Exploiting high throughput DNA sequencing data for genomic analysis

Markus Hsi-Yang Fritz
Darwin College

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

14th October 2011
Markus Hsi-Yang Fritz
EMBL-European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge, CB10 1SD
United Kingdom

e-mail: mhyf3@cam.ac.uk
This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

No part of this work has been submitted or is currently being submitted for any other qualification.

This document does not exceed the word limit of 60,000 words\textsuperscript{1} as defined by the Biology Degree Committee.

Markus Hsi-Yang Fritz
14th October 2011

\textsuperscript{1} excluding bibliography, figures, appendices etc.
To an exceptional scientist
— my Dad.
Exploiting high throughput DNA sequencing data for genomic analysis

Summary

Markus Hsi-Yang Fritz
Darwin College

The last few years have witnessed a drastic increase in genomic data. This has been facilitated by the shift away from the Sanger sequencing technique to an array of high-throughput methods — so-called next-generation sequencing technologies.

This enormous growth of available DNA data has been a tremendous boon to large-scale genomics studies and has rapidly advanced fields such as environmental genomics, ancient DNA research, population genomics and disease association. On the other hand, however, researchers and sequence archives are now facing an enormous data deluge. Critically, the rate of sequencing data accumulation is now outstripping advances in hard drive capacity, network bandwidth and processing power.

In the first part of this thesis, I present an efficient compression method to store and transmit DNA sequencing data. Given a reference genome, sequencing reads are stored as pointers into the reference and a list of differences, thus exploiting the redundancy the reads exhibit to the reference.

The second part deals with methods for the detection and analysis of low-copy number repeats (segmental duplications). First, I present a method that computes duplicated regions in assembled sequence through efficient self-alignment. Tests are carried out on assemblies of the fruitfly and human genomes. Then, a method is introduced that aligns sequence probes of known duplications to short reads, allowing direct interrogation of duplications in unassembled data and genotyping across multiple individuals. This method is tested on DGRP (Drosophila Genetic Reference Panel) and 1000 Genomes data and association studies to SNPs and phenotypes are carried out on the results.
ACKNOWLEDGEMENTS

First and foremost, thanks are due to my supervisor Ewan Birney. His guidance and enormous enthusiasm were the fuel for this work and made all of it possible.

I had the great pleasure of sharing my office with a bunch of friendly and helpful folks over the years. These include current and former members of Ewan’s research group (Sander Timmer, Mikhail Spivakov, Dace Ruklisa, Daniel Zerbino, Alison Meynert and Michael Hoffman), visiting students (Florian Gnad and Marcel Schulz) and members of Paul Flicek’s research group (Andre Faure, Petra Schwalie and Albert Vilella).

The EBI has a colourful and lively community of PhD students and it was a great experience to be part of it. In particular, I would like to thank Michele Mattioni, Greg Jordan, Judith Zaugg and Julia Fischer, who were part of my year’s cohort and with whom I spent two great holidays and numerous other fun events.

I was fortunate to have a very friendly and helpful thesis advisory committee: Paul Bertone, Jan Korbel and Matthew Hurles. All of them showed great interest in my work and provided valuable criticism.

I am extremely grateful to those who reviewed portions of this work: Hans-Joachim Fritz, Steven Wilder, Nicolas Delhomme, France Audrey Peltier and Mikhail Spivakov. Thank you for all the useful comments and discussions.

Many co-workers on campus made my everyday work so much easier. Thanks to the EBI systems group, technical issues were promptly resolved. Nicolas Rodriguez did a great job in managing software and installed numerous tools and libraries for me. Shelley Goddard, Sally Wedlock and Peggy Nunn were fantastic in sorting out countless formalities for me. A big
thank you to Marc Folland from the graphics department and the always helpful library staff.

Cambridge is an exciting place to be a student and an important part of this experience is the college one is member of, Darwin College in my case. This is to all the interesting people I met while living in college accommodation and being part of the rowing club and to the friendly college staff.

Mom and Dad — I could not ask for better parents. Thank you for being there for me. To my sister and nephew, thanks for putting a smile on my face every time I see you.

Above all, I would like to thank you, Fa, for your endless love, support and for enduring four years of long distance relationship. I am really happy I can finally move in with you again. I love you.
CONTENTS

Summary 7
Acknowledgements 9
List of Figures 17
List of Tables 19
List of Abbreviations 21

1 INTRODUCTION 25
  1.1 DNA sequencing 25
  1.2 Next-generation sequencing technologies 26
  1.3 The impact of next-generation sequencing 28
  1.4 Scope of this thesis 30
  1.5 Data Compression 31
    1.5.1 Importance of compression 32
    1.5.2 Types of compression 33
    1.5.3 Delta encoding 34
    1.5.4 Quantifying compression 34
    1.5.5 DNA data compression 35
  1.6 Segmental Duplications 36
    1.6.1 Properties of segmental duplications 38
    1.6.2 Human (Homo sapiens) 38
    1.6.3 Mouse (Mus musculus) 39
    1.6.4 Chimpanzee (Pan troglodytes) 40
    1.6.5 Cattle (Bos taurus) 41
    1.6.6 Fruitfly (Drosophila melanogaster) 41
    1.6.7 Mechanisms of segmental duplications 42
    1.6.8 Impact of segmental duplications 45
    1.6.9 Experimental, hybridization-based, segmental duplication detection methods 46
    1.6.10 Computational, sequencing-based, segmental duplication detection methods 47
    1.6.11 Segmental duplication terminology 52
Contents

IV APPENDIX 177
A PUBLICATIONS 179
A.1 Papers resulting from work presented in this thesis 179
A.2 Other papers published during PhD 179
B INPUT FILES 181
B.1 Compression 181
B.2 Assembly-based segmental duplication detection 181
B.3 Reads-based segmental duplication detection 182
Bibliography 185
# List of Figures

| Figure 1.1 | Costs of sequencing a human genome | 27 |
| Figure 1.2 | Duplications and reciprocal deletions resulting from NAHR | 43 |
| Figure 1.3 | Read-depth method for SD discovery | 52 |
| Figure 1.4 | Split-read method for SD detection | 53 |
| Figure 1.5 | Paired-end mapping method for SD detection | 54 |
| Figure 1.6 | Example of segmental duplications | 55 |
| Figure 2.1 | Schematic of the reference-based compression method | 61 |
| Figure 2.2 | Compression efficiency for simulated datasets | 64 |
| Figure 2.3 | Compression storage components for three parameterisations of simulated data | 65 |
| Figure 2.4 | Example of a phred-scaled base quality score distribution | 69 |
| Figure 2.5 | Compression storage costs for different quality budgets | 71 |
| Figure 2.6 | File structure of a compressed file | 77 |
| Figure 3.1 | Overview of the assembly-based SD detection method | 84 |
| Figure 3.2 | Example of an alignment | 85 |
| Figure 3.3 | Example of a maximal exact match | 87 |
| Figure 3.4 | Example of a supermaximal repeat | 87 |
| Figure 3.5 | Example of the chaining procedure | 91 |
| Figure 3.6 | Number of pairwise alignments for a duplication of copy number four | 93 |
| Figure 3.7 | Connected components of a graph | 94 |
| Figure 3.8 | Clustering of SD duplicons | 95 |
| Figure 4.1 | Circular plot of pairwise duplications in the dm2 assembly sequence | 101 |
| Figure 4.2 | Percent identity values of pairwise duplications of the dm2 assembly sequence | 102 |
List of Figures

Figure 4.3  Sizes of duplication clusters of the dm2 assembly sequence 103
Figure 4.4  Lengths of SDs of the dm2 assembly sequence 104
Figure 4.5  Lengths of SDs from Fiston-Lavier data 107
Figure 4.6  SD overlap between this work and data from Fiston-Lavier 108
Figure 5.1  Circular plot of pairwise duplications in the hg18 assembly sequence 116
Figure 5.2  Percent identity values of pairwise duplications of the hg18 assembly sequences 117
Figure 5.3  Lengths of SDs of the hg18 assembly sequence 118
Figure 5.4  SD overlap between this work and data from the Segmental Duplication DB 122
Figure 6.1  Overview of the read-based SD detection method 126
Figure 6.2  Nomination of probe candidates 127
Figure 7.1  Sequencing coverages of the DGRP data 132
Figure 7.2  Sequencing coverage vs. genome coverage 133
Figure 7.3  Minor allele frequency spectrum of the dm2 SD probes 135
Figure 7.4  Expected MAF spectrum 136
Figure 7.5  Scatter plot comparing dm2 SD probes of same duplicons 138
Figure 7.6  Scatter plot comparing dm2 SD probes of different duplicons and sample clusters 139
Figure 7.7  Scatter plot comparing dm2 SD probes of different clusters 140
Figure 7.8  FDRs of correlations between dm2 SD probes and SNPs 143
Figure 7.9  Distances of dm2 SD probes to SNPs for all significant correlations 144
Figure 7.10  Distance vs. p-value for all significant correlations between dm2 SD probes and SNPs 145
Figure 7.11  FDRs of correlation between nearby SNPs 147
Figure 7.12  Distances between SNPs for all significant correlations between nearby SNPs 148
Figure 7.13  Distance vs. p-value for all significant correlations between nearby SNPs 149
Figure 7.14  Distances between SNPs for perfect correlations between nearby SNPs  150
Figure 7.15  FDRs of dm2 chr3L SD probes to SNP correlations  151
Figure 7.16  Distances of dm2 SD chr3L probes to SNPs for all significant correlations  152
Figure 7.17  Distance vs. p-value for all significant correlations between dm2 chr3L SD probes and SNPs  153
Figure 7.18  p-values for all correlations between dm2 SD probes and three phenotypes  156
Figure 7.19  FDRs of correlations between dm2 SD probes and three phenotypes  157
Figure 7.20  Sequencing coverages of the 1000 Genomes data  159
Figure 7.21  1000 Genomes data sequencing coverages compared to DGRP data  160
Figure 7.22  Minor allele frequency spectrum of the hg18 SD probes  162
Figure 7.23  Minor allele frequency simulation for the 1000 Genomes data  163
Figure 7.24  Coefficients of correlations between hg18 SD probes  165
Figure 7.25  FDRs of correlations between hg18 SD probes and SNPs  167
Figure 7.26  Distances of hg18 SD probes to SNPs for all significant correlations  168
Figure 7.27  Distance vs. p-value for all significant correlations between hg18 SD probes and SNPs  169
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table 2.1</th>
<th>Compression efficiency for two real datasets</th>
<th>68</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.2</td>
<td>Error rates for the two experimental data sets</td>
<td>70</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Running times for compression and decompression</td>
<td>79</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Number MEMs vs. SMRs</td>
<td>90</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Chromosome lengths of the \textit{Drosophila melanogaster} dm2 assembly</td>
<td>98</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Counts of pairwise dm2 SD alignments</td>
<td>100</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>dm2 SD counts and extent per chromosome</td>
<td>105</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Run times and memory consumption for dm2 SD detection</td>
<td>106</td>
</tr>
<tr>
<td>Table 4.5</td>
<td>dm2 SD overlap with genes and isoforms</td>
<td>109</td>
</tr>
<tr>
<td>Table 4.6</td>
<td>Duplicon counts of SDs with copy number 2 of the dm2 assembly</td>
<td>110</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Chromosome lengths of the Human hg18 assembly</td>
<td>113</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Counts of pairwise hg18 SD alignments</td>
<td>115</td>
</tr>
<tr>
<td>Table 5.3</td>
<td>Sizes of duplication clusters of the hg18 assembly sequence</td>
<td>119</td>
</tr>
<tr>
<td>Table 5.4</td>
<td>hg18 SD counts and extent per chromosome</td>
<td>120</td>
</tr>
<tr>
<td>Table 5.5</td>
<td>Run times and memory consumption for hg18 SD detection</td>
<td>121</td>
</tr>
<tr>
<td>Table 5.6</td>
<td>hg18 SD overlap with genes and isoforms</td>
<td>123</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

aCGH ............ Array comparative genomic hybridization

ASCII ............ American Standard Code for Information Interchange

BAC ............... Bacterial artificial chromosome

bp ................. Base pairs

CERN ............... Organisation Européenne pour la Recherche Nucléaire

CNV ............... Copy number variant

CPU ................. Central processing unit

DGRP ............... Drosophila Genetic Reference Panel

DNA ............... Deoxyribonucleic acid

DSB ................. DNA double-strand break

EMBL ............... European Molecular Biology Laboratory

ENA ............... European Nucleotide Archive

ESA ............... Enhanced suffix array

FDR ............... False discovery rate

FoSTeS ............... Fork stalling and template switching
List of Tables

FTP ............ File Transfer Protocol
Gb ............. Giga base pairs
HR ............. Homologous repair
ICGC ............ International Cancer Genome Consortium
kb ............. Kilo base pairs
LD ............ Linkage disequilibrium
LINE ............ Long interspersed nuclear element
MB ............. Megabyte
Mb ............. Mega base pairs
MEM ............ Maximal exact match
MMBIR ............ Microhomology-mediated break-induced repair
MUM ............ Maximal unique match
NAHR ............ Non-allelic homologous recombination
NCBI ............ National Center for Biotechnology Information
NGS ............. Next-generation sequencing
NHEJ ............ Non-homologous end-joining
RNA ............. Ribonucleic acid
SD ............ Segmental duplication

22
List of Tables

SINE ............. Short interspersed nuclear element

SMR ............. Supermaximal repeat

SNP ............. Single-nucleotide polymorphism

SRA ............. Short Read Archive

SV ............. Structural variant

Tb ............. Tera base pairs

UCSC ............. University of California, Santa Cruz

WGAC ............. Whole-genome assembly comparison

WSSD ............. Whole-genome shotgun sequence detection
INTRODUCTION

1.1 DNA SEQUENCING

DNA sequencing is the act of determining the nucleotide sequence of given DNA molecules — from a short segment of a single molecule, such as a regulatory region or a gene, up to collections of entire genomes.

In the early 1970s, the first DNA sequences were obtained through extremely laborious techniques. An example is the sequencing of the two dozen base pairs of the lac operator (Gilbert and Maxam 1973). The first revolution in the DNA sequencing field took place in the second half of the 1970s with methods published by Allan Maxam and Walter Gilbert (Maxam and Gilbert 1977) and Frederick Sanger and colleagues (Sanger et al. 1977). Both techniques greatly increased the throughput of sequencing DNA. The method from Gilbert and Maxam, however, was more complex and involved the use of hazardous chemicals. The Sanger method, on the other hand, offered overall higher efficiency after a series of optimisations, in particular switching from radioactive to dye labelling of nucleotides and using capillary electrophoresis instead of slab gels. This technique dominated DNA sequencing for the following decades and lead to the determination of a complete¹ human reference genome sequence at the

¹ technically, the human reference genome sequence is still not completed, as certain parts are missing that cannot be obtained by current sequencing methods, such as regions of the highly repetitive centromeres and telomeres
INTRODUCTION

turn of the millennium (The International Human Genome Sequencing Consortium 2001; Venter et al. 2001).

Nevertheless, some disadvantages remained with the Sanger method: overall the process was still rather labour-, reagent- and time-consuming and thus involved major expenses. The costs of sequencing the genome of Craig Venter (Levy et al. 2007), for example, using an automated Sanger platform, is estimated at 70 million US dollars (Metzker 2010). This triggered the investigation and engineering of alternative, more efficient methods in the 2000’s — the so-called next-generation sequencing (NGS) or second-generation sequencing methods. Their efficient design, in particular with respect to the labour and reagents needed, and the competition between several vendors has triggered a steady drop in sequencing costs since the introduction of those technologies. At the time of writing this thesis (mid 2011) the cost of sequencing a single human genome is about 15,000 US dollars and, if current trend holds, will drop to 1,000 US dollars by the end of next year (Fig. 1.1).

1.2 NEXT-GENERATION SEQUENCING TECHNOLOGIES

Numerous NGS platforms (reviewed in Metzker 2010; Pareek et al. 2011) have been launched or are announced. The first three platforms, which currently are still the most prevalent ones, are, in chronological order of their publication: 454 (Margulies et al. 2005), an array-based pyrosequencing approach, Illumina (Bentley 2006), another sequencing-by-synthesis method and SOLiD (Valouev et al. 2008), performing sequencing-by-ligation. Without going into the details of the individual methods some fundamental features they share are as follows:

1. Cell-free template amplification using emulsion PCR or solid phase amplification

2 automated Sanger sequencing is now referred to as first-generation sequencing
3 at sufficiently large sequence coverage to obtain a high quality assembly
Figure 1.1.: Costs of sequencing a human genome in US dollars. Note the log_{10} scale of the y-axis and the radical shift in costs starting end of 2007, which coincides with the transitioning of the sequencing centres to NGS technology. The price is calculated by multiplying the costs of sequencing a single base-pair with the human genome size and the sequence coverage needed to obtain a high-quality assembly. Data taken from [http://www.genome.gov/sequencingcosts](http://www.genome.gov/sequencingcosts).
2. Immobilization of templates to some solid structure, which allows massive parallel processing

3. Imaging of nucleotides being incorporated into synthesized molecules (sequencing-by-synthesis) or probe hybridisation to templates (sequencing-by-ligation)

Newer platforms, such as Pacific Biosciences (Eid et al. 2009) or Helicos (Helicos BioSciences Corporation) generally bypass the amplification step and use single-molecule templates. Yet other methods, such as the one from Oxford Nanopore (Oxford Nanopore Technologies Ltd), employ single-molecule sequencing as well, but in addition interrogate the molecule directly and not through complementary synthesis or hybridisation.

1.3 THE IMPACT OF NEXT-GENERATION SEQUENCING

The possibility of sequencing large amounts of DNA at considerably low cost has been a tremendous boon to genomics and has lead to the advancement and establishment of several sub-disciplines.

The main driver for cheap genome-scale sequencing has been the medical sector. The compilation of detailed catalogues of alleles underlying human diseases will help to dissect their mechanisms and will lead to better treatment, particularly making possible personalised medicine (Hingorani et al. 2010). Traditionally, variant discovery was undertaken by blanket sequencing of a small pool of individuals. These variants were then used to construct assay chips which made economically feasible the genotyping of thousands of additional individuals. Such variation data were successfully employed in several large-scale case-control studies, e.g. (Wellcome Trust Case Control Consortium 2007, 2010; Barrett et al. 2008). This strategy, however, can only capture common variants and additionally suffers from sampling bias. Crucial for the success of this approach is that untyped causal alleles are in linkage disequilibrium (LD) with typed alleles. Blanket sequencing of thousands of individuals can provide a detailed catalogue of human variation down to very low allele frequencies. Importantly, such
data make possible imputation to infer untyped variants (as an example, refer to Day-Williams et al. 2011). Compilation of such repositories of dense whole-genome variation in humans is precisely the aim of the 1000 Genomes project (The 1000 Genomes Project Consortium 2010) and the UK10K project (The UK10K Project). With ever plummeting sequencing costs, genotyping is now shifting from the traditional, targeted approach to whole genome sequencing.

Besides its implication in disease association, sequencing of entire genomes of individuals is also greatly benefiting population genetics studies. Data from the 1000 Genomes project have already been used to infer mutation rates (Conrad et al. 2011) and coalescent times (Li and Durbin 2011) in humans.

Cheap whole genome sequencing has also recently led to the sequencing of multiple strains of animal model organisms. In (Keane et al. 2011), for example, the sequences of 17 inbred strains of laboratory mice are reported. The resulting, extensive catalogue of genotype variation provides an extremely valuable resource for studying genotype-phenotype associations. Another example is the sequencing of the genome of the spontaneously hypertensive rat (Atanur et al. 2010).

The feasibility to blanket sequence all DNA contained in a complex environmental sample has established the field of metagenomics. Recent landmark studies have been the reconstruction of the genome of Neanderthal, an extinct hominin, from fossil bone samples (Green et al. 2010) and the sequencing of a human gut microbial gene catalogue (Qin et al. 2010). One of the ultimate goals of environmental sequencing is transferring the information extracted from the genomes of organisms to synthetic life forms. These could be engineered in various ways to benefit mankind and reduce ecological problems, serving as medicine, fuel or fertiliser to name a few applications.

A final example on this non-exhaustive list of the high impact of cheap DNA sequencing is the development of several high-throughput readout assays that are now routinely used. In particular, these are ChIP-Seq for the interrogation of genome-wide DNA-to-protein binding (Johnson et al. 2007),
and RNA-Seq for the analysis of transcriptomes (Mortazavi et al. 2008). Other assays that rely on DNA sequencing at some stage are Methyl-seq for analysing DNA methylation state (Brunner et al. 2009) and DNase-Seq (Crawford et al. 2006) and FAIRE-Seq (Giresi et al. 2007) for identifying regions in the genome that are characterised by open chromatin.

1.4 Scope of this thesis

The thesis at hand is split into two parts. In the first part, a novel, efficient compression method is presented to reduce the storage requirements of NGS data. It is a differential compression method that is termed reference-based compression. This method exploits the redundancy NGS reads exhibit to assembled sequences (the references). Reads are stored as pointers into the reference and a list of edit operations (differences to the references). A thorough analysis on simulated data reveals how the compression gains efficiency with increasing read length and sequence coverage and it is demonstrated how this gain can be tuned by the amount of additional information kept, in particular quality information and unmapped sequence. Some experimental data sets are compressed to show how the method is expected to perform in the real world. Lastly, it is discussed how this method fits into a larger framework of lossy compression of biomolecular data, a mindset that will become increasingly important in the coming years.

In the second part, two computational methods for the detection of segmental duplications (SDs) are presented. The first method takes as input an assembled sequence and computes SDs of given minimal length and sequence identity through efficient self-alignment. Whole genome wide SD detection was carried out on Drosophila melanogaster and Human genome assemblies and compared to published results. The second method compiles duplcon-specific probes from alignments of known SDs and then matches those probes against NGS reads of an individual, thus allowing for the direct interrogation of SDs bypassing any assembly or mapping step of the reads. For data of multiple individuals this method can be used to genotype the individuals across the SDs. Analysis was carried
out on the Drosophila Genetic Reference Panel (DGRP) and 1000 Genomes low-coverage individuals. Lastly, association studies between polymorphic SDs and available single-nucleotide polymorphism (SNP) data were carried out. For Drosophila whole-genome associations between SDs and SNPs of the DGRP were computed. Furthermore, SDs were correlated with three phenotypes: starvation response, chill coma and startle response. For the human data, SD detection was carried out on a set of the low-coverage individuals. A subset of detected SDs were then associated with SNPs.

The remainder of the introduction will provide the relevant background information for the two main parts of the thesis.

1.5 DATA COMPRESSION

While the big surge in genomic data has mostly been a blessing for genomics researchers, the downside of the enormous amount of data being produced is their storage and transmission requirements. Critically, the current doubling time of sequenced DNA, estimated at less than 6 months (Stein 2010), is now outstripping improvement in several fields important for the handling of the data:

1. processing power: the performance of microchips doubles roughly every one and a half years (Moore’s law) (Stein 2010)

2. network bandwidth: the capacity of optical fiber is doubling every nine months (Butter’s law) (Stein 2010)

3. hard disk capacity: storage density of magnetic disk doubles roughly annually (Kryder’s law) (Stein 2010)
1.5.1  Importance of compression

With hard drive costs exponentially decreasing, the need for data compression might not be obvious. However, we live in an era of big data\(^4\) and data production often outstrips hard drive advancement. The Large Hadron Collider at CERN, for example, will produce roughly 15 petabytes of data per year when fully active (CERN: LHC Computing). Many other scientific fields, such as eco-sciences, neurobiology and healthcare are collecting data at an unprecedented rate (Hey et al. 2009). Genomics is experiencing an enormous surge in data as well, with DNA sequencing contributing a considerable fraction. Large sequence archives that collect data in a centralised location are particularly affected by this development. In practical terms this means that sequence archives cannot maintain stable storage budgets anymore and need progressively growing budgets if they want to keep up with the data flood. This hardly is a sustainable solution and renders this approach very unattractive. Another solution is to abstain storing some or all of the data. This scenario includes filtering data as they come off the machine and discarding whole data sets after completion of their analysis. However, the underlying assumption, in particular for the latter case, is that it is feasible to re-sequence any sample at any given point of time. This creates problems for samples that are difficult or impossible to renew, such as cancer samples and other human samples. Thus, electronic archival of those data might be the only viable solution for permanent availability. Furthermore, many large scale international projects run for several years and even after completion, follow-up studies typically are carried out which require easy access to the original data. A third, alternative, approach is data compression, which is the process of converting data into a different representation of smaller size. Out of the three approaches, compression is certainly the most attractive, as it allows the long term storage of data, but with a drastically reduced and tunable (lossy compression, see below) hard disk footprint.

\(^4\) large data that are difficult to handle (store, transmit, analyse, . . . ) with on-hand technology
1.5 Data Compression

1.5.2 Types of compression

There are two basic ways to achieve compression:

Lossless compression represents redundancy in data more compactly which can be fully restored to its original representation during decompression, i.e. no actual information is lost. A text containing many repeated words, for example, can be compressed without loss by storing a dictionary of the most used words and replacing the instances of those words in the text with pointers into that dictionary.

Lossy compression, on the other hand, removes actual information from data. An example, familiar to many, is MP3 compression of audio data, which, in essence, removes components of sound that are deemed to lay outside of the realm of human perception.

Central to compression is the notion of encoding, which is how parts of the input (typically single characters or substrings for text) are represented in the (compressed) output. One important distinction is between fixed-length and variable-length codes. Computers read and write data in fixed-sized chunks. Thus, the encodings that are most natural and easiest to work with are of fixed-size as well. The well known ASCII codes represent characters from the English alphabet, along with other characters, such as digits, punctuation and control characters as 7-bit codes. These are typically padded with one additional bit to form an 8-bit code (an octet), a length handled naturally by computers of common architecture. This, however, is in most cases an inefficient representation, as certain symbols in data often have a higher frequency of occurrence than others and thus could be assigned a shorter code, leading to an overall smaller representation. In the English language, the letter ‘e’, for example, occurs two orders of magnitudes more often than the letter ‘z’ (Wikipedia: Letter Frequency). Another example are integers which are often represented in fixed 32 or 64 bits (the so-called word size of the CPU). If small integers are much more likely to occur than larger one, again, it makes sense to employ variable-length codes, assigning shorter codes to smaller integers.
1.5.3 Delta encoding

Delta encoding (or differential compression) is a general compression technique when dealing with sequential data, in which successive values do not differ too much. In such a scenario it is sensible to encode those values as differences from each other. An example is a version control system in which successive versions of a file are stored. Such successive versions will often have only few modifications while most of the document remains unchanged. As such, many version control systems will only store the modifications and will use the previous versions as a template to apply those changes to. In this setup, delta encoding is also referred to as file differencing.

Another example is a sequence of numbers, that do not differ much, for example, coming off a telemetry device. In this case, the first value can be stored as an absolute value, and each successive value is then encoded by subtracting its absolute value from the absolute value of the previous one. Differences will generally be smaller numbers than the absolute numbers and thus require fewer bits to represent. In this context, delta encoding is also referred to as relative encoding.

1.5.4 Quantifying compression

To quantify the efficiency of a given compression run, several measures can be applied:

Compression ratio = \( \frac{\text{Compressed size}}{\text{Uncompressed size}} \)

Compression factor = \((\text{compression ratio})^{-1} = \frac{\text{Uncompressed size}}{\text{Compressed size}} \)

Data savings = \((1 - \text{compression ratio}) \times 100 \)

As an example, if a 10 MB file is compressed to 2 MB, then the compression ratio is 0.2, the compression factor 5 and the data savings 80%.
DNA sequences are naturally represented as strings over the alphabet \{A, C, G, T\}, the four nucleotides. Due to the small alphabet size and bases typically occurring with similar frequencies, a straightforward encoding of each base in fixed 2 bits often leads to a compact representation which is hard to improve on. Principally, a variety of general-purpose compressors, routinely used for text compression, are readily available, such as gzip or bzip2. However, these methods mostly fail to detect and hence to eliminate DNA-specific redundancy and ultimately result in representations that require more than 2 bits per character. Therefore, tailored DNA compression methods are needed and several methods have been proposed to date.

Most of the earlier DNA compression methods were concerned with the redundancy within a given DNA string (i.e. repeated substrings) and were essentially dictionary-based methods. Dictionary compression is a technique in which a special data structure (the dictionary) is used to hold a static (predefined) or dynamic dictionary (changing over the course of compression). Occurrences of those strings in the text are then replaced by pointers into that dictionary. Often, such methods additionally are statistical-based. This means that they use frequencies of input symbols or strings of symbols to construct variable-length codes, assigning short codes to frequent (strings of) symbols. A method could, for example, in a first pass over a text determine the frequencies of words. The dictionary would then be constructed by storing the words in descending order of their frequencies, giving the high frequency words small indices and thereby short codes.

One of the first methods proposed for compressing DNA was biocompress (Grumbach and Tahi 1993) which detects exact repeats and palindromes of arbitrary distance to each other and replaces them with pointers to previous occurrences.

An improvement of this idea was given in (Chen et al. 2002). Here, it was proposed to use approximate repeats instead of exact ones. In a
first pass such repeats are identified using a quick and sensitive heuristic and in a second pass those repeats are encoded as pointers to previous instances along with a description of the differences. This method performs significantly better than biocompress due to the typically high occurrence of slightly degenerate repeats within genomes.

With the advent of individual-level resequencing, focus shifted to the redundancy between sequences and a delta compression method was proposed in (Christley et al. 2009). In particular, given a reference assembly, a second assembly can be compressed as a series of differences to that reference. An impressive reduction in size from about 700 MB uncompressed to 4 MB compressed was achieved on the assembly of James Watson’s genome sequence against the human reference genome sequence.

1.6 Segmental duplications

Genetic variation between species or individuals of one species are differences in their genomes introduced through the process of mutation.

Historically, much attention has been paid to single nucleotide changes. If a given base is the same in all individuals of a species it is said to be fixed, otherwise it is said to be polymorphic. Accordingly such intra-species variants are called single nucleotide polymorphisms (SNPs). Once a SNP has been discovered in a pool of individuals, other individuals’ DNA can be interrogated for its specific variant. This process is known as genotyping.

Another class of variants are so called structural variants (SV). SVs describe gross changes in the genome, in particular large insertions and deletions, duplications, inversions and translocations. The subset of duplications and deletions is commonly termed CNVs (copy number variants).

Arguably, the most interesting class of SVs are large duplications, also termed segmental duplications (SDs). Such duplicated blocks can contain genes or other functional elements that were copied with the duplication event. Already in 1970, Susumo Ohno postulated in his now classic book
“Evolution by gene duplication” that duplication events are expected to play a major role in evolution as duplicated genes can diversify under relaxed selection constraints (Ohno 1970). Since then many examples have been found, such as the the evolution of the Hox multi-gene cluster (Carroll 1995) or the emergence of the gene underlying trichromatic vision in the old world primate lineage (Hunt et al. 1998).

As with single nucleotide variants, there are SDs that are polymorphic in a population of individuals. Given segments can occur in different copy number and at different locations across individuals if independent duplication events took place or if duplicated blocks or parts of them were deleted after the duplication event.

In the last decade, with the availability of many whole genome sequences, it has become apparent to what great a degree SDs have shaped species and population diversity.

The human draft genome sequence demonstrated the large amount of duplicated segments in the human genome. In (The International Human Genome Sequencing Consortium 2001), the total amount of SDs is estimated at 5% of the total genome, i.e. comprising far more base pairs than single nucleotide polymorphisms (SNPs) (estimated at 1.4 million at that time).

When the chimpanzee reference genome sequence was published (The Chimpanzee Sequencing and Analysis Consortium 2005), a companion paper addressed the overlap of SDs between the species (Cheng et al. 2005). It was shown that about a third of large SDs between the close evolutionary cousins were not shared, totalling about 2.5% of genetic difference (compared to about 1% of difference in single nucleotide changes).

Finally, the advent of NGS technology made economically feasible individual level resequencing. Even though array-based studies had already demonstrated that polymorphic SDs exist in great number, the resolution of this polymorphism has increased considerably with next-generation sequencing (Itsara et al. 2010; Kato et al. 2010; Conrad et al. 2010). As such they contribute to phenotype diversity amongst humans, including diseases (as reviewed in Mefford and Eichler 2009; Girirajan and Eichler 2010;
Stankiewicz and Lupski 2010). Examples are autism (Ullmann et al. 2007; Weiss et al. 2008; Cukier et al. 2011), macrocephaly (Brunetti-Pierri et al. 2008) and mental retardation (Lisi et al. 2008) for cases of duplication and schizophrenia (The International Schizophrenia Consortium 2008; Kirov et al. 2009) and developmental delay (Ballif et al. 2007; van Bon et al. 2011) for cases of (reciprocal) deletion.

1.6.1 Properties of segmental duplications

Genome-wide SD discovery has been undertaken for numerous species, including human (Bailey et al. 2002), mouse (Cheung et al. 2003b), chimpanzee (Cheng et al. 2005), Drosophila melanogaster (Fiston-Lavier et al. 2007), and domesticated cattle (Bos taurus) (Liu et al. 2009). Some striking differences in extent and pattern of duplicated sequence have been observed. Here, a short review is given with the key findings for each of the species mentioned above.

1.6.2 Human (Homo sapiens)

In the human genome, about 5.5% (>159 Mb) is made up of large (≥1 kb), highly identical (≥90% sequence identity) duplicate sequence (She et al. 2008). One characteristic feature is the large amount of interspersed duplications (i.e. either intrachromosomal duplications with a distance of more than a megabase or interchromosomal duplications) (Marques-Bonet et al. 2009a). Furthermore, the duplications exhibit a highly non-random distribution across the genome and are particularly common in the vicinity of the subtelomeric and pericentromeric regions (Marques-Bonet et al. 2009a).

About 400 large genomic regions have been identified that have undergone many rounds of duplication and have led to complex, mosaic structures
1.6 Segmental Duplications

(Jiang et al. 2007). Those blocks contain almost all duplications associated with recurrent structural variation and genomic disorders.

More than three quarters of the duplicated bases were found to be part of alignments of sequence identity $>95\%$ pointing to a rather recent origin (She et al. 2006). A more recent study, focusing on primates, uncovered a big surge in duplication activity taking place in the ancestor of the African great apes (Marques-Bonet et al. 2009b). Interestingly, this took place at a time when other mutational processes, such as transposition and single base pair changes were slowing down\(^5\) and thus SDs are now generally believed to be one of the key players in hominin evolution.

1.6.3 Mouse (Mus musculus)

One of the first genome-wide comparisons of segmental duplications between human and another species was with mouse (Cheung et al. 2003b). In this study, it was determined that about 1.2% (33.6 Mb) of the mouse genome was duplicated. However, only very large (>5 kb) duplications were considered and additionally the study suffered from a low-quality shotgun assembly. Interestingly, shifting to an ordered BAC-based approach lead to a much better assembly in terms of duplication resolution. Using this assembly, a second study determined that about 4.94% (141.4 Mb) of the genome was comprised of SDs $\geq 1$kb and $\geq 90\%$ percent identity (She et al. 2008).

Even though, after all, the percentage of duplicated regions in the mouse and human genome is quite comparable, the distribution and content of SDs are strikingly different (She et al. 2008):

1. in mouse, SDs map to fewer genomic locations and are typically 50-80% larger

\(^5\) DNA sequence analysis identifies mutations that established themselves in a population, i.e. were not weeded out immediately. Corresponding mutation rates are related to but not identical with rates at which according DNA structural changes occur prior to selection.
2. intrachromosomal duplications have much lower sequence identity in mouse than in human (95% compared to >99%) suggesting an older origin

3. large enrichment of tandem duplications in mouse compared to enrichment of interspersed duplications in human

4. in mouse, SDs are enriched for LINE (Long interspersed nuclear element) and depleted in SINE (Short interspersed nuclear element) transposons (the opposite is true for human; this difference might be a key factor in distribution of SDs as LINEs/SINEs have particular distribution in genomes themselves and play a role in non-allelic recombination that leads to SD formation, see section 1.6.7)

5. SDs are depleted in both genes and spliced isoforms (in human SDs are enriched for spliced isoforms)

1.6.4 Chimpanzee (Pan troglodytes)

When the genome sequence of chimp, the closest living evolutionary relative to humans, became available, one of the first questions addressed was how the SDs differed between those two species (Cheng et al. 2005).

One of the key findings was that a third of SDs detected (>94% sequence identity, >20 kb), were not shared between human and chimp, suggesting an extremely young age for this considerable fraction of human duplications. Furthermore, 177 complete or partial genes were detected in human-specific duplications, in contrast to 94 genes in chimp-specific genes. Looking at a smaller set of duplications (>20 kb), 17 full length genes were found in human-specific SDs (Marques-Bonet et al. 2009b). These genes showed evidence of positive selection. Several genes associated with human adaptation were found to have higher copy number in humans than in chimp — such as amylase AMY1, important for starch digestion and aquaporin7, implicated in sperm function. Amongst the duplications which are shared between human and chimp but are absent in non-apes, genes associated
with neuronal and muscular activities were enriched (Marques-Bonet et al. 2009b). Finally, several human disease-related duplicated segments were identified as having a single copy in chimp (Cheng et al. 2005).

1.6.5 Cattle (Bos taurus)

One of more recent genome-wide SD analysis looked at the duplications (>1 kb, >90% sequence identity) in the genome of domesticated cattle (Liu et al. 2009). 94.4 Mb duplicated sequence was identified in the genome (3.1%). One of the key findings was that most SDs were organized as local tandem duplications. Together with findings in mouse (She et al. 2008), rat (Tuzun et al. 2004) and dog (Nicholas et al. 2009), which all showed similar enrichment of tandem SDs, this suggested that tandem duplications are the archetypical mammalian SD pattern. Great apes seem to be an exception from this, as large amounts of interspersed duplications can be found in their genomes.

1.6.6 Fruitfly (Drosophila melanogaster)

Results of a genome-wide SD analysis carried out on the fruitfly genome were presented in (Fiston-Lavier et al. 2007). 1.66 Mb of the genome (about 1.4%) were found to be duplicated, with about half of the pairwise alignments having a rather high (>97%) sequence identity. The genome of the fruitfly seems particularly poor in large (>10 kb), interchromosomal and multi-copy (>5) duplications. About half of the intrachromosomal duplications are in close proximity to each other (<14 kb).
INTRODUCTION

1.6.7 *Mechanisms of segmental duplications*

Several mechanisms have been proposed that can lead to the formation of segmental duplications:

1. **Non-allelic homologous recombination**

2. **Homologous Repair with Non-homologous end-joining**

3. **Fork Stalling and Template Switching/Microhomology-mediated break-induced replication**

*Non-allelic homologous recombination*

Most recurrent SDs (i.e. those of common size and breakpoints) are believed to be formed by non-allelic homologous recombination (NAHR) between large low-copy number repeats, i.e. SDs themselves (Stankiewicz and Lupski 2002). Due to their high sequence similarity, non-allelic copies can align during meiosis or mitosis instead of the allelic pairs. If the copies are in direct orientation, subsequent crossover can lead to duplication on one chromosome arm and reciprocal deletion on the other, either on the same chromosome, but different chromatid (intrachromosomal duplication) or on different chromosomes (interchromosomal duplication). Intrachromatid NAHR occurs as well, but always leads to a deletion (see Fig 1.2). As recombination relies on DNA double-strand breaks (DSBs), sequence prone to DSBs is often found near NAHR hotspots. Such characteristic sequences include transposons, palindromes and minisatellites (Gu et al. 2008). As duplications formed by NAHR occur in both germ cells and somatic cells, they can be transmitted to offspring, thereby manifesting phenotypical traits, including heritable diseases, as well as leading to abnormal cells in a body, including ones that can lead to the onset of cancer (Fridlyand et al. 2006; Campbell et al. 2008).
Figure 1.2.: Duplications and reciprocal deletions resulting from non-homologous allelic recombination. (a) Interchromosomal, interchromatid recombination leading to an interchromosomal duplication and reciprocal deletion (b) Intrachromosomal, interchromatid recombination leading to an intrachromosomal duplication and reciprocal deletion (c) Intrachromatid recombination leading to deletion only. Adapted from (Gu et al. 2008).
INTRODUCTION

**Homologous Repair with Non-homologous end-joining**

Non-homologous end-joining (NHEJ) (Weterings and van Gent 2004) and homologous repair (HR) (Kanaar et al. 1998) are two different DSB repair pathways. As their names suggest, HR makes use of a homologous template, while NHEJ is independent of homology. In more details the two mechanisms work as follows.

During NHEJ, enzymatic machinery first detects a DSB which is followed by the formation of a molecular bridge holding the DNA ends together. Modification of the ends follows that make them compatible and ligatable. Finally, the ends are directly ligated without homologous sequence needed to guide repair.

During HR, single-strand ends are produced, one of which can invade an intact DNA template (sister chromatid or homologous chromosome) and prime DNA synthesis guided by homologous sequence on the template in order to repair the DSB.

It has been proposed that those two mechanisms in interplay can mediate duplications. In particular, during HR one of the broken ends invades and copies from the sister chromatid, leading to a duplication, while NHEJ then rejoins the ends (Gu et al. 2008). This mechanism is responsible for non-recurrent duplications, i.e. of differing size and breakpoints.

**Fork Stalling and Template Switching/Microhomology-mediated break-induced replication**

A third, rather recently proposed mechanism is Fork Stalling and Template Switching (FoSTeS) which was put forward to explain some non-recurrent rearrangements in human (Lee et al. 2007). FoSTeS is a replication error, by which a replication fork that stalls, instead of resuming at the same position, switches to a different template with complementary microhomology to anneal and prime DNA replication. Switching back to a fork located upstream will then result in a duplication.
FoSTeS has been generalized into a model termed microhomology-mediated break-induced repair (MMBIR) which may underlie genome rearrangements from all domains of life (Hastings et al. 2009).

1.6.8 Impact of segmental duplications

The impact of segmental duplication on the evolution of genomes is in essence two-fold:

1. major mechanism of gene births
2. source of genetic instability

Birth of genes through duplication

The emergence of novel genes is, for obvious reasons, a powerful source for evolutionary innovation. Most new genes seem to have arisen through different genome rearrangements, in particular whole gene duplication events, as well as creating mosaic genes from exon shuffling and gene fusion or fission (Long et al. 2003). Recently, the de-novo origin of three human protein-coding genes was reported (Knowles and McLysaght 2009), but this mode of gene innovation seems rare, judging by the few known examples.

Duplications, on the other hand are widely accepted as one of the major sources of new genes and many examples are known (as reviewed in Prince and Pickett 2002; Taylor and Raes 2004).

Once a gene gets duplicated, three outcomes are possible:

1. pseudogenization: one gene copy acquires a degenerative mutation in its coding or regulatory region that leads to the loss of its function.

2. neofunctionalization: one copy acquires changes in its coding or regulatory region which leads to a gene with novel function.
3. **subfunctionalization**: mutations in regulatory or coding region cause duplicates to carry out a different subset of functions of the ancestral single copy or to act in a tissue- or organelle-specific manner.

**Segmental duplications as a source of genetic instability**

Highly similar low copy number repeats can trigger non-allelic non-homologous recombination and thus mediate genome plasticity. NAHR can lead to different genome rearrangements, including deletion, inversion and duplications. Thus SDs can act somewhat auto-catalytically. Structural variants in turn can lead to a change in copy number of functional elements (genes, regulatory regions), or introduce evolutionary breakpoints that can ultimately lead to speciation.

When duplications occur in the germ line they can manifest themselves as genomic disorders, i.e. diseases that result from genomic rearrangements. The well-known Charcot-Marie Tooth disease type 1A (CMT1A) and hereditary neuropathy with liability to pressure palsies (HNPP) are genomic disorders that result from the duplication and reciprocal deletion of a 1.4 Mb region on chromosome 17 respectively (Stankiewicz and Lupski 2010).

If duplication events occur during mitosis, somatic cells will then be polymorphic with respect to their genome architecture. One possible fate of somatic-specific duplication is cancer, as indeed many cancers have been associated with genomic rearrangements in somatic cells (Campbell et al. 2008).

**1.6.9  Experimental, hybridization-based, segmental duplication detection methods**

There are two methods which are routinely used for experimental duplication detection, both of which rely on hybridization:

1. Array comparative genomic hybridization
2. SNP microarrays

Array comparative genomic hybridization

Array comparative genomic hybridization (aCGH) (Pinkel et al. 1998) is a method by which two individually labelled samples (test and reference) are cohybridized to an array containing target sequences. The ratio in signal intensities can then be interpreted as a proxy for copy number, i.e. if there is a significant color asymmetry in any given signal, this indicates loss or gain of DNA in the test sample at that specific genomic location.

SNP microarrays

SNP arrays can be utilized to detect duplications as well (Cooper et al. 2008). Those arrays work by hybridizing a single sample to an array containing target probes. Ratios are then generated from intensities at each probe across multiple samples. SNP arrays offer the possibility to use allele-specific probes, but suffer from reduced signal-to-noise ratio compared to aCGH (Alkan et al. 2011a).

1.6.10 Computational, sequencing-based, segmental duplication detection methods

Several computational methods have been proposed to date to detect segmental duplications in genome sequences. Generally, they can be classified into two groups:

1. Assembly-based methods: these methods take as input assembled sequence and identify duplications by self-alignment of that sequence.

2. Read-based methods: these methods use sequencing reads to detect duplications. Typically, the second input is an assembled reference sequence to which the reads are mapped.
Assembly-based segmental duplication detection

Assembly-based SD methods interrogate an assembled genome through self-alignment. The input is a set of genomic sequences, i.e. contiguous regions of a genome up to entire chromosome sequences. All pairs of sequences undergo computation of local alignments to detect repeated segments.

The two main obstacles for this strategy are as follows:

1. Common repeats: high-copy number repeats (transposons, satellite DNA) should not be mistaken for segmental duplications and must be eliminated at some stage. On the other hand, SDs can contain embedded common repeats. Depending on the repeat elimination strategy chosen, care has to be taken to characterize duplications with repeats as contiguous regions.

2. SDs can contain large indels that arose in one copy after the duplication event. This needs to be accounted for in the alignment strategy.

The most-widely used assembly-based method is WGAC (whole-genome assembly comparison), propagated by Evan Eichler and colleagues (Bailey et al. 2001). The algorithm can be summarized as follows:

1. Common, high-copy number repeats are detected by aligning the genome to repeat consensus sequences and are cut out from the assembly.

2. The remaining sequence is partitioned into tractable segments (typically of 400 kb size).

3. All-vs-all alignment of segments by BLAST software, using relaxed affine gap costs.

4. For each high-scoring sequence pair, common repeats are re-inserted and a global alignment is computed.
5. Alignments are filtered based on length and sequence identity (typically $\geq 1$ kb and $\geq 90\%$ respectively).

The central components of this method is the removing of common repeats from the very start and using relaxed gap costs when computing the local alignments to traverse large indels. Removal of high-copy number repeats (called fuguization by the authors, referring to the compact genome of the puffer fish *Fugu rubripes*) offers numerous benefits:

1. Common repeats as a source of false positives are excluded from the beginning.

2. Input sequences are shortened (almost halved in the case of the human genome) which trivially leads to reduced time and space requirements of the alignment phase. This, however, is particularly true for the removal of common repeats as these generate large numbers of high-scoring pairwise alignments.

3. Easier detection of duplication breakpoints which are often flanked by common repeats.

Using BLAST, which does not scale well to large chromosome sizes, this method relies on partitioning the input sequences into small, tractable chunks. This can lead to a large number of jobs to run. Half the human genome (after fuguization is segmented into roughly 4000 sequences of 400 kb each, which results in about 8 million pairwise alignments. Also, additional effort has to be taken to detect and connect alignments spanning segment boundaries.

Another strategy based on all-against-all BLAST alignments of partitioned sequences is given in (Fiston-Lavier et al. 2007). Using software originally designed for transposon detection (Quesneville et al. 2003, 2005) all repeats, including common ones, are detected and in a second step connected to bridge large indels. Finally, duplications that consist of only one common repeat copy are removed from the result set. This approach suffers as well from the potentially large amounts of jobs to run and additionally is likely
to compute many undesired alignments of common repeats particularly on big, repeat-rich genomes, like the human genome.

A different approach is discussed in (Abouelhoda et al. 2008). Here, the authors perform whole chromosome alignments, which results in far fewer runs. To make this tractable they employ a suffix-array sequence index which allows fast computation of exact match alignment seeds (also called anchors) and an efficient algorithm and tool to connect (so-called fragment chaining) these anchors to larger alignments. To allow large indels they employ a two-step chaining procedure, where they first detect chains made up of small fragments that are close to each other (to align the shorter, high-similarity regions of a duplication). This first set of chains then serves as input to a second chaining run allowing large gaps to traverse big indels. However, it is only demonstrated how to detect very large, interspersed duplications. As such, they use gap sizes up to 70 kb in the second chaining run. Filtering for large (≥50 kb) duplications at the end, common repeats are excluded naturally. It is not discussed how repeats should be handled when using smaller gap and length constraints. A problem with large gap size is that this is prone to connect independent duplications which are near to each other, i.e. tandem duplications.

In general, the major advantage of assembly-based methods is that they reveal the structural details of the duplications: the exact genomic positions and breakpoints. Obviously, the key factor for accuracy is the quality of the assembly. Repeated regions are notoriously difficult to assemble correctly. This is especially true for short read length due to the high number of ambiguous overlaps between reads. A common property of short-read shotgun assembly is that distinct repeat copies are collapsed into one copy (Alkan et al. 2011b), thus making it impossible for an assembly-based method to detect this duplication. Another problem that can occur is that a polymorphic single copy region in a diploid organism can be mistaken for a very recent segmental duplication. The reason for this is that virtually all current assemblers treat genomes as being haploid (Kelley and Salzberg 2010). Such spurious duplications can also occur due to sequencing errors.
1.6 Segmental Duplications

Read-based methods

Following read-based methods have been published so far:

1. Read-depth method
2. Split-read approach
3. Paired-end mapping

Read-depth method

The first read-based method proposed was the so-called WSSD (whole-genome shotgun sequence detection) method (Bailey et al. 2002). This method starts by aligning sequence reads against a reference genome. It then proceeds to identify regions across the reference in which the read-depth of mapped sequence reads is significantly higher or lower than expected, indicating a duplication or deletion in the test sample respectively, see Fig. 1.3. This method requires randomly distributed reads and that a duplicated region is present in at least one copy in the reference. Note that this method handles collapsed regions in the reference, although it then cannot distinguish between a polymorphic and fixed duplication. Another advantage is that absolute copy numbers can be predicted. The disadvantages of this method is that only very large duplications (>20 kb) can be detected with high accuracy (at 20x sequence coverage) (Alkan et al. 2009) and that no structural details of copies are revealed.

Split-read method

This group of methods searches for reads that span breakpoints (also called junctions) of duplications (the first such method was presented in Mills et al. 2006). The signal looked for is a read that maps to non-continuous regions of a reference (hence the name split-read approach). Fig. 1.4 illustrates this. With this approach, breakpoints can be identified directly, but the disadvantage is that this method does not work well for very short reads and for repeat-rich regions, because of the difficulty of aligning such reads. Worth noting is the tool Pindel (Ye et al. 2009), which typically drastically reduces the amount of reads to
Figure 1.3: Read-depth method for SD discovery. Sample reads (red) are aligned to a reference (black). Regions with significantly high read depth indicate a duplication in the sample (a), while regions with significantly low read depth indicate a deletion in the sample (b).

investigate by only considering reads of mate pairs where one read does not map to the reference.

**Paired-end mapping** Paired-end mapping makes use of read pair libraries. Such libraries contain pairs of reads, sampled at a given genomic distance (the so-called insert size). Mapping a pair to a reference can then reveal structural variants, as pairs that span SVs or are contained in SVs will align at different distance from the insert size, to different chromosomes or in different orientation than the original pair. The patterns identifying duplications are illustrated in Fig. 1.5.

1.6.11 *Segmental duplication terminology*

This section aims to define some terminology to be used when referring to segmental duplications. Fig. 1.6 illustrates the concepts. Pairwise duplications are detected through pairwise alignments, which might overlap in one sequence. To define broader regions of duplication activity (“hotspots”), sequences can be merged on coordinate overlap. The merging of sequence
1.6 SEGMENTAL DUPLICATIONS

Figure 1.4: Split-read method for SD detection. The signal that is looked for is a read whose two halves map to non-continuous regions on the reference. Note that according to the mapping pattern interspersed duplications (a) can be distinguished from tandem duplications (b). Adapted from (Alkan et al. 2011a).

Also allows the easy determination of the total amount of duplicated, non-overlapping sequence in a genome. Any single, merged sequence is termed a duplcon. Duplcons that share a common origin can be grouped into one segmental duplication. The number of duplcons of a SD is also called the copy number of that SD. When referring to the SDs of a genome the complete set of SDs/duplcons is meant.

If a segmental duplication is present in the genome of all individuals of a given population, this SD is said to be fixed in that population. If the genomes of different individuals harbour different sets of SD duplcons (that is there are missing or additional copies), that SD is said to be polymorphic.
Figure 1.5.: Paired-end mapping method for SD detection. Mate pairs are shown in the same color (black or blue). The signal looked for is change in orientation and relative distance of the mapped reads. According to the pattern observed, interspersed duplications (a) can be distinguished from tandem duplications (b). Adapted from (Alkan et al. 2011a).
Figure 1.6: (a) 3 pairwise duplications observed as 3 pairwise alignments (1 between chromosomes A and B and two between chromosomes A and C). (b) 5 sequences (so-called duplicons) result after merging sequences of pairwise alignments on coordinate overlap. These duplicons make up a SD of copy number 3 (left, blue) and one of copy number 2 (right, left).
Part I

COMPRESSSION OF HIGH-THROUGHPUT DNA SEQUENCING DATA
2

REFERENCE-BASED COMPRESSION OF DNA SEQUENCING DATA

2.1 INTRODUCTION

In this chapter a novel compression method is presented that efficiently compresses high throughput DNA sequencing data. The method exploits the redundancy that these sequences exhibit to some reference sequence. The compression scheme shows efficiency gains as read length and reference coverage increase and is tunable with respect to additional read information kept, in particular base qualities and unaligned sequence. This method provides a means to overcome serious bottlenecks in storage and transmission of next-generation sequencing data that sequence archives and research groups are currently facing.

Most of the methods and results presented here have been published in (Hsi-Yang Fritz et al. 2011).

2.2 REFERENCE-BASED COMPRESSION FRAMEWORK FOR DNA SEQUENCING DATA

What follows is a differential compression scheme that can be applied to high throughput DNA sequencing data.
2.2.1 Lossless compression of sequence bases

The basic idea is that once a read has been aligned (mapped) to a reference sequence, one can then represent the read sequence as a pointer into the reference (positional offset) and a list of variation (edit operations). For unmapped reads, one can employ a de-novo assembler (for example velvet ([Zerbino and Birney 2008])) to create contigs. These can then, in turn, be used as additional references onto which the previously unmapped reads can be aligned and compressed. An overview of the overall compression strategy is illustrated in Fig. 2.1.

In more detail, the method works as follows:

1. A pointer into the reference is stored for every read (i.e. the first position of its mapping). Lengths of reads are compressed using Huffman coding ([Huffman 1952]) in case of varying read lengths across a file. For constant read lengths, the length is stored once in the file header.

2. Assuming that the read ordering in a file carries no information, reads are re-ordered with respect to their mapping position. This sorting allows for efficient relative coding of positions, where one does not store the absolute starting positions or reads, but the difference between them. This will map “many” (depending on reference coverage and mapping distribution) absolute positions to the same, smaller relative values, thus creating redundancy. A Golomb integer code ([Golomb 1966]) is chosen to compress this positional information. The Golomb code is parameterized on expected read offset (given by the quotient of read length and coverage) or on the actual median offset determined by a first pass over a given file.

3. Any variation of the read to the reference is stored as the relative offset into the read, along with base identities (for substitutions and insertions) or length (for deletions). Positional values, again, are encoded using a Golomb code parameterized on expected (or actual) distance between successive variation positions.
Figure 2.1.: Schematic of the reference-based compression method. (a) Reads are first aligned to an established reference. (b) Unaligned reads are then pooled to create a specific “compression framework” for this data set. (c) The base pair information is stored as offsets of reads on the reference, with substitutions, insertions or deletions encoded in separate data structures.
4. In the compressed stream, every read will have a natural index. Read pairing is encoded as a rank offset from the positionally lower to the positionally higher read, which 3 additional bits present to indicate the relative orientation of the reads and the strands from which they were sequenced. Rank offsets, again, are assigned a Golomb code, parameterized on their offsets.

More details on the compression method and resulting file structure are given in section 2.5.

2.3 Results

2.3.1 Simulations

Some initial results were obtained on simulated reads. As a reference the genome of *Escherichia coli* K-12/MG1655 (accession NC_000913) was used. The reason to choose a small genome was to make high coverage data sets easily tractable. Note that due to the use of relative encoding of positions, absolute positions and hence genome length is irrelevant. Therefore, simulations of bigger genomes would have yielded the same results. All combinations of following data properties were generated:

1. Read lengths 25, 50, 100, 200, 400.

2. Error rates 0.01%, 0.1%, 1%, made up of

   a) 90% substitutions

   b) 5% insertions

   c) 5% deletions

Errors were generated uniformly across reads, i.e. agnostic of a specific sequencing platform. Indel lengths were chosen uniformly between 1 and 3 bases.
3. Reference coverages 0.5x, 1x, 5x, 10x, 25x, 50x.

4. Unpaired and paired reads, latter with an insert size of 500 bp.

Figure 2.2 shows the storage requirements for all data sets, expressed in bits/base. The worst performing data set uses 0.66 bits/base (paired 25 bp reads, 0.5x coverage, 1% error rate), while the best case uses 0.02 bits/base (unpaired 400 bp reads, 50x coverage, 0.01% error rate). For comparison, a straight-forward, minimal and uncompressed representation of the bases \{A, C, G, T, N\} would require 3 bits per base, storing them in a plain text file would require 8 bits (1 byte) per base and applying efficient bzip2 compression uses about 1 bit/base for a typical high coverage data set.

The compression shows an increase in efficiency with higher read lengths at any given coverage. This is due to the smaller number of reads one has to store for longer reads at a given coverage. Similarly, there is a marked increase in efficiency with higher coverage, though this flattens out once ~30x coverage is reached. Higher coverage leads to higher read density, and thus relative encoding of reads gets more efficient.

Figure 2.3 illustrates how at different error rates and coverages the storage requirements for individual components change. As the data properties change from low error rate and low coverage over high error rate and low coverage to high error rate and high coverage, the proportion of storage needed for variation progressively takes over the total storage requirements. As explained above, as coverage increases the storage needed for read positions decreases relatively due to higher read density and more efficient relative encoding.

2.3.2 Experimental data sets

Two experimental (“real-world”) data sets were explored as well, and compression results on their alignable portion (mapped reads) are given in Table 2.1, while Table 2.2 shows the data sets’ error rates. Compression achieved is close to the simulations and is between 5-fold and 54-fold smaller than
Figure 2.2.: Compression efficiency for simulated datasets. The plot shows storage of DNA sequence expressed as bits/base stored on the Y axis (log scale) vs coverage of datasets (X axis) for different read lengths (the different colours) after reference-based compression. The different columns indicate different simulated error rates (0.01%, 0.1%, 1.0%). The left three panels show this for unpaired data, the right three for paired data.
Figure 2.3.: Compression storage components for three parameterisations of simulated data: 0.1% error and 1x coverage (left panel), 1% error and 1x coverage (middle) and 1% error and 25x coverage (right). `readpos` and `readflags` is the storage of the read positions and read flags (strand, exact match) respectively. Variation storage for substitutions (`subst`), insertions (`insert`) and deletions (`del`) is split into positional information (`pos`), flags (`flags`) and bases (`bases`, for substitutions and insertions) or length (`len`, for deletions). The pie charts show overall storage requirements, where `readinfo` sums over read positions and read flags and `variation` is the sum over all variation storage components.
compressed FASTA or BAM respectively. As the numbers in parentheses in the table reveal, distributing a copy of the reference along the read data adds only little overhead. One key aspect is handling of so-called soft clipped bases. Clipping refers to the process of removing synthetic sequence adaptors from the reads. Soft clipping is the usual standard by which these adaptors are identified, but are kept and flagged. This allows for further analysis of these sequences and/or refinement of adaptor boundaries. On sequence level, soft-clip bases typically add only a small space overhead. For reference-based compression, however, they can add substantial overhead, as, most-often, soft clip bases won’t match the reference and thus create an edit structure that needs to be stored. This inflation in file size is very profound for the human chromosome 20 data set (cf. table 2.1). Interestingly, keeping soft clip bases can, in some instances, lead to an improvement in compression efficiency: in the bacterial data set all reads have the same length when soft clip bases are kept. Thus the read length does not have to be stored explicitly with every read. On the other side, the number of soft clip bases is extremely small (8 bases in total) and therefore is outweighed by the savings gained from the constant read length.

2.3.3 Unmapped reads

For obvious reasons, reference-based compression only handles reads that can be mapped to a reference. However, often a substantial amount of reads (10% to 40%) (E. Birney, personal communication) remains unmapped. For some data sets, such as RNA sequencing this percentage can even be higher (60% to 70%). The proposed methodology for handling such cases is to pool unmapped reads from a variety of “similar” experiments, e.g. from the same species or individual and then using a de-novo sequence assembler to generate additional references (the contigs or scaffolds). Previously unmapped reads can now be aligned against these “secondary” references and can be compressed with reference-based compression. In the case of the human chromosome 20 data set, Cortex (Caccamo and Iqbal), a large-scale De Bruijn graph assembler was used to assemble a scaffold of a subset of the unmapped reads. 45MB of novel, additional continuous sequence could be
generated, to which 14% of the originally unmapped reads aligned to. These reads compressed to 0.26 bits/base if stored without paired-end information. Furthermore, the still remaining unmapped reads were aligned against a second human genome sequence, that of Craig Venter, (Levy et al. 2007), and all bacterial and viral genomes available at that time. The reasoning was potential occurrences of structural variation and/or alignment artifacts in the human reference (thus the alignment against Venter’s genome) and potential laboratory or sample contamination (thus the alignment against the bacterial and viral sequences). In this step, an additional 3% of the reads could be compressed at a similar rate. The remaining reads exhibited no strong k-mer redundancy: 57% of 21-mers were unique in the dataset and <1% were present more than 10 times. Additionally, applying bzip2 on the raw, unmapped reads, yielded a compression of 2.12 bits/base: higher than for average sequencing reads. This strongly suggests that there is no systematic source of high coverage sequence contained in the set of remaining, unmapped reads. Handling of unmapped sequence is further discussed in section 2.4.

2.3.4 Quality bases

All established sequencing platforms output a continuous value for every base, quantifying the confidence of a particular base being correct. Typically, these values are then converted to a phred or log-scale quality value (Ewing and Green 1998). Phred scores often compress reasonably well due to small “alphabet” size and non-uniform distribution (refer to Figure 2.4 for an example). For the human chromosome 20 data set, quality scores compress to about 4 bits/quality score using Huffman coding, which is far higher than the lossless compression achieved for bases, as described before.

The scheme proposed is to only store quality values of variation bases and additionally a user-defined percentage of bases identical to the reference. The sum of these quality values is what I term “quality budget”. For any given data set, researchers can then make informed decisions of how best to put this budget into practice. As an example, the positions
### Table 2.1.: Compression efficiency for two real datasets.

<table>
<thead>
<tr>
<th>Compression method</th>
<th>NA12878 chrom20</th>
<th>Pseudomonas syringae pathovar syringae B728a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw FASTQ</td>
<td>19.96</td>
<td>21.04</td>
</tr>
<tr>
<td>Raw FASTA</td>
<td>11.45</td>
<td>12.37</td>
</tr>
<tr>
<td>Bzip2 FASTQ</td>
<td>6.64</td>
<td>4.77</td>
</tr>
<tr>
<td>Bzip2 FASTA</td>
<td>1.84</td>
<td>1.66</td>
</tr>
<tr>
<td>Bzip2 Raw sequence</td>
<td>1.09</td>
<td>0.99</td>
</tr>
<tr>
<td>BAM</td>
<td>17.48</td>
<td>7.02</td>
</tr>
<tr>
<td>Bzip2 BAM</td>
<td>17.55</td>
<td>7.03</td>
</tr>
<tr>
<td>Hard Clipped, Reference Based, Sequence Only</td>
<td>0.32 (0.37)</td>
<td>0.21 (0.28)</td>
</tr>
<tr>
<td>Bzip2 Hard Clipped, Reference Based, Sequence Only</td>
<td>0.30 (0.35)</td>
<td>0.16 (0.22)</td>
</tr>
<tr>
<td>Soft Clipped, Reference Based, Sequence only</td>
<td>0.41 (0.46)</td>
<td>0.19 (0.25)</td>
</tr>
<tr>
<td>Bzip2 Soft Clipped, Reference Based, Sequence only</td>
<td>0.37 (0.42)</td>
<td>0.16 (0.22)</td>
</tr>
<tr>
<td>Hard Clipped, Reference Based, 2% quality budget</td>
<td>0.59 (0.64)</td>
<td>0.47 (0.53)</td>
</tr>
<tr>
<td>Bzip2 Hard Clipped, Reference Based, 2% quality budget</td>
<td>0.56 (0.61)</td>
<td>0.39 (0.45)</td>
</tr>
<tr>
<td>Soft Clipped, Reference Based, 2% quality budget</td>
<td>0.79 (0.84)</td>
<td>0.44 (0.50)</td>
</tr>
<tr>
<td>Bzip2 Soft Clipped, Reference Based, 2% quality budget</td>
<td>0.69 (0.74)</td>
<td>0.39 (0.45)</td>
</tr>
</tbody>
</table>

Under all conditions our reference based compression method is significantly more efficient than standard compression techniques. The bracketed numbers shows bits/base when a bzip2-compressed copy of the reference sequence is stored with the dataset.
Figure 2.4: Example of a phred-scaled base quality score distribution. Data taken from the chromosome 20 sequencing read data of individual NA12878 from the 1000 Genomes project.
Table 2.2.: Error rates for the two experimental data sets (chromosome 20 of the individual NA12878 from the 1000 Genomes project and P. syringae whole genome data). Numbers are given in percent.

<table>
<thead>
<tr>
<th></th>
<th>NA12878 chrom20</th>
<th>P. syringae</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hard-clipped</td>
<td>Soft-clipped</td>
</tr>
<tr>
<td>Substitution rate</td>
<td>1.3756</td>
<td>2.9606</td>
</tr>
<tr>
<td>Insertion rate</td>
<td>0.0089</td>
<td>0.0080</td>
</tr>
<tr>
<td>Deletion rate</td>
<td>0.0088</td>
<td>0.0079</td>
</tr>
<tr>
<td>Error rate total</td>
<td>1.3933</td>
<td>2.9765</td>
</tr>
</tbody>
</table>

chosen could be all known SNP positions of a given species or known regions of copy-number variants. Here, the notion of lossy compression becomes pronounced. The more positions that are thrown out, the better the compression gets, but the chances of discarding qualities of potentially “interesting” or “important” positions rises. One approach imaginable is being overly pessimistic at first (i.e. keeping many positions in the budget) and then progressively reducing the budget over time after some sort of validation has been undertaken.

Figure 2.5 shows compression efficiencies for simulated reads and different quality budgets. For these data sets random positions where chosen for the budget. Note that in all cases, even for budgets of 5%, the storage required is below 1 bit/base. For obvious reasons, decreasing the quality budget leads to better compression. Additionally, however, for lower quality budgets there is an absolute shift in the compressibility of the data and the data becomes more compressible at longer read lengths. The reason for this behaviour is that a main property leading to efficient compression is having reads that match the reference exactly. For a given rate, positions kept will be distributed on proportionally more independent reads of short length, while the chances of them clustering on same, individual reads is higher for long sequence length. As mentioned before, random positions were chosen for the simulations and thus any heterogeneity in positions kept in practice will lead to better compression results. Table 2.1 illustrates results for the
Figure 2.5.: Compression storage costs for different quality budgets. Storage costs including quality information in bits/base (Y-axis) are shown for different read lengths (X-axis) for fixed coverage (10x) simulated data. Note that not only do lower quality budgets compress better, but also the compression efficiency improves proportionally more at lower quality budgets for higher read lengths. Quality budgets are the percentage of base pairs in the dataset for which quality scores are retained. Quality values were chosen randomly from the distribution shown in 2.4.
experimental data sets, using a 2% quality budget. Here, positions with the overall lowest quality values were kept. As shown, 10-30 fold improvement compared to compressed FASTQ and BAM respectively is achieved.

2.4 DISCUSSION

DNA sequencing data is increasing at a tremendous rate and a critical gap is opening up between data generation and affordable storage. Sequence archives are currently facing the dilemma of ever-increasing annual hard disk budgets. As a consequence of this, the NCBI temporarily stopped accepting submissions to the SRA (Short Read Archive) (GB Editorial Team 2011). The need for a sustainable solution to the storage problem, has led to the investigation and subsequently engineering of lossless compression of base information and lossy compression of additional information (quality values, unmapped reads). With informed decisions being made for particular data sets, this compression framework can then be utilized to tailor the loss of precision to available storage.

The reference-based compression method presented here provides efficient lossless compression of read sequences aligning to a references with few differences (down to 0.02 bits per base in simulations). The most critical decision therefore will be the loss of precision of quality values and unaligned data. While it was demonstrated that an appreciable amount of unaligned data can potentially be assembled and utilized as an additional reference, it is expected that the majority of them will remain unaligned. A decision whether or not to keep these raw data then needs to be made on a case-by-case basis, given available resources and budget and source of the sample, e.g. tumour sample. For quality information, a quality budget has to be assigned. This again, will involve informed decision and will depend highly on the factors given above as well.

A number of standard compression techniques (such as Golomb and Huffman codes) have been employed as sub-components of the reference-based compression method. Other encodings, more specifically tailored to the
actual distribution of values, might yield improved compression efficiency. Furthermore, some additional, orthogonal techniques could be used to improve the compression, e.g. detecting positions where most reads disagree with the reference. In this case one should store a single edit of the reference to the common read base. Especially for resequencing data, the error rate of the sequencing machines is often higher than for true biological polymorphism. One could apply pre-processing to the reads to recognize likely errors and subsequently remove them. Lastly, one could incorporate external data sources (known SNPs, common repeat sequences) as illustrated in (Christley et al. 2009).

My method already achieves a many-fold better compression than standard approaches. One central property of reference-based compression is that it improves with increasing read length. This is extremely promising as on every next-generation sequencing platform read length increases with newer models. Also, the increase of compression efficiency is a function of the employed quality budget. This suggests that budgets can be tailored to match sequence intake to storage capacity.

Upcoming sequencing platforms (“third-generation” generation) seem mostly improving on read length as well (i.e. longer read length). On the other hand, it is possible that they exhibit higher error rates or other different properties, which would render them sub-optimal for reference-based compression.

Recently, (Daily et al. 2010) published a similar method for compressing DNA sequencing reads. They share the basic idea of storing reads as differences to a reference sequence and employ similar standard compression techniques for storing positions and edits. However, they do not discuss many relevant issues such as handling of unaligned data, paired-end data, quality scores and clipped bases. Furthermore, they don’t consider insertions and deletions and restrict to a maximum of 2 substitutions. With those limitations, they achieve a similar compression rate (~0.35 bits per base) for a high-coverage human data set.

The main target for the method presented is whole genome shotgun data, where a substantial amount of reads can be mapped to reference sequences.
At the time of writing, these data are by far the biggest component in the SRA and is assumed to stay the largest sector of sequence growth for the next decade (E. Birney, personal communication). This is mainly due to the rapid growth in interest in medical genetics, specifically cancer research, for which a high number of samples per cancer (\(~500\)) and high sequencing coverage (\(30x\)) is typically needed. Many large-scale cancer-related projects are either already ongoing or have been declared, and the ICGC (The International Cancer Genome Consortium 2010), for example, currently coordinates 35 projects (icgc.org). Other prominent sequencing data sources are ChIP-seq and RNA-seq data. These currently occupy less than 20% of the SRA by bases and will probably decrease in proportion given the surge in medical genetics sequencing. While reference-based compression is directly applicable to those data, some limitations do however apply. ChIP-seq data can be challenging due to the very specific distribution of reads across a genome. One approach could be partitioning the genome into peak and non-peak regions and applying different codes for these different parts. Particular for proteins binding at regular intervals (e.g. histones) this might be a feasible approach. RNA-seq data are more challenging due to the typically high amount of unaligned reads. This is due to the fact that mRNAs are typically created by cutting and pasting non-contiguous parts of the (transcribed) genome (RNA splicing). One possibility would be using known transcript sequences as additional references. It is still expected, however, that new RNA-seq experiments would discover an appreciable amount of novel splice junctions. Other data sets, such as previously unstudied organisms or metagenomics data sets will bring their own challenges as well. With those data the main drawback for reference-based compression will, again, be unaligned reads. Here, an obvious approach would be to identify the most closely related species and using those as references. The compression efficiency should then be inversely proportional to evolutionary distance.

DNA has become the first bio-molecular data source, for which the storage requirements have become a significant proportion of overall analysis costs. Parallels can be drawn to imaging and particle physics where loss of information is an integral part of data handling. With an efficient compression framework offering controlled loss of precision, individual groups can
make informed choices on any particular data set. They can decide what information to keep and thereby tailor data size to available resources. The difficult decision whether to discard entire data sets can thus be prevented or, for the very least, be postponed.

2.5 CURRENT IMPLEMENTATION

To allow rapid prototyping and exploration of different methods, I wrote a suite of scripts in the python programming language¹:

1. `mz_bam_convert.py` takes a BAM file as input and produces a file holding read IDs, read positions, strand information and variation.

2. `mzip.py` takes a file, such as produced by `mz_bam_convert.py` as input and write out a compressed file. It performs two passes over the file. First it collects statistics, such as read and variation distribution. These statistics are used for choosing sensible parameters for Golomb encoding and to construct the Huffman codes. In the second pass, the actual compression is performed.

3. `munzip.py` decompresses files produced by `mzip.py` and produces a similar file that served as input to `mzip.py` (however, without read IDs and all bases are output in lower case).

A prototypical reference-based compressed file stores:

1. Magic number 0xFA220484 for verifying file type
2. Global flags indicating
   a) if read lengths are constant across the file
   b) if paired-end reads are present
   c) if base identities are present (cf. quality budget TODO add ref)
   d) which compression method (see section 2.5.1) was used to encode
      i. read positions
      ii. variation positions

¹ currently available at [http://www.ebi.ac.uk/~markus/mzip/](http://www.ebi.ac.uk/~markus/mzip/)
iii. paired-end offsets

3. Set of compression parameters
   a) for Golomb/Rice codes, the parameter $m$ (see section 2.5.1) is stored

4. Read lengths
   a) single value if all reads have the same length
   b) read length counts (used for re-construction of Huffman codes (see section 2.5.1)

5. Set of read records
   a) starting positions w.r.t. to the reference. Positions are relative encoded, i.e. the differences between successive reads instead of their absolute values are stored
   b) bit storing strand of reference to which read maps
   c) bit storing if read matches without errors to reference

6. (optional) Variation of read to reference
   a) variation positions on read; relative encoded
   b) type of variation: substitution (S)/insertion (I)/deletion (D); 1 bit of storage used in case of S, 2 bits for I or D
   c) additional variation information
      i. S: change of base w.r.t. the reference is stored in 2 bits (see section 2.5.1); in the case of quality budgets, a bit is stored indicating whether the base is identical to the reference, possibly followed by 2 bits encoding the change of base as above
      ii. I: bases are stored in 2 or 3 bits (see section 2.5.1)
      iii. D: length is stored Gamma encoded (see section 2.5.1)
   d) (optional) Paired-end information
      i. bit indicating if read is paired (only stored with the 1st mate)
      ii. strand (in 1 bit) from which read was sequenced (stored with both mates)
      iii. orientation: 1st mate was sequenced upstream from 2nd mate (only stored with the 1st mate)
      iv. offset to mate record (Golomb code; only stored with the 1st mate)

7. Magic number 0xFA220484 as a sentinel to check for file truncation

A graphical representation of a compressed file is given in Fig. 2.6
Figure 2.6.: File structure of a compressed file.
2.5.1 **Encodings**

**Golomb/Rice codes**

Golomb codes are a family of codes depending on a parameter \( m \). The subset of codes where \( m \) is a power of 2 is called Rice Codes. Let \( n \) be a non-negative integer, \( \alpha(n) \) its unary representation and \( \beta(n) \) its binary representation. The Rice code \( R \) for \( n \) and parameter \( m \) is then

\[
\alpha(q) \oplus 0 \oplus \beta(r)
\]

where \( q = \lfloor n/m \rfloor \) is the quotient of \( n \) and \( m \), \( r = n \mod m \) is the remainder of the division of \( n \) and \( m \) and \( \oplus \) is the concatenation operator. An example is given below.

\[
n = 40, \ m = 32 \\
q = 1, \ r = 8 \\
R(n, m) = 1 \oplus 0 \oplus 1000 = 101000
\]

**Huffman codes**

Given the number of expected or actual occurrences of symbols in a text, Huffman coding constructs a variable-length code for each symbol. The length of a given code is inversely proportional to the number of times the according symbol occurs, i.e. symbols with high frequencies get assigned short code words.

**Elias Gamma codes**

Let \( n \) be a positive integer, \( \beta(n) \) its binary representation and \( |\beta(n)| \) the length of \( \beta(n) \) in bits. The Gamma code of \( n \) is then \( |\beta(n)| - 1 \) zeroes followed by \( \beta(n) \). This code is scribed in (Elias 1975).
### 2.5 CURRENT IMPLEMENTATION

<table>
<thead>
<tr>
<th>Step</th>
<th>NA12878 chrom20</th>
<th>P. syringae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAM conversion</td>
<td>0:31:48</td>
<td>0:02:28</td>
</tr>
<tr>
<td>Actual compression</td>
<td>1:40:15</td>
<td>0:07:34</td>
</tr>
<tr>
<td>Decompression</td>
<td>1:46:19</td>
<td>0:07:06</td>
</tr>
</tbody>
</table>

Table 2.3: Running times for compression and decompression. The results were obtained using a computer with a Xeon processor (2.66 GHz, 32 GB RAM). Times, given in hours:minutes:seconds, were averaged over three runs.

*Base substitution codes*

Let the codes $B$ of bases $\{A, C, G, T, N\}$ be $[0, 4]$ respectively. The code of a reference base $x$ substituted by a read base $y$ is then

$$((B(y) - B(x)) \mod 5) - 1$$

*Insertion base codes*

Truncated binary encoding is used that maps the bases $\{A, C, G, T, N\}$ to $00, 01, 100, 101, 110$ respectively. $111$ serves as an *end-of-sequence* code.

#### 2.5.2 Running Time

Table 2.3 gives running times of all three scripts for the experimental data sets. It should be pointed out that the implementation was not optimised for either run time or memory usage. Nevertheless, given the big fold reduction in file size, this compares very favourable to the time needed to transmit the uncompressed files over a network. Also, currently an optimised and extended program is being written in the Java programming language which improves on running time by an order of magnitude (refer to chapter 8).
Part II

LARGE-SCALE DETECTION AND ANALYSIS OF SEGMENTAL DUPLICATIONS
While there are many publications about segmental duplications, only a small number of different methods exist for assembly-based SD detection (refer to section 1.6.10). In particular, nearly all of the published results on SDs detected in assembled sequence have originated from one group with one methodology. The goal of the work illustrated below was to take conceptual elements from these methods and to combine them into an efficient, scalable pipeline. In particular, these ideas were implemented:

1. Efficient whole chromosome comparison using tools utilized in the pipeline of (Abouelhoda et al. 2008).

2. Use of fuguization to remove common repeats from the very start, as proposed in (Bailey et al. 2001). As an alternative, delayed repeat elimination was implemented to allow using the method of (Abouelhoda et al. 2008) on smaller duplications and smaller gaps with according post-processing.

3. Recursive chaining to allow bridging of large indels.

4. Single-linkage clustering of pairwise alignments based on sequence overlap to define duplicons and SDs, inspired by (Fiston-Lavier et al. 2007)

An overview of the pipeline is given in Fig. 3.1.
Figure 3.1: Overview of the assembly-based SD detection method.
3.1 BASIC NOTIONS AND DEFINITIONS

This section aims to define some concepts for readers unfamiliar with basic sequence analysis.

3.1.1 Alignments

Given two sequences \( s \) and \( t \), a (pairwise) alignment of \( s \) and \( t \) is a sequence of so-called edit operations that transforms \( s \) into \( t \). Edit operations are insertion of characters (bases in the case of DNA), deletion and substitution, the latter of which can either be a match or a mismatch\(^1\). Fig. 3.2. illustrates this. An alignment between more than two input sequences is called multiple sequence alignment. Work for this thesis only included the use of pairwise alignments. Thus, for the remainder of this document alignment and pairwise alignment is used interchangeably.

Figure 3.2.: Alignment of sequences \( s = ACTAAGCT \) and \( t = AGAGTCT \). The sequence of edit operations is as follows. 1: substitution of \( A \) to \( A \) (match), 2: substitution of \( C \) to \( G \) (mismatch), 3: deletion of TA, 4: substitutions of \( A \) to \( A \) and \( G \) to \( G \) (matches), 5: insertion of \( T \), 6: substitutions of \( C \) to \( C \) and \( T \) to \( T \) (matches).

\(^1\) Note that these terms are used differently in molecular genetics, where “substitution” denotes a particular kind of mutation, i.e. the replacement of one nucleotide (pair) by a different one. A “match”, in molecular genetics terminology, refers to a canonical Watson-Crick base pair in a double stranded nucleic acid, whereas a “mismatch” is an opposition of two nucleotides not conforming to the Watson-Crick base pairing rules.
If \( s \) and \( t \) are aligned from front to end, the alignment is called \textit{global}. A \textit{local} alignment aligns any two substrings of \( s \) and \( t \).

### 3.1.2 Heuristic local alignment

Sequence alignment is a well-studied field and algorithms are known for the computation of optimal global (Needleman and Wunsch 1970) and optimal local (Smith and Waterman 1981) alignments. Having time complexity proportional to the product of the sequence lengths, those methods, however, are too slow for whole genome comparison.

A well known and often employed algorithm for computing local, heuristic DNA alignments is (gapped) \textsc{BLAST} (Altschul et al. 1997). The core of the algorithm is fast computation of exact matches of a given, fixed size between two sequences (the so-called seeds). If two such matches are found in close proximity, the regions around those seed matches are aligned to give rise to a local alignment.

### 3.1.3 Exact matches as alignment seeds

Most heuristic alignment methods perform computation of exact matches as an initial step to find regions of high similarity. The matches used by \textsc{BLAST} are of a fixed size \( k \) and are thus called \( k \)-mer matches. However, the number of \( k \)-mer matches grows rapidly with sequence length and degree of similarity between the input sequences. Given that typically many \( k \)-mers overlap and form longer stretches of matching bases, a more efficient type of seed is maximal exact matches (MEMs). These are exact matches of a given minimal length, which cannot be extended to either side (as illustrated in Fig. 3.3). While the term seed is the commonly used for \( k \)-mer matches, the term anchor is typically used instead for MEMs. The use of MEMs often yields far fewer elements to compute and handle. However, in the presence of many high-copy number repeats, as are present in many genomes, the number
of MEMs can quickly become intractable as well. Maximal unique matches (MUMs), as employed by the MUMmer software (Delcher et al. 2002), are MEMs that only occur once. However, such matches might fail to produce enough anchors due to their extreme rareness. Recently rare MEMs were introduced (Ohlebusch and Kurtz 2008) which are MEMs that occur at most a given time (the so-called rareness value). For the case of self-comparison, i.e. detecting intrachromosomal duplications, supermaximal repeats (SMRs) have been employed (Abouelhoda et al. 2008). These are maximal exact repeats that are not a substring of any other maximal exact repeat (Fig. 3.4). It can be shown that a SMR corresponds to a maximal exact repeat with rareness value 4 (Abouelhoda et al. 2008). For the remainder of this thesis, I will use the term rare4-MEM to refer to both supermaximal repeats within a single sequence and rare MEMs with rareness value 4 between 2 sequences.
For computing maximal exact matches a sequence index has to be employed that facilitates their rapid computation. A sequence index that makes this both efficient and easy is a suffix tree (part 2 of Gusfield 1997) or an equivalent data structure like the enhanced suffix array (ESA) (Abouelhoda et al. 2004).

3.2 SEGMENTAL DUPLICATION DETECTION PIPELINE

3.2.1 Fuguization

As mentioned earlier, fuguization is the process of removing high-copy number repeats (such as transposable elements and satellite DNA sequence) from a given genomic sequence. Widely used tools for detecting common repeats are RepeatMasker (Smit et al. 1996-2010) and censor (Kohany et al. 2006). Both tools align query sequences against a database of known repeat consensus sequences in order to detect instances of such repeats in the queries.

For many assembled and publicly available genome sequences, the online resources Ensembl (Flicek et al. 2011) and UCSC Genome Browser database (Fujita et al. 2011) offer pre-computed RepeatMasker annotation for download. For the work at hand, a Python script was written that reads in a genomic sequence, cuts out all repeats that are listed in a given annotation file and outputs the concatenated remaining sequences.

If annotation is not readily available, RepeatMasker or censor can be used to compute the annotation on the fly.

It is worth noting that the aforementioned repeat detection methods can only discover repeats that have already been characterised, as they rely on a database of repeat consensus sequences, such as Repbase (Jurka et al. 2005). For less-studied genomes, de-novo repeat detection approaches (such as Edgar and Myers 2005; Ellinghaus et al. 2008) have to be employed. As the
3.2.2 Initial anchor computation

The pipeline uses either MEMs or rare4-MEMs as initial alignment anchors. In a preprocessing step, an enhanced suffix array is built for each chromosome. The default is to use compacted chromosomes, which have undergone fuguization. For ESA construction the tool mkvtree is utilized. For the actual match detection vmatch is used to compute all MEMs of a given minimal length between all pairs of sequences. When rare4-MEMs are used a combination of vmatch and ramaco is employed as vmatch only computes SMRs, i.e. intrachromosomal duplications and only those in direct orientation. For all other cases (reverse complemented intrachromosomal duplications and both, direct and reverse complemented interchromosomal duplications) ramaco is used to compute rare MEMs with rareness value 4.

It is worth noting that ramaco can additionally compute rare MEMs of any given rareness value. However, at the time of writing ramaco cannot perform efficient self-comparison as it was not designed for this task. Generally, the sparseness of rare MEMs, especially for low rareness values, can lead to loss of sensitivity when not enough anchors are produced in near proximity. Therefore, by default, MEMs are used as they offer the highest density which typically increases sensitivity. Crucially, the use of MEMs often is only feasible due to the fuguization step as the number of MEMs can become very large in the presence of common repeats. Using rare4-MEMs, as suggested by (Abouelhoda et al. 2008), alleviates this problem to some extent as such rare matches are not necessarily expected to be generated by high-copy number repeats. Table 3.1 compares the number of MEMs and SMRs of minimal length 20 that are found in the sequence of human chromosome 12 (which amongst the human chromosomes is of average size). Note that the number of SMRs contained in the uncompacted sequence is about 30 times higher than in the compacted one, which shows that common repeats generate a large amount of supermaximal matches. This
### Table 3.1:

<table>
<thead>
<tr>
<th></th>
<th>MEM count</th>
<th>SMR count</th>
</tr>
</thead>
<tbody>
<tr>
<td>fuguization</td>
<td>122,504</td>
<td>38,554</td>
</tr>
<tr>
<td>no fuguization</td>
<td>821,999,253</td>
<td>1,124,678</td>
</tr>
</tbody>
</table>

Number MEMs vs. SMRs of minimal length 20 of human chromosome 12.

This means that even when using rare 4-MEMs as anchors, repeat elimination has to be undertaken in post processing. Also, the number of SMRs in the uncompacted sequence is about 9 times higher than the number of MEMs in the compacted one. This means increased time and space requirements for computing, storing and chaining the anchors.

#### 3.2.3 Recursive chaining

After the initial match detection phase, matches have to be connected to bridge gaps and form longer alignments. For this the tool CHAINER is used that performs efficient match chaining. The chaining algorithm computes optimal local chains of non-overlapping, co-linear matches. The latter constraint means that the substrings of the matches have to appear in the same relative order in their respective sequence. For an example, refer to Fig. 3.5.

The two main parameters for chaining are maximal gap size and a weighting constant for the matches. The latter counterbalances gap costs and makes long, high-scoring chains possible.

After transforming the output from vsmatch/ramaco into a format compatible with CHAINER, a first round of chaining is performed. In this step, typically, a small, maximal gap size and the default weight is used. This facilitates bridging small indels and regions of mismatch and thus focuses on short, high similarity regions. The resulting chains are then input into CHAINER a second time, i.e. every chain is taken as one contiguous fragment. The
Figure 3.5.: Four matches 1-4 are shown between two sequences $s$ and $t$ (top). Matches 1, 3 and 4 are non-overlapping and co-linear and can thus form a chain. Match 2 overlaps match 3 and furthermore is not co-linear with match 1 and 3 and thus cannot be part of the same chain. Another possible chain is 1, 2 and 4. However match 3 is longer than 2, while the gap sizes are comparable and thus the optimal chain is 1, 3, 4 in this example (bottom).
second call is used to bridge larger indels. The maximal gap size is set accordingly (typically 1-2 kb) and a higher weighting factor is used.

When not employing fuguization and using rare4-MEM seeds, larger gap sizes have to be used as embedded common repeats might fail to generate seeds. As such, there is an increased possibility to connect two distinct duplications in near proximity, i.e. tandem duplications.

3.2.4 Global alignment and filtering

After the recursive chaining procedure, the first step is re-insertion of all common repeats into the segmental duplication candidates. A Python script was written that reads in the result from the final CHAINER run together with the repeat annotation and original chromosome sequences. Based on the repeat annotation a mapping is computed from sequence coordinates in the compacted sequences to the original sequences. The script then iterates over the chains, translates the coordinates and outputs the according substrings from the original sequences.

If fuguization was not employed, all sequences can be directly extracted from the original input sequences. In this scenario, all sequences then undergo repeat detection using the censor software. For a given segmental duplication candidate, if any of the two sequences match with more than 95% of their length to a single copy of a common repeat, the candidate is discarded.

Besides traversing large indels, chaining also might bridge large mismatching sequence. Thus, it is important to filter out sequence pairs with overall low similarity. Additionally, one might be interested only in recent duplications and consequently highly similar sequence pairs.

Therefore global alignments of all sequence pairs are computed. The tools utilized are needle and stretcher from the EMBoss software package (Rice et al. 2000). Both programs compute optimal global alignments, with the difference that stretcher is tailored for memory-efficient handling of large
sequences at the cost of increased running time. Per default, if the product of the sequence lengths is greater than $10^8$, the pipeline engineered for this work utilizes stretcher, otherwise needle.

Once the pairwise alignments are computed, the percent identity of the sequences is calculated and used for filtering. To be consistent with output from the WGAC method, the percent identity is calculated on substitutions only, i.e. gaps excluded (Bailey et al. 2001). To eliminate alignments consisting of large amounts of gaps, an additional filter was implemented based on the percentage of gaps in the alignment. Per default, alignments with more than 30% gaps are discarded.

### 3.2.5 Clustering

At this stage a set of pairwise duplications has been computed. However, regions might undergo multiple rounds of duplications and be present in higher copy number than 2. If a given segment has copy number $n$ in the genome, it is expected that $\binom{n}{2}$ pairwise alignments will be computed, with regions participating in several alignments (for an example refer to Fig. 3.6).

![Figure 3.6](image-url)

**Figure 3.6:** Number of pairwise alignments for a duplication of copy number four. Duplicated segments are shown as grey rectangles. Up to $\binom{4}{2} = 6$ pairwise alignments (illustrated as arcs and small, red squares) will be reported.
Therefore, sequences across pairwise alignments should be merged on coordinate overlap. This allows the detection of multi-step duplications and to define the segmental duplications as broader regions of recurring duplication activity.

To this end a clustering procedure was implemented. The algorithm performs single linkage clustering based on sequence overlap. This means, for a given cluster and a given pairwise duplication of this cluster, one of the two sequences must overlap at least one sequence of any other pairwise duplication in the same cluster.

A Python script was written that performs the clustering using a graph-based approach. A pairwise duplication is represented as a node in a graph. Iteratively, overlaps are detected and the nodes of duplications with overlapping sequences are connected with an edge. In the end, the connected components (Fig. 3.7) of the graph are computed, each of which corresponds to one cluster. For every cluster, overlapping sequences are

![Graph with 5 connected components (A, B, C, D, E). Formally, a connected component is a subgraph where any two nodes are connected to each other by a path.](image-url)
merged into a single region to define the duplicons and SDs. For an example of the clustering procedure, refer to Fig. 3.8.

Figure 3.8: (a) 3 pairwise duplications are shown: 1 duplicated segment between chromosomes A and B and two duplicated segments between chromosomes A and C. (b) Pairwise duplications are represented as nodes of a graph and are connected if they overlap in one of their sequence. Two connected components are the result. (c) Overlapping sequences are merged and each connected component forms one cluster (bottom).
ASSEMBLY-BASED DETECTION RESULTS:
DROSOPHILA MELANOGASTER

This chapter describes the results obtained on the assembled genome of Drosophila melanogaster using the assembly-based segmental duplication detection method described in chapter 3.

A published data set of D. melanogaster segmental duplications for the R4/dm2 assembly (Fiston-Lavier et al. 2007) was used to draw some comparisons.

4.1 INPUT FILES

Sequence files for chromosome arms 2L, 2R, 3L, 3R, 4 and X, and the RepeatMasker annotation of the Drosophila melanogaster assembly R4/dm2 were obtained from the UCSC Genome Browser Database (refer to appendix B.2 for details). As in (Fiston-Lavier et al. 2007) the heterochromatic sequence files were not analysed.
Table 4.1: Chromosome lengths of the dm2 assembly before and after applying fuguization.

<table>
<thead>
<tr>
<th>chromosome</th>
<th>original (bp)</th>
<th>compacted (bp)</th>
<th>compacted (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2L</td>
<td>22,407,834</td>
<td>20,814,877</td>
<td>92.89</td>
</tr>
<tr>
<td>2R</td>
<td>20,766,785</td>
<td>18,844,690</td>
<td>90.74</td>
</tr>
<tr>
<td>3L</td>
<td>23,771,897</td>
<td>21,843,648</td>
<td>91.88</td>
</tr>
<tr>
<td>3R</td>
<td>27,905,053</td>
<td>26,352,669</td>
<td>94.43</td>
</tr>
<tr>
<td>4</td>
<td>1,281,640</td>
<td>927,125</td>
<td>72.33</td>
</tr>
<tr>
<td>X</td>
<td>22,224,390</td>
<td>19,964,656</td>
<td>89.83</td>
</tr>
</tbody>
</table>

4.2 PIPELINE RUN

4.2.1 Fuguization

Common repeats were removed from the chromosome sequences using the RepeatMasker annotation. Table 4.1 shows the length of chromosome arms before and after repeat removal. The resulting sequences are on average about 10% reduced in size.

4.2.2 Computation of initial alignment anchors

For all 21 chromosome pairs (including the self comparisons), maximal exact matches of minimal length 15 were computed, both in direct and reverse complemented orientation, using vmatch. A total number of 72,529,811 MEMs were detected.
4.2.3 Recursive chaining

The maximal exact matches then served as input to CHAINER. Chains of minimal average length 30 were computed using a maximal gap size of 200 and a fragment weighting factor of 10. This resulted in a total of 43,528 chains.

A second chaining run was then executed on the chains from the previous step. This time, chains were filtered on minimal average length 300 using a maximal gap size of 1500 and a weighting factor of 12. 5,148 chains were the result of the recursive chaining.

4.2.4 Repeat reinsertion and global chain alignments

Using the RepeatMasker annotation and the original chromosome sequences, repeats were re-inserted into the chains.

In the data of (Fiston-Lavier et al. 2007), the minimal segmental duplication unit is 347 bp long. To make results more comparable, only chains were kept in which both sequences had a minimal length of 350 bp.

For each remaining chain, a pairwise global alignment was computed using needle and stretcher. For both programs gap parameters were chosen to favour few long gaps over many short gaps and stretches of mismatched bases. For needle penalties for opening a gap and extending a gap were set to 10.0 and 0.5 respectively, for stretcher parameters 16 and 1 were chosen.

The minimal percent identity of duplications in the data of (Fiston-Lavier et al. 2007) is 85\% (Supplementary Fig. S1). Therefore, alignments were filtered for percent identity $\geq$85\% as well. Furthermore, a maximal gap percentage of 30\% was used for filtering.

The combination of relaxed gap parameters and a constrained gap percentage threshold will discard some duplications that have undergone extensive
As a final step, the 3,386 pairwise alignments were clustered on overlapping coordinates and merged. 825 segmental duplication units split across 335 clusters were the result. Fig. 4.3 shows the distribution of cluster sizes (duplication copy numbers).

The smallest segmental duplication reported is 350 bp (as per filter criterion) and the maximal length is 126,911 bp. The distribution of all lengths is given in Fig. 4.4. All segmental duplications combined make up a total of 2,022,416 bp (1.70% of the genome analysed). Table 4.3 gives the count and extent (in bp) of segmental duplications per chromosome.
Figure 4.1.: Circular plot showing the pairwise SD alignments of the dm2 assembly as arcs between chromosomes. The different colours of the arcs encode the sequence lengths: grey (350-1,000 bp), light blue (1,001-5,000 bp), green (5,001-10,000 bp) and red (>10,000 bp).
Figure 4.2: Histogram of percent identity values for each pairwise alignment of the dm2 SD analysis.
Figure 4.3.: Histogram of number of duplicons per SD cluster (cluster size) of the dm2 assembly. In most cases the cluster size corresponds to the copy number of the according SD.
Figure 4.4.: Histogram of SD lengths (in bp) for the dm2 SD analysis. Note that the x-axis is on log10 scale.
### 4.3 Comparison with Published Results

<table>
<thead>
<tr>
<th>chromosome</th>
<th>count</th>
<th>total number bp</th>
<th>fraction of chrom (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2L</td>
<td>135</td>
<td>461,998</td>
<td>2.06</td>
</tr>
<tr>
<td>2R</td>
<td>154</td>
<td>380,825</td>
<td>1.83</td>
</tr>
<tr>
<td>3L</td>
<td>131</td>
<td>246,049</td>
<td>1.03</td>
</tr>
<tr>
<td>3R</td>
<td>151</td>
<td>295,207</td>
<td>1.05</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>50,185</td>
<td>3.91</td>
</tr>
<tr>
<td>X</td>
<td>232</td>
<td>588,152</td>
<td>2.64</td>
</tr>
</tbody>
</table>

Table 4.3: SD counts and extent (in bp) for each chromosome of the dm2 assembly.

The running times and peak memory consumption for all steps of the pipeline are given in Table 4.4. The most time-consuming step is the alignment of duplications. The tools used, needle and stretcher, perform optimal global alignment. Using a heuristic should considerably speed up that part of the pipeline. Also it should be noted that most steps can be run parallelised. Using a peak of 150 CPU cores the complete analysis was performed in 7 minutes.

### 4.3 Comparison with Published Results

This section draws some comparison to the results reported in (Fiston-Lavier et al. 2007). The data given is either quoted from the manuscript or extracted from the SD coordinates provided as supplementary material (refer to appendix B.2).

The analysis results of (Fiston-Lavier et al. 2007) reports about 1.6 Mb total duplicated sequence split across 444 segmental duplication. This is about 80% of total content and a bit more than half of SD units that are reported here (about 2 Mb, 825 SDs). The length distribution of the SDs found in (Fiston-Lavier et al. 2007) (Fig. 4.5) reveals that, in particular, many fewer short segmental duplication (of a few hundred bp) were found than in
### Assembly-Based Detection Results: Drosophila melanogaster

<table>
<thead>
<tr>
<th>step</th>
<th>time (min)</th>
<th>memory peak (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuguization of chromosome sequences</td>
<td>0.21</td>
<td>2</td>
</tr>
<tr>
<td>Indexing of sequences</td>
<td>0.61</td>
<td>3</td>
</tr>
<tr>
<td>MEM computation</td>
<td>13.08</td>
<td>339</td>
</tr>
<tr>
<td>First chaining</td>
<td>13.85</td>
<td>659</td>
</tr>
<tr>
<td>Second chaining</td>
<td>0.02</td>
<td>2</td>
</tr>
<tr>
<td>Extraction and uncompacting of sequences</td>
<td>8.34</td>
<td>3</td>
</tr>
<tr>
<td>Global alignments of final chains</td>
<td>40.54</td>
<td>719</td>
</tr>
<tr>
<td>Filtering of final chains</td>
<td>0.09</td>
<td>8</td>
</tr>
<tr>
<td>Clustering of duplications</td>
<td>0.03</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.4.: Run times (in minutes) and peak memory consumption (in MB) for the dm2 SD analysis.

The analysis carried out for this thesis (Fig. 4.4). The range of SD lengths, however, is very comparable. It should be noted that using a maximal gap size of 1,500 bp in the second chaining step will split duplications with very large indels in favour of not joining unrelated duplications in near proximity. This could be part of the reason for the much smaller number of SDs reported in (Fiston-Lavier et al. 2007) if larger indels were allowed in their analysis.

Segmental duplications across the two data sets were intersected based on their coordinates. Fig. 4.6 shows the result. It is worth noting that both methods miss a substantial fraction of SDs detected by the according other method.

One source of possible false positives in the result of the analysis done for this thesis are common repeats as (Fiston-Lavier et al. 2007) employ a more comprehensive repeat detection step using de-novo repeat finding. However, common repeats would lead to clusters with very high copy numbers, something that is not observed (Fig. 4.3).
Figure 4.5.: Histogram of SD lengths (in bp) for the dm2 SD analysis of (Fiston-Lavier et al. 2007). Note that the x-axis is on log10 scale.
Figure 4.6: Venn diagram showing the overlap of SDs detected for this work (a) with the data published by (Fiston-Lavier et al. 2007) (b). The intersection gives two numbers: 262 for this work and 250 for the work of (Fiston-Lavier et al. 2007). The reason for this is that multiple (fragmented) SDs can overlap a single (merged) SD.
4.3 Comparison with Published Results

<table>
<thead>
<tr>
<th></th>
<th>Genes</th>
<th>Isoforms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count Normalised</td>
<td>Count Normalised</td>
</tr>
<tr>
<td>This work</td>
<td>426 0.76</td>
<td>729 1.30</td>
</tr>
<tr>
<td>Fiston-Lavier et al.</td>
<td>82 0.42</td>
<td>211 1.09</td>
</tr>
</tbody>
</table>

Table 4.5.: Count of genes and their isoforms that overlap with SDs that were either detected by this work or the work of (Fiston-Lavier et al. 2007). Counts are given in absolute numbers and normalised by the number of SDs.

The SDs that were only detected by one of the two methods were analysed for overlap with known genes. First, the coordinates of the duplications were converted from the dm2 assembly to dm3 using the Flybase Coordinates Converter web tool. For the data of this work, coordinates of 3 SDs could not be converted, which left 560 SDs; for the data of Fiston-Lavier 2 SDs could not be converted, which left 192 SDs. These coordinates were then uploaded to the Biomart Ensembl web server to compute the overlap to the Ensembl release 63 BDGP5.25 Drosophila melanogaster gene set. Table 4.5 gives the counts of genes and isoforms overlapping the SDs. Either work misses a considerable number of genes and hence potential targets for follow-up studies. The use of different parameters is likely to increase sensitivity and thus a careful investigation concerning this matter will be undertaken as future work. It is possible that big discrepancies in the results of both methods remain due to some fundamental differences. In this case, the two methods should be used in parallel to complement each other, something that is already routinely done for reads-base d methods (Alkan et al. 2011a).

Table 3 of (Fiston-Lavier et al. 2007) shows the number of SDs per chromosome for all duplications of copy number 2. Table 4.6 gives the according counts for the results of this work. The relative numbers are comparable, although numbers are generally twice as high in the data obtained for this thesis, for reasons given above.

Finally, in (Fiston-Lavier et al. 2007) an excess of high-similarity duplications and duplication enrichment in chromosome arm ends (subtelomeric and
### Table 4.6.

Duplicon counts of SDs with copy number 2 of the dm2 assembly. For interchromosomal duplications the number of dupli
cation events is given, whereas for intrachromosomal duplication the number of duplicons is given (this is consistent with Table 3 in (Fiston-Lavier et al. 2007)).

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>2L</th>
<th>2R</th>
<th>3L</th>
<th>3R</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>98</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2L</td>
<td>56</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2R</td>
<td>66</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3L</td>
<td>54</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3R</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

pericentromeric regions) is noted. These characteristics are observed in the data of this work as well (Figs. 4.2 and 4.1).

In summary, the general properties of the SDs of both sets are very comparable. The only possible substantial bias is the seeming lack of small SDs in (Fiston-Lavier et al. 2007), although this could, at least in part, be due to a different strategy of combining alignments in particular with respect to gap sizes. As pointed out, both methods do detect a considerably different set of SDs. This is probably due to the alignment algorithm and parameters, possibly in combination with the different treatment of common repeats taken.
ASSEMBLY-BASED DETECTION RESULTS: HOMO SAPIENS

This chapter describes the results obtained on the assembled human genome.

Results were validated against the key finding of several publications. Segmental duplications for the hg18 assembly from the Segmental Duplication DB were used as a reference set.

5.1 INPUT FILES

Sequences for chromosomes 1-22, X and Y, and RepeatMasker annotation of the hg18 assembly were downloaded from the UCSC Genome Browser Database (confer appendix B.2 for details). Files of unfinished or unmapped sequence (chrN_random files) and haplotype-specific sequence files were excluded from the analysis.
5.2 PIPELINE RUN

5.2.1 Fuguization

The first step was to remove all annotated common repeats from the chromosome sequences. Due to the high amount of high copy number repeats in the human genome, the sequences were significantly compacted with a reduction in length of about a half on average. (Table 5.1).

5.2.2 Computation of initial alignment anchors

After constructing an ESA index for each of the compacted sequences using mkvtree, maximal exact matches of minimal length 17 bp were computed with vmatch. This was done for all 300 sequence pairs both in direct and reverse complemented orientation. Across all sequence pairs a total of 1,247,350,530 MEMs were detected.

5.2.3 Recursive chaining

The MEMs then served as input to an initial chaining procedure. Following CHAINER parameters were used for this step: minimal average length 34, maximal gap size 200 and fragment weighting factor 10. A total of 1,609,217 were computed.

These chains then used as the input of a second chaining run. This time chains were computed using minimal average length 500, maximal gap size 1500 and weighting factor 12. A total of 177,297 chains were the result.
<table>
<thead>
<tr>
<th>chromosome</th>
<th>original (bp)</th>
<th>compacted (bp)</th>
<th>compacted (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>247,249,719</td>
<td>137,315,434</td>
<td>55.54</td>
</tr>
<tr>
<td>2</td>
<td>242,951,149</td>
<td>132,507,662</td>
<td>54.54</td>
</tr>
<tr>
<td>3</td>
<td>199,501,827</td>
<td>105,551,327</td>
<td>52.91</td>
</tr>
<tr>
<td>4</td>
<td>191,273,063</td>
<td>99,368,840</td>
<td>51.95</td>
</tr>
<tr>
<td>5</td>
<td>180,857,866</td>
<td>95,168,069</td>
<td>52.62</td>
</tr>
<tr>
<td>6</td>
<td>170,899,992</td>
<td>91,676,326</td>
<td>53.64</td>
</tr>
<tr>
<td>7</td>
<td>158,821,424</td>
<td>84,024,144</td>
<td>52.90</td>
</tr>
<tr>
<td>8</td>
<td>146,274,826</td>
<td>77,176,667</td>
<td>52.76</td>
</tr>
<tr>
<td>9</td>
<td>140,273,252</td>
<td>81,939,640</td>
<td>58.41</td>
</tr>
<tr>
<td>10</td>
<td>135,374,737</td>
<td>73,623,792</td>
<td>54.39</td>
</tr>
<tr>
<td>11</td>
<td>134,452,384</td>
<td>69,987,304</td>
<td>52.05</td>
</tr>
<tr>
<td>12</td>
<td>132,349,534</td>
<td>67,421,365</td>
<td>50.94</td>
</tr>
<tr>
<td>13</td>
<td>114,142,980</td>
<td>70,179,406</td>
<td>61.48</td>
</tr>
<tr>
<td>14</td>
<td>106,368,585</td>
<td>63,755,426</td>
<td>59.94</td>
</tr>
<tr>
<td>15</td>
<td>100,338,915</td>
<td>61,850,368</td>
<td>61.64</td>
</tr>
<tr>
<td>16</td>
<td>88,827,254</td>
<td>50,097,383</td>
<td>56.40</td>
</tr>
<tr>
<td>17</td>
<td>78,774,742</td>
<td>41,724,226</td>
<td>52.97</td>
</tr>
<tr>
<td>18</td>
<td>76,117,153</td>
<td>42,429,428</td>
<td>55.74</td>
</tr>
<tr>
<td>19</td>
<td>63,811,651</td>
<td>32,032,638</td>
<td>50.20</td>
</tr>
<tr>
<td>20</td>
<td>62,435,964</td>
<td>33,328,923</td>
<td>53.38</td>
</tr>
<tr>
<td>21</td>
<td>46,944,323</td>
<td>31,168,187</td>
<td>66.39</td>
</tr>
<tr>
<td>22</td>
<td>49,691,432</td>
<td>32,922,552</td>
<td>66.25</td>
</tr>
<tr>
<td>X</td>
<td>154,913,754</td>
<td>65,374,264</td>
<td>42.20</td>
</tr>
<tr>
<td>Y</td>
<td>57,772,954</td>
<td>41,857,553</td>
<td>72.45</td>
</tr>
</tbody>
</table>

Table 5.1.: Chromosome lengths of the hg18 assembly before and after applying *fuguization*. 
5.2.4 **Repeat reinsertion and global chain alignments**

After repeat insertion only chains were kept in which both sequences had a minimal length of 1,000 bp.

Pairwise global alignments were produced with `needle` and `stretcher` and were filtered for percent identity $\geq 90\%$ and a maximal gap percentage of $\leq 40\%$.

After filtering, a total of 29,988 pairwise duplications remained. Table 5.2 shows the number of pairwise alignments per chromosome pair, Fig. 5.1 illustrates their distribution along the chromosomes and Fig. 5.2 gives an overview of the percent identity values for each alignment.

5.2.5 **Duplication clustering**

The pairwise alignments were clustered on overlapping coordinates and merged. 7,316 segmental duplication units split across 2,238 clusters were the result. Table 5.3 gives counts for copy numbers.

A total amount of 148,180,604 (about 4.8% of the genome analysed) duplicated sequence was detected. The lengths of SDs range from 1000 bp (as per filtering) to 1,946,491 bp (Fig. 5.3). The largest segment is located on chromosome Y and contains 75 genes, the majority of which (56) are pseudogenes.

The SD counts and total amount of duplicated sequence per chromosome are given in Table 5.4.

The running times and peak memory consumption for all steps of the pipeline are given in Table 5.5. As mentioned in chapter 4, the most time-consuming step, the global alignment of final chains, can be sped up by using a heuristic alignment software. The analysis was run on about 600 CPU cores and completed within a few hours.
|    | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  | 20  | 21  | 22  | X   | Y   |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1  | 2,709 | 178 | 133 | 153 | 134 | 86  | 307 | 90  | 144 | 74  | 94  | 55  | 57  | 32  | 85  | 140 | 44  | 53  | 74  | 20  | 30  | 35  | 89  | 75  |
| 2  | 743  | 132 | 127 | 67  | 81  | 161 | 97  | 234 | 102 | 186 | 65  | 72  | 117 | 141 | 188 | 58  | 73  | 48  | 17  | 61  | 113 | 88  | 53  |
| 3  | 260  | 155 | 100 | 70  | 120 | 186 | 87  | 51  | 155 | 90  | 43  | 47  | 66  | 46  | 26  | 20  | 19  | 20  | 19  | 22  | 20  | 66  | 23  |
| 4  | 365  | 53  | 44  | 149 | 195 | 142 | 88  | 121 | 60  | 40  | 26  | 24  | 21  | 37  | 21  | 21  | 121 | 26  | 27  | 10  | 41  | 24  |
| 5  | 937  | 173 | 59  | 41  | 39  | 36  | 31  | 37  | 22  | 25  | 58  | 52  | 26  | 19  | 26  | 46  | 13  | 250 | 47  | 13  |
| 6  | 151  | 62  | 55  | 47  | 37  | 39  | 30  | 38  | 15  | 7   | 29  | 29  | 20  | 18  | 25  | 17  | 10  | 42  | 11  |
| 7  | 1,227| 92  | 130 | 100 | 150 | 56  | 37  | 35  | 82  | 251 | 80  | 35  | 40  | 54  | 29  | 24  | 41  | 37  |
| 8  | 439  | 46  | 45  | 150 | 103 | 38  | 23  | 21  | 53  | 14  | 23  | 19  | 10  | 15  | 8   | 48  | 10  |
| 9  | 1,045| 122 | 65  | 58  | 56  | 65  | 76  | 115 | 43  | 60  | 50  | 65  | 49  | 69  | 48  | 48  |
| 10 | 655  | 45  | 21  | 22  | 34  | 62  | 87  | 56  | 38  | 14  | 10  | 19  | 34  | 40  | 20  |
| 11 | 289  | 72  | 41  | 29  | 25  | 32  | 15  | 18  | 16  | 6   | 15  | 11  | 51  | 8   |
| 12 | 141  | 31  | 26  | 21  | 16  | 14  | 7   | 10  | 19  | 13  | 12  | 44  | 9   |
| 13 | 221  | 30  | 69  | 23  | 12  | 23  | 8   | 1   | 40  | 25  | 39  | 19  |
| 14 | 245  | 66  | 34  | 19  | 30  | 4   | 10  | 23  | 53  | 20  | 4   |
| 15 | 1,460| 187 | 41  | 25  | 21  | 12  | 17  | 43  | 65  | 285 |
| 16 | 973  | 68  | 80  | 19  | 3   | 20  | 49  | 35  | 34  |
| 17 | 743  | 15  | 16  | 24  | 6   | 18  | 27  | 4   |
| 18 | 59   | 10  | 5   | 38  | 21  | 16  | 6   |
| 19 | 468  | 10  | 5   | 10  | 28  | 5   |
| 20 | 47   | 2   | 36  | 10  | 5   |
| 21 | 22   | 28  | 3   |
| 22 | 272  | 10  | 18  |
| X  | 854  | 204 |
| Y  | 837  |
Figure 5.1.: Circular plot showing the pairwise SD alignments of the hg18 assembly as arcs between chromosomes. The different colours of the arcs encode the sequence lengths: light blue (5,001-20,000 bp), green (20,001-50,000 bp) and red (>50,000 bp).
Figure 5.2.: Histogram of percent identity values for each pairwise alignment of the hg18 SD analysis. Values for interchromosomal duplications are shown in light grey, values for intrachromosomal duplications in dark grey.
Figure 5.3.: Histogram of SD lengths (in bp) for the hg18 SD analysis. Note that the x-axis is on log10 scale.
5.3 COMPARISON WITH PUBLISHED RESULTS

<table>
<thead>
<tr>
<th>cluster size</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>1,626</td>
</tr>
<tr>
<td>3</td>
<td>264</td>
</tr>
<tr>
<td>4</td>
<td>97</td>
</tr>
<tr>
<td>5-9</td>
<td>107</td>
</tr>
<tr>
<td>10-19</td>
<td>26</td>
</tr>
<tr>
<td>20-49</td>
<td>12</td>
</tr>
<tr>
<td>50-100</td>
<td>5</td>
</tr>
<tr>
<td>&gt;100</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.3.: Histogram of number of duplicons per SD cluster (cluster size) of the hg18 assembly. In most cases the cluster size corresponds to the copy number of the according SD.

5.3 COMPARISON WITH PUBLISHED RESULTS

Several studies have published detailed analysis on human segmental duplications (Bailey et al. 2002; Cheung et al. 2003a; Zhang et al. 2005). Using different assembly versions and methods, they report about 3.5% to 5% duplicated sequence in total. 5% is a number typically quoted and is close to the 4.8% that the analysis of the work at hand detected.

The amount of duplicated sequence has been reported to vary greatly between the different chromosomes, from about 1.5% to 27%, which is consistent with the results obtained for this thesis (Table 5.4). In (Bailey and Eichler 2006) it is reported that chromosomes 7, 9, 10, 15, 16, 17, 22, X and Y are the most enriched chromosomes for duplications, again consistent with the results that I obtain.

Compared to Drosophila there are far more inter-chromosomal duplications in the human genome. In the study quoted by (Bailey and Eichler 2006), almost half of the alignments are inter-chromosomal (very similar to what
<table>
<thead>
<tr>
<th>chromosome</th>
<th>count</th>
<th>total number bp</th>
<th>fraction of chrom (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>572</td>
<td>11,684,751</td>
<td>4.73</td>
</tr>
<tr>
<td>2</td>
<td>439</td>
<td>9,450,760</td>
<td>3.89</td>
</tr>
<tr>
<td>3</td>
<td>350</td>
<td>2,978,337</td>
<td>1.49</td>
</tr>
<tr>
<td>4</td>
<td>284</td>
<td>4,596,486</td>
<td>2.40</td>
</tr>
<tr>
<td>5</td>
<td>341</td>
<td>5,686,175</td>
<td>3.14</td>
</tr>
<tr>
<td>6</td>
<td>321</td>
<td>3,225,381</td>
<td>1.89</td>
</tr>
<tr>
<td>7</td>
<td>568</td>
<td>12,532,560</td>
<td>7.89</td>
</tr>
<tr>
<td>8</td>
<td>214</td>
<td>2,837,237</td>
<td>1.94</td>
</tr>
<tr>
<td>9</td>
<td>283</td>
<td>14,195,687</td>
<td>10.12</td>
</tr>
<tr>
<td>10</td>
<td>380</td>
<td>8,734,962</td>
<td>6.45</td>
</tr>
<tr>
<td>11</td>
<td>285</td>
<td>5,201,058</td>
<td>3.87</td>
</tr>
<tr>
<td>12</td>
<td>293</td>
<td>2,757,363</td>
<td>2.08</td>
</tr>
<tr>
<td>13</td>
<td>226</td>
<td>2,834,719</td>
<td>2.48</td>
</tr>
<tr>
<td>14</td>
<td>189</td>
<td>2,664,097</td>
<td>2.50</td>
</tr>
<tr>
<td>15</td>
<td>315</td>
<td>7,919,435</td>
<td>7.89</td>
</tr>
<tr>
<td>16</td>
<td>231</td>
<td>7,700,209</td>
<td>8.67</td>
</tr>
<tr>
<td>17</td>
<td>388</td>
<td>6,658,364</td>
<td>8.45</td>
</tr>
<tr>
<td>18</td>
<td>112</td>
<td>1,804,892</td>
<td>2.37</td>
</tr>
<tr>
<td>19</td>
<td>317</td>
<td>3,759,184</td>
<td>5.89</td>
</tr>
<tr>
<td>20</td>
<td>108</td>
<td>1,421,632</td>
<td>2.28</td>
</tr>
<tr>
<td>21</td>
<td>83</td>
<td>1,794,537</td>
<td>3.82</td>
</tr>
<tr>
<td>22</td>
<td>219</td>
<td>3,987,613</td>
<td>8.02</td>
</tr>
<tr>
<td>X</td>
<td>598</td>
<td>10,335,818</td>
<td>6.67</td>
</tr>
<tr>
<td>Y</td>
<td>200</td>
<td>13,419,347</td>
<td>23.23</td>
</tr>
</tbody>
</table>

Table 5.4.: SD counts and extent (in bp) for each chromosome of the hg18 assembly.
Table 5.5: Run times (in minutes) and peak memory consumption (in MB) for the hg18 SD analysis.

<table>
<thead>
<tr>
<th>step</th>
<th>time (min)</th>
<th>memory peak (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuguization of chromosome sequences</td>
<td>4.70</td>
<td>1240</td>
</tr>
<tr>
<td>Indexing of sequences</td>
<td>9.56</td>
<td>1626</td>
</tr>
<tr>
<td>MEM computation</td>
<td>1555.17</td>
<td>1669</td>
</tr>
<tr>
<td>First chaining</td>
<td>156.58</td>
<td>1456</td>
</tr>
<tr>
<td>Second chaining</td>
<td>0.32</td>
<td>4</td>
</tr>
<tr>
<td>Extraction and uncompacting of sequences</td>
<td>729.71</td>
<td>1964</td>
</tr>
<tr>
<td>Global alignments of final chains</td>
<td>3186.98</td>
<td>766</td>
</tr>
<tr>
<td>Filtering of final chains</td>
<td>4.85</td>
<td>10</td>
</tr>
<tr>
<td>Clustering of duplications</td>
<td>0.46</td>
<td>31</td>
</tr>
</tbody>
</table>

can be observed in Table 5.2. Most of the duplications with the highest sequence identity, however, are intra-chromosomal SDs and have been associated with a recent expansion in the great ape lineage (She et al. 2006). Fig. 5.2 is clearly in line with this finding.

As for enrichment of duplications along chromosomes, duplication hubs were observed and particularly pericentromeric and subtelomeric regions were found to have high duplication frequencies (Zhang et al. 2005; Bailey and Eichler 2006). This can be observed in Fig. 5.1.

Finally, segmental duplication coordinates were obtained from the Human Segmental Duplication Database (refer to appendix B.2) and compared to the results of this work (Fig. 5.4). To a large extend, both methods discover the same set of duplications. The WGAC method, however, with the parameters used, detects an overall larger set of duplications. Notably there is a considerable number of duplications found by the method presented here and not by WGAC.

For those SDs that were only detected by one of the two methods, the overlap with known genes was computed. Duplication coordinates were uploaded
Figure 5.4: Venn diagram showing the overlap of SDs detected for this work (a) with data from the Segmental Duplication DB (b). The intersection gives two numbers: 6687 for this work and 6389 for the Segmental Duplication DB data. The reason for this is that multiple (fragmented) SDs can overlap a single (merged) SD.
Table 5.6.: Count of genes and their isoforms that overlap with SDs that were either detected by this work were taken from the Segmental Duplication DB. Counts are given in absolute numbers and normalised by the number of SDs.

to the Biomart Ensembl web server and compared to Ensembl release 54 NCBI36 Homo sapiens genes set. Table 5.6 gives the counts of genes and isoforms overlapping the SDs. Both methods detect a considerable number of unique SDs that overlap genes. As discussed in chapter 4, follow-up work will be carried out to study the influence of parameters on sensitivity and if big differences in the results of the two approaches remain, those methods should then be used in parallel, complementing each other.
READ-BASED DETECTION AND GENOTYPING METHOD

This chapter describes a method to detect polymorphic segmental duplications in sequencing read data, bypassing any assembly or reference mapping step. For data of a single individual, this method genotypes that particular individual for given duplications. When applied to data of multiple individuals, allele frequencies can be derived for the duplications that are analysed.

The core idea is to compile duplicon-specific probes of known duplications and align such probes to read sequences. In more detail, the method works as follows:

1. Pairwise alignments of known duplications are analysed for short, dissimilar stretches that have the potential to uniquely tag a particular region of a duplicon.

2. Such “probe candidates” are extracted from the alignment and then aligned against the whole reference genome to ensure their uniqueness.

3. Probe candidates that align more than once, i.e. to other loci other than their own, are discarded.

4. The remaining “tag probes” are aligned to short read sequences.

An overview of the process is illustrated in Fig. 6.1.
Figure 6.1.: Overview of the read-based SD detection method.
6.1 NOMINATION OF PROBE CANDIDATES

In the first step pairwise alignments of known duplications are analysed to find short substrings that are unique to one of the sequences. Such pairwise alignments can be the alignments of the final chains from the assembly-based method for example (refer to chapter 3). Also, databases such as the Segmental Duplication Database provide the coordinates of pairwise alignments of duplications, which can be used to construct global alignments.

A Python script was written that reads in an alignment in needle or stretcher format. It then outputs all substrings of a given length that have a minimal, given edit distance to the aligned substring. Fig. 6.2 illustrates this.

![Figure 6.2: Nomination of probe candidates. 2 probe candidates of length 10 and edit distance 3 are shown in grey rectangles. DNA bases are depicted as coloured circles, gaps as small, black circles, matches as vertical bars and mismatches and indels as red crosses.](image)

6.2 FILTERING OF PROBE CANDIDATES

Once the probe candidates have been compiled, they are aligned against a complete reference genome sequence. This step is necessary to ensure the probe is unique across the genome and hence diagnostic for a given duplicon. The short read mapping software bwa is used for efficient alignment. In downstream analysis, probes will be aligned to short read sequences. In
that step one typically wants to allow some mismatches and indels in the alignment due to sequencing errors and individual variation. Therefore, in the filtering phase one has to make sure that the second best alignment of a probe (i.e. the one next to the exact match to its own locus) has a higher edit distance than the edit distance allowed during the detection phase. To illustrate this, given probes of length 25 bp, one might want to allow an edit distance of 1 when aligning to short reads. In the filtering phase if the second best alignment of a given probe is either a perfect match or has an edit distance of 1 to the reference, then this probe has to be discarded as it would not be diagnostic.

6.3 Alignment to short reads

In the final step all remaining tag probes are aligned to a set of sequencing reads. Again, bwa is used, this time treating the reads as a collection of references and mapping the probes to them. As discussed before, the maximal edit distance allowed in this step has to be set to a lower value than the threshold used for the filtering phase.

When used in a large-scale scenario bwa produces a large amount of output. To avoid storing these data, bwa is instructed to write its output to the terminal and a custom Python script is used to read in this stream, only keeping track of the number of times a probe aligns to a read.

The final output of the method is an alignment count for every probe, i.e. how often this probe was detected in the short read data. A probe with a count greater than zero indicates that the duplicon region that particular probe tags, is present. It is important to note that a count of zero does not necessarily mean that the tagged region is absent from the genome of the analysed individual, instead it could also be due to missing sequencing data or a large number of mutations in that region.
6.4 USE OF PROBE COUNTS IN ASSOCIATION STUDIES

Using this method on data of multiple individuals, SDs that are polymorphic in a population can be determined along with an estimation of their allele frequency. Furthermore, the counts for a probe across individuals can then be associated to other genotype or phenotype data.

Given that many association studies have already discovered SNPs that have a strong correlation to diseases and other phenotypes, one particular interesting analysis is how these SNPs in turn associate with duplicated segments. Given that duplications have been already linked to various diseases, it is probable that a subset of SNPs that have been associated with a some phenotype are actually not the causal allele, but are simply in LD with a duplication that confers the phenotype. Therefore, association of a putative SNP marker for a disease, or any other phenotype, to duplications has the potential to reveal a duplication in LD with the marker as the true causal allele.

In the case of individuals which have been both sequenced and measured for phenotypes, such as the fruit fly lines of the DGRP, one can directly associate the SD probes to those phenotypes.
READ-BASED DETECTION AND GENOTYPING RESULTS

7.1 DROSOPHILA MELANOGASTER: HIGH COVERAGE DATA FROM THE DGRP

The read-based segmental duplication detection and genotyping method was tested on sequence data of 176 inbred Drosophila Melanogaster lines from the DGRP.

Three association studies were performed on the resulting alignment counts of the probes: 1. association of probes to each other 2. association of probes to known SNPs and 3. association of probes to three phenotypes (chill coma, startle response and starvation resistance).

7.1.1 Input

Whole genome sequencing files, comprising a total of 716 Gb raw sequence, were obtained from ENA (refer to appendix B.3). All lines have been sequenced to high depth, with a minimal sequencing coverage of 11x (Fig. 7.1). For such high sequencing coverage of an isogenic genome (i.e. lines are homozygous for every variant), it is expected to achieve a genome coverage of more than 99.99%. Fig. 7.2 plots sequencing coverage against expected genome coverage.
Figure 7.1.: Histogram of sequencing coverages of the 176 DGRP lines used in this work.
Figure 7.2: Sequencing coverage plotted against expected genome coverage, assuming that a Poisson process underlies the sampling of sequencing reads. The genome coverage reaches 99.99% at 10x sequencing coverage.
Probes candidates of length 25 were extracted from the 3386 pairwise alignments that resulted from the assembly-based analysis described in chapter 4. Probe candidates were then aligned to the Drosophila melanogaster dm2 assembly sequence, including the available heterochromatic and unmapped sequence. Probes were filtered for having a single alignment of edit distance <2 across the whole genome. This resulted in a total of 286,122 probes.

Using bwa and a maximal edit distance of 1, these tag probes were then aligned to all sequencing read data of the 176 Drosophila melanogaster DGRP lines. For each probe and line, the total number of alignments found was recorded and normalized by the sequence coverage of that particular line. Fig. 7.3 shows the distribution of non-zero to zero count ratios of probes (taking the minimum of the non-zero to zero count ratio and its reciprocal) for all probes with a least one zero count, i.e. excluding probes of sequences fixed across all individuals. For probes with only true non-zero (tagged segment is present in the genome and at least one alignment was found) and true zero (tagged segment is absent in the genome and no alignment was found) counts, this ratio corresponds to a minor allele frequency (MAF). Fig. 7.4 illustrates an expected MAF spectrum with a characteristic exponential decay. The empirical MAF distribution of probes matches this expected MAF spectrum quite closely. For simplicity the ratio of non-zero to zero counts will be referred to as MAF, ignoring false non-zero and false zero counts.

All probes with MAF <0.05 were discarded for all subsequent association studies. This left a total of 75,167 probes.
Figure 7.3.: Minor allele frequency (MAF) spectrum of the dm2 SD probes.
Figure 7.4.: Expected MAF spectrum for 176 individuals under a constant effective population size neutral model. This model predicts that variants with allele count $k$ are observed in a sample with frequency $\theta/k$, where $\theta$ is the population-scaled mutation rate. Here, $\theta$ is set to an arbitrary value and the y-axis shows relative frequencies.
7.1 DROSOPHILA MELANOGASTER: HIGH COVERAGE DATA FROM THE DGRP

7.1.3 Association between probes

To assess how consistently different probes tag a given segment and to carry out some controls, the normalized alignment counts of probes were correlated to each other. In particular three classes were investigated:

1. Probes of the same duplication cluster and the same duplicon: about 28 million pairs of alignment counts were sampled and correlated.

2. Probes of the same duplication cluster but different duplicon. In this class about 18 million pairs of alignment counts were sampled and correlated.

3. Probes of different duplication clusters: about 38 million pairs were sampled and correlated.

Figs. 7.5-7.7 plot the sampled alignment counts for pairs of probes.

Probes of the same cluster and the same duplicon

In this class, generally, strong, positive correlation is expected. Along a duplicon, different tag probes should align a similar number of times across individuals. As Fig. 7.5 shows counts of probes of the same duplicon generally correlate well, however an appreciable proportion of probe counts with weak correlation is present as well. The overall Pearson correlation coefficient of the sampled count pairs is 0.37.

Weak correlation can have several sources:

1. In more complex scenarios, duplicons are merged across various duplications events. In particular, hotspots can become merged into one broad duplicon. Such a segment will harbour multiple independent events with different degrees of polymorphism across individuals.
Figure 7.5.: Scatter plot with smoothed densities of alignment counts for pairs of probes of the same SD cluster and same duplicon. Density is shown as a gradient from blue (low density) over orange (medium density) to red (high density).
7.1 Drosophila melanogaster: High Coverage Data from the DGRP

Figure 7.6: Scatter plot with smoothed densities of alignment counts for pairs of probes of the same SD cluster but different duplcon. Density is shown as a gradient from blue (low density) over orange (medium density) to red (high density).
Figure 7.7: Scatter plot with smoothed densities of alignment counts for pairs of probes of different SD clusters. Density is shown as a gradient from blue (low density) over orange (medium density) to red (high density).
2. Both during chaining as well as clustering, nearby tandem duplications are often merged into one duplicon. Such segments, again, might contain parts with different degrees of polymorphism.

3. Genomic rearrangements, e.g. deletions, within a duplicon in a subset of individuals. Probes covering such a rearrangement will have different signatures to probes outside the rearrangement.

4. Multiple false non-zero and/or false zero alignment counts across a single probe. Given the high sequencing coverage across all individuals, this scenario is expected to occur rarely.

*Probes of the same cluster but different duplicon*

In this class, different scenarios are expected, depending on the underlying series of duplication events and allele frequencies of polymorphic copies. Fig. 7.6 shows that the overall correlation is weaker compared to the set of probe pairs of same duplicons (Fig. 7.5). The Pearson correlation coefficient of the sampled probe count pairs is 0.26.

*Probes of different clusters*

In this class, probe pairs are expected to have no correlation as duplications of different clusters should be independent from each other. As Fig. 7.7 illustrates, this holds true for the analysed data. The Pearson correlation coefficient of the probe count pairs is 0.02.

**7.1.4 Association of probes to SNPs**

For this analysis, LD between probes and known SNPs was studied. The SNPs of the DGRP July 2010 data freeze (refer to appendix B.3) were filtered for SNPs with a MAF \( \geq 5\% \) and having no ambiguous bases which resulted in a set of 99,749 SNPs used for a first round of association. Only 162 DGRP
lines out of the 176 lines used for probe detection had available SNP data and were used for this analysis.

Probe coordinates were converted from the dm2 assembly to dm3 using the sequence coordinates converter web tool from Flybase (Flybase: Sequence Coordinates Converter). A total of 72,217 probes could be converted.

For every probe, correlations were carried out with SNPs 10 kb upstream of the probe start and 10 kb downstream of the probe end. 44,972 probes (62% of the total) had at least one SNP in the 20 kb window. In total 785,273 Spearman Rank Tests were carried out.

To obtain false discovery rates (FDRs; the expected proportion of false positives) 10,000 permutation runs were computed. For each permutation run, one random order of probe counts was fixed. In the end, the most significant p-value was chosen for every probe-to-SNP correlation across all 10,000 runs. These values were then used to derive FDRs. For a given p-value threshold, let n be the number of permutation correlations above the threshold and r the number of correlations of the real associations above the threshold. The FDR is then \( \frac{n}{n+r} \). Fig. 7.8 plots FDRs against different \( -\log_{10} \) p-value thresholds. A \( -\log_{10} \) p-value threshold of 8.2 was chosen, corresponding to an FDR of 5%. With this threshold 5,915 probes were identified with at least one strongly correlating SNP (13.15% of the probes with a at least one SNP in the 20 kb window). In view of the facts that the analysis was based on sparse, filtered SNP data and that generally little LD is observed for distances above 1-2 kb in *Drosophila melanogaster* (Takano-Shimizu et al. 2004; Itoh et al. 2010), the analysis was re-done using regions 1 kb upstream from probe starts and 1 kb downstream from probe ends. 3,475 probes were detected with at least one significant correlation (with an FDR of 5%) which corresponds to 21.57% of all probes with at least one SNP in the 2 kb window.

Fig. 7.9 shows the distance of probe to SNP for each significant correlation and Fig. 7.10 plots those distances against the corresponding p-values. The number of SNPs and p-values decrease with longer distances. This is expected as LD decays with distance.
Figure 7.8: FDRs for different $-\log_{10} p$-value thresholds for the correlations of dm2 SD probes with nearby SNPs.
Figure 7.9.: Distances of probe to SNP for all significant correlations of dm2 SD probes with nearby SNP.
Figure 7.10.: Distance (in bp) vs. $-\log_{10}$ p-value for all significant correlations of dm2 SD probes with nearby SNPs.
In order to be able to interpret the low proportion of probes having strong correlation to a nearby SNP, the same experimental setup was used to associate SNPs to other SNPs. More specifically, each of the 99,749 filtered SNPs that were used for the association to probes was correlated to all SNPs 10 kb upstream and 10 kb downstream of it. 99,285 SNPs had at least one other SNP in the 20 kb window and a total of 4,314,674 correlations were computed.

10,000 permutation runs were performed to obtain FDRs. Fig. 7.11 plots FDRs for different p-value thresholds. When using a $-\log_{10} p$-value threshold of 11.5, corresponding to an FDR of 5%, 73,770 SNPs (74.3%) had at least one significant association to another SNP nearby, a far higher proportion than observed for significant correlation between probes and SNPs. Fig. 7.12 plots the distance of SNPs with significant correlation, while Fig. 7.13 shows the p-values of the significant correlations as a function of distance. In latter plot, all perfect correlations with Spearman $\rho$ 1 or -1 were excluded as no p-values were reported for those. Fig. 7.14 shows the distance of SNPs with perfect correlation.

Finally, probes of chromosome arm 3L were associated with a much denser set of SNPs of the same chromosome. For this analysis, SNPs were filtered for MAF $\geq$ 5% and additionally up to 10% ambiguous bases were allowed. In the associations, the ambiguous bases and corresponding probe counts were removed. This set of chromosome 3L SNPs contained 382,714 SNPs, with an average density of one SNP every 64 bp.

To obtain FDRs, 10,000 permutation runs were carried out. Fig. 7.15 plots FDRs for different p-value thresholds. A $-\log_{10} p$-value threshold was chosen which corresponds to an FDR of 5%. Fig. 7.16 shows the distribution of distances between probes and strongly correlated SNPs and Fig. 7.17 plots these distances against the corresponding p-values.

3,283 out of 8,022 (about 41%) probes with nearby SNPs had at least one significant correlation with a SNP.

To exclude probes with false non-zero or false zero counts, the set of probes was restricting to those having at least 10 strong, positive correlations.
Figure 7.11.: FDRs for different $-\log_{10}$ p-value thresholds for the correlation of SNPs to other, nearby SNPs.
Figure 7.12.: Histogram of distances (in bp) between nearby SNPs having a significant correlation.
Figure 7.13.: Distance (in bp) vs. $-\log_{10}$ p-value for all significant correlations between nearby SNPs. Points were binned using hexagonal binning. Perfect correlations (Spearman’s $\rho$ of 1 or -1) are excluded due to missing p-values.
Figure 7.14.: Histogram of distances (in bp) between nearby SNPs having a perfect correlation (Spearman’s ρ of 1 or -1).
Figure 7.15: FDRs for different $-\log_{10}$ p-value thresholds for the correlations of dm2 chr3L SD probes with nearby SNPs.
Figure 7.16.: Distances of probes to SNPs for all significant correlations of dm2 chr3L SD probes with nearby SNPs.
Figure 7.17.: Distance (in bp) vs. $-\log_{10} p$-value for all significant correlations of dm2 chr3L SD probes with nearby SNPs. Points were binned using hexagonal binning.
(Spearman’s $\rho > 0.9$) to other probes of the same duplicon. However, no higher proportion of probes in strong LD with a SNP was obtained.

Discussion

Using a dense set of SNPs, a considerable number of probes (41%) show a significant association with at least one nearby SNP. Such data provide a valuable resource, facilitating the imputation of duplications into other genotype data. Nevertheless, a large number of apparently non-imputable duplications remains. In particular, this proportion is much bigger than for the associations between SNPs, even when using relatively sparse SNP data. Several sources for this mismatch are possible:

1. Complex multi-copy duplications: for multi-copy duplications, a tag sequence that is unique in one individual (the reference) can be present in multiple copies, thereby tagging independent duplications. This will lead to a confounding probe count signature as counts are summed over the different events.

2. Gene conversion: tagged sequence can be transferred from one duplcon to others via gene conversion (Chen et al. 2007). Similar to complex multi-copy duplications, this will lead to a confounding probe count signature.

3. Difficult SNP calling: by far the highest proportion of significant correlations between SNPs are over very short distance (a few hundred bp; refer to Fig. 7.12 and Fig. 7.14). However, identification of SNPs in the near vicinity of any structural variant is difficult due to poorly mapping sequencing reads. This can lead to a reduced number of SNP calls and/or poor quality genotype data in these regions, where the power to detect an association is the highest. This may explain the reduced proportion of detected short-range associations between probes and SNPs (Fig. 7.16).
7.1.5 Association to three phenotypes

Based on the discussion above, correlating phenotypes directly with SDs without imputing them from SNP data is likely to increase the sensitivity of an analysis as some SDs might not be in strong LD with a known SNP.

For this work, SD probes were associated with quantitative phenotype data obtained from the Trudy Mackay lab (NCSU). The following phenotypes were measured across 168 DGRP lines in both male and female fruit flies:

1. Chill-coma
2. Startle response
3. Starvation resistance

The mean values of the phenotype measurements were correlated to the normalized probe counts using Spearman’s Rank Correlation Test. The data of 11 lines were removed due to missing data points, leaving the data of 157 lines. A total of 459,979 correlations were carried out.

Fig. 7.18 shows the p-value distribution of the results.

10,000 permutations were computed to obtain FDRs. Fig. 7.19 plots p-value thresholds against corresponding FDR. No significant correlations could be detected with any reasonable FDR. This suggests that no polymorphic SDs for which tag probes were obtained nor any SNPs in linkage disequilibrium with the SDs are responsible for differences in the three phenotypes studied.

Besides the three phenotypes studied for this work, measurements of hundreds of additional phenotypes have been announced for the near future (BCM: DGRP). The read-based SD detection and genotyping method will allow for direct comparison of these phenotypes to known SDs once the data become available.
Figure 7.18: Boxplots of p-values for the correlations of dm2 SD probes with three phenotypes: starvation resistance (starv), startle response (startle) and chill coma (chill). Distributions are shown for both male (m) and female (f) fruit flies.
Figure 7.19.: FDRs for different $-\log_{10}$ p-value thresholds for the correlations of dm2 SD probes with three phenotypes.
7.2 HOMO SAPIENS: LOW COVERAGE DATA FROM THE 1000 GENOMES PROJECT

The read-based segmental duplication detection and genotyping method was additionally tested on data of 86 human individuals from the 1000 Genomes project.

Two association studies were performed on the resulting alignment counts of the probes: 1. association of probes to each other and 2. association of probes to known SNPs.

7.2.1 Input

The individuals to analyse were chosen as being part of the HapMap project (The International HapMap Consortium 2003), having low coverage sequencing data available from the 1000 Genomes project and having one ancestry (Northern and Western European).

Sequencing data were obtained from the EBI 1000 Genomes FTP server (refer to appendix B.3) totalling 1.57 Tb raw sequence data. The individuals studied have been generally sequenced to low coverage with a minimal sequencing coverage of 2x. (Fig. 7.20). Fig. 7.21 compares the sequencing coverages of the DGRP data set analysed with the 1000 Genomes data. Given that most genomes were sequenced to low coverage and that the human individuals are heterozygous, this analysis is expected to be far more challenging than for the DGRP individuals.

7.2.2 Analysis

The pairwise alignments resulting from the analysis described in chapter 5 were not used for extracting probe candidates. Instead the results from an
Figure 7.20.: Histogram of 1000 Genomes data sequencing coverages.
Figure 7.21.: Boxplots comparing the 1000 Genomes data sequencing coverages with the DGRP data sequencing coverages.
earlier run were used. This particular analysis proceeded in four steps as follows:

1. Computation of rare4-MEMs of all uncompacted (i.e. no fuguization was undertaken) chromosome sequence pairs of the hg18 assembly.

2. Chaining of rare4-MEMs, selecting chains of minimal length 34 bp using a gap constraint of 300 bp and a fragment weighting factor of 9.

3. Recursive chaining using a gap constraint of 6 kb and a weighting factor of 20, filtering for minimal chain length of 1 kb.

4. All remaining chains were analysed for common repeats using annotation from RepeatMasker. Chains for which any of the two sequences were made up of more than 95% common repeats were removed.

This analysis resulted in 25,469 pairwise alignments of final chains, 8,908 SDs and 2,561 duplication clusters.

Probe candidates of length 30 were extracted from the 25,469 pairwise alignments and aligned to the hg18 human reference assembly sequence. Probes with a second-best hit of edit distance <3 were discarded. The result were 18,908,089 tag probes.

Tag probes were aligned to all sequencing read data of the 86 1000 Genomes individuals allowing a maximal edit distance of 2. Resulting alignment counts were normalised by the according sequencing coverage.

Fig. 7.22 shows the distribution of non-zero to zero count ratios of probes with at least one zero count (taking the minimum of the non-zero to zero ratio and its reciprocal). In the case of Drosophila Melanogaster the frequency spectrum matched closely the expected MAF spectrum with an exponential decay. As discussed above, this is expected for high-coverage sequencing data. In the human spectrum, there seems to be a considerable amount of false zero counts due to missing sequencing data. This can be seen in Fig. 7.22 as inflated frequency values, particularly noticeable for the frequency range between 0.1 and 0.25. Despite the apparently large number of false
Figure 7.22.: Minor allele frequency (MAF) spectrum of the hg18 SD probes.
probe counts, the ratio of counts will still be referred to as MAF for sake of simplicity.

Fig. 7.2.3 illustrates how 1000 Genomes low coverage data is expected to skew the MAF spectrum due to missing sequencing data. Two scenarios were simulated using the same number of individuals and sequencing coverages as the original data.

Figure 7.2.3.: MAF simulation of 1000 Genomes data. (a) Assuming that all individuals are homozygous for a tagged segment (b) Assuming all individuals are heterozygous for a tagged segment (coverages are halved).
In one scenario it is assumed that probes tag sequences that are fixed and homozygous across all individuals. A virtual sequencing experiment is computed whereby for the 86 individuals reads are generated from a genome of 1 Mb length. The number of reads is calculated from the according sequencing coverage of that individual’s original data. Then, 10,000 probes of 20 bp are randomly generated from the 1 Mb genome and for every probe alignment counts are recorded. Probe MAFs are shown in Fig. 7.23(a).

In a second simulation it is assumed that all individuals carry one and only one copy of a tagged segment, i.e. they are heterozygous for it. In this scenario all coverages are halved. Resulting probe MAFs are shown in Fig. 7.23(b).

Both simulated scenarios show that the low coverage data will produce a considerable number of false zero counts, especially in the case of heterozygous alleles.

Probes with a MAF $\geq 5\%$ were used for further analysis. This set comprised 14,451,694 probes.

### 7.2.3 Association between probes

As for *Drosophila melanogaster*, normalised counts of probe alignments were correlated to each other using Spearman Rank Tests. Again, following 3 classes were investigated:

1. Probes of the same duplication cluster and the same duplcon.
2. Probes of the same duplication cluster but different duplcon.
3. Probes of different duplication clusters.

For each class 20 million probe pairs were randomly sampled. Fig. 7.24 shows the distribution of Spearman’s $\rho$ correlation coefficients.
Figure 7.24. Density plots of Spearman’s $\rho$ correlation coefficient for three classes: a) same duplication cluster and same duplicon b) same duplication cluster but different duplicon c) different duplication cluster.
Notably, there are much fewer probes with strong, positive correlation across a given duplkon compared to Drosophila, as illustrated in Fig. 7.24(a). This is due to the high amount of false probes given the low coverage, as discussed before. In addition, there are many more large duplications and hotspots found in humans, potentially harbouring many independent duplications. As discussed earlier, this can lead to probes with weak correlation as well. The mean Spearman’s ρ values of the 3 classes are 0.2771, 0.2514 and 0.2216, a trend which is in line with the results obtained for Drosophila melanogaster.

7.2.4 Association to SNPs

Genotype data for the 1000 Genomes individuals were downloaded from the EBI 1000 Genomes FTP server (refer to appendix B.3). Data were available for 79 out of the 86 individuals for which probe detection had been carried out. SNPs with MAF < 0.05 were discarded. This left 6,483,639 SNPs in total. The 98,027 probes of chromosome 21 were used for association with the 103,498 probes of the same chromosome.

Probe coordinates were converted from the hg18 assembly to hg19 using the batch coordinate conversion (LiftOver) web tool of the UCSC Genome Browser (UCSC: LiftOver).

For every probe, correlations were computed between its alignment counts and the genotypes of every SNP 100 kb upstream of the probe start and 100 kb downstream of the probe end. For all 103,498 probes there was at least one SNP in the 200 kb window with a mean of 636 SNPs. A total of 65,899,316 correlations were computed.

To obtain FDRs, 1,000 permutation runs were carried out. Fig. 7.25 plots FDRs for different $-\log_{10}$ p-value cutoffs.

A $-\log_{10}$ p-value threshold of 14.6 was chosen, corresponding to an FDR of 5%. In total, 46 probes had at least one significant correlation, with a
Figure 7.25.: FDRs for different $-\log_{10}$ p-value thresholds for the correlations of hg18 SD probes with nearby SNPs.
total of 214 significant correlations (mean of 4.65 significant correlations per probe).

Fig. 7.26 shows the distribution of distances of probe to SNP for all significant correlations, while Fig. 7.27 plots these distances against the p-values of the according associations.

Figure 7.26.: Distances of probes to SNPs for all significant correlations of hg18 SD probes with nearby SNPs.
Figure 7.27.: Distance (in bp) vs. $-\log_{10}$ p-value for all significant correlations of hg18 SD probes with nearby SNPs.
7.2.5 Discussion

The low sequencing coverage and the heterozygous nature of the human data make the genotyping strategy presented here challenging. Due to missing data from the sequencing experiments many probes will generate false zero counts, thereby creating a false signal.

Some polymorphic duplications in strong LD with SNPs could be detected in the 1000 Genomes data, but these still were much less numerous than for the high-coverage homozygous *Drosophila melanogaster* lines.

The correlation analysis between probes is confounded by the nature of the 1000 Genomes data as well. Nevertheless, the basic trend (mean values of Spearman’s $\rho$) is consistent with the results obtained for *Drosophila melanogaster*. It might be possible to leverage the inter-probe correlation information and combine probe counts across different probes. This is planned as future work. The method in its current form, however, is largely limited to high-coverage data. With many more emerging high-coverage datasets (including new 1000 Genomes data), this limitation is likely to become less critical in the near future.
Part III

CONCLUSION
For this thesis two main projects were carried out. The first was to tackle the enormous, ongoing accumulation of DNA sequencing data by means of data compression algorithms. The second was to engineer methods for the large-scale detection and genotyping of segmental duplications in assembled sequences and sequencing read data.

For the first project a novel compression framework for DNA sequencing data, termed reference-based compression, was designed and implemented. This framework exploits the redundancy that sequencing reads exhibit to assembled references. Read sequences can thus be efficiently represented as pointers (positional offsets) into a reference and a list of variation to the reference. The compression method was applied to simulated sequencing read data of a variety of length, error rate and sequencing coverage. For reads that align to the reference with few or no differences, the compression of simulated data was as efficient as 0.02 bits/base. Storing only sequence information, 96-98% data savings were achieved on experimental data sets compared to the de facto standard storage format BAM. For storing base quality scores, the concept of a quality budget is introduced. This is a lossy compression scheme which only retains quality scores of a subset of bases and encodes those with efficient Huffman coding. For a given data set an informed decision can then be made how large the quality budget should be and how it should be distributed.
A crucial property of the compression scheme is that it is more efficient for longer read lengths. This is a promising feature as read lengths are steadily increasing with new rounds of sequencer models. The concepts and algorithms of this work form the core of an ongoing project to engineer production ready reference-based compression software, the CRAM toolkit. This software is being developed at the EMBL-EBI/ENA. At the time of writing all features of the prototype that was written for the work at hand have been re-implemented in the Java programming language. This has led to a speed-up of an order of magnitude compared to the prototype. Some additional feature have been implemented as well, such as handling of paired-end reads in BAM files. Yet other features, such as a file index for rapid random access or support of fragmented reads, such as strobe sequencing reads from Pacific Biosciences (Ritz et al. 2010), will be implemented in the near future. In addition, a complementary method to reference-based compression is planned that processes mapped reads and characterises regions on the reference based on coverage and error/variation distribution of the reads. This information can then be used as a pre-processing step to the compression to identify the following features allowing for an even more efficient compression:

1. Variation that occurs consistently in most or all reads. In this case, instead of storing this variation multiple times it can be recorded only once. This can be viewed as storing an edit operation on the reference, making it consistent with the read sequences.

2. High-confidence sequencing read errors, i.e. mismatched bases with low quality scores. Such read positions can then be treated as a match to the reference and can be excluded from the quality budget.

For the second project, an efficient, scalable pipeline for SD detection in assembly sequences was engineered and a method was developed to detect and genotype SDs from sequencing read data.

The assembly-based method combines different conceptual ideas from published methods into a comprehensive pipeline. It covers handling of common repeats and large indels and offers the important post-processing step of duplication clustering, which allows the identification of duplication
hotspots and multistep duplications such as the formation of multigene clusters. The pipeline was run on the *Drosophila melanogaster* dm2 and *Homo sapiens* hg18 assemblies. Comparison with published results show that an appreciable amount of SDs are missed by the different methods. Especially in light of the fact that most results published so far have been obtained by a single methodology, this suggests complementing methods or adapting them after investigating systematic biases.

Currently, most of the available assembled genome sequences have been obtained by shotgun sequencing using reads of short length. Reconstructing duplicated sequences from such short reads is notoriously difficult and often leads to the collapse of duplications into single regions (Alkan et al. 2011b). A different type of mis-assembly can lead to false SDs, constructed from allelic, heterozygous segments in polyploid organisms. This means that current application of assembly-based SD detection methods is severely limited. The problem, however, lies solely in the difficulties of faithful de novo assembly. Given that assembly-based methods are the most comprehensive and accurate way to determine duplications and other SVs (Alkan et al. 2011a), the long term goal should be reliable de novo assembly. The ever growing read lengths of NGS technology will contribute to this naturally, but additionally algorithms have to be engineered to tackle repeat resolution for assembly data, such as (Zerbino et al. 2009; Kelley and Salzberg 2010).

The read-based method compiles duplicon-specific tag probes from known duplications and then aligns those probes to short sequencing reads. When targeting multiple individuals, this approach allows for genotyping SDs across those individuals. The method was tested on high coverage data of homozygous individuals (*Drosophila melanogaster* lines from the DGRP) and low coverage data of heterozygous human individuals from the 1000 Genomes project. In the case of *Drosophila melanogaster* the genotyping method performs well and suggests that some duplications are not in strong LD with known SNPs. Thus, associating phenotypes directly with SD genotype data might lead to more comprehensive results than relying on imputation from associations with SNPs. In the case of the 1000 Genomes data, the strategy in its current form is less satisfactory for genotyping SDs, but still holds promise as many more high-coverage datasets are emerging.
FINAL REMARKS AND FUTURE DIRECTIONS

One possibility to make the method applicable to low coverage data is combining alignment counts across several probes of a duplicon. This will be investigated in the near future. For low coverage data of single individuals, on the other hand, the current methodology is applicable for interrogating the presence of given SDs.

One important piece of future work will be the modelling of the true copy number genotype of a given segment and a given individual. Some steps in that direction could include GC-normalisation of alignment counts to counteract sequence technology bias and using statistical approaches to convert the alignment counts into copy-number genotypes, as discussed in (Waszak et al. 2010). Using confident genotypes, this method should then be readily applicable for large-scale population-level analyses. This will be of great scientific interest as the relationship between duplications and phenotype is not well elucidated. Interesting directions would be applying the method to eQTL studies, i.e. comparing copy number states with expression levels or analysing fundamental biological processes, such as sleep and longevity, using phenotype resources as the DGRP.
Part IV

APPENDIX
A.1 PAPERS RESULTING FROM WORK PRESENTED IN THIS THESIS


A.2 OTHER PAPERS PUBLISHED DURING PHD

INPUT FILES

B.1 COMPRESSION

The file NA12878.chrom20.ILLUMINA.bwa.CEU.high_coverage.20100311.bam was downloaded from ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data/NA12878/alignment. The reference human chromosome 20 file was obtained from http://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes.

The Pseudomonas syringae files ERR005143_1.fastq and ERR005143_2.fastq were downloaded from http://www.ebi.ac.uk/ena/data/view/ERR005143, the reference genome file from http://www.ncbi.nlm.nih.gov/nuccore/NC_007005. The read files were concatenated and mapped to the reference using bwa 0.5.8 (Li and Durbin 2009) with default options.

B.2 ASSEMBLY-BASED SEGMENTAL DUPLICATION DETECTION

For the Drosophila melanogaster SD analysis of the dm2 genome assembly, the chromosome sequences were downloaded from ftp://hgdownload.cse.ucsc.edu/goldenPath/dm2/bigZips and RepeatMasker annotation files from ftp://hgdownload.cse.ucsc.edu/goldenPath/dm2/database. The segmental duplication coordinates from the study of (Fiston-Lavier et al. 2007)
INPUT FILES

were obtained from http://genome.cshlp.org/content/17/10/1458/suppl/DC1 (Supp Table T1).

For the human SD analysis of the hg18 genome assembly, chromosome files were downloaded from ftp://hgdownload.cse.ucsc.edu/goldenPath/hg18/bigZips and RepeatMasker annotation files from ftp://hgdownload.cse.ucsc.edu/goldenPath/hg18/database. The SD coordinates for comparison were obtained from http://humanparalogy.gs.washington.edu/build36/build36.htm.

B.3 READS-BASED SEGMENTAL DUPLICATION DETECTION

Sequencing read files for 176 Drosophila melanogaster lines (see below) were downloaded from http://www.ebi.ac.uk/ena/data/view/SRP000694 in January 2011. SNP data was obtained from http://www.hgsc.bcm.tmc.edu/projects/dgrp/freeze1_July_2010/snp_calls/Illumina/ in May 2011.

B.3 reads-based segmental duplication detection

The identifiers with an asterisk symbol (*) are of those lines for which no SNP data was available. Consequently these lines were excluded from the correlations between probes with SNPs. The identifiers that are underlined are of those lines for which no phenotype data was available or that had missing data points. Consequently these lines were excluded from the correlations between probes and three phenotypes.

1000 Genomes sequencing read files for the 86 human individuals (see below) were obtained from ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data/.

Genotype data from the 1000 Genomes project for the individuals were downloaded from ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20101123/interim_phase1_release

The identifiers of the human individuals are: NA06984, NA06986, NA06989, NA06994, NA07000, NA07037, NA07051, NA07056, NA07346, NA07347, NA07357, NA10847, NA11829, NA11830, NA11831, NA11832∗, NA11840∗, NA11843, NA11881∗, NA11892, NA11893, NA11894, NA11918, NA11919, NA11920, NA11930, NA11931, NA11992, NA11993, NA11994, NA11995, NA12003, NA12006, NA12043, NA12044, NA12045, NA12144, NA12154, NA12155, NA12156∗, NA12249, NA12272, NA12273, NA12275, NA12282, NA12283, NA12286, NA12287, NA12340, NA12341, NA12342, NA12347, NA12348, NA12383, NA12399, NA12400, NA12413, NA12489, NA12546, NA12716, NA12718, NA12748, NA12749, NA12750, NA12751, NA12761, NA12763, NA12775, NA12776∗, NA12777, NA12778, NA12812, NA12813∗,
INPUT FILES

NA12814, NA12815, NA12827, NA12828*, NA12829, NA12830, NA12842, NA12843, NA12872, NA12873, NA12874, NA12889, NA12890.

The identifiers with an asterisk symbol (*) are of those individuals for which no SNP data was available. Consequently these individuals were excluded from the correlations between probes and SNPs.


Bibliography


BCM: DGRP. http://www.hgsc.bcm.tmc.edu/project-species-i-Drosophila_genRefPanel.hgsc.


Bibliography


Bibliography


Bibliography


Bibliography


The 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073,


UCSC: LiftOver. http://genome.ucsc.edu/cgi-bin/hgLiftOver.


Bibliography


Wellcome Trust Case Control Consortium. Genome-wide association study of cnvs in 16,000 cases of eight common diseases and 3,000 shared controls. *Nature*, 464(7289):713–720, Apr 2010. doi: 10.1038/nature08979. URL http://dx.doi.org/10.1038/nature08979.


