a few interspersed SNPs are called, paradoxically *because* the real SNP density is very high.

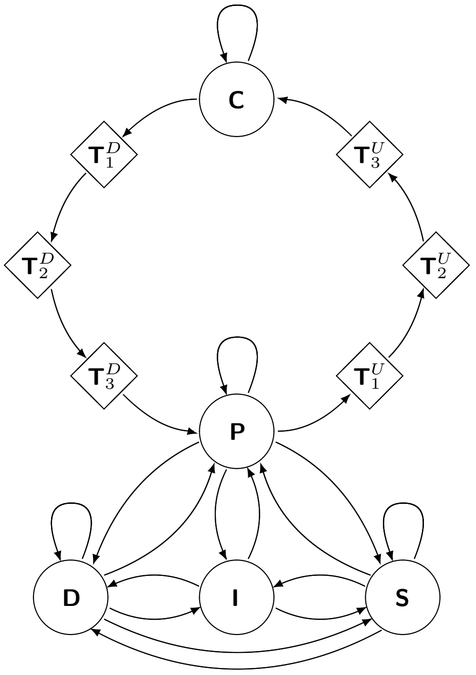

**Figure 5-12**
Topology of a state model for the detection of clustered sequence polymorphisms. Transition states have diamond shapes, all other states are depicted by circles. Transitions between the states are indicated by arrows. For a description of the different states see text.

The topology of such a state model for the resequencing data could look like the one presented in figure 5-12. It contains two main states, one modelling conserved sequences with normal hybridization intensity (**C**), the other one modelling depressed intensities characterizing sequence polymorphisms (**P**). From this second state there are transitions into states corresponding to actual single nucleotide (**S**), insertion (**I**) or deletion polymorphisms (**D**). As the transitions between normal intensity and depressed intensity at polymorphic sites are usually not abrupt but extend over a few nucleotides, there are also several states to model these gradual transitions ($\mathbf{T}^D$ states for decreasing intensities around polymorphisms, $\mathbf{T}^U$ for increasing intensities up to normal level). The actual number of these transition states can be varied a little in order to perfectly model the observed length of this gradual decrease or increase of intensities, respectively.

Associated with such a topology are probabilities for state transitions and for emissions of the actual nucleotide and of the intensities of the corresponding probe quartets. The training of such an HMM could be done on the more polymorphic ones among the sequenced fragments which in several cases contain dense clusters of SNPs and indels and apparently on those fragments that contain deletions (with a length of at least 25 bp). After training, test sequences together with intensities are threaded through the model in order to obtain the path, i.e. the sequence of states, with the highest probability for the given sequences and intensities. This path can be computed using the Viterbi algorithm.

Instead of an HMM, one can use the same state transition model, but replace the probabilities by real valued scores. These scores are optimized such that one obtains a large margin classifier. This means to maximize the score difference between the true path (the sequence of states defined by the polymorphisms in the sequenced fragment) and all false paths with a different sequence of states during the training on labelled (i.e. known) sequences. Instead of considering *all* paths one only uses a small number of false paths whose scores are greater than those of all other false paths, which is conceptually similar to support vectors that are close to the decision boundary. False paths with a score higher than all others can be efficiently computed by modifying the Viterbi algorithm such that it returns a certain number of paths with (almost) maximal score. Being able to assign scores to real valued intensities requires to learn a—preferably smooth—piecewise linear function that maps intensities to scores. With the same approach other features, e.g. quality scores or sequence properties like GC content, discussed in chapter 2, can be added to improve the performance of the classifier. This kind of classifier could also be modified into one that is trained on data from multiple ecotypes and then used to predict common polymorphisms.

The benefits from a learning algorithm incorporating this state transition model could be the ability to detect dense clusters of indel polymorphisms and SNPs which go undetected by SNP calling algorithms in most cases. This information would be helpful to get an idea of the overall density of polymorphisms, even if the classification result into SNP and indel positions was not completely correct. The prediction of many deletions, too small to be detected by the deletion calling heuristic, could be possible. Finally the ability to use resequencing data for the detection of insertions (and the distinction from SNPs with high confidence) would be novel.

# 6 Conclusion

For this diploma thesis a huge set of resequencing data from 20 *Arabidopsis* ecotypes has been analyzed. As the resequencing technique has been developed relatively recently, properties of these data had to be investigated prior to high-level analyses. To facilitate visual inspections a visualization tool has been implemented.

One of the key observations in the resequencing data is large variability in intensity values. Because of this, base calling is impossible for a considerable fraction of sites with extremely weak intensities. In part, intensity variability is caused by experimental conditions, in part by physicochemical properties of the probes.

Investigations of hybridization intensity and probe sequence revealed that there are sequence properties such as GC content and sequence entropy which are positively correlated to hybridization intensity. In part, the failure of probes with unfavorable hybridization characteristics can be explained by an excess of A/T nucleotides or self-complementarity which allows to form stable hairpins.

As a future goal one could attempt to predict hybridization intensity with support vector regression on the basis of probe sequences and their properties. The challenge for such a prediction would be to cope with large intensity variability and limited reproducibility of hybridization intensity. It would, however, be very useful as a means of normalizing intensities to correct for the huge variability across different wafers and to facilitate comparisons between ecotypes.

In order to assess the probability of cross-hybridization at a given site, a program has been written to systematically search for 25-mer sequences that occur multiple times in the *Arabidopsis* genome. It is based on a linear sort algorithm that can deal with a limited number of mismatches between two 25-mer sequences. The genome-wide search resulted in more than 900 million matches which allow to estimate the cross-hybridization potential of every probe on the resequencing arrays. These matches were successfully applied to reduce the number of false positive SNP calls computed by Perlegen and they have facilitated the development of new SNP calling algorithms which take cross-hybridizing probes into account. The collection of 25-mer matches has also made it possible to call large deletions. Without filtering for repetitive regions, predictions of large deletions would likely be split up into small fragmented ones in many cases. 25-mer matches are thus essential for deletion calling.

The value of this collection of 25-mer matches would be higher, though, if different types of mismatches could be weighted to obtain better estimates of the true cross-hybridization potential. The concept of dominating $k$-mer matches could be improved with a weighting scheme and consequently, $k$-mer matches could be applied to SNP calling more specifically.

A drawback of the $k$-mer analysis might be that 25-mer duplexes with small bulges are not searched for, although 1-bp-bulges might have a smaller effect on duplex stability than mismatches [15]. In the future, the $k$-mer analysis should thus be improved to model bulges as 1-bp-indels between matching $k$-mers.

A novel heuristic for deletion calling has been proposed. Its sensitivity has been shown to increase with deletion length. Although specificity cannot be assessed directly since the set of known deletions is incomplete, indirect evaluations revealed a number of overlaps to known deletions. Some of the deletions are predicted very accurately and a systematic comparison suggests that in most cases predicted deletion boundaries should be accurate enough to sequence the break points. However, there are a few cases where predictions are split, the predicted boundaries are diffuse or the real break points are estimated badly.

A future goal for deletion calling would be to implement a learning algorithm with a state transition model trained to obtain a large margin classifier. In addition to the advantage of being founded in statistical learning theory, such a learning algorithm could be used to detect deletions smaller than 200 bp as well as regions where many polymorphisms are clustered. Such regions are observed often enough in the fragments from dideoxy sequencing to generate sufficient training data. To discover those regions would be especially valuable as SNP calling is difficult whenever polymorphisms are closer than 12 bp to each other, which results in drastic underestimation of SNP density in highly polymorphic regions. The state machine would thus supplement SNP calling in an ideal way.

In summary, the resequencing of 20 *Arabidopsis* ecotypes has provided an insight into the naturally occurring genome-wide genetic variation of *Arabidopsis* with unprecedented resolution. In this diploma thesis it has been demonstrated for large deletions, how biologically relevant information can be extracted from the huge amount of resequencing data—although it has yet to be experimentally validated how specific the deletion calls are. SNP calling could only be sketched in the scope of this diploma thesis, although SNPs are the most important information contained in the resequencing data. Finally, a few months are hardly enough to obtain a complete picture of sequence variation that can be inferred from a data set of this magnitude—there are many more lessons about genome evolution that can be learned from the resequencing data of *Arabidopsis*.

## References

[1] Mohamed Ibrahim Abouelhoda, Stefan Kurtz, and Enno Ohlebusch. The enhanced suffix array and its applications to genome analysis. In *WABI '02: Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, pages 449–463, London, UK, 2002. Springer-Verlag.

[2] James K. Bonfield, Kathryn F. Beal, Matthew J. Betts, and Rodger Staden. Trev: a DNA Trace Editor and Viewer. *Bioinformatics*, 18(1):194–195, 2002.

[3] James K. Bonfield and Rodger Staden. ZTR: A New Format for DNA Sequence Trace Data. *Bioinformatics*, 18(1):3–10, 2002.

[4] Justin O. Borevitz, David Liang, David Plouffe, Hur-Song Chang, Tong Zhu, Detlef Weigel, Charles C. Berry, Elizabeth Winzeler, and Joanne Chory. Large-Scale Identification of Single-Feature Polymorphisms in Complex Genomes. *Genome Res.*, 13(3):513–523, 2003.

[5] Aravinda Chakravarti. Population Genetics—Making Sense out of Sequence. *Nat. Genet.*, suppl_21:56–60, 1999.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 2nd edition.* MIT Press, 2001.

[7] David J. Cutler, Michael E. Zwick, Minerva M. Carrasquillo, Christopher T. Yohn, Katherine P. Tobin, Carl Kashuk, Debra J. Mathews, Nila A. Shah, Evan E. Eichler, Janet A. Warrington, and Aravinda Chakravarti. High-Throughput Variation Detection and Genotyping Using Microarrays. *Genome Res.*, 11(11):1913–1925, 2001.

[8] Brent Ewing and Phil Green. Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities. *Genome Res.*, 8(3):186–194, 1998.

[9] John E Gill, Mark D Adams, Ana J Carrera, Todd H Creasy, Howard M Goodman, Chris R Somerville, Greg P Copenhaver, Daphne Preuss, William C Nierman, Owen White, Jonathan A Eisen, Steven L Salzberg, Claire M Fraser, and J Craig Venter. Sequence and Analysis of Chromosome 2 of the Plant Arabidopsis thaliana. *Nature*, 402(6763):761–768, 1999.

[10] Joseph G. Hacia. Resequencing and Mutational Analysis Using Oligonucleotide Microarrays. *Nat. Genet.*, suppl_21:42–47, 1999.

[11] Doeke Hekstra, Alexander R. Taussig, Marcelo Magnasco, and Felix Naef. Absolute mRNA Concentrations from Sequence-specific Calibration of Oligonucleotide Arrays. *Nucl. Acids Res.*, 31(7):1962–1968, 2003.

[12] David A. Hinds, Laura L. Stuve, Geoffrey B. Nilsen, Eran Halperin, Eleazar Eskin, Dennis G. Ballinger, Kelly A. Frazer, and David R. Cox. Whole-Genome Patterns of Common DNA Variation in Three Human Populations. *Science*, 307(5712):1072–1079, 2005.

[13] The Arabidopsis Genome Initiative. Analysis of the Genome Sequence of the Flowering Plant Arabidopsis thaliana. *Nature*, 408(6814):796–815, 2000.

[14] Georg Jander, Susan R. Norris, Steven D. Rounsley, David F. Bush, Irena M. Levin, and Robert L. Last. Arabidopsis Map-Based Cloning in the Post-Genome Era. *Plant Physiol.*, 129(2):440–450, 2002.

[15] Mazen W. Karaman, Susan Groshen, Chi-Chiang Lee, Brian L. Pike, and Joseph G. Hacia. Comparisons of Substitution, Insertion and Deletion Probes for Resequencing and Mutational Analysis Using Oligonucleotide Microarrays. *Nucl. Acids Res.*, 33(3):e33, 2005.

[16] Maarten Koornneef, Carlos Alonso-Blanco, and Dick Vreugdenhil. Naturally occurring genetic variation in arabidopsis thaliana. *Annual Review of Plant Biology*, 55(1):141–172, 2004.

[17] Stefan Kurtz, Enno Ohlebusch, Chris Schleiermacher, Jens Stoye, and Robert Giegerich. Computation and visualization of degenerate repeats in complete genomes. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 228–238. AAAI Press, 2000.

[18] Inhan Lee, Alan A. Dombkowski, and Brian D. Athey. Guidelines for Incorporating Non-perfectly Matched Oligonucleotides into Target-specific Hybridization Probes for a DNA Microarray. *Nucl. Acids Res.*, 32(2):681–690, 2004.

[19] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *SIAM J. Comput.*, 22(5):935–948, 1993.

[20] Rui Mei, Earl Hubbell, Stefan Bekiranov, Mike Mittmann, Fred C. Christians, Mei-Mei Shen, Gang Lu, Joy Fang, Wei-Min Liu, Tom Ryder, Paul Kaplan, David Kulp, and Teresa A. Webster. Probe Selection for High-density Oligonucleotide Arrays. *PNAS*, 100(20):11237–11242, 2003.

[21] David W. Meinke, J. Michael Cherry, Caroline Dean, Steven D. Rounsley, and Maarten Koornneef. Arabidopsis thaliana: A Model Plant for Genome Analysis. *Science*, 282(5389):662–682, 1998.

[22] Todd C. Mockler and Joseph R. Ecker. Applications of DNA Tiling Arrays for Whole-genome Analysis. *Genomics*, 85(1):1–15, 2005.

[23] Magnus Nordborg, Tina T. Hu, Yoko Ishino, Jinal Jhaveri, Christopher Toomajian, Honggang Zheng, Erica Bakker, Peter Calabrese, Jean Gladstone, Rana Goyal, Mattias Jakobsson, Sung Kim, Yuri Morozov, Badri Padhukasahasram, Vincent Plagnol, Noah A. Rosenberg, Chitiksha Shah, Jeffrey D. Wall, Jue Wang, Keyan Zhao, Theodore Kalbfleisch, Vincent Schulz, Martin Kreitman, and Joy Bergelson. The Pattern of Polymorphism in Arabidopsis thaliana. *PLoS Biology*, 3(7):1289–1299, 2005.

[24] Y. L. Orlov and V. N. Potapov. Complexity: an Internet Resource for Analysis of DNA Sequence Complexity. *Nucl. Acids Res.*, 32(suppl_2):628–633, 2004.

[25] Hugh Salamon, Midori Kato-Maeda, Peter M. Small, Jorg Drenkow, and Thomas R. Gingeras. Detection of Deleted Genomic DNA Using a Semiautomated Computational Analysis of GeneChip Data. *Genome Res.*, 10(12):2044–2054, 2000.

[26] Detlef Weigel and Magnus Nordborg. Natural Variation in Arabidopsis. How Do We Find the Causal Genes? *Plant Physiol.*, 138(2):567–568, 2005.

[27] Yiping Zhan and David Kulp. Model-P: A Basecalling Method for Resequencing Microarrays of Diploid Samples. *Bioinformatics*, 21(suppl_2):ii182–189, 2005.

[28] Michael Zuker. Mfold Web Server for Nucleic Acid Folding and Hybridization Prediction. *Nucl. Acids Res.*, 31(13):3406–3415, 2003.

# Appendix

## Comparison of sequenced deletions and predicted deletions

Column headings for table 6-1:

| | |
|---|---|
| Chr | chromosome |
| D start | start point of the sequenced deletion |
| D end | end point of the sequenced deletion |
| D len | length of the sequenced deletion |
| %rep | percentage of repetitive site in the sequenced deletion |
| P start | start point of the 5' boundary of the predicted deletion |
| Core start | start point of the core of the predicted deletion |
| Core end | end point of the core of the predicted deletion |
| P end | end point of the 3' boundary of the predicted deletion |
| $l_B$ | estimated length of the predicted deletion including boundaries |
| $\triangle l_B$ | difference between predicted deletion length $l_B$ and length of the sequenced deletion |
| $\triangle l_B/L$ | as $\triangle l_B$ but divided by the length of the sequenced deletion $L$ |
| $l_C$ | estimated length of the predicted deletion core |
| $\triangle l_C$ | difference between predicted core length $l_C$ and length of the sequenced deletion |
| $\triangle l_C/L$ | as $\triangle l_C$ but divided by the length of the sequenced deletion $L$ |
| $o_1$ | offset between the start of the 5' boundary and the start of the sequenced deletion |
| $o_2$ | offset between the start of predicted core and the start of the sequenced deletion |
| $o_3$ | offset between the end of predicted core and the end of the sequenced deletion |
| $o_4$ | offset between the end of the 3' boundary and the end of the sequenced deletion |

For further explanations see chapter 5, section "Results an evaluation".

## Table 6.1 Comparison of sequenced deletions and predicted deletions

| Chr | D start | D end | D len | %rep | P start | Core start | Core end | P end | $l_B$ | $\triangle l_B$ | $\triangle l_B/L$ | $l_C$ | $\triangle l_C$ | $\triangle l_C/L$ | $o_1$ | $o_2$ | $o_3$ | $o_4$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3774272 | 3775239 | 967 | 7 | 3774256 | 3774412 | 3774717 | 3775290 | 1035 | 68 | 0.07 | 306 | -661 | -0.68 | 140 | -522 | -16 | 51 |
| 1 | 5370123 | 5370806 | 683 | 12 | 5370116 | 5370274 | 5370600 | 5370601 | 486 | -197 | -0.29 | 327 | -356 | -0.52 | 151 | -206 | -7 | -205 |
| 1 | 18704036 | 18704526 | 490 | 6 | 18704012 | 18704027 | 18704366 | 18704533 | 522 | 32 | 0.07 | 340 | -150 | -0.31 | -9 | -160 | -24 | 7 |
| 1 | 20088824 | 20093921 | 5097 | 20 | 20088820 | 20089792 | 20093571 | 20093927 | 5108 | 11 | 0.00 | 3780 | -1317 | -0.26 | 968 | -350 | -4 | 6 |
| 1 | 22072903 | 22073530 | 627 | 5 | 22072864 | 22072895 | 22073479 | 22073537 | 674 | 47 | 0.07 | 585 | -42 | -0.07 | -8 | -51 | -39 | 7 |
| 1 | 23066314 | 23066850 | 536 | 0 | 23066274 | 23066572 | 23066855 | 23066856 | 583 | 47 | 0.09 | 284 | -252 | -0.47 | 258 | 5 | -40 | 6 |
| 1 | 23418620 | 23418962 | 342 | 16 | 23418610 | 23418681 | 23418873 | 23418970 | 361 | 19 | 0.06 | 193 | -149 | -0.44 | 61 | -89 | -10 | 8 |
| 1 | 25162702 | 25167156 | 4454 | 13 | 25162637 | 25162920 | 25166312 | 25166315 | 3679 | -775 | -0.17 | 3393 | -1061 | -0.24 | 218 | -844 | -65 | -841 |
| 1 | 25242007 | 25242694 | 687 | 0 | 25242008 | 25242012 | 25242630 | 25242632 | 625 | -62 | -0.09 | 619 | -68 | -0.10 | 5 | -64 | 1 | -62 |
| 1 | 26891074 | 26892024 | 950 | 2 | 26891069 | 26891104 | 26891822 | 26891823 | 755 | -195 | -0.21 | 719 | -231 | -0.24 | 30 | -202 | -5 | -201 |
| 2 | 530243 | 537748 | 7505 | 34 | 529990 | 531854 | 532312 | 533190 | 3201 | -4304 | -0.57 | 459 | -7046 | -0.94 | 1611 | -5436 | -253 | -4558 |
| 2 | 1366861 | 1377066 | 10205 | 39 | 1366209 | 1374780 | 1375114 | 1377068 | 10860 | 655 | 0.06 | 335 | -9870 | -0.97 | 7919 | -1952 | -652 | 2 |
| 2 | 1543012 | 1543324 | 312 | 7 | 1542904 | 1542915 | 1543179 | 1543327 | 424 | 112 | 0.36 | 265 | -47 | -0.15 | -97 | -145 | -108 | 3 |
| 2 | 1561640 | 1562857 | 1217 | 10 | 1561817 | 1562061 | 1562559 | 1562880 | 1064 | -153 | -0.13 | 499 | -718 | -0.59 | 421 | -298 | 177 | 23 |
| 2 | 2944210 | 2944960 | 750 | 31 | 2944133 | 2944226 | 2944376 | 2944968 | 836 | 86 | 0.11 | 151 | -599 | -0.80 | 16 | -584 | -77 | 8 |
| 2 | 4903776 | 4905053 | 1277 | 44 | 4903761 | 4903764 | 4904735 | 4908160 | 4400 | 3123 | 2.45 | 972 | -305 | -0.24 | -12 | -318 | -15 | 3107 |
| 2 | 5731349 | 5733014 | 1665 | 15 | 5730932 | 5731376 | 5731835 | 5734480 | 3549 | 1884 | 1.13 | 460 | -1205 | -0.72 | 27 | -1179 | -417 | 1466 |
| 2 | 7784131 | 7791392 | 7261 | 10 | 7783979 | 7788901 | 7790745 | 7791577 | 7599 | 338 | 0.05 | 1845 | -5416 | -0.75 | 4770 | -647 | -152 | 185 |
| 2 | 9194571 | 9202364 | 7793 | 46 | 9194549 | 9195588 | 9196327 | 9202656 | 8108 | 315 | 0.04 | 740 | -7053 | -0.91 | 1017 | -6037 | -22 | 292 |
| 2 | 10469947 | 10470435 | 488 | 0 | 10469897 | 10469975 | 10470449 | 10470456 | 560 | 72 | 0.15 | 475 | -13 | -0.03 | 28 | 14 | -50 | 21 |
| 2 | 10710511 | 10711134 | 623 | 0 | 10710348 | 10710487 | 10711134 | 10711139 | 792 | 169 | 0.27 | 648 | 25 | 0.04 | -24 | 0 | -163 | 5 |
| 2 | 10713064 | 10713458 | 394 | 5 | 10712869 | 10713024 | 10713428 | 10713638 | 770 | 376 | 0.95 | 405 | 11 | 0.03 | -40 | -30 | -195 | 180 |
| 2 | 11874258 | 11875335 | 1077 | 44 | 11874276 | 11874828 | 11875338 | 11875434 | 1159 | 82 | 0.08 | 511 | -566 | -0.53 | 570 | 3 | 18 | 99 |
| 2 | 18332143 | 18334260 | 2117 | 21 | 18332244 | 18333625 | 18333968 | 18334309 | 2066 | -51 | -0.02 | 344 | -1773 | -0.84 | 1482 | -292 | 101 | 49 |
| 2 | 18658335 | 18662919 | 4584 | 2 | 18658249 | 18658377 | 18662903 | 18662981 | 4733 | 149 | 0.03 | 4527 | -57 | -0.01 | 42 | -16 | -86 | 62 |
| 3 | 129164 | 130760 | 1596 | 8 | 129161 | 129165 | 129897 | 130519 | 1359 | -237 | -0.15 | 733 | -863 | -0.54 | 1 | -863 | -3 | -241 |
| 3 | 1652107 | 1652691 | 584 | 0 | 1652087 | 1652107 | 1652692 | 1652700 | 614 | 30 | 0.05 | 586 | 2 | 0.00 | 0 | 1 | -20 | 9 |
| 3 | 3558516 | 3564101 | 5585 | 12 | 3558899 | 3558899 | 3562763 | 3562764 | 3866 | -1719 | -0.31 | 3865 | -1720 | -0.31 | 383 | -1338 | 383 | -1337 |
| 3 | 9205126 | 9206502 | 1376 | 2 | 9205124 | 9205132 | 9206574 | 9206838 | 1715 | 339 | 0.25 | 1443 | 67 | 0.05 | 6 | 72 | -2 | 336 |
| 3 | 10495591 | 10502214 | 6623 | 28 | 10497568 | 10498062 | 10501527 | 10502217 | 4650 | -1973 | -0.30 | 3466 | -3157 | -0.48 | 2471 | -687 | 1977 | 3 |
| 3 | 10495591 | 10502214 | 6623 | 28 | 10495601 | 10496262 | 10497008 | 10497009 | 1409 | -5214 | -0.79 | 747 | -5876 | -0.89 | 671 | -5206 | 10 | -5205 |
| 3 | 11123945 | 11128918 | 4973 | 7 | 11123938 | 11123938 | 11128914 | 11128919 | 4982 | 9 | 0.00 | 4977 | 4 | 0.00 | -7 | -4 | -7 | 1 |
| 3 | 11885866 | 11887008 | 1142 | 6 | 11885857 | 11886061 | 11887006 | 11887007 | 1151 | 9 | 0.01 | 946 | -196 | -0.17 | 195 | -2 | -9 | -1 |
| 3 | 17133603 | 17134410 | 807 | 8 | 17133565 | 17133587 | 17134189 | 17134578 | 1014 | 207 | 0.26 | 603 | -204 | -0.25 | -16 | -221 | -38 | 168 |
| 4 | 6467404 | 6473064 | 5660 | 24 | 6467367 | 6470863 | 6472377 | 6473085 | 5719 | 59 | 0.01 | 1515 | -4145 | -0.73 | 3459 | -687 | -37 | 21 |
| 4 | 10171528 | 10175310 | 3782 | 49 | 10173177 | 10174274 | 10175083 | 10175318 | 2142 | -1640 | -0.43 | 810 | -2972 | -0.79 | 2746 | -227 | 1649 | 8 |
| 4 | 10341395 | 10341767 | 372 | 4 | 10341374 | 10341610 | 10341776 | 10341797 | 424 | 52 | 0.14 | 167 | -205 | -0.55 | 215 | 9 | -21 | 30 |
| 4 | 10684535 | 10685919 | 1384 | 10 | 10684535 | 10684535 | 10685864 | 10685865 | 1331 | -53 | -0.04 | 1330 | -54 | -0.04 | 0 | -55 | 0 | -54 |
| 4 | 10766976 | 10767332 | 356 | 0 | 10767011 | 10767011 | 10767126 | 10767338 | 328 | -28 | -0.08 | 116 | -240 | -0.67 | 35 | -206 | 35 | 6 |
| 4 | 10803145 | 10803944 | 799 | 23 | 10802669 | 10803785 | 10803900 | 10803902 | 1234 | 435 | 0.54 | 116 | -683 | -0.85 | 640 | -44 | -476 | -42 |
| 4 | 12847590 | 12848439 | 849 | 4 | 12847584 | 12847586 | 12848442 | 12848465 | 882 | 33 | 0.04 | 857 | 8 | 0.01 | -4 | 3 | -6 | 26 |
| 4 | 14979927 | 14980393 | 466 | 31 | 14979852 | 14979928 | 14980370 | 14980398 | 547 | 81 | 0.17 | 443 | -23 | -0.05 | 1 | -23 | -75 | 5 |
| 5 | 1688276 | 1690968 | 2692 | 11 | 1689187 | 1690143 | 1690933 | 1690997 | 1811 | -881 | -0.33 | 791 | -1901 | -0.71 | 1867 | -35 | 911 | 29 |
| 5 | 4320162 | 4326251 | 6089 | 41 | 4320156 | 4320159 | 4322285 | 4322362 | 2207 | -3882 | -0.64 | 2127 | -3962 | -0.65 | -3 | -3966 | -6 | -3889 |
| 5 | 4320162 | 4326251 | 6089 | 41 | 4322462 | 4323080 | 4324871 | 4326247 | 3786 | -2303 | -0.38 | 1792 | -4297 | -0.71 | 2918 | -1380 | 2300 | -4 |
| 5 | 5383158 | 5384294 | 1136 | 23 | 5383590 | 5383590 | 5384191 | 5384192 | 603 | -533 | -0.47 | 602 | -534 | -0.47 | 432 | -103 | 432 | -102 |
| 5 | 8803849 | 8803954 | 105 | 0 | 8803830 | 8803830 | 8804032 | 8804186 | 357 | 252 | 2.40 | 203 | 98 | 0.93 | -19 | 78 | -19 | 232 |
| 5 | 8836280 | 8838245 | 1965 | 2 | 8836276 | 8836360 | 8838252 | 8838253 | 1978 | 13 | 0.01 | 1893 | -72 | -0.04 | 80 | 7 | -4 | 8 |
| 5 | 9920351 | 9933945 | 13594 | 45 | 9922858 | 9925172 | 9930349 | 9933173 | 10316 | -3278 | -0.24 | 5178 | -8416 | -0.62 | 4821 | -3596 | 2507 | -772 |
| 5 | 13004716 | 13005879 | 1163 | 17 | 13005224 | 13005511 | 13005895 | 13005956 | 733 | -430 | -0.37 | 385 | -778 | -0.67 | 795 | 16 | 508 | 77 |
| 5 | 14538357 | 14538920 | 563 | 5 | 14538335 | 14538335 | 14538928 | 14538951 | 617 | 54 | 0.10 | 594 | 31 | 0.06 | -22 | 8 | -22 | 31 |
| 5 | 14563839 | 14566063 | 2224 | 17 | 14563840 | 14563881 | 14565655 | 14565656 | 1817 | -407 | -0.18 | 1775 | -449 | -0.20 | 42 | -408 | 1 | -407 |
| 5 | 14757140 | 14757664 | 524 | 3 | 14757125 | 14757136 | 14757670 | 14757671 | 547 | 23 | 0.04 | 535 | 11 | 0.02 | -4 | 6 | -15 | 7 |
| 5 | 15816198 | 15816573 | 375 | 1 | 15816166 | 15816179 | 15816575 | 15816578 | 413 | 38 | 0.10 | 397 | 22 | 0.06 | -19 | 2 | -32 | 5 |
| 5 | 17253936 | 17254398 | 462 | 11 | 17253919 | 17253949 | 17254361 | 17254397 | 479 | 17 | 0.04 | 413 | -49 | -0.11 | 13 | -37 | -17 | -1 |
| 5 | 18159382 | 18164129 | 4747 | 10 | 18162726 | 18162726 | 18163512 | 18163513 | 788 | -3959 | -0.83 | 787 | -3960 | -0.83 | 3344 | -617 | 3344 | -616 |
| 5 | 18159382 | 18164129 | 4747 | 10 | 18159336 | 18159502 | 18162434 | 18162435 | 3100 | -1647 | -0.35 | 2933 | -1814 | -0.38 | 120 | -1695 | -46 | -1694 |
| 5 | 19375599 | 19381296 | 5697 | 3 | 19375476 | 19377761 | 19381157 | 19381366 | 5891 | 194 | 0.03 | 3397 | -2300 | -0.40 | 2162 | -139 | -123 | 70 |
| 5 | 20820568 | 20821899 | 1331 | 7 | 20820576 | 20820627 | 20821849 | 20821945 | 1370 | 39 | 0.03 | 1223 | -108 | -0.08 | 59 | -50 | 8 | 46 |
| 5 | 21000840 | 21001017 | 177 | 0 | 21000833 | 21000834 | 21001021 | 21001022 | 190 | 13 | 0.07 | 188 | 11 | 0.06 | -6 | 4 | -7 | 5 |
| 5 | 22645207 | 22648667 | 3460 | 4 | 22645551 | 22645751 | 22648684 | 22648685 | 3135 | -325 | -0.09 | 2934 | -526 | -0.15 | 544 | 17 | 344 | 18 |
| 5 | 24272110 | 24273109 | 999 | 2 | 24272106 | 24272774 | 24273033 | 24273107 | 1002 | 3 | 0.00 | 260 | -739 | -0.74 | 664 | -76 | -4 | -2 |
| 5 | 25862612 | 25862928 | 316 | 1 | 25861801 | 25862597 | 25862930 | 25862932 | 1132 | 816 | 2.58 | 334 | 18 | 0.06 | -15 | 2 | -811 | 4 |