Probabilistic methods for processing high-throughput sequencing signals

Lasse Maretty

July 7, 2016

Department of Biology
University of Copenhagen
Denmark

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen
Til min mor.

At turde
er at miste fodfæstet et øjeblik.
Ikke at turde
er at miste sig selv.

— Søren Kierkegaard
Preface

The work presented in this thesis was conducted at the Section for Computational and RNA Biology at the Department of Biology, University of Copenhagen from February 2013 to July 2016. The work was supervised by Professor Anders Krogh.

The original work in this thesis is composed of three manuscripts of which the first is published and the last two were in review at the time of writing. Chapters 1 and 2 describe new methods for transcriptome assembly and variant graph genotyping, respectively. Chapter 3 describes an application of the method described in chapter 2 to call variants in the GenomeDenmark project.

I contributed to all parts of the two methodological studies, whereas my main contribution to chapter 3 was the application of our method to call variants as well as the analysis and writing related to this part of the project. I note that most of the theoretical work described in chapter 1 was developed as part of my Master’s thesis together with Jonas Andreas Sibbesen. During my PhD studies, I further made significant contributions to two original articles and one review article that are listed under *additional research papers*.

Lasse Maretty
July 7, 2016
Acknowledgements

I want to thank Anders Krogh for providing me with the opportunity to spend the past three and a half years in an exceptionally inspiring environment. Science emerges when curiosity meets freedom—you have provided plenty of both. I can think of no better place to have done my PhD.

I want to thank Jonas Andreas Sibbesen for four years of excellent collaboration. Most of all I thank you for the daily scientific discussions. Some led to the work presented in this thesis. Others provided just a brief ride in the rollercoaster of science with immediate excitement followed by depression. But it was all great fun. I really hope that we will be able to keep the discussion going in the future.

I thank the office core: Mireya, Peter, Simon and Jonas for a great time in the office—not least the lunch-time discussions. Thanks to Siyang for your extraordinary kindness, you are great. I thank all past and present kgb members (including Jes, Kim, Anders Skanderup, Anne) for great discussions and for making PhD life so much more enjoyable.

I thank the members of the GenomeDenmark consortium (Mikkel, Palle, Jacob, Laurits, Søren, Simon) for being great collaborators.

I also thank the entire Section for Computational and RNA biology for making coffee machine and other encounters a pleasure. Special thanks go to outer binf (Anders Albrecht-ersen, Hans, Rasmus, Lemvig, Ida, Rute, Line, Peter), upper binf (Kristoffer, Robin, Maria, Morana), wet binf (Jeppe, Line, Christel, Lukasz), Ole’s binf (Ole, Nikos, Christoffer) and structure binf (Thomas, Lubo, Jesper) for making the section a very social place. Thanks to Henriette and Camilla for always being ready to help and to Hanne and Emil for being great system administrators.

I thank the Novo Nordisk Foundation and the Danish National Advanced Technical Foundation for providing funding for this work.

My final thanks go to my friends and family. Special thanks go to Elin—you are an exceptional woman. And thanks to Vera, your arrival into the world made the last three months so much more enjoyable.
Abstract

High-throughput sequencing has the potential to answer many of the big questions in biology and medicine. It can be used to determine the ancestry of species, to chart complex ecosystems and to understand and diagnose disease. However, going from raw sequencing data to biological or medical insights is far from trivial.

A key challenge is that these methods cannot read the input sequences in their entirety. Due to technological constraints, they instead provide the sequences of very many fragments of the input molecules. Furthermore, not all nucleotides in these fragments are measured correctly and the final output of a typical experiment thus consists of hundreds of millions of error-containing sequence fragments.

This thesis concerns the development of methods for transforming such a raw sequencing signal into a simpler representation from which biological inferences can then be made. Importantly, the fact that the fragments are short and contain errors implies that there may be significant uncertainty associated with the signal. By using probabilistic models, we are able to quantify this uncertainty and propagate it to downstream analyses.

The first chapter describes a new method for reconstructing transcript sequences from RNA sequencing data. The method is based on a novel sparse prior distribution over transcript abundances and is markedly more accurate than existing approaches. The second chapter describes a new method for calling genotypes from a fixed set of candidate variants. The method queries the reads using a graph representation of the variants and hereby mitigates the reference-bias that characterise standard genotyping methods. In the last chapter, we apply this method to call the genotypes of 50 deeply sequencing parent-offspring trios from the GenomeDenmark project. By estimating the genotypes on a set of candidate variants obtained from both a standard mapping-based approach as well as de novo assemblies, we are able to find considerably more structural variation than previous studies.
Resumé

Ny teknologi gør det muligt at aflæse sekvensen af DNA og RNA molekyler i meget stor skala og åbner således for et væld af applikationer indenfor både biologi og medicin. Teknologien kan imidlertid ikke aflæse sekvenserne i deres fulde længde, men kun små fragmenter ad gangen og aflæsningerne er ikke altid korrekte. De rå data fra et typisk eksperiment udgøres således af hundrede af millioner sekvens fragmenter indeholdende fejl.

Denne afhandling beskriver to nye metoder til at transformere et sådant råt sekventeringssignal til en simplere repræsentation, der så kan bruges som udgangspunkt for at besvare biologiske spørgsmål. På grund af den begrænsede fragmentlængde og fejl vil signalet være behæftet med en hvis usikkerhed. En vigtig fællesnævner for vores metoder er brugen af sandsynlighedsmodeller, som gør det muligt at estimere disse usikkerheder til brug for videre analyser.

## Contents

Preface ................................................................. i
Acknowledgements ........................................ iii
Abstract .............................................................. v
Resumé ................................................................. vii

### Introduction

- Genomes and transcriptomes ........................................ 1
- Sequencing .......................................................... 3
  - Shotgun sequencing ............................................. 4
  - High-throughput sequencing .................................. 5
  - The phase problem .............................................. 5
- Genomics ............................................................ 6
  - Analysing genome sequencing data ............................ 7
  - Mapping-based approaches .................................. 7
  - Assembly-based approaches ................................ 11
  - Hybrid approaches ............................................ 11
  - Variant graph alignment .................................... 12
- Contribution: Unbiased genotyping of complex variation using exact alignment of k-mers to variant graphs ....................... 12
- Contribution: De novo assembly of 150 Danish genomes reveals rich structural complexity ........................................ 13

### Transcriptomics

- RNA-seq .......................................................... 14
- Analysing RNA-seq data ....................................... 15
- Transcriptome assembly ....................................... 18
- Contribution: Bayesian transcriptome assembly .............. 21

### Conclusion

................................................................. 23
Perspectives 25
References 27
Additional research papers 33
Chapter 1: Bayesian transcriptome assembly 35
Chapter 2: Unbiased genotyping of complex variation using exact alignment of k-mers to variant graphs 67
Chapter 3: De novo assembly of 150 Danish genomes reveals rich structural complexity 95
Introduction

High-throughput sequencing (HTS) technology is transforming biology as it allows researchers to accurately, rapidly and cheaply determine the sequences of very many DNA or RNA molecules in parallel. However, the machines do not simply spit out entire sequences. Instead, they provide hundreds of millions of small, error-containing fragments of the input sequences from which the desired information needs to be extracted. The processing of these data is typically done in two stages. First, the data is reduced from billions of sequenced nucleotides to an intermediate representation such as genotypes or transcript sequences, which is then used as a basis for answering the biological hypotheses in question.

This thesis concerns the development of two new methods for processing raw DNA and RNA sequencing signals to genotypes and transcript sequences, respectively, as well as an application of the former to estimate genotypes in the GenomeDenmark project. Importantly, by using probabilistic models, we not only provide point estimates, but also provide measures of confidence in these estimates that can then be used in downstream analyses.

I will first introduce the macromolecule ensembles that we are studying and then proceed to motivating why these are interesting study objects. I will then turn to discuss more technical aspects of how these can be studied using HTS and introduce our contributions in this context.

Genomes and transcriptomes

Nucleic acids constitute a core component of all known living organisms. Deoxyribonucleic acid (DNA) constitutes the medium that stores essentially all inheritable information about an organism. This includes the recipe for turning a fertilised egg into an adult that itself is able to collect nutrients, fight infections and peers before it eventually starts the cycle all over again by reproducing itself.

DNA is a polymer composed of four nucleotides that contain either adenosine (A), cytosine (C), guanine (G) or thymine (T) base. Each polymer is coiled around another polymer of equal length such that each nucleotide in one sequence forms hydrogen bonds with the
nucleotide at the same position in the other sequence. In the canonical pattern, A can form hydrogen bonds with T, and C with G implying that it is sufficient to know the sequence of one the polymers as the other is completely determined by this complementary principle.

Many organisms distribute the inheritable information onto multiple DNA polymers termed chromosomes; the collection of all chromosomes of an organism will be referred to as the organism’s genome. The complexity of an organism’s genome differs widely. Bacteria generally have a single circular chromosome, which typically contains approximately $10^6$ nucleotides. Higher organisms, especially those that reproduce by sexual reproduction, tend to have a more complicated arrangement of their genome. For instance, the human genome is composed of 46 chromosomes (ignoring the mitochondrial genome) that contain a total of $6 \times 10^9$ nucleotides. The 46 chromosomes can be further divided into 22 pairs of autosomes and two sex chromosomes (XX in females and XY in males). The chromosomes in each autosomal pair share the same overall organisation meaning that they have approximately the same length and share significant amounts of sequence; the two members of a pair are referred to as homologous chromosomes.

Reproduction occurs by randomly selecting one member from each pair as well as a sex chromosome and placing them in a special cell called a gamete that, if successful, will combine its genetic material with that of a gamete from another organism to produce a new organism. Importantly, the genome of this new organism will not merely be a composition of chromosomes from the parental genomes as the process of forming the gametes induces changes in the chromosomal sequences. First, the gametes are the result of a long developmental trajectory that entails multiple rounds of DNA replication that in turn may give rise to errors like substituting an A with a G or inserting or deleting a nucleotide. Second, during the last stage of development, the sequences of homologous chromosomes may be recombined to form new, chimeric sequences that are then passed on to the offspring. Finally, a range of other mutational processes may change the DNA sequences. For instance, it is estimated that nearly half of the human genome sequence derives from so-called autonomous or selfish elements that can autonomously insert one or multiple copies of themselves at random locations in the genome [1, 2].

Ribonucleic acid (RNA) is responsible for turning much of the information encoded in DNA into actions. Like DNA, RNA is composed of nucleotides, albeit slightly modified, that can form pairs through hydrogen bonding with bases in RNA and DNA. In RNA, a nucleotide containing the Uracil base is used in place of the thymine base with a corresponding change to the base-pairing rules (now A-U, C-G). RNA is produced (or transcribed) by an enzyme that runs along a segment of DNA (a gene) and produces an RNA sequence containing the complementary sequence based on the base-pairing rules introduced above (e.g. a C in the DNA becomes a G in the RNA, an A a U etc.).
The immediate RNA product contains a nucleotide for every transcribed nucleotide in the DNA. However, in eukaryotes, segments of the RNA may be cut out and the flanks spliced together to form an RNA that contains only a subset of the transcribed nucleotides. The segments that remain after splicing are referred to as exons and the removed parts called introns. Intriguingly, RNAs from the same gene can differ in which introns are cut out and most genes produce different transcripts by alternative splicing [3]. Further diversity in the transcriptional output is achieved through the use of alternative transcription start- or end-sites within a gene [4, 5].

The canonical role of the resulting RNAs is to act as an intermediate step in the translation of DNA sequences into amino acids sequences. These proteins in turn serve multiple roles like catalysing the replication of DNA, transcribing RNA, importing nutrients etc. RNA has long been known to also serve other roles for instance as part of the ribosome, where it is partly responsible for catalysing the synthesis of protein based on an RNA messenger. However, more recently, a range of new biological functions for RNA have been discovered including its direct involvement in regulating transcription, the stability of other RNAs (i.e. post-transcriptional regulation) and translation (reviewed in [6]).

The expression level of transcripts varies both within and across genes as some are needed in large quantities and others in minute amounts depending on their biological function. The set of all RNAs and their abundance levels in a sample is called the transcriptome.

The ability to study genomes and transcriptomes is key to modern biology. Studying DNA variations for instance makes it possible to answer questions about the evolution of different species or human migration patterns. An individual’s genetic make-up also influences her predisposition to different diseases. Hence, comparisons of genomes between individual’s suffering from a disease and suitable controls can provide important clues about patho-genetic mechanisms. RNA measurements provide information about the current expression state of the cells in a sample. This information can for instance be used to understand the functional consequences of genetic variation or to study the dynamics of gene expression e.g. along a developmental trajectory or in response to a disease or drug.

Sequencing

The ultimate goal in any genome or transcriptome study is to determine the sequences of all molecules in a sample that can then serve as a basis for answering biological questions. Hence, for a human DNA sample, we want to determine the exact sequences of all 46 chromosomes (i.e. the genome). For RNA, we want to determine the sequences of all transcripts (including alternative splice variants) as well as their abundance in the sample (i.e. the transcriptome). As RNA can be reverse transcribed to the corresponding DNA using
Introduction

a specific enzyme, any method capable of sequencing DNA can also be used to sequence RNA. Assuming that RNAs in a sample are sequenced in proportion to their abundance, we can use the number of times each molecule is sequenced as a measure of its (relative) abundance in the sample.

The first widely applied method was devised by Frederick Sanger and later improved by others to yield what is generally known as Sanger sequencing [7, 8]. In this method, a short, synthetic sequence that is complementary to a segment of the target sequence is used to prime the enzymatic replication of the sequence by DNA polymerase. Rather than only providing the standard four nucleotides as substrates for the replication reaction, a set of specially designed nucleotides are combined with the standard ones. These have two properties. First, they are chemically modified to ensure that the DNA replication reaction terminates once one of these are incorporated by the polymerase. Second, they have a nucleotide-specific fluorescent label attached. By performing the DNA replication using both standard and modified nucleotides, the result is a mixture of DNA fragments terminating at different positions of the template. The mixture is then separated by size using electrophoresis and the exact sequence determined from the colour at each position. This trace of colours at each position is then converted to a sequence read along with a quality score, which can be converted to the probability that the wrong nucleotide was reported in the read through

\[ P(\text{error}) = 10^{-q} \]  \hspace{1cm} (1)

as originality defined in [9].

Shotgun sequencing

A key issue with the Sanger approach is that it cannot produce reads longer than approximately 1,000 nucleotides as individual nucleotides cannot be resolved by electrophoresis for longer fragment lengths. Hence, it is impossible to cover even the smallest viral genome in a single read. One approach to overcome this issue is to randomly fragment the target DNA sequences and then assemble the target sequence by searching for overlaps between the fragments. This approach is known as shotgun sequencing. This was initially only possible for small genomes like a bacteriophage on which this approach was first tested by Sanger [10]. Simultaneous developments in the Sanger sequencing technology, computing power and algorithms enabled the first bacterial genome to be sequencing using a pure shotgun strategy [11]. Finally, the complete human genome sequence was obtained mainly based on the shotgun approach [12, 13]. Simultaneously with the genome sequencing efforts, the shotgun approach was also employed to sequence RNAs [14]. However, the resulting fragments (called expressed sequence tags) were not assembled to full length mRNAs, but rather used to identify expressed segments of the genome.
High-throughput sequencing

Yet another issue with Sanger sequencing is cost. The electrophoresis-based detection system is hard to scale up and thus constitutes a barrier to reducing costs. The scaling issue was dramatically improved by the advent of a range of new sequencing technologies. Ideas from different labs converged into the three major sequencing platforms: 454, SOLID and Illumina (Solexa) that were later joined by the IonTorrent platform [15, 16, 17, 18]. These are collectively referred to as next-generation or second-generation sequencing (SGS) methods. In all of these methods, the DNA is randomly sheared, amplified using the polymerase chain reaction (PCR) and the resulting fragments isolated either in solution (using lipid emulsions) or on solid surfaces. In isolation, each individual fragment is amplified using PCR to generate a set of identical sequences that are then sequenced from one end; this occurs simultaneously for all clones. The method used for sequencing the clones differs between platforms resulting in differences in read lengths, error profiles and throughput. However, the read lengths for all methods remain below that of Sanger sequencing with the 454 platform achieving the longest reads (up to 700 nts). After sequencing one end of the fragment, it is generally possible to also obtain a sequencing read from the other end. The output is then a pair of potentially non-overlapping reads that is known to originate from the same fragment. These are referred to as paired-end reads and are important for reasons that will become clear below.

More recently, methods capable of achieving markedly longer read lengths have been described such as the PacBio and Oxford Nanopore platforms [19, 20]. However, these methods remain constrained by accuracy and throughput and these barriers will have to be overcome before these methods can be widely adopted. Hence, the primary data in DNA and RNA sequencing studies remains large numbers of short reads.

The phase problem

Reconstructing the original sequences from short read data is difficult - and often not possible. The difficulty arises when the same sequence occurs at multiple places in the sequence set. Indeed, if we take two sequences (e.g. chromosomes or transcripts) and insert an identical piece of sequence that is longer than the read length in the middle of each sequence, we can no longer reconstruct the target sequence as shown in figure 1. Instead, we can only reconstruct the contiguous segments between these repeats and the repeats themselves and how they connect to neighbouring sequences.

We can represent this information as a graph structure as shown in figure 1. In this graph, vertices represent contiguous pieces of genome sequence and an edge between to vertices indicate that these are connected in sequence. The correct assembly then corresponds to a set of paths that fully explains the graph. In the current graph, there are four different end-
**Figure 1: The phase problem.** The two sequences contain a repeated piece of sequence (R) and four unique segments (A-D). After sequencing these using short read sequencing, we cannot determine how the four subsequences A-D were phased in the original sequences as no reads are long enough to span the repeated sequence R. We can therefore only reconstruct the sequences up to a graph as shown below.

to-end paths through the graph and at least two are needed to explain the graph. However, there is no information available to determine which pair of paths is correct; the phase information is lost. Paired-end reads can partially alleviate this problem. More specifically, the phase can be completely recovered when the length of the fragment that was sequenced is long enough to span the repetitive sequence.

The phase problem arises in both DNA and RNA sequencing studies, but for different reasons.

**Genomics**

SGS methods have been widely adopted for genome sequencing and have been used to answer biological questions ranging from the composition of microbial communities to understanding human history and disease. For instance, a number of large-scale collaborative efforts aimed at cataloguing human genomic variation have been initiated such as the 1000 genomes project (1000G), the Genome of the Netherlands (GoNL) and the UK10K project [21, 22, 23, 24]. These serve both as a basis for population genetic studies as well as a resource for disease studies.
Analysing genome sequencing data

An intermediate step in most analyses of genome sequencing data is to obtain a map of variation between the individuals in question. From this map, we can for instance reconstruct phylogenies or search for associations between variants and disease. The map could theoretically be obtained by directly comparing the raw reads or assembled sequences, but it is typically more convenient to compare every individual with a reference sequence (if one of sufficient quality is available) and do analyses in this space instead.

The human reference sequence is composed of a single representative sequence for each of the 22 autosomal pairs and the two sex chromosomes. In this coordinate system, an individual’s genome is typically represented by an unordered set of alleles (the genotype) at each polymorphic position. For instance, say for a given autosomal position that the nucleotides A and T have been observed. The individual can then have three different genotypes, two homozygotic configurations (AA, TT) and one heterozygotic (AT). Hence, this representation carries no information about which alleles at heterozygotic positions co-occur on the same DNA strand. For a given chromosome, we can expand this representation by ordering the heterozygotic state such that the first element always originates from the same parent. We will refer to this as phased genotypes and denote the collection of variant alleles from the same parent a haplotype. Hence, for any given autosomal or female X chromosome, each individual carries two haplotypes.

The haplotype information is generally lost due to the phase problem introduced above as the reads (even paired-end) are generally too short to bridge heterozygotic sites over longer distances as required to retain the phase. The phase can be partially recovered using genotype information from many individuals or if the genotypes of the parents of an individual are known (reviewed in [25]). Hence, obtaining accurate estimates of the genotypes constitutes an important step in most analyses.

Mapping-based approaches

The standard approach is to first align the short reads to the reference genome, which can be done efficiently using dedicated read alignment tools such as BWA-MEM or STAMPY [26, 27]. To call genotypes, we can start by obtaining a set of candidate alleles across all sites by searching for mismatches and insertions/deletions in the alignments. Some of these candidate alleles are the result of PCR, sequencing and alignment errors and may thus make a homozygote look like a heterozygote. Conversely, we may also be in a situation where we only observe one allele at a heterozygotic site because no reads were sampled from the
other allele simply due to chance. Hence, methods for estimating genotypes should be able to take these uncertainties into account.

For a collection of reads \( R \), let \( P(R|G) \) be the probability of observing the reads given the genotype \( G \in \{(x,x), (x,y), (y,y)\} \), where \( x, y \) are the two possible alleles at a biallelic site. This quantity is typically referred to as the genotype likelihood. Assuming that the reads are conditionally independent given the genotype, the genotype likelihood can be written as

\[
P(R|G) = \prod_{r \in R} P(r|G)
\]

(2)

where

\[
P(r|G = xy) = \frac{P(r|x) + P(r|y)}{2}
\]

(3)

where \( P(r|x) \) denotes the probability of generating the read \( r \) given the allele \( x \).

For single nucleotide substitutions (SNVs), a simple model can be build by only considering bases aligned on top of this position rather than modelling entire reads. In this model, the probability of observing an aligned base \( z \) can be written as

\[
P(z|x) = \begin{cases} 
1 - P(error) & \text{if } z = x \\
\frac{P(error)}{3} & \text{if } z \neq x 
\end{cases}
\]

(4)

where \( P(error) \) is obtained from the quality scores using equation 1. This is the model used in the Genome Analysis Toolkit (GATK) SNV genotyping module [28, 29]. This model works well for SNVs in regions with sparse variation and non-repetitive sequence as most reads will align correctly in this case. However, assuming that the alignments are correct is problematic in other cases.

Incorporating alignment uncertainty into the likelihood means accounting for the fact that an aligned read may also align elsewhere in the individual’s genome or differently at the current position than what has been reported by the aligner. A read mapper will generally report the best alignment, but other alignments may be nearly as good. Also, the alignment results are highly dependent on the specific parameters used for the alignment such as the gap penalties. Finally, the read mapper considers only the reference sequence and not the actual genome sequence from which the read originated. Hence, once variations from the reference are considered, other alignments than the ones reported may be better. It follows that there exists a circular dependency between alignments and genotypes. More specifically, we need to know the individual’s genotype at all positions to correctly estimate which alignments are likely (i.e. the alignment uncertainty), but at the same time our genotype
estimate depends on our alignment estimates. Together, this suggests that in the optimal case we would have to co-estimate alignments and genotypes. This in turn implies that the genotype estimates at different positions become correlated. While it may be possible to solve this by iteratively estimating genotypes and alignments, it comes with significant computational costs.

Instead, approximations are typically used. To derive approximations, it is convenient to distinguish between global and local uncertainty, where global uncertainty is about whether the read is anchored at the right position in the genome and local uncertainty deals with alternative alignments at the current position. To estimate global uncertainty, Li et al. proposed a probabilistic framework for read mapping where each aligned read is associated with a mapping quality that represents the probability that the read is aligned at the correct position [30]. This is computed assuming ungapped alignments to the reference sequence taking only quality scores into account. While this represents a rather crude approximation, it will deal with cases when there are closely related sequences elsewhere in the genome. Similar probabilistic approaches are used in the Stampy program [27]. The estimated mapping error probability can then be incorporated into the genotype likelihood as done by Li et al. and in the Dindel method [30, 31]. Other methods like GATK and Platypus simply filter reads with mapping quality below a certain threshold.

For more complex variation such as insertions and deletions or dense SNVs, local alignment uncertainty becomes important as reads containing the same indel may align differently depending on where in the read the variant is located. If located towards the end, the optimal alignment may be to accept a couple of mismatches rather than inserting a large gap depending on the alignment parameters. Most importantly, many of these misalignments reflect that the alignment is to the reference rather than the alternative allele sequence. Hence, the alignments need to be re-estimated by aligning the reads also to the alternative alleles. To completely mitigate such reference-bias, it is necessary to also consider other variants if they occur less than the read length away from the current variant. This corresponds to realigning the reads to the set of possible local haplotypes formed by these variants.

The haplotype realignment approach is used by most state-of-the-art methods. In the GATK pipeline, reads are realigned to haplotypes around all indels and clusters of mismatches (i.e. putative SNVs) [29]. In their approach, reads are aligned to the possible haplotypes and the originally reported alignment for a read replaced by its best haplotype alignment in the downstream analysis and genotype inference done at the individual variant level as for SNVs. Rather than only using the haplotypes to mitigate reference bias, other methods such as Dindel and Platypus redefine the entire model to operate in (local) haplotype space rather than genotype space [31, 32]. Hence, in this formulation, the objective is to estimate the individual’s pair of haplotypes (i.e. the diplotype). This allows
them to properly account for dependencies between variants in the local haplotype in the estimation, while still being able to obtain a marginal genotype estimate directly from the diplotype estimate. Both methods use a Hidden Markov model (or similar) to realign reads to the haplotypes. This enables them to properly account for indels errors in the sequence and directly provides them with the conditional probability of observing a read given a haplotype as obtained using the forward algorithm.

The genotype can be estimated directly by finding the genotype (or diplotype) with the highest likelihood or using Bayesian inference. When more samples are available, one approach is to introduce a prior distribution over genotypes that allows for sharing of information between individuals and to infer the genotypes of all individuals in the population simultaneously. A widely used choice is to introduce a variable that models the frequency of each allele (or haplotype) in the population, where the probability of a diploid genotype is given by the probability of drawing its two constituent alleles from the categorical distribution defined by the allele frequencies (assuming Hardy-Weinberg equilibrium). Other priors can also be considered (reviewed in [33]). Finally, it is noted that some methods such as Platypus use separate procedures for determining whether a variant is called (i.e. segregates in the population) and for estimating the genotypes [32].

This model can be directly applied to the case where local diplotypes rather than genotypes are estimated. However, as the alleles will typically be tightly linked, the model needs to account for the fact that only a subset of the possible haplotypes will have non-zero frequency. Estimation of the haplotype frequencies can be done using regularised Expectation-Maximisation as in the Dindel and Platypus methods and diplotype estimates for instance obtained as the maximum a posteriori estimate given the frequencies [31, 32]. The posterior probability distribution over genotypes for every variant in the haplotypes can be directly obtained by marginalisation.

A major issue with the mapping-based approach is that it assumes that the information of interest can be retrieved from the reads that mapped to the reference genome. While this holds true for simple variation like isolated SNVs, reads containing complex variation from the reference such as insertions, deletions or strongly linked SNVs will often fail to align to the reference genome using standard methods. Some of these reads can be retrieved using split-read mapping approaches, where multiple co-linear local alignments are combined to an alignment of the entire read as implemented for instance in BWA-MEM [26]. This improves the sensitivity for reads containing deletions, but reads where the deletion occurs towards the end of the read will still not align properly. Hence, we may be able to see that there is a potential deletion, but we cannot assess the complete read support for the variant. Finally, this approach does not give any improvements for longer insertions as reads located within the insertion cannot be aligned at all. An alternative solution is therefore to first do de novo assembly of the reads and then align the assemblies back to the reference to
identify variants.

**Assembly-based approaches**

*De novo* genome assembly from short read sequencing data is difficult due to the phasing problem. First, the phasing issue implies that we cannot reconstruct the sequences of the autosomes independently and the assembly objective is therefore typically defined as to assemble a single representative sequence for each autosomal pair as well as the two sex chromosomes. Second, even this problem is intractable as the high frequency of repetitive sequence elements (e.g. transposons) that are longer than the read length implies that it in many cases is impossible to identify which sequences are connected across a repeat. Hence, the underlying sequences can generally only be reconstructed up to a graph, where the vertices represent contiguous segments of sequence (contigs) and edges represent connections between such contigs. In the case that paired-end information is available, it can be used to simplify the graph by combining contigs into longer segments called scaffolds.

The graph is typically constructed using either a *de Bruijn* graph approach (e.g. SOAPdenovo2, AllPaths-lg) or using the later string graph approach (SGA) that operates directly with reads rather than k-mers [34, 35, 36, 37, 38]. Various heuristics are then applied to simplify the graph from which scaffolds are then constructed using paired-end information if available.

To call variants from the assemblies, the scaffolds need first to be aligned to the reference sequence. Alignment can be done using dedicated long sequence alignment tools such as LAST [39]. Both assemblies and assembly alignments are prone to error and hence variants cannot simply be called by searching for mismatches or gaps in the alignments. As in the mapping-based approach, these can instead be used as candidate variants and the raw data queried for support for the variant. Again, a simple reference alignment of the reads will not suffice as many of the reads supporting more complex variant alleles will be lost. Liu et al. proposed to base a genotype estimate on an alignment of the reads to both the scaffold and the reference sequence [40]. While this approach is reasonable for isolated variation, it may not work well when multiple variants occur within shorter distances as heterozygotic sites are arbitrarily collapsed to single alleles by most assemblers, which may result in an incorrect phase in the scaffold.

**Hybrid approaches**

Mapping-based genotyping generally works well for simple variation, but it is biased towards the reference sequence for complex variation. Assembly-based genotyping mitigates the reference-bias, but comes at the cost of noise from the assemblers and alignment. Hence, it may be better to combine candidate alleles from both sources and then call variants by re-interrogating the raw data. Moreover, as candidates may be missed in an individual e.g.
due to low coverage, sensitivity is improved by combining candidates both across sources and individuals in the study population. Indeed, Platypus does genotyping based on a reference map, but increases sensitivity by doing local de novo assembly using the mates of reads anchored to a particular region and then combines variants across both sources and individuals [32]. However, by basing the estimate solely on reads anchored to the reference it is still biased towards the reference sequence both in the identification of candidate variants and in the genotype estimates.

Hence, to maximise sensitivity, variants likely need to be combined across both mapping-based and de novo assembly-based approaches as well as across individuals. And to ensure accurate genotypes estimates, an unbiased method for querying the reads for variant support is needed.

**Variant graph alignment**

The candidate variants and the reference sequence can be represented as a graph, where vertices represent reference or variant allele sequence and edges represent connections between them. Querying the reads for variant support then corresponds to aligning the reads against the graph structure which corresponds to finding a path in the graph that matches the query sequence. This problem is notoriously difficult as the number of possible paths grows exponentially in the path length. The problem has received some attention recently, however it remains unclear whether these methods are practical for genome-wide genotyping [41, 42, 43, 44]. Recently, Dilthey et al. described a variation graph method for improved genotyping in the Human Leukocyte Antigen (HLA) locus, a variant dense region of the human genome, however their method does not scale to genotyping an entire genome [45].

**Contribution: Unbiased genotyping of complex variation using exact alignment of k-mers to variant graphs**

In chapter 2, we introduce a new method, BayesTyper, for genotyping a population of individuals on a fixed set of arbitrarily complex variants through efficient re-interrogation of the raw sequencing data. The motivation for developing the method came from a need to integrate and genotype variants called using both mapping-based and assembly-based methods in the GenomeDenmark project as described further below.

The method first constructs a variant graph from the input variants and reference sequence, queries the reads for variant support using the graph and uses this information to estimate genotypes. More specifically, all k-mers in the sequencing reads are first counted for each individual. Next, loci less than k nucleotides apart are joined to form a variant graph in which all possible haplotypes are enumerated except for larger graphs, where a heuristic is used to limit the number of haplotypes. Finally, the multiset containing all k-mers found in
haplotypes are enumerated and combined with the corresponding sample table to provide a matrix containing the occurrences of haplotype k-mers for each sample.

This count matrix then serves as a basis for estimating the genotypes using a fully probabilistic model. For each individual, the observed count vector is modelled as generated by combining counts obtained from the individual’s haplotype pair (i.e. diplotype) with counts originating from a noise process. To share information across individuals, each individual’s diplotype is in turn modelled as drawn from a shared population of haplotypes whose frequencies are modelled using a novel sparse prior. The posterior distribution over genotypes is inferred using collapsed Gibbs sampling of diplotypes, haplotype frequencies and noise parameters. The posterior distribution over genotypes is obtained from the posterior distribution over diplotypes by marginalisation.

To test our method, we first simulated genome sequencing data from ten individuals from the 1000 genomes project. Using these data, we demonstrate that BayesTyper shows similar performance to state-of-the-art tools for small variants like SNVs, whereas it is markedly more sensitive, while being as precise, for larger variants. We also investigated the accuracy of BayesTyper as a function of the divergence between the reference sequence and the individual’s genome (estimated in a window around each variant). Importantly, BayesTyper shows high accuracy independently of the amount of divergence, whereas the accuracy of the other methods decline as divergence increases.

**Contribution: De novo assembly of 150 Danish genomes reveals rich structural complexity**

In chapter 3, we present a comprehensive catalogue of genomic variation in the Danish population as represented by 50 parent-offspring trios. Each individual was sequenced to high depth (\(\sim 78\times\)) and using multiple insert size libraries (up to 20 kb) to enable the construction of high-quality de novo assemblies from each individual.

We adopted the hybrid strategy for variant calling described above. First, a set of candidate variants was generated by combining permissively filtered calls from a standard mapping-based approach (HaplotypeCaller) with structural variant calls obtained from alignment of de novo assembled scaffolds across all 150 individuals. We then applied the method described in chapter 2 to genotype the study population on this set of candidate variants.

The quality of the resulting call-set was assessed using Mendelian error rates and by subjecting a small subset of the called variants across different length classes to experimental validation. Mendelian error rates were generally low and better than for the calls made by HaplotypeCaller (shown in chapter 2). The overall experimental validation rate was 0.79 corresponding to a true positive rate of 0.9 after taking into account the number of variants.
in each length class; highest validation rates were observed for short and non-repetitive variants.

Our variant calling approach resulted in a significantly more complete call-set as compared with previous population sequencing projects. For instance, we discovered significantly more insertion variants as reflected in a highly symmetric variant size spectrum, which contrasts the deletion bias observed in most previous studies.

This study provides a vast, population-specific resource of genomic variation that we expect will increase both imputation accuracy and increase power to identify causal variants in future genome-wide association studies.

### Transcriptomics

The ultimate objective in transcriptomics is to determine the complete set of RNAs that are expressed in a sample and their abundances. One of the first approaches for high-throughput probing of the transcriptome was the expressed sequence tag (EST) approach, where randomly selected transcripts are subjected to shotgun sequencing [14]. However, the objective was to build a catalogue of transcribed parts of the human genome rather than quantification due to the low throughput of Sanger sequencing. To enable quantification with low sequencing throughput, methods that concatenate and sequence a number of small *tags* derived only from the start or end of transcripts were devised [46, 47]. These approaches were complemented by microarray approaches, where cDNA fragments are allowed to hybridize with immobilised oligo-nucleotides from known genes [48]. The amount of hybridized cDNA at each spot is then used as an estimate of abundance of that particular gene.

A number of consortium-based efforts to build comprehensive transcriptomic maps arose out of these technologies. For instance, the Functional Annotation of the Mammalian Genome (FANTOM) consortium used various tag-based approaches to catalogue the mouse transcriptome [49]. Later, the Encyclopedia of DNA Elements (ENCODE) consortium combined EST, tag and microarray technologies to build a transcriptional map of 1% of the human genome [50]. While tag- and array-based methods have been instrumental in providing the first genome-wide measurements of transcriptomes, they suffer from important limitations. More specifically, the tag-based methods cannot detect changes in splicing and the short size of tags implies that many of them cannot be uniquely aligned to the genome. Similarly, array-based methods provide no or little information about alternative splicing and can only quantify expression within a modest dynamic range.
RNA-seq

The advent of second-generation sequencing technologies provided the means to use random shotgun sequencing of full-length RNA (RNA-seq) for probing the transcriptome. RNA-seq methods were published almost simultaneously by different labs studying both more technical properties of the method as well as mapping the transcriptomes of different biological systems [3, 51, 52, 54, 55]. For instance, Marioni et al. demonstrated that gene expression levels estimated using RNA-seq is strongly correlated among independent experiments conducted on the same sample (i.e. technical replicates) and suggested an improvement in the dynamic range and accuracy over expression microarrays. Similarly, a recent large scale study suggested that RNA-seq provides improved power for detecting differentially expressed genes over microarrays [56].

Following these initial, proof-of-concept studies, RNA-seq has been widely adopted in both small-scale studies as well as large-scale consortium efforts to further characterise the human transcriptome such as later ENCODE projects and the Genotype-Tissue Expression (GTEx) consortium [57, 58].

Analysing RNA-seq data

The typical objective of an RNA-seq analysis is to identify changes in the transcriptome between conditions. For species for which a high-quality reference genome is available, the first step in an analysis is typically to align all reads to the reference genome using dedicated RNA-seq mappers such as TopHat2, STAR or HiSat2 to allow for mapping across splice-junctions and then use the reference as a scaffold for the analysis [59, 60, 61]. However, in some cases a reference-free (i.e. de novo) approach may be warranted.

The analysis can be performed at the gene-, exon- or transcript-level depending on the objectives of the study and it can rely only on existing annotations to define these features or obtain them from the data. I will first introduce the different levels of analysis assuming an annotation is provided and then turn to the case where these features are estimated from the data.

The gene-level approach is likely the most common method and requires a set of gene definitions, where a gene is typically defined as a set of disjoint genomic intervals. One commonly employed method for obtaining gene definitions is taking the union of all exons from transcripts originating from the same gene according to an annotation. Provided with gene definitions, the relative expression level of a gene is then estimated simply as the number reads aligned to a particular gene definition and normalised to enable comparison across samples [63] (figure 2). The discrete nature of the measurements implies that stan-
Figure 2: Gene-level analysis. The gene-level approach compares the number of reads falling within each gene across conditions, however it cannot detect changes in the relative distribution of reads among splice-variants if it does not change the overall transcriptional output. Adapted from [62].

Standard statistical tests cannot be directly applied to these data and a wealth of dedicated methods for performing tests for differential expression from RNA-seq data are now available. One set of methods explicitly model the observed counts using the negative binomial distribution and provide either pairwise or more general tests based on generalised linear models [64, 65, 66]. Others mitigate the heteroscedasticity problem with count data using transformations and reweighing of the data, which enables them to use more standard tests [67].

A major issue with the gene-level approach is that it cannot detect changes that only affect the relative distribution of splice variants within a gene as shown in figure 2. One simple solution is to do differential expression at the exon-level by effectively treating individual exons as genes [68]. While this approach does increase the resolution for splicing changes, exon expression estimates are less stable than gene expression estimates (exons typically cover only a few hundred nucleotides), are hard to interpret and discard information about how exons co-occur in transcripts.

The final option is to look for differential expression at the transcript level. In contrast to the two former approaches, we cannot simply count the number of reads falling within the exons of a transcript as many reads can be explained equally well by different transcripts (figure 3). If we assume that reads are sampled uniformly (or from some other distribution that we can estimate) along the transcript sequence, we can construct a model that allows
Figure 3: Transcript-level analysis. The transcript-level approach requires the abundance of each transcript to be estimated using algorithms that take into account that reads may be explained by multiple transcripts.

us to estimate the transcript expression levels. Let $R$ be a set of aligned reads, $S$ be the set of transcript sequences and let $\mathbb{P}(r|s)$ be the conditional probability of observing read $r \in R$ given transcript $s \in S$. We can define

$$\mathbb{P}(r|s) = \sum_{p=1}^{\mid s \mid - \ell + 1} \mathbb{P}(r|p,s)\mathbb{P}(p|s)$$

(5)

where $\mid s \mid$ is the length of transcript $s$, $\ell$ is the read length, $\mathbb{P}(r|p,s)$ is the probability of observing read $r$ given that it starts at position $p$ in transcript $s$ and $\mathbb{P}(p|s)$ is the probability of picking start position $p$. Hence, the former term captures how well the read is explained by the sequence and a simple model can be obtained from the sequence and quality scores equivalent to the expression used for genotyping. The latter term captures the distribution of reads along a transcript and assuming uniformity, we can use $\mathbb{P}(p|s) = (\mid s \mid - \ell + 1)^{-1}$. Importantly, the marginalisation is generally not performed over all possible positions, but only over positions to which the read has been aligned (typically one). Finally, this model can be generalised to paired-end reads by introducing an additional position variable.

Provided with this likelihood, the objective is to estimate a vector of relative expression values $\mathbf{e}$, where $\mid \mathbf{e} \mid = \mid S \mid$ and $\sum_{\mathbf{e}} = 1$. The complete model can be derived by assuming that reads are generated in a two-step procedure. For each read, a transcript assignment $t \in \{x|x \in \mathbb{N}, 1 \leq x \leq \mid S \mid\}$ is first sampled from the categorical distribution specified by $\mathbf{e}$. Second, conditioned on the transcript assignment, a read is sampled from the corresponding transcript using the read likelihood defined above. The full likelihood is then

$$\mathbb{P}(R|S, \mathbf{e}) = \prod_{r \in R} \sum_{t=1}^{\mid S \mid} \mathbb{P}(r|t,S)\mathbb{P}(t|\mathbf{e})$$

(6)

where $\mathbb{P}(t|\mathbf{e}) = e_t$.

The typical approach to estimating $\mathbf{e}$ is to use the Expectation-Maximisation algorithm, where transcript expression levels and transcript assignments (or continuous transcript responsibilities) are estimated iteratively. The estimate $\hat{\mathbf{e}}$ corresponds to an estimate of the
fraction of reads explained by each transcript. The quantity of interest is the relative expression level of each transcript (i.e. fraction of transcripts not fraction of reads) and the final estimates therefore have to be adjusted for the fact that longer transcripts produce more reads. This can be done by dividing the estimates with the effective transcript length $|s| - \ell + 1$ and then renormalise the estimates.

This model and EM-inference approach was initially developed by Li et al. and implemented in the RSEM method partly based on earlier ideas by Jiang and Wong [69, 70, 71]. A similar model is used in the abundance estimation algorithm implemented in Cufflinks, however numerical optimization rather than EM is used for estimation in their implementation [72]. These algorithms all require reads to be aligned to the reference (or directly to the transcripts), which is computationally demanding. Recent approaches such as Sailfish and Kallisto mitigate the runtime problem by using exact matching of k-mers rather than alignment as the basis for estimation [73, 74]. To obtain optimal estimates, the model needs to be modified to take into account that different steps in the sequencing protocol introduce biases that in turn result in a non-uniform read distribution along transcripts [75, 76].

A key difference between working at the gene/exon-level and transcript-level is that the downstream tests have to take into account the additional uncertainty arising from many reads being shared by multiple transcripts. Dedicated methods for conducting differential expression analysis at the transcript level such as BitSeq and EBSeq that take this uncertainty into account therefore need to be applied [77, 78].

**Transcriptome assembly**

The annotation-based approach has been used quite extensively, but it does not make full use of the data as we can only look at genes, exons and transcripts that we already know. Alternatively, transcripts can be reconstructed directly from the raw data and the analysis based on this assembly (the analysis can still be done at either level as explained further below).

Transcriptome assembly is non-trivial due to the phasing problem introduced earlier. More specifically, when exons (or longer exon chains) are longer than the read length (or fragment length for paired-end data), we loose information about exon connectivity as these act as phase breakers analogously to the repeats in genome assembly (figure 4). Again, we can represent the exon context information contained in the read sequences by a graph, where vertices represent exons and junctions represent exon-exon junctions as shown in the figure. The transcripts that generated the reads can then be represented by a set of paths in this graph and the objective in transcriptome assembly is to find these paths.
Introduction

**Figure 4: Transcriptome assembly.** Two transcripts, of which one is twice as abundant as the other, are subjected to RNA-seq. The sequence information from the short reads provides only enough information to construct a splice-graph as the central exon, which is shared between the two transcripts, is longer than the read length. The correct transcripts are represented by a set of paths in this graph and these can under some conditions be identified by using information about the read coverage. Here, solution A conforms with a uniform sequencing model, whereas the implied read coverage of the transcripts in solution B is non-uniform, suggesting that solution A is correct.

In contrast to the genome assembly problem, the fact that alternative splice variants often differ in expression levels implies that we have additional phase information. For instance, in the example in figure 4, two different solutions each composed of two paths (i.e. transcripts) may explain the graph. But, given the read coverage at each vertex, one of the solutions does not look very likely under a uniform sequencing model, whereas the other does. This suggests that the read coverage can be used to guide the assembly and essentially implies that transcripts and their abundances need to be co-estimated. While some of the lost phase information may be retrieved using this idea, information about the number of expressed transcripts is also lost. For instance, in the example in figure 4, the solution composed of all four transcripts (i.e. the union of the two solutions) explains the graph and coverage at least as well as the correct solution. Hence, we further need to impose a sparsity criterion to be able to identify the correct assembly.
The transcriptome assembly problem is typically approached in two stages. In the first stage, a splice graph is constructed either by looking at overlapping read alignments on the reference genome or \textit{de novo} using approaches similar to those employed for genome assembly. The reference-based approach to graph construction is advantageous when a high-quality reference genome is available as transcripts may differ widely in abundance implying that there will be insufficient read coverage to assemble many transcripts \textit{de novo}. However, there may be cases where the inherent biases in current methods for mapping RNA-seq makes it impossible to detect certain transcripts of interest (e.g. fusion transcripts) and where a \textit{de novo} approach therefore works better. In both cases, it is possible to include previously annotated transcripts to bridge gaps resulting from uncovered segments.

The next stage deals with identifying the assembly (i.e. transcripts and abundances) from the splice-graph. It is useful to distinguish between two different approaches for identifying the paths. The first class uses ideas from graph optimization to find the optimal assembly directly from the splice-graph. For instance the Cufflinks assembler uses a graph optimization algorithm to find the minimal number of paths (starting and ending anywhere in the graph) that can explain the splice-graph [72]. However, the strict minimum path cover objective implies that it will not perform well when transcripts share longer stretches of sequence or when the number of expressed transcripts is greater than the minimal number required to explain the graph. Also, their approach uses only a fraction of the phase information from coverage. Others rely on more greedy approaches. For instance, StringTie iteratively applies a heuristic algorithm to find the highest expressed path in the graph and removes the reads explained by the fragment [79]. Related ideas are used in the earlier Traph assembler [80]. While these approaches relax the strict sparsity criterion imposed by Cufflinks, the greedy approach comes with obvious risks. More generally, working directly on the graph renders it difficult to define probabilistic models of the read generating process and hence makes it difficult to incorporate non-uniform read distributions and to assess confidence in an assembled transcript. Finally, it is not trivial to include previously annotated transcripts in this approach. \textit{De novo} assemblers such as Trinity, Oases and Bridger generally use either heuristics or graph optimization approaches to infer assemblies from the splice-graph [81, 82, 83].

The second class first enumerates a large set of candidate transcripts from the graph using heuristics and then does assembly by estimating the abundances of these transcripts subject to a constraint that only some of them should have non-zero abundance. This problem can for instance be solved by adding a regularisation term to equation 6. IsoLasso and SLIDE both use the Lasso penalty, however this may not work well under the combined positivity and summation constraints of a relative abundance measure [84, 85]. CEM combines a sparse prior on the relative abundances and a heuristic modification to the EM algorithm [86]. Finally, iReckon uses a custom non-linear regulariser that penalises lowly expressed
transcripts, but also relies on a rather heuristic modification of the objective function [87]. Importantly, these methods generally do not consider all possible paths in the splice-graph, but only end-to-end paths implying that these approaches cannot detect internal transcription start- or end-sites.

If the analysis is to be performed at the gene- or exon-level, the splice-graph and splice-graph vertices can serve as gene and exon definitions, respectively. Alternatively, gene expression estimates may be obtained by summing the abundances of the transcripts assigned to each gene. In fact, Trapnell et al. suggested that this provides a more accurate estimate of gene expression than the naive count-based estimate [62]. A similar estimation method could be obtained for individual exons by summing the abundances of all transcripts that contain the exon in question.

**Contribution: Bayesian transcriptome assembly**

In chapter 1, we describe a new method for transcriptome assembly. The key idea in our method is to use a spike-and-slab prior distribution over candidate expression levels that forces some candidates to have exactly zero abundance (i.e. not expressed), while imposing no prior knowledge on the transcripts with positive abundance (i.e. expressed transcripts) in contrast to current regularisation-based approaches.

We devised a collapsed Gibbs sampling method that does inference in this model by iteratively sampling whether a candidate is expressed, the abundances of expressed transcripts and read assignments to candidates. We then use the number of iterations a candidate is expressed (i.e. has non-zero abundance) as a confidence measure that in turn is used to select candidates for the final assembly. Importantly, the collected expression estimates are averaged across samples from the posterior and thus marginalise out the effects of assembly size and composition.

We demonstrate that the method improves both sensitivity and precision over current methods (both graph- and candidate-based) as judged using simulations and three different accuracy metrics applied to real data assemblies.
Conclusion

The information output from high-throughput sequencing experiments is massive and fairly standardised analysis pipelines exist that will allow us to extract a lot of this information. We know how to map the short reads to a reference genome with high accuracy and speed. For genome sequencing data, we know how to proceed from such a map to accurate genotypes of SNVs and shorter indels. For transcriptome data, we know how to find genes that differ in expression between conditions with reasonable accuracy—at least when it comes to already annotated genes. But the analysis problem is far from solved as there is more to be learned from these data.

For genotyping, mapping-based approaches are biased towards the reference sequence implying that certain types of variation is invisible in this approach. We devised a new genotyping approach that uses a graph representation of variants to mitigate reference-bias. This in turn allows researchers to obtain better genotype estimates for complex variants by pooling information across individuals and different sources of candidate variants such as standard mapping-based approaches and de novo assembly. We applied our method to estimate genotypes from mapping- and assembly-based variant candidates in the Genome-Denmark project. The combination of very deep sequencing, de novo assembly and variant graph genotyping resulted in a call-set that is more complete in terms of complex variation (especially insertions) than those from previous large-scale sequencing projects.

In transcriptomics, RNA-seq provides much more information about expression at the individual transcript-level than previous tag- or microarray-based approaches. Yet gene-level approaches still seem to dominate the analysis of these data. As most functional information is only available at the gene-level, this may be a reasonable approach. But on the other hand, to learn about the functional differences between say alternative splice variants, we also do need to study them. Another barrier for such analyses is the lack of methods for accurately reconstructing transcripts from RNA-seq data. To enable such an analysis, we therefore propose a new and more accurate method for transcriptome assembly. Using a fully probabilistic approach based on a sparse prior distribution over transcript abundances, we are able to reconstruct transcripts with markedly better accuracy than previous methods. Importantly, by also providing a confidence estimate in each assembled transcript, we
enable researchers to rank transcripts by confidence in the downstream analysis.

In conclusion, we present two new methods that we hope will enable more researchers to study complex variation from genome sequencing data and to study transcript-level effects from RNA-seq data.
Perspectives

The main motivation behind our genotyping method was to enable the use of different sources of candidate variants to improve sensitivity, while still being able to get accurate genotypes (and thus variant calls). Optimally, candidates identified in a study would be combined also with previously annotated variants to maximise sensitivity. However, this is currently not possible as many graphs become prohibitively large. One way to solve this is to genotype each variant independently and only consider variants within k as the haplotype context. Alternatively, graphs can be split when there are longer tracts of variants that do not segregate in the population as judged using a heuristic.

Future work also includes a closer look at the assumption of independence between k-mer counts within a locus. One idea is to estimate an offset in coverage for each locus across individuals and use this to capture some of the covariance. Alternatively, a sequential model of the coverage could be considered, but this would require more complicated bookkeeping of the k-mer information.

Finally, computational requirements, especially memory, need to be reduced. A major issue is the significant amount of memory consumed by singleton k-mers arising from sequencing errors. One straightforward way to improve this is to use the "fuzzy" k-mer counting facilities in kmc2. In this mode, k-mers are counted as continuous variables such that the count is incremented with the probability that all bases in the read are called correctly. By using a threshold on this fuzzy count, we expect to be able to dramatically decrease the number of singleton k-mers with little impact on accuracy.

On the transcriptome side, we were motivated by a need for better methods to study transcripts from RNA-seq. While our method performs markedly better than the other methods, the absolute performance as judged from simulated data leaves room for improvement. A key challenge for improving sensitivity is to reduce fragmentation of the splice-graphs due to coverage dips. A simple idea is to combine information across samples and from annotations in the graph construction phase. This can be taken one step further to also include pooling at the candidate enumeration stage after which the Gibbs sampler can then be run independently for each sample. This provides an assembly estimate for each sample that is
based on the same set of candidates. We hypothesize that this will provide a better basis for transcript-level differential expression tests than current approaches. This pooling strategy has already been implemented in the Bayesembler and will be a part of the next release.

Future work on the Bayesembler also includes improvements to the candidate generation mechanism. The current method ignores paired-end information until the Gibbs sampling stage and we are considering ways of using this information already at the candidate stage. Finally, a major issue with the current Gibbs sampler is that it mixes very slowly in some cases. We plan to improve this by augmenting the current sampling scheme with a move that swaps expression levels of transcripts. This in turn should allow the sampler to move more easily between modes.
References


[40] Liu, S. *et al.* Discovery, genotyping and characterization of structural variation and novel sequence at single nucleotide resolution from de novo genome assemblies on a population scale. *Gigascience* **4**, 64 (2015).


[50] States, D. J. Faculty of 1000 evaluation for identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project.


Additional research papers

The following list contains additional research papers to which I have made significant contributions during my PhD:


Chapter 1

Bayesian transcriptome assembly
Bayesian transcriptome assembly

Lasse Maretty†, Jonas Andreas Sibbesen† and Anders Krogh*

Abstract
RNA sequencing allows for simultaneous transcript discovery and quantification, but reconstructing complete transcripts from such data remains difficult. Here, we introduce Bayesembler, a novel probabilistic method for transcriptome assembly built on a Bayesian model of the RNA sequencing process. Under this model, samples from the posterior distribution over transcripts and their abundance values are obtained using Gibbs sampling. By using the frequency at which transcripts are observed during sampling to select the final assembly, we demonstrate marked improvements in sensitivity and precision over state-of-the-art assemblers on both simulated and real data. Bayesembler is available at https://github.com/bioinformatics-centre/bayesembler.

Background
The massive throughput of second-generation sequencing technologies is rapidly changing our ability to explore complex transcriptomic landscapes as it can reveal both sample-specific transcript variants and their abundances (i.e. expression levels). However, due to the combination of alternative splicing and the short sequencing fragments characteristic of these methods, it is often not possible to determine directly which exons are linked in splice variants over longer sequence distances. Instead, due to variation in abundance between alternative splice variants, read coverage along exons and splice junctions can be used to infer the most likely exon combinations.

We define transcriptome assembly as the problem of determining the set of expressed transcripts and their abundance levels in a sample from a set of RNA sequencing (RNA-seq) reads. Current assembly algorithms generally proceed by first extracting exon boundary and splice junction information from the RNA-seq reads, which is then used to build a set of splice graphs representing all possible splice variants [1]. This problem can generally be solved efficiently using either a reference-based strategy, where a reference genome is used as a scaffold in splice graph assembly, or by de novo construction. Given a graph, the challenge is then to determine which combination of transcripts – represented by paths in the graph – and associated abundances best explains the data. However, as shorter sequencing fragments give rise to larger numbers of putative transcripts, we know a priori that we should generally search for solutions that are sparse relative to the total number of paths in the graph. The inference objective is thus to find solutions sufficiently rich to explain both the graph and its read coverage without overfitting by predicting too many transcripts.

The widely used Cufflinks [2] method estimates splice variants and their abundances sequentially and solves the former problem by searching for the smallest set of transcripts that can explain the graph guided by splice junction coverage information. However, the use of only local coverage information makes it susceptible to noise and the search for the minimal set of transcripts lacks a biological foundation as more complex solutions may better explain the full graph coverage. More recent assemblers like IsoLasso [3], SLIDE [4], CEM [5] and iReckon [6] co-estimate splice variants and abundances using regularisation-based methods. But these approaches achieve sparsity by effectively thresholding transcript abundances and thus implicitly penalise lowly abundant transcripts. The Mitie [7] assembler avoids the thresholding approach to regularisation by using a greedy variant of mixed-integer programming, which, however, comes at the risk of only finding suboptimal solutions. Similarly, the Trapk [8] assembler pursues the simplest possible transcript solution also using a greedy graph-optimisation algorithm. More generally, most assemblers rely on a number of hard-to-tune hyperparameters and heuristic thresholds, which suggests that the methods may not generalise well across, for example, species and
RNA-seq protocols. Finally, even under sparse estimation conditions, model identifiability issues and noise may still give rise to uncertainty about the correct combination of expressed transcripts, thus motivating a fully probabilistic approach to the assembly problem.

Here, we present a novel probabilistic approach to transcriptome assembly based on an efficient Gibbs sampling method for inference in a Bayesian model of the RNA sequencing process. By modelling a subset of paths in the graph – or transcript candidates – as having point-zero abundance, the Bayesian formulation allows us to model the prior expectation of the number of expressed transcripts (sparsity) without penalising lowly expressed ones. The frequency at which transcript candidates are observed to have positive abundance in a set of RNA-seq samples then serves as a confidence metric for each transcript. These confidence estimates are in turn used to determine the final assembly and further provide a means for prioritising assembled transcripts for downstream validation. Our method is implemented in C++ as a complete transcriptome assembly package under the name *Bayesembler*.

**Results**

We first present an informal overview of our model and inference method; a formal description is provided under Materials and methods. We then compare the performance of our assembler with a panel of state-of-the-art transcriptome assembly methods on a number of different datasets.

**The Bayesembler**

The algorithm first constructs a set of splice graphs from a TopHat2 [9] map of the RNA-seq reads using the graph construction routine in the CEM [5] assembly package (Figure 1(1)). Next, for each splice graph, a set of candidate transcripts is constructed by iteratively traversing paths in the graph and pruning the edges with lowest coverage until the total number of candidates does not exceed 100 (Figure 1(2)). Provided with a set of transcript candidates, we use Bayesian inference to determine the most likely combination of candidates and corresponding abundance levels.

Our inference method is built on a generative model of the RNA sequencing process. In the model, each candidate is associated with a binary random variable, which models whether the candidate is expressed (i.e. has non-zero abundance); this construct allows us to model our prior expectation of sparsity in the number of expressed candidates. Each expressed transcript is further associated with a real-valued random variable, which models its relative abundance. Finally, we assume that the binary variables share a Bernoulli prior distribution, which controls the number of expressed candidates, and further assume a symmetric Dirichlet prior distribution on the abundances of the expressed transcripts. Intuitively, this construct decouples the distribution of candidate expression from the distribution over abundance levels and hereby contrasts with most current approaches by not penalising lowly abundant transcripts. To specify a complete generative model of the RNA sequencing process, we assume that for each transcript a binary variable is first drawn from the Bernoulli distribution, followed by a draw of abundance values from the Dirichlet distribution for the expressed transcripts. For each paired-end read to be generated, a transcript is then drawn from the categorical distribution specified by the relative abundances, followed by sampling of a paired-end read from the transcript, essentially as described by Pachter [10].

A Gibbs sampling method was derived to infer the joint posterior distribution over expressed candidates, their abundances and assignments of reads to candidate transcripts, where the latter represents a latent variable in our model (Figure 1(3–5)). The Gibbs sampler is initialised by randomly sampling a candidate assignment for each read and proceeds as follows (Figure 1(3)). First, candidate expression values (i.e. binary values indicating whether a transcript has non-zero abundance) are sampled conditioned on an assignment of reads (Figure 1(4a)). Next, abundance values are sampled conditioned on the set of expressed transcripts and read assignments (Figure 1(4b)). Finally, each read is assigned to a candidate conditioned on the expression and abundance values of all candidates, and the probabilities of observing the read given each of the candidates (Figure 1(4c)). The number of iterations of these three steps needed to explore the posterior distribution sufficiently is then calculated as an affine function of the number of candidates. The main output of the algorithm is the fraction of iterations in which a candidate transcript is expressed (our confidence metric) and a posterior mean abundance estimate for each candidate (Figure 1(5a,b)). Candidates with a confidence above 0.5 and at least 12 expected paired-end read counts are included in the final assembly; the latter threshold is enforced to filter out putative transcript fragments (Figure 1(6)). The hyperparameter that controls the prior distribution over the number of expressed transcripts was estimated using a greedy minimum set cover method.

**Performance evaluation**

The performance of our method was compared with state-of-the-art assemblers on both simulated data and data from the human K562 erythroleukaemia and H1 embryonic stem cell lines [11] as well as mouse dendritic cells [12]. We sought to compare against all assemblers that do not require a genome annotation and that are capable of handling paired-end data. Furthermore, we required that
Figure 1 Outline of the Bayessembler algorithm. (1) A splice graph is first constructed from the RNA-seq data. (2) Transcript candidates are subsequently enumerated by exhaustively searching paths in the graph. (3–5) Gibbs sampling is then used to infer the posterior distribution over expressed candidates, their abundances and assignments of reads to candidates. (3) The sampler is initialised by randomly sampling a candidate assignment for each paired-end read and proceeds as follows: (4a) First, candidate expression values (i.e. binary values indicating whether the candidate has non-zero abundance) are sampled conditioned on an assignment of reads. (4b) Next, abundance values are sampled conditioned on the expressions and read assignments. (4c) Finally, read-to-candidate assignments are sampled conditioned on the transcript expression and abundance values, and the conditional probabilities of observing the reads given the candidates. (5a, b) The fraction of iterations a candidate transcript is expressed during sampling and its mean abundance level across expressed iterations are then used to estimate candidate confidence and abundance levels, respectively. (6) The final assembly is produced by selecting the transcript candidates with highest confidence.
assemblers were stable and efficient enough to complete assembly on at least two of the employed datasets within a week of computation on a server with 40 CPUs. The Cufflinks, IsoLasso, CEM and Traph assemblers were selected based on these criteria. To retain a fair ground for comparison, a single set of optimal Bayesembler hyperparameters was estimated across datasets using partitions of the simulated, K562 and mouse dendritic cell data reserved for this purpose. Hence, the presented performance estimates for Bayesembler were obtained on hold-out partitions of these datasets together with the complete H1 dataset, which was reserved solely for testing purposes.

Simulated data
For the simulation study, a dataset of approximately 80 million paired-end strand-specific RNA-seq reads were simulated from the UCSC Known Genes annotation [13] using the Flux Simulator [14]. The main measures of performance were sensitivity, defined as the fraction of simulated transcripts that were assembled correctly, and precision, defined as the fraction of assembled transcripts found in the set of simulated transcripts.

Our method exhibited both higher sensitivity and precision than all other methods (Figure 2a,b). More specifically, Bayesembler assembled 3,528 more correct transcripts, while producing 9,427 less incorrect ones on the data simulated from 40,496 annotated transcripts than the runner-up assembler, IsoLasso. Importantly, both sensitivity and precision remained higher for Bayesembler relative to the other assemblers independent of transcript abundance levels (Figure 2c,d). Moreover, the length distribution of transcripts predicted by Bayesembler resembled the length distribution of the simulated transcripts in contrast with the other assemblers, which tended to produce shorter transcripts (Figure 2e). Furthermore,
Bayesembler was better at estimating the number of expressed splice variants for each predicted gene than the other assemblers, which may partially explain the observed improvements in assembly accuracy (Figure 2f, Figure S1 in Additional file 1). Finally, we assessed the accuracy of the transcript abundance estimates produced by the assemblers (Figure 2g, Figure S2 in Additional file 1). Here, the estimates produced by Bayesembler exhibited marginally better agreement with the simulated values compared with all other assemblers.

**Real data**

To assess performance on real data, the assemblers were tested on paired-end strand-specific RNA-seq data from two biological replicates from the K562 and H1 cell lines, and a single replicate of mouse dendritic cells. Importantly, these datasets represent different species, tissues, library construction protocols and sequencing depths. Of note, it was not possible to run the Traph assembler on the K562 and H1 data due to instability of the program. As there is no gold standard transcriptome reference for real data, we combined three complementary validation strategies to assess performance of the different assemblers.

We first evaluated assembler performance by estimating the number of predicted transcripts that could be confirmed using the UCSC Known Genes annotation against the total number of predicted transcripts. To adjust for any abundance bias in the annotation and between assemblers, we calculated the number of confirmed transcript predictions across a sequence of thresholds on abundance and plotted it against the corresponding number of predicted transcripts. Hence, the final performance metric is a curve, where the height and slope represent sensitivity and precision, respectively. For both the K562 and H1 data, the Bayesembler curve extended higher and ascended more steeply for both replicates than any of the other assemblers thus indicating both better sensitivity and precision of our method (Figure 3a,b, Figure S3a,b in Additional file 1). Importantly, similar results were observed for data from mouse dendritic cells, which in turn suggests that the results are robust across species, tissues and library construction protocols (Figure 3c). Interestingly, we also observed that the transcript lengths produced by Bayesembler were closer to the length distribution of annotated transcripts than the other assemblers, which again tended to produce shorter transcripts (Figure S4a–e in Additional file 1).

Next, we evaluated both replicate assemblies of the H1 transcriptome using data from a recent study by Au et al. [15]. In this study, RNA from the same cell line was sequenced on the Pacific Biosciences (PacBio) platform (Pacific Biosciences, California, USA), which produces significantly longer reads than standard second-generation sequencing platforms at the expense of lower throughput. As sampling bias implies that lowly abundant transcripts have a lower probability of being verified by a PacBio read, we again used the curve-based metric introduced above to adjust for transcript abundance bias between assemblers. Hence, we computed the number of transcript predictions that could be verified by a PacBio read against the corresponding full number of predictions across a sequence of thresholds on abundance (Figure 4a, Figure S5 in Additional file 1). In agreement with the annotation-based results, Bayesembler found more verified transcripts both in absolute numbers and relative to
the total number of predicted transcripts for each threshold, thus demonstrating better sensitivity and precision, respectively.

Finally, we leveraged the large degree of overlap expected between the transcriptomes of biological replicates to evaluate the different assemblers. We defined stable transcripts as multi-exonic transcripts that were present in both replicate assemblies. Again, we used a curve-based metric to correct for abundance bias between assemblers as lowly abundant transcripts are expected to be less stable. More specifically, for each pair of K562 and H1 replicates, assembled transcripts were ranked according to abundance for each of the replicates and the number of stable transcripts above a certain rank was plotted against the corresponding number of predicted transcripts across a sequence of thresholds on transcript rank, where the ranks were obtained by sorting transcripts in each replicate in decreasing order of abundance (decreasing rank threshold from left to right).

**Discussion**

We have devised a new probabilistic approach to transcriptome assembly. Our primary result is the derivation of a Bayesian model of the RNA sequencing process, which uses a novel prior distribution over transcript abundances to model the number of expressed transcripts for each gene. The model thus provides a statistically consistent way of combining a priori knowledge about sparsity in the number of expressed transcripts with information from the sequencing data. Importantly, in the model, only the read distribution along transcript sequences – and not transcript abundance levels – influences which transcripts are inferred as expressed. This contrasts with most current assemblers, which use abundance – either implicitly through regularisation or explicitly by truncating assemblies at an abundance cut-off – as a proxy for transcript confidence. This is despite the fact that abundance contains only limited information about whether a transcript is expressed in a sample.

The main advantage of our fully probabilistic approach is the ability to quantify our degree of confidence in a transcript given the data. To our knowledge, all previously published assemblers output assembly point estimates and thus provide no means of prioritising newly discovered variants for downstream validation. In contrast, **Bayesembler** provides both a confidence and an abundance estimate for each transcript. To evaluate our confidence metric, we used a simple threshold on confidence – and expected read count – to determine the most likely assembly from one simulated and five real datasets and compared our performance with those of state-of-the-art assemblers. Measuring the accuracy of transcriptome assembly algorithms without extensive experimental validation is difficult. More specifically, simulated data provides a known ground truth at the cost of less realistic splicing and noise patterns, whereas real-data
benchmarks suffer from the lack of a ground truth reference. We therefore used a combination of simulations and three complementary metrics applied across different real datasets to gauge performance.

For simulated data, Bayesembler markedly outperformed all other assemblers on both sensitivity and precision, with IsoLasso coming in as runner-up. Interestingly, Bayesembler was markedly better at estimating the number of expressed transcripts for each gene, thus lending confidence to our sparsity model. In contrast, Cufflinks markedly underestimated the number of transcripts for many genes, suggesting that its strong sparsity objective negatively affects both sensitivity and precision. Real-data benchmarking was performed by comparing assemblers with both transcript annotations, PacBio long-read sequencing data and independent assemblies made for biological replicates to adjust for biases in the individual verification methods. As the latter two reference sets (and likely also annotations) all favour highly abundant transcripts, we used a novel curve-based metric to take into account any possible abundance bias between assemblers. Bayesembler consistently outperformed all other assemblers, with Cufflinks coming in as runner-up for all metrics for all datasets. Clearly, the validity of the real-data performance metrics is strengthened because the same ranking of assembler performance was observed across metrics and datasets. The performance ranking of IsoLasso and Cufflinks was inconsistent between the benchmarks for simulated and real data. We speculate that annotation-based simulations may give rise to genes with more complex splicing patterns (i.e. more variants per gene), which may have impaired the performance of Cufflinks. Abundance estimation accuracy was also investigated, with Bayesembler and Cufflinks performing better than CEM and IsoLasso, but the results did not allow for performance ranking of the former two assemblies.

Importantly, the gain in accuracy provided by Bayesembler does not come at the cost of increased computation time. Indeed, our program took approximately 5.5 hours on 16 CPU cores to assemble the deepest benchmark dataset (approximately 125 million paired-end reads) with a maximum memory footprint of 1.7 GB. This is both faster and more memory efficient than the widely used Cufflinks assembler.

Looking ahead, we believe that our method will also benefit fields outside of reference-based transcriptome assembly. First, we note that our probabilistic inference method is also directly applicable to de novo assembled splice graphs and could easily be implemented as a post-processing routine in packages like Trinity [12] and Oases [16]. Second, the ability to consistently average quantities that depend on the transcript structure provides an additional advantage of our probabilistic approach. Indeed, a recent study highlighted the influence of transcript structure on gene expression estimates [17] and we speculate that gene expression estimates averaged across assemblies may improve the accuracy of differential expression tests.

Conclusions
RNA-seq is rapidly becoming the de facto standard method for expression analysis. However, despite vast amounts of available data, the reconstruction of complete transcripts from such data remains a fundamental challenge in computational biology.

We present here Bayesembler, a statistical approach to transcriptome assembly based on a novel Bayesian model and inference method. Using our approach, we observe a marked and consistent improvement both in assembly sensitivity and precision over state-of-the-art assemblers as judged using several independent robust measures applied across several datasets. Moreover, the use of a fully probabilistic approach allows us to provide a confidence (and an abundance) estimate for each transcript, which we expect will aid in prioritising newly discovered transcripts for experimental validation.

Materials and methods
Our transcriptome assembly algorithm proceeds by first constructing a set of splice graphs [1] (i.e., a directed acyclic graph where vertices represent exons and edges represent exon–exon junctions) from an alignment of RNA-seq reads. Next, for each splice graph, transcript candidates are enumerated by exhaustively searching paths in the graph. Bayesian inference is then used to identify the most probable transcript candidates and associated abundance levels given the reads using a generative model of the RNA sequencing process.

Splice graph construction and generation of transcript candidates
Provided with a TopHat2 [9] alignment of RNA-seq reads, putative PCR duplicates and multimapping reads are first removed. Furthermore, only reads mapping in a proper paired-end fashion are retained. For strand-specific data, the originating strand is inferred for each paired-end read and splice graphs are constructed separately for each strand using the processsam utility from the CEM (v0.9.1) [5] transcriptome assembly package with the -d <strand> option. For unstranded data, the originating strand is inferred from splice site information by running processsam with the -d. option. All graphs for which a strand cannot be inferred are discarded by the assembler. For both stranded and unstranded data, processsam is run with a minimum gap length between genes of one and a minimum number of two reads per gene (-g 1 -c 2).
A set of candidate transcripts is then generated for each splice graph using the following iterative procedure. The algorithm is first initialised with an edge-coverage threshold of one read. For each splice graph, all source-to-sink paths are then enumerated using a depth-first search until the search is complete or the number of candidates exceeds 100. In the latter case, the edge-coverage threshold is incremented by one and all edges with a coverage below the threshold removed from the graph; the graph-search and edge-pruning steps are iterated until the path search has been completed without exceeding the threshold or all edges have been removed. To model the presence of pre-mRNA in the sample, a candidate with a single exon spanning the entire genomic interval of the splice graph is included. Next, to leverage paired-end information, only candidates with paired-end read coverage across all splice junctions are retained. Finally, to improve detection of pre-mRNA material, candidates from overlapping splice graphs are combined and the resulting candidate set used as bases for the probabilistic inference method.

A generative model of the RNA-sequencing process

We will use sequencing fragment to denote a read pair. The overall process of generating a set \( F \) of \( n \) sequencing fragments given a set \( S \) of \( m \) candidate transcripts proceeds as follows. First, a set of relative transcript abundances taking values in \([0,1]\) is drawn from a prior distribution. The inclusion of point-zero abundances in the sample space reflects prior knowledge that only a subset of the candidates are likely to be expressed. For each fragment to be generated, a transcript candidate is then sampled conditioned on the abundance values followed by sampling the fragment’s two sequencing reads conditioned on the selected candidate.

To define a prior distribution that allows for point-zero transcript abundances, let first \( \mathbf{z} \) be a vector of \( m \) independent random variables drawn from the Bernoulli distribution with parameter \( \pi \) such that:

\[
\mathbb{P}(\mathbf{z}\mid \pi) = \pi^{z} (1 - \pi)^{m - z} K_{0}
\]

where:

\[
b_{z} = \sum_{j=1}^{m} z_{j} \quad \text{and} \quad K_{0} = \frac{1}{1 - (1 - \pi)^{m}}
\]

(Supplementary methods 1 in Additional file 1). Then, let \( z_{j} = 0 \) and \( z_{j} = 1 \) indicate that transcript \( s_{j} \) has point-zero abundance (i.e. \( s_{j} \) is not expressed) and positive abundance (i.e. \( s_{j} \) is expressed), respectively. Next, let \( \mathbf{e}^{+} \) be a vector containing the relative abundance levels for the expressed transcripts such that:

\[
|\mathbf{e}^{+}| = b_{z} \quad \text{and} \quad \sum_{k=1}^{b_{z}} e_{k}^{+} = 1,
\]

and let the values be symmetrically Dirichlet distributed such that:

\[
\mathbb{P}(\mathbf{e}^{+}\mid \mathbf{z}, \gamma) = \frac{\Gamma(b_{z}\gamma)}{\Gamma(\gamma)^{b_{z}}} \prod_{k=1}^{b_{z}} (e_{k}^{+})^{\gamma - 1}
\]

where \( \Gamma \) is the gamma function. Finally, define \( \mathbf{e} \) to be the vector of transcript abundances taking values in \([0,1]\) and let it be completely specified by \( \mathbf{z} \) and \( \mathbf{e}^{+} \) such that \( e_{j} = 0 \) when \( z_{j} = 0 \) and \( e_{j} = e_{j}^{+} \) when \( z_{j} = 1 \), where:

\[
k = \sum_{j=1}^{m} z_{j}
\]

Then, to generate a set of abundance values, \( m \) binary values are first drawn from the Bernoulli distribution conditioned on the parameter \( \pi \) followed by sampling of \( \mathbf{e}^{+} \) from the \( b_{z} \)-dimensional symmetric Dirichlet distribution. Conditioned on the set of abundances, each fragment \( f \in F \) is then generated by first sampling a transcript index \( t \) from the categorical distribution \( \mathbb{P}(t|\mathbf{e}) \) defined by \( \mathbf{e} \) (i.e. \( \mathbb{P}(t|\mathbf{e}) = e_{t} \)). Finally, conditioned on the candidate index, the fragment sequences are sampled from \( \mathbb{P}(f|t, q, S, \mu, \sigma) \), where \( q \in Q \) represent the observed quality scores for fragment \( f \), and \( \mu \) and \( \sigma \) denote the mean and standard deviation of the fragment length distribution, respectively, essentially as proposed by Pachter [10].

The joint distribution over a set of \( n \) fragments \( F \), transcript indices \( \mathbf{t} \) and transcript abundances \( \mathbf{e} \) conditioned on the set of transcript candidates, quality scores and hyperparameters then factorises as

\[
\mathbb{P}(F, \mathbf{t}, \mathbf{e}|Q, S, \mu, \sigma, \pi, \gamma) =
\mathbb{P}(\mathbf{e}|\pi, \gamma) \prod_{i=1}^{n} \mathbb{P}(f|t_{i}, q_{i}, S, \mu, \sigma) \mathbb{P}(t_{i}|\mathbf{e})
\]

with

\[
\mathbb{P}(\mathbf{e}|\pi, \gamma) = \mathbb{P}(\mathbf{e}^{+}, \mathbf{z}|\pi, \gamma) = \mathbb{P}(\mathbf{e}^{+}|\mathbf{z}, \gamma) \mathbb{P}(\mathbf{z}|\pi)
\]

The corresponding graphical model is shown in Figure 5. Further details of the model derivation are provided in Supplementary methods 1 in Additional file 1.

Approximate inference using Gibbs sampling

To apply the model to transcriptome assembly, we need to infer the posterior distribution over abundance levels \( \mathbf{e} \). This will in turn provide information on both whether a given transcript candidate \( s_{j} \) is expressed (i.e. when \( e_{j} > 0 \)) and its abundance value if expressed. We treat the vector of transcript indices \( \mathbf{t} \) as a nuisance variable to be inferred.
to make inference tractable, and $P(t|e,F,Q,S,\mu,\sigma,\pi,\gamma)$ thus becomes the target posterior distribution. Samples from this joint distribution are obtained by iteratively drawing samples from $P(e|t,\pi,\gamma)$ and $P(t|e,F,Q,S,\mu,\sigma)$. To do this, let $e$ denote the vector of occurrences of each transcript index in $t$ (i.e. $|e| = m$ and $\sum_{j=1}^{m} e_j = n$). It follows from the model definition that $e$ is a sufficient statistic for $t$ with respect to the posterior distribution of $e$. From this and our definition of the prior distribution over abundances, it follows that:

$$P(e|t,\pi,\gamma) = P(e^+|z,c,\gamma)P(z|c,\pi,\gamma)$$

A sample from $P(e|t,\pi,\gamma)$ can then be obtained by first drawing the number of expressed transcripts $b_x$ from:

$$P(b_x|c,\pi,\gamma) = \frac{\binom{b_x}{|j^+|} \Gamma(b_x + \sum_{j=1}^{m} |j^+|) \Gamma(b_x + \sum_{j=1}^{m} j^+)}{\prod_{j=1}^{m} \Gamma(b_x + j^+ + \mu_j + \sigma_j^2/2) \Gamma(b_x + \sum_{j=1}^{m} j^+ + \sigma_j^2/2) \Gamma(n + b_x \gamma) / \Gamma(n + \gamma)}$$

where $j^+$ is the subset of transcript indices $j$ for which $c_j > 0$, $j^0$ is the subset of indices for which $c_j = 0$ and $b_x$ is constrained such that $j^+ \leq b_x \leq m$. Conditioned on $b_x$, a binary vector $z$ is first generated such that $z_j = 1$ for all $j \in j^+$. The remaining transcripts are then allocated by sampling $b_x - |j^+|$ transcript indices $H$ uniformly from $j^0$ and setting $z_h = 1$ for all $h \in H$. Equivalent to the definition of $e^+$, let $\mathbf{e}^+$ denote the vector of occurrences $c_j$ for which $z_j = 1$. The abundance levels $e^+$ are then sampled from:

$$P(e^+|z,c,\gamma) = \frac{\Gamma(n + b_x \gamma) \prod_{k=1}^{b_x} \Gamma(e^+_k + \gamma)}{\prod_{k=1}^{b_x} \Gamma(e^+_k + \gamma + \mu_k)}$$

Finally, it follows from conditional independence of the elements in $t$ given $e$ that a sample from $P(t|e,F,Q,S,\mu,\sigma)$ can be obtained by sampling the individual elements of $t$ from:

$$P(t_e|e,F,Q,S,\mu,\sigma) = \frac{\prod_{j=1}^{m} P(t_j|q_j,S,\mu,\sigma)}{\sum_{j=1}^{m} P(t_j|q_j,S,\mu,\sigma) P(t_e)}$$

The sampler is initialised by randomly sampling a fragment assignment $t$. A detailed derivation of the Gibbs sampling updates is provided in Supplementary methods 2 in Additional file 1. To estimate $\pi$, the minimum number of transcript candidates required to explain all paired-end reads $m_{min}$ is obtained using a greedy method and $\pi$ estimated using:

$$\pi = \frac{m_{min}}{m}$$

(Supplementary methods 3 in Additional file 1). The fragment lengths of all paired-end reads mapping uniquely to transcripts at least 2500 nt long from single-transcript graphs are used to estimate the parameters of the Gaussian distribution used to model fragment lengths. To minimise the influence of outliers, we use the median and median absolute deviance as estimators of $\mu$ and $\sigma$, respectively (Supplementary methods 3 in Additional file 1). Finally, an uninformative choice of $\gamma = 1$ corresponding to the uniform symmetric Dirichlet distribution is used for the prior over relative abundances. The number of burn-in iterations required for each graph is calculated as $60 \times m + 1000$, where $m$ is the number of candidates; the number of subsequent iterations (i.e. the sample size) is set to 10 times the number of burn-in iterations.

The samples from $P(e|t,\pi,\gamma)$ are subsequently used to estimate both a confidence and a mean abundance estimate for each transcript candidate, where the confidence estimate is calculated as the fraction of iterations in which a candidate is expressed. The posterior mean abundance estimate is calculated as the average of the sampled abundances after they have been normalised to the effective transcript length and total library size.
(Supplementary methods 2 in Additional file 1). The final assembly is produced by selecting all transcript candidates with a confidence above 0.5 and excluding transcripts with an expected paired-end read count below 12; the latter threshold was implemented to filter out putative transcript fragments.

Implementation and availability
The inference algorithm is implemented in C++ as a complete transcriptome assembly package under the name Bayesembler and supports multi-threading. The program is freely available under the MIT licence [18]. The algorithm takes a TopHat2 [9] map as input and outputs an assembly in GTF format and is thus compatible with downstream analysis tools like CuffDiff2 [17].

Benchmarking
The Flux Simulator [14] (v1.2) was used to simulate approximately 80 million strand-specific paired-end reads from the human UCSC Known Genes annotation [13] (hg19 version downloaded 10 May 2013) with read lengths of 100 nt and a mean fragment length of 249 nt. Reads were simulated without variation in transcription start and end sites using the fragmentation-first protocol, where fragmentation is performed prior to reverse transcription with all other parameters set to their default values. Paired-end strand-specific RNA-seq data from two biological replicates of the K562 and H1 cell lines [11], and a single replicate of mouse dendritic cells [12] together with PacBio reads [15] from the H1 cell line were downloaded from the NCBI Short Read Archive and the Gene Expression Omnibus database (Table 1). Simulated and real RNA-seq reads were mapped to either the hg19 or mm10 reference genomes using TopHat2 [9] (v2.0.8, default settings) together with Bowtie2 [19] (v2.1.0). The simulated, K562 and mouse data were divided into separate validation and test sets each consisting of data from half of the chromosomes. The H1 data were reserved for testing purposes only. The validation sets were used for estimating the maximum number of candidates, how the number of Gibbs iterations should scale with the number of candidates as well as the transcript confidence and expected count thresholds. Hence, the test datasets were used exclusively to produce the benchmarks presented in this paper.

The Cufflinks [2] (v2.1.1), IsoLasso [3] (v2.6), CEM [5] (v0.9.1) and Traph [8] (v0.7) assemblers were used in the performance evaluation. All assemblers were run with default parameters except that bias estimation was enabled for Cufflinks, IsoLasso and CEM, and Cufflinks was also run with multimap correction. In addition, the fragment length mean and standard deviation estimates used by Bayesembler were provided as input to IsoLasso and CEM. Bayesembler (v1.1.1) was used in the performance evaluation.

The main benchmark criteria for the simulated data were sensitivity, defined as the fraction of simulated transcripts that were predicted correctly, and precision, defined as the fraction of predicted transcripts that were found in the simulation set. Here, a transcript match was defined as a complete intron-chain match between an assembled transcript and a simulated transcript (i.e. identical intron coordinates between the two transcripts). Single exon transcripts were considered matches if they were contained in and covered at least 75% of a simulated single-exon transcript. As neither CEM, IsoLasso nor Traph provides full support for strand-specific data, transcripts were not matched by strand to provide a conservative estimate of our performance relative to these assemblers. Of note, the abundance estimates of all other assemblers were renormalised to the total library size to allow for visual comparison with the Bayesembler estimates to produce Figure 2d and Figures S2, S6 and S7 in Additional file 1. Spearman’s rank correlation coefficient was used to assess the accuracy of the abundance estimates for each assembler; this metric was selected to provide robustness to the scale of abundances and outliers. To adjust for assembly size, only abundance values of transcripts assembled correctly by all assemblers (i.e. the intersection between assemblies and the simulation set) were used to compute the correlations.

Performance on the five real datasets was first assessed by plotting the number of assembled transcripts that could be confirmed using the UCSC Known Genes annotations (hg19 version downloaded 10 May 2013; mm10 version downloaded 1 October 2013) against the corresponding number of predicted transcripts in each assembly. These estimates were computed across a sequence of lower-bound thresholds on abundance to adjust for

<table>
<thead>
<tr>
<th>Sample</th>
<th>Platform</th>
<th>Approximate depth</th>
<th>Accession</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human K562 cells (replicate 1)</td>
<td>Illumina</td>
<td>125 million</td>
<td>[SRX110318]</td>
<td>[11]</td>
</tr>
<tr>
<td>Human K562 cells (replicate 2)</td>
<td>Illumina</td>
<td>88 million</td>
<td>[SRX110318]</td>
<td>[11]</td>
</tr>
<tr>
<td>Human H1 cells (replicate 1)</td>
<td>Illumina</td>
<td>41 million</td>
<td>[SRX082572]</td>
<td>[11]</td>
</tr>
<tr>
<td>Human H1 cells (replicate 2)</td>
<td>Illumina</td>
<td>37 million</td>
<td>[SRX082572]</td>
<td>[11]</td>
</tr>
<tr>
<td>Human H1 cells</td>
<td>Illumina</td>
<td>8 million</td>
<td>[GSE51861]</td>
<td>[15]</td>
</tr>
<tr>
<td>Mouse dendritic cells</td>
<td>Illumina</td>
<td>53 million</td>
<td>[SRX062280]</td>
<td>[12]</td>
</tr>
</tbody>
</table>

[1] NCBI Sequence Read Archive.
any abundance bias in the confirmation method using the same transcript matching criteria as defined above for the simulated transcripts. The same metric was also used for further evaluation of the assemblers’ performance on the H1 datasets using PacBio reads for confirmation. Finally, performance was also estimated by assessing the overlap between replicate assemblies from the K562 and H1 datasets. To do this, we defined stable transcripts as multi-exonic transcripts that were present in both replicate assemblies. Assembled transcripts were ranked according to abundance for each of the two replicates, and the number of stable transcripts was plotted against the corresponding number of predicted transcripts above a certain rank, thus producing a curve with a point for each rank. Finally, Spearman’s rank correlation coefficient was computed for the predicted abundances of transcripts assembled for both replicates. Only the intersection between stable transcripts of the different assemblers was used to compute the correlations to adjust for assembly size.

**Additional file**

Additional file 1: Supplementary information. Supplementary figures S1–S7 and Supplementary methods 1–3.

**Abbreviations**
PacBio, Pacific Biosciences; PCR, polymerase chain reaction; RNA-seq, RNA sequencing.

**Competing interests**
The authors declare that they have no competing interests.

**Authors’ contributions**
LM and JAS contributed equally to this work. LM, JAS and AK derived the model and inference method. LM and JAS implemented the method and performed the benchmarking. LM, JAS and AK wrote the manuscript. All authors read and approved the final manuscript.

**Acknowledgements**
We thank Ole Winther for his advice on the model and Jes Frellsen and Lubomir Antonov for their assistance on implementation-related issues. This work was supported by grants from the Novo Nordisk Foundation and the Danish National Advanced Technical Foundation.

Received: 24 January 2014  Accepted: 9 October 2014
Published online: 31 October 2014

**References**


Cite this article as: Maretty et al: Bayesian transcriptome assembly. Genome Biology 2014 15:501.
Bayesian transcriptome assembly

Supplementary information

Lasse Maretty*1, Jonas Andreas Sibbesen*1 and Anders Krogh1

*Equal contributors.
1The Bioinformatics Centre, Department of Biology and Biotech Research and Innovation Centre (BRIC), University of Copenhagen, Ole Maaløes Vej 5, 2200 Copenhagen, Denmark.
Correspondence should be addressed to A.K. (krogh@binf.ku.dk).

Contents

Supplementary figures
Figure S1: Difference in predicted and simulated number of variants per gene... 2
Figure S2: Abundance estimation accuracy on simulated data ................. 3
Figure S3: Assembler performance estimates on K562 and H1 replicate 2 - annotation-based measure ............................................. 4
Figure S4: Length distribution of assembled transcripts ............................. 5
Figure S5: Assembler performance estimates on H1 replicate 2 - PacBio-based measure ................................................................. 6
Figure S6: Inter-replicate correspondence of abundance estimates on K562 .... 7
Figure S7: Inter-replicate correspondence of abundance estimates on H1 .... 8

Supplementary methods
1 A generative model of the RNA-sequencing process ...................... 9
1.1 A prior distribution over transcript candidate abundances .............. 9
1.2 A generative model of the sequencing process of a transcript ......... 10
1.3 The joint probability distribution ............................................. 11
2 Approximate inference using Gibbs sampling .............................. 12
2.1 Sampling abundance levels .................................................. 12
2.2 Sampling fragment assignments ............................................. 14
2.3 Abundance normalisation .................................................... 15
3 Hyperparameter estimation .................................................... 16
3.1 Fragment length distribution estimation .................................. 16
3.2 Sparsity estimation .......................................................... 16

Supplementary references 17
Supplementary figures

Figure S1: Difference in predicted and simulated number of variants per gene. RNA-sequencing of transcripts from the UCSC Known Genes annotation [1] was simulated using the Flux Simulator [2]. The figure shows the distribution over the differences in the number of predicted transcripts and the number of simulated transcripts for each predicted gene.
Figure S2: Abundance estimation accuracy on simulated data. RNA-sequencing of transcripts from the UCSC Known Genes annotation [1] was simulated using the Flux Simulator [2]. The figures show predicted abundance estimates against the corresponding simulated abundances for transcripts that were predicted correctly by all five assemblers. $r_s$ denotes the Spearman’s rank correlation coefficient.
Figure S3: Assembler performance estimates on K562 and H1 replicate 2 - annotation-based measure. The figures show the number of assembled transcripts from K562 (replicate 2) \cite{3} (a) and H1 (replicate 2) \cite{3} (b) that were confirmed using the UCSC Known Genes annotation \cite{1} against the corresponding number of predicted transcripts across a sequence of abundance thresholds (decreasing abundance threshold from left to right).
**Figure S4: Length distribution of assembled transcripts.** The figures show the length distribution of the assembled transcripts from K562 (replicate 1 and 2) [3] (a,b), H1 (replicate 1 and 2) [3] (c,d) and mouse dendritic cells [4] (e) together with the length distribution of the UCSC Known Genes annotation [1] (logarithmic binning).
Figure S5: Assembler performance estimates on H1 replicate 2 - PacBio-based measure. The figure show the number of assembled transcripts from H1 (replicate 2) [3] that were verified by a PacBio read [5] against the corresponding number of predicted transcripts across a sequence of abundance thresholds (decreasing abundance threshold from left to right).
Figure S6: Inter-replicate correspondence of abundance estimates on K562. The figures show the predicted abundances of transcripts assembled in both K562 replicates [3] by all assemblers. $r_s$ denotes the Spearman’s rank correlation coefficient.
Figure S7: Inter-replicate correspondence of abundance estimates on H1. The figures show the predicted abundances of transcripts assembled in both H1 replicates by all assemblers. $r_s$ denotes the Spearman’s rank correlation coefficient.
Supplementary methods

1 A generative model of the RNA-sequencing process

We here present a model for the process of generating a set of paired-end sequencing reads from a set of transcript candidates. Let \( F = \{f_1, f_2 \ldots f_n\} \) denote a set of \( n \) sequenced fragments (i.e. paired-end reads) and let \( I = \{1, 2, \ldots, i, \ldots, n\} \) define an index set on \( F \). Let \( S = \{s_1, s_2 \ldots s_m\} \) denote a set of \( m \) transcript candidate sequences and let \( J = \{1, 2, \ldots, j, \ldots, m\} \) define an index set on \( S \). Finally, let \( t \) be a vector of transcript candidate indices for the fragments in \( F \) such that \( t_i \in J \) and \( i \in I \). The process of generating a set of fragments \( F \) from a set of transcript candidates \( S \) then proceeds as follows.

First, a vector of relative abundances \( e \) for the transcripts in \( S \) is sampled from its prior distribution \( P(e) \). \( F \) is then generated by for each fragment first sampling a candidate index \( t \in J \) from \( P(t|e) \) followed by sampling of a fragment \( f \) from \( P(f|t) \). By further assuming that fragments are generated independently, this provides the following factorisation of the joint distribution over sets of fragments \( F \), candidate index vectors \( t \) and candidate abundances \( e \)

\[
P(F, t, e) = P(e) \prod_{i=1}^{n} P(f_i|t_i)P(t_i|e)
\] (1)

First define \( P(t|e) \) to be the categorical distribution with parameter vector \( e \)

\[
P(t|e) = e_t
\] (2)

The other two terms in equation (1) will be derived in the following two sections.

1.1 A prior distribution over transcript candidate abundances

Define the elements of an abundance vector \( e \) to be random variables taking values in \( \{0, \mathbb{R}^+\} \). To construct a prior distribution on \( e \), first let \( z \) be a binary vector of length \( m \), which models which transcripts have non-zero abundance (i.e. are expressed). Furthermore, let each element \( z_j \) in \( z \) be independent and Bernoulli distributed with parameter \( \pi \) and let each element \( e_j \) in an abundance vector \( e \) depend on \( z_j \) in \( z \) such that

\[
e_j = 0 \text{ if } z_j = 0 \\
e_j > 0 \text{ if } z_j = 1
\] (3)

(4)

It follows from independence of the individual elements that the prior probability of a binary vector \( z \) is given by

\[
P(z|\pi) = \pi^{b_z}(1 - \pi)^{m-b_z}
\] (5)
where $b_z = \sum_{j=1}^{m} z_j$. Next, let $e^+$ denote the vector of relative abundances for the expressed transcripts such that $e_j = e_k^+$ when $z_j = 1$, where $k = \sum_{l=1}^{j} z_l$ and $\sum_{k=1}^{b_z} e_k^+ = 1$. Finally, let the elements of $e^+$ be distributed according to the symmetric, Dirichlet distribution with density function

$$
P(e^+ | z, \gamma) = \frac{\Gamma(\gamma_{b_z})}{\Gamma(\gamma)} \prod_{k=1}^{b_z} (e_k^+)^{\gamma-1}
$$

where $\Gamma$ is the gamma function. The prior density of an abundance vector $e$ is then given by the factorisation

$$
P(e | \pi, \gamma) = P(e^+, z | \pi, \gamma) = P(e^+ | z, \gamma)P(z | \pi)
$$

$$
= \frac{\Gamma(\gamma_{b_z})}{\Gamma(\gamma)^{b_z}} \left( \prod_{k=1}^{b_z} (e_k^+)^{\gamma-1} \right) K_{z_0} \pi^{b_z} (1 - \pi)^{m-b_z}
$$

where $K_{z_0} = \frac{1}{1 - P(z_0)} = \frac{1}{1 - (1-\pi)^m}$ with $z_0$ being the all-zeros vector. The latter term is needed as the Dirichlet distribution is not defined on the simplex of size zero and we thus need to define $P(z | \pi) = 0$ and $P(e | z, \gamma) = 0$ when $b_z = 0$ to arrive at a valid density function.

### 1.2 A generative model of the sequencing process of a transcript

A modified version of the model described by Pachter is used to model the sequencing of transcripts corresponding to the indices in $t$ [6]. More specifically, given a transcript index $t$, the three-prime position $p$ of the fragment is first sampled from

$$
P(p | t, S, \mu, \sigma) = \frac{\sum_{l=1}^{p} P(l | \mu, \sigma)}{\sum_{s=1}^{|s_t|} \sum_{l=1}^{q} P(l | \mu, \sigma)} \quad \text{s.t.} \quad p \leq |s_t|
$$

where $|s_t|$ denotes the length of sequence $s_t$ and $P(l | \mu, \sigma)$ denotes the Gaussian distribution, which is used to model the length of the sequencing fragments.

Given a three-prime position, a fragment length - and hence a five-prime position - is then sampled from the renormalised fragment length distribution

$$
P(l | p, \mu, \sigma) = \frac{P(l | \mu, \sigma)}{\sum_{k=1}^{p} P(k | \mu, \sigma)} \quad \text{s.t.} \quad l \leq p
$$

Finally, a sequenced fragment $f$ is generated conditioned on the sub-sequence specified by $t$, $p$ and $l$, and the corresponding observed quality scores $q$. The quality score at position $k$ in sequencing read one of fragment $f$ is defined as

$$
q(1,k) = -10 \log_{10}(\epsilon(1,k))
$$
where $\epsilon_k$ designates the probability of a sequencing error at position $k$ [7]. The probability of an error at position $k$ given a quality score is then given by

$$P(\text{error} | q_1, k) = \epsilon_{(1,k)} = 10^{-\frac{q_{(1,k)}}{10}} \quad (12)$$

Let the individual positions in a sequencing read be independent and let the distribution over the three alternative bases given an error be uniform. The joint distribution over positions $k$ in the left read and right reads, $r_1$ and $r_2$, respectively, of a fragment $f$ is then given by

$$P(r_1 | t, q_1, S, p, l) = \prod_{k=1}^{\vert r_1 \vert} \left\{ \frac{1 - P(\text{error} | q_{(1,k)})}{P(\text{error} | q_{(1,k)})} \right\}^{3} \cdot \frac{1}{3} \quad (13)$$

$$P(r_2 | t, q_2, S, p, l) = \prod_{k=1}^{\vert r_2 \vert} \left\{ \frac{1 - P(\text{error} | q_{(2,k)})}{P(\text{error} | q_{(2,k)})} \right\}^{3} \cdot \frac{1}{3} \quad (14)$$

where $\vert r \vert$ designates the length of the read. The joint distribution over all sequenced positions in a fragment is then given by

$$P(f | t, q, S, p, l) = P(r_1 | t, q_1, S, p, l)P(r_2 | t, q_2, S, p, l) \quad (15)$$

Finally, a pair of sequenced reads can then be obtained by first sampling whether an error has occurred for each position conditioned on the quality score as given by equation (12). If an error is to be introduced, a substitution is sampled uniformly over the alternative base identities.

### 1.3 The joint probability distribution

It follows from the preceding sections that the full joint probability distribution is given by the factorisation

$$P(F, t, l, p, e | Q, S, \mu, \sigma, \pi, \gamma) = P(F | Q, S, l, p, t)P(l | p, \mu, \sigma)P(p | t, S, \mu, \sigma)P(t | e)P(e | \pi, \gamma)$$

$$= P(e | \pi, \gamma) \prod_{i=1}^{n} P(f_i | t_i, q_i, S, l_i, p_i)P(l_i | p_i, \mu, \sigma)P(p_i | t_i, S, \mu, \sigma)P(t_i | e) \quad (16)$$

The corresponding graphical model is shown in figure 5 in the main manuscript.
2 Approximate inference using Gibbs sampling

The objective is to infer the posterior distribution

\[ P(e, t|F, Q, S, \mu, \sigma, \pi, \gamma) \]  

(17)

We can draw samples from this distribution by iteratively drawing samples from the two conditional distributions

\[ P(e|t, \pi, \gamma) \]  

(18)

\[ P(t|e, F, Q, S, \mu, \sigma) \]  

(19)

where the restricted conditioning follows from conditional independences given by factorisation (16). First, the conditional distribution of \( e \) is derived followed by a derivation of the conditional distribution for \( t \).

2.1 Sampling abundance levels

From the definition of \( P(e|\gamma, \pi) \) in equation (8) it follows that the posterior distribution over \( e \) can be factorised as

\[ P(e|t, \pi, \gamma) = P(e^+|z, t, \gamma)P(z|\pi) \]  

(20)

To derive \( P(z|t, \pi, \gamma) \), first note from Bayes’ rule

\[ P(z|t, \pi, \gamma) = \frac{P(t|z, \gamma)P(z|\pi)}{P(t|\pi, \gamma)} \]  

(21)

where \( Z \) denotes the set of \( m \) long binary vectors excluding the all zeros-vector \( z_0 \). Next, let \( c \) be a count-vector of length \( m \) such that \( c_j = \sum_{i=1}^{n} 1(t_i = j) \). Using, that \( P(z|t, \pi, \gamma) \) only depends on \( t \) through \( c \), we can reexpress equation (21) as

\[ P(z|c, \pi, \gamma) = \frac{P(c|z, \gamma)P(z|\pi)}{\sum_{z'\in Z} P(c|z', \gamma)P(z'|\pi)} \]  

(22)

To obtain \( P(c|z, \gamma) \), first note that it follows from definition (3) and (4) that

\[ P(c_j > 0|z_j = 0, \gamma) = 0 \] for all \( z_j \in z \)  

(23)

Provided with this notion, let \( J^0 \subseteq J \) be the subset of \( J \) that contains the indices of the transcripts that have zero counts in \( c \) (i.e. \( J^0 = \{ j \in J|c_j = 0 \} \)). Correspondingly, let \( J^+ \subseteq J \) contain the indices of the transcripts that have positive counts in \( c \) (i.e. \( J^+ = \{ j \in J|c_j > 0 \} \)).
$J^+ = \{ j \in J | c_j > 0 \})$. Using this, define $U$ to be the set of feasible binary vectors given by condition (23)

$$U = \{ z \in Z | z_j = 1 \ \forall j \in J^+ \}$$  \hfill (24)

Using this condition, it follows that $P(c|z, \gamma)$ can be expressed as

$$P(c|z, \gamma) = \begin{cases} \int_{e^+} P(c|e^+, z)P(e^+|z, \gamma) \ dc^+ & \text{if } z \in U \\ 0 & \text{if } z \notin U \end{cases}$$  \hfill (25)

First, it follows from the definition of $P(t|e^+, z)$ in equation (2) and from conditional independence of the elements in $t$ given $e$ that $P(c|e^+, z)$ is equivalent to the multinomial distribution. As $P(e^+|z, \gamma)$ represents the symmetric Dirichlet distribution, it follows that $P(c|z, \gamma)$ is equivalent to the Dirichlet-multinomial distribution. Hence, we have that

$$P(c|z, \gamma) = \begin{cases} \frac{\Gamma(n+1)\Gamma(b_z \gamma)}{\Gamma(n+b_z \gamma)} \prod_{j \in J^+} \frac{\Gamma(c_j + \gamma)}{\Gamma(c_j + 1) \Gamma(\gamma)} & \text{if } z \in U \\ 0 & \text{if } z \notin U \end{cases}$$  \hfill (26)

where again $b_z = \sum_{j=1}^{m} z_j$.

To derive a method of sampling from the posterior distribution given by equation (22), first define $\sim$ to be an equivalence relation on binary vectors such that $z_p \sim z_q$ when $b_{z_p} = b_{z_q}$. Using this, define $Y = U/ \sim$ (i.e. $U$ collapsed to its quotient by $\sim$) to obtain the set of feasible equivalence classes. The posterior distribution over equivalence classes is then given by

$$P([z]|c, \pi, \gamma) = \begin{cases} \frac{P(c|z, \gamma)P([z]|\pi)}{\sum_{y \in Y} P(c|y, \gamma)P([y]|\pi)} & \text{if } [z] \in Y \\ P(c|y, \gamma)P([y]|\pi) & \text{if } [z] \notin Y \end{cases}$$  \hfill (27)

From distribution (5) and (26) it follows that $P(z|\pi)$ and $P(c|z, \gamma)$ are equal for all $z \in [z]$ when $[z] \in Y$. Using this, the sum over $z \in [z]$ can thus be replaced by the cardinality of $[z] \in Y$. Hence, the posterior distribution over equivalence classes becomes

$$P([z]|c, \pi, \gamma) = \begin{cases} \frac{(b_{[z]} - |J^+|) P(c|z, \gamma)P([z]|\pi)}{\sum_{y \in Y} (b_{[y]} - |J^+|) P(c|z, \gamma)P([y]|\pi)} & \text{if } [z] \in Y \\ P(c|y, \gamma)P([y]|\pi) & \text{if } [z] \notin Y \end{cases}$$  \hfill (28)

where $z \in [z]$, $z_y \in y$ and $b_{[z]}$ and $b_y$ are the simplex sizes defined by $z$ and $z_y$, respectively.
From the definition of the equivalence class it follows that a sum over equivalence classes corresponds to a sum over simplex-sizes. By substituting in equations (5) and (26), we can re-express the posterior distribution as

\[
\mathbb{P}(z | c, \pi, \gamma) = \begin{cases} 
\frac{\Gamma(b_0 z) \pi^{b_0 - b_z} \Gamma(b_0 + b_z) \pi^{b_0 + b_z} \Gamma(b_0 + b_z + 1) \pi^{b_0 + b_z + 1} - 1}{\sum_{b = 0}^{m} \frac{\Gamma(b_0 + b_z + 1) \Gamma(b_0 + b_z + 1) \pi^{b_0 + b_z + 1} - 1}{\sum_{b = 0}^{m} \frac{\Gamma(b_0 + b_z + 1) \Gamma(b_0 + b_z + 1) \pi^{b_0 + b_z + 1} - 1}}}
& \text{if } [z] \in Y \\
0 & \text{if } [z] \notin Y
\end{cases}
\] (29)

where the correction term \(K_{z_0}\) in equation (5) and multiple terms in equation (26) vanishes as they are independent of \(b_z\) when \([z] \in Y\).

Using that all \(z \in [z]\) has equal probability, we have from distribution (29) that the posterior distribution over binary vectors \(z \in U\) is given by

\[
\mathbb{P}(z | c, \pi, \gamma) = \frac{\Gamma(b_0 + b_z + 1) \pi^{b_0 + b_z + 1} - 1}{\sum_{b = 0}^{m} \frac{\Gamma(b_0 + b_z + 1) \Gamma(b_0 + b_z + 1) \pi^{b_0 + b_z + 1} - 1}{\sum_{b = 0}^{m} \frac{\Gamma(b_0 + b_z + 1) \Gamma(b_0 + b_z + 1) \pi^{b_0 + b_z + 1} - 1}}}
\] (30)

By conjugacy of the Dirichlet distribution to the multinomial distribution, we have that the posterior probability of the non-zero expression values \(e^+\) are given by the Dirichlet distribution. Equivalent to the definition of \(e^+\), let \(c^+\) denote the vector of occurrences \(c_j\) for which \(z_j = 1\). The posterior density is then given by

\[
\mathbb{P}(e^+ | z, c, \gamma) = \frac{\Gamma(n + b_x + 1) \pi^{b_x + 1} - 1}{\prod_{k=1}^{b_x + 1} \Gamma(c_k + 1)} \prod_{k=1}^{b_x + 1} (c_k + 1)^{\gamma - 1}
\] (31)

From equation (29) and (31) it follows that samples from the posterior distribution over expression vectors \(e\) can be obtained using the following hierarchical sampling scheme

1. Sample an equivalence class \([z]\) given \(c\), \(\pi\) and \(\gamma\) from distribution (29)

2. Sample a binary vector \(z\) uniformly from the equivalence class \([z]\)

3. Sample an expression vector \(e^+\) given \(z\) and \(c\) from distribution (31)

Samples from the Dirichlet distribution in (31) was obtained by transforming variables drawn from the gamma distribution as described by Devroye [8].

2.2 Sampling fragment assignments

From conditional independence of the individual elements in \(t\) given \(e\), we have that

\[
\mathbb{P}(t | e, F, Q, S, \mu, \sigma) = \prod_{i=1}^{n} \mathbb{P}(t_i | e, F, Q, S, \mu, \sigma)
\] (32)
Using Bayes’ theorem, the posterior probability of an individual fragment-transcript map \( t \) is given by

\[
P(t | e, F, Q, S, \mu, \sigma) = \frac{P(t | e)P(F | t, Q, S, \mu, \sigma)}{\sum_{j=1}^{m} P(j | e)P(F | j, Q, S, \mu, \sigma)}
\]  

where \( P(t | e) \) is given by distribution (2) and

\[
P(F | t, Q, S, \mu, \sigma) = \sum_{p=1}^{s_t} \sum_{l=1}^{l} P(p | t, S, \mu, \sigma)P(l | p, \mu, \sigma)P(F | t, Q, S, p, l)
\]

where \( |s_t| \) designates the length of transcript \( s_t \). The individual probabilities in equation (34) are defined in section 1.2.

### 2.3 Abundance normalisation

The abundance estimates generated by the Gibbs sampler corresponds to the fraction of all paired-end reads a transcript explains. Thus, in order to provide a measure of transcript abundance these values need to be normalised to transcript length. Therefore, define

\[
|\tilde{s_j}| = \sum_{q=1}^{s_j} \sum_{l=1}^{l} P(l | \mu, \sigma)
\]

where \( |s_j| \) denotes the sequence length of transcript \( s_j \) and \( P(l | \mu, \sigma) \) denotes the Gaussian distribution used to model fragment lengths. In addition to transcript length, the abundances is normalised to the total library size \( |F| \) in order to make the estimates comparable across different sequencing depths. The relative abundance \( a_j \) of transcript \( s_j \) is then defined to be

\[
a_j = \frac{e_j 10^9}{|s_j||F|}
\]

This estimate is used to produce the FPKM attribute in the Bayesembler output.
3 Hyperparameter estimation

3.1 Fragment length distribution estimation

The standard Gaussian distribution with mean $\mu$ and standard deviation $\sigma$ is used to model the fragment length distribution. The fragment lengths of all paired-end reads mapping uniquely to transcripts at least 2500 nt long from single-transcript graphs were used to estimate the parameters. To minimise the influence of outliers, we used the median and median absolute deviance ($MAD$) as estimators of $\mu$ and $\sigma$, respectively, with $\sigma$ estimated using $\sigma = MAD \times 1.4826$.

3.2 Sparsity estimation

To estimate $\pi$, the minimum number of candidates required to explain all paired-end reads is first estimated using a greedy minimum set cover method. Formally, let $A$ be an $n \times m$ binary matrix, where the rows and columns represent paired-end reads and candidate transcripts, respectively, and the values indicate whether a transcript contains a paired-end read or not. Then, let $x$ denote a binary vector of length $m$ such that $x_j = 1$ for each candidate transcript $s_j$ in the minimum cover. The problem is then given by

$$\arg\min_x \; c^T x \quad \text{s.t.} \; Ax \geq a$$

where $a$ and $c$ are $n$ and $m$ long vectors of ones, respectively. Finding the optimal solution for this problem requires evaluating all column combinations, which becomes infeasible for large values of $m$. For this reason, we used a greedy algorithm, where the transcript containing the largest amount of uncovered reads is chosen at each stage as earlier described by Johnson [9]. Finally, $\pi$ is estimated using

$$\pi = \frac{\sum_{j=1}^{m} x_j}{m}$$
Supplementary references


Chapter 2

Unbiased genotyping of complex variation using exact alignment of k-mers to variant graphs
Chapter 2: Unbiased genotyping of complex variation

Unbiased genotyping of complex variation using exact alignment of k-mers to variant graphs
Jonas Andreas Sibbesen*1, Lasse Maretty*1, The GenomeDenmark Consortium & Anders Krogh1
*These authors contributed equally to this work.
1The Bioinformatics Centre, Department of Biology, University of Copenhagen, Ole Maaløes Vej 5, 2200 Copenhagen, Denmark.
Correspondence should be addressed to A.K. (krogh@binf.ku.dk).

Abstract
Genotype estimates from genome sequencing data are generally based solely on alignments of reads to a reference genome, but reads from regions with complex variation often fail to align. Missed variation can be partially retrieved using e.g. split-read alignment or alignment of de novo assemblies, but no single approach covers the entire variant spectrum. It is therefore necessary to combine variants from different methods, but some of these approaches produce many false positives and most provide only partial information about variant support in the data. Optimally, we therefore need to realign all reads against all variants to obtain an accurate genotype estimate, but this is currently not possible.

We here present BayesTyper, a new method that uses exact alignment of read k-mers to a graph representation of the reference and variant sequences to efficiently obtain unbiased information about the read support for any type of variant. This is then used to estimate genotypes using a probabilistic model. We show that our method significantly improves the genotyping of complex variation as judged using simulations as well as mendelian error rates and experimental validations from the GenomeDenmark project.

Introduction
Genotyping single nucleotide variants (SNVs) or short indels from second-generation genome sequencing data can be done accurately by mapping reads to a reference genome (using e.g. BWA-MEM) and then do genotyping (using e.g. HaplotypeCaller). Yet, genotyping larger variants or regions with dense variation remains non-trivial, because we cannot reliably align reads containing such variants to the reference genome using standard approaches.

This alignment problem can be partially mitigated by allowing for larger edit distances e.g. by combining multiple local alignments of a read as used in the split-read approach in BWA-MEM. Still, insertions longer than the read length (minus the length of anchors required for
aligning both ends) cannot be recovered from alignments of single reads alone as we need information from multiple reads to reconstruct the inserted sequence. The latter problem can be addressed by assembly-based approaches. For instance, Platypus\(^3\) uses “local assembly” of reads anchored either by local alignment or through the alignment of a mate-read within a genomic window. Alternatively, one can first de novo assemble scaffolds and then realign these back to the reference genome; this should give better results for longer insertions, but comes at the cost of increased computation time and noise. In most large-scale projects, like the 1000 genomes project\(^4,5\) and the Genome of the Netherlands project\(^6\), several approaches are used to ensure sensitivity. While the combination of these approaches allows us to discover variants not visible from standard reference mapping alone, they do not provide complete information about the read support for a particular variant. Hence, significant mapping-bias persists even after applying these measures.

It is therefore necessary to realign the reads to a database containing both the reference sequence and the variants discovered using the approaches outlined above. The set of putative variants can be augmented with already known variants or variants discovered in other individuals in the same study to further improve sensitivity. Rather than re-interrogating the entire set of reads for information about all putative variants, most methods - including the GATK indel-realigner\(^2\) and Platypus\(^3\) - reduce the problem to merely improving the alignment of reads already anchored to a particular region. However, this causes loss of information especially for larger insertions or when multiple variants occur within the same read. Importantly, even expanding the realignment to all reads that did not align in the first attempt, would still ignore important information as reads that did align initially may turn out to also align to an included variant. Optimally, we thus need to align the entire collection of reads to the variants and reference in an unbiased way.

This problem corresponds to aligning reads to a graph representation of the reference and variants in which nodes represent sequences (i.e. reference or alternative allele sequence) and edges represent junctions between them. However, while the graph alignment problem has received significant attention recently, new approaches are needed\(^7,8\) as the few available methods are too computationally demanding for our purpose\(^9-11\).

We here present a new method, BayesTyper, that uses exact alignment of read k-mers to a graph representation of variants to efficiently obtain unbiased information about the read support for any type of variant. It then uses this as a basis for estimating the genotypes of a population using a novel probabilistic model that allows us to estimate the posterior probabilities of the possible genotypes. We demonstrate that the method is accurate using both simulations and by applying the tool to generate the final call-set in the GenomeDenmark project in which a subset of the calls were experimentally validated (Sibbesen et al., manuscript submitted).
As our method allows researchers to accurately genotype complex variation using a combination of different variant signals, we expect that the BayesTyper will serve as a key component in making calling of complex variation a routine procedure that can be applied to any genome sequencing project.

Results

We first present an overview of the BayesTyper; a detailed description is provided in Online Methods. We then show tests of the method based on extensive simulations and compare with a number of state-of-the-art methods. Finally, we provide estimates of real data accuracy from parent-offspring mendelian error rates and experimental validations obtained from the application of our program to generate the final call-set in the GenomeDenmark project (Sibbesen et al., manuscript submitted).

BayesTyper

The objective is to query the sequencing reads for support for a set of candidate variants in a way that is unbiased with respect to the reference sequence and then use this information as a basis for genotyping. The overall idea is to compare k-mers of a certain preset size in the reads from an individual with k-mers from all known variants in a database. The posterior distribution over possible genotypes is then estimated from the counts of variant k-mers in the reads using a Bayesian model.

Our method thus requires a database with k-mer occurrences in the variants (including the reference sequence) and the sequencing reads. To generate a database of variant k-mers, we first observe that variants that are less than k nucleotides apart will share a k-mer and we thus need to co-estimate their genotypes. We therefore cluster variants that are less than k nucleotides apart and construct a variant graph for each such cluster. Co-estimating the genotypes of all variants in such a cluster corresponds to estimating the pair of underlying haplotypes (i.e. the “diplotype”), from which the genotypes of the constituent variants can be obtained. Optimally, we would simply enumerate all possible haplotypes (i.e. all possible paths in the cluster graph), collect the k-mer counts for these haplotypes in the sequencing reads, and use this to obtain a genotype estimate. However, as the number of haplotypes grows exponentially in the number of variants, this is not feasible for the large clusters. We therefore devised an iterative scheme as shown in Figure 1a-c.

First, variants are clustered using a k-mer size of 55 and graphs constructed for each cluster (Figure 1a). All k-mers (small-mers) are then enumerated in each graph using a value of k so small (k=18) that this task is feasible even in variant dense regions. Next, the occurrences of full-size k-mers (k=55) in the sequencing reads are counted for each individual using the
KMC2 program\textsuperscript{12}, but only reads that can be completely covered by small-mers are kept (Figure 1b). For smaller graphs, all possible haplotypes are then enumerated, whereas a heuristic algorithm that uses read k-mer information from each sample to select the most likely haplotypes are used for larger graphs (Figure 1c). Provided with a set of candidate haplotypes, we then enumerate k-mers (k=55) found in at least one haplotype and generate a table with sample counts for each of these k-mers.

We model the vector of these observed k-mer counts for each sample as generated by combining counts obtained from an individual’s diplotype, modelled using the negative binomial distribution, with counts originating from a noise process (Figure 1d). An individual’s diplotype is in turn modelled as drawn from a shared population of haplotypes whose frequencies are modelled using a sparse prior. Under this model, the posterior distribution over genotypes is then inferred using collapsed Gibbs sampling of diplotypes, haplotype frequencies and noise parameters. The posterior distribution over genotypes is then obtained directly from the diplotype posterior and the maximum a posteriori genotype provided as the genotype call together with the full posterior probability distribution.

### Simulated data

To assess the accuracy of our method and compare it with state-of-the-art mapping-based methods, we simulated 10x and 40x paired-end sequencing of ten female individuals from the CEU population in the 1000 genomes project\textsuperscript{4,5}. Read k-mer counts from KMC2 and reference alignments from BWA-MEM\textsuperscript{1} were used as sample input to BayesTyper and the mapping-based methods, respectively. A set of 14,760,269 candidate variants was obtained by combining all variants called in the entire CEU population (99 individuals) with variants predicted using six different, mapping-based variant discovery approaches. The latter were included to ensure a realistic level of genotyping complexity by providing plausible, but incorrect variants (e.g. originating from sequencing errors) that can give rise to false positive calls. All methods were evaluated using two different measures: allele sensitivity and precision. Allele sensitivity was defined as the fraction of alternative alleles in the CEU genotypes that were genotyped in the call-set. Allele precision was defined as the fraction of alternative alleles in the call-set genotypes that matched an allele in the corresponding CEU genotype.

On 40x data, BayesTyper and FreeBayes\textsuperscript{13} exhibit high and similar sensitivity for SNVs, whereas Platypus\textsuperscript{3} and HaplotypeCaller\textsuperscript{2} calls markedly fewer of the alternative alleles in the test population (Figure 2a). The reverse trend was observed for SNV precision, where Platypus and HaplotypeCaller rank highest and BayesTyper ranks lowest, but with only a small margin between the highest and lowest ranking (Figure 2b). For insertions and deletions, BayesTyper exhibits high sensitivity across the entire size spectrum (except for a single deletion bin), whereas the other methods generally have significantly lower sensitivity.
(Figure 2c). Importantly, the increase in sensitivity relative to mapping-based methods also comes with an increase in precision across nearly the entire size spectrum with the exception of a marked dip in precision for deletions around 100 nts (Figure 2d). On 10x data, BayesTyper generally exhibits higher sensitivity for both SNVs and SVs, but with precision either lower (for SNVs) or on par (for SVs) with the other methods; again a dip in both sensitivity and precision is observed for medium-sized deletions (Supplementary Figure 1a-d).

The advantage of the variant graph approach over mapping-based methods is expected to increase with distance between the sequenced haplotypes and the reference genome. We therefore estimated the correlation between genotyping sensitivity/precision and edit distance (Figure 2e-f). As seen in the figure, all other method exhibited significant declines in both sensitivity and precision for larger edit distances, whereas the BayesTyper maintained high sensitivity and precision independently of edit distance for both SNVs and SVs.

**Real data**

BayesTyper was used to generate the final set of genotype calls in the GenomeDenmark project, where 50 parent-offspring trios were sequenced to an average depth of 78x distributed across multiple library insert sizes. First, 21,234,298 candidate variants were obtained by combining permissively filtered variant calls from the HaplotypeCaller with structural variants called from alignments of high-quality de novo assemblies from all 50 trios. These were then provided as input to BayesTyper together with the sample k-mer counts and run in batches of five trios each. A detailed analysis of the resulting call-set is provided in Sibbesen et al. (manuscript submitted).

To assess the performance of BayesTyper on real data, we calculated the trio concordance rate (1 - mendelian error rate) on the GenomeDenmark calls for all trio genotype sets for which no genotypes were filtered and at least one genotype contained the alternative allele. The same procedure was repeated for the high confidence variants from the HaplotypeCaller (a subset of the variants provided as input to BayesTyper). The median trio concordance rate for BayesTyper is higher across all variant types compared to HaplotypeCaller with the largest differences observed for structural and multi-allelic variants (Figure 3a). A modified version of our Gibbs sampling method was devised for "nested" variants such as SNVs occurring within a deletion, but far from the breakpoints, as our standard inference is often not tractable for such arrangements (see Online Methods). Importantly, a relatively high concordance rate is observed for such variants and thus suggests that our approach to handling such variants is effective. We note that there are a few outliers with lower trio concordance (e.g. BayesTyper SNV calls for trio 1426), which likely reflect that both old and newly collected DNA was used for the sequencing of these trios. Importantly, the BayesTyper concordance rate remains relatively high across the structural variant size...
spectrum, whereas the concordance rate of the HaplotypeCaller decreases markedly for longer variants (Figure 3b). Finally, we computed the same measure on the subset of variants that were evaluated by both methods to ensure that the observed differences not only reflect differences in which variants were considered by the two callers (Figure 3c-d). As seen in the figure, the BayesTyper concordance rate remains highest across variant classes albeit with a smaller margin between the two methods.

Finally, a subset of 200 variants selected randomly in different size and allele frequency bins were experientially validated by cloning and sequencing in the GenomeDenmark project. Of these 79% of the variants validated corresponding to a true positive rate of 0.9 after correcting for the number of variants in each bin.

Discussion

We here present a new method for population genotyping based on a variant catalogue and a collection of sequencing reads. Our method solves a fundamental problem in structural variant calling, where multiple variant signals like split-read mapping and aligned de novo assemblies, need to be integrated to a variant call - optimally by realigning the entire set of sequencing reads against the variants and reference sequence. Indeed, both the 1000 genomes\(^5\) and Genome of the Netherlands\(^6\) project used an ensemble of different methods to call structural variants with sufficient sensitivity, but used either a simple consensus between callers or applied a genotyping method that does not perform complete realignment to generate the final call-set.

Our simulations indicate that for simple variation, our method has the same false positive rate as state-of-the-art methods, but finds many more variants. While it is possible that better results for the other methods could be obtained by carefully optimizing variant filters, this is not easily done on simulated data without adding bias and we therefore used the recommended filters without further optimization. Not least because BayesTyper, in contrast to most of the other methods, relies only on a single hard filter (requiring three k-mers with a positive count for each called allele), and thus leaves little room for overfitting to the simulation data.

For more complex variation our method generally exhibits both higher sensitivity and precision than all other methods and remains accurate across variant lengths and edit distances. Importantly, our performance on real data as judged using Mendelian error rates and experimental validation corroborated the results obtained on simulated data. Together, these results suggest that our method mitigates the reference-bias problem. This is important not only for structural variants, but also in regions with complex linkage disequilibrium structure such as the Human Leukocyte Antigen locus that is of critical
importance for medical genetics, but where current genotyping methods fail\textsuperscript{14}. Indeed, Dilthey et al. recently devised a new k-mer based genotyping method customly designed for HLA and showed improved genome inference in this region compared to a mapping-based approach\textsuperscript{15}. However, it is unclear whether their method can be to full genome genotyping.

In summary, we present an accurate and unbiased probabilistic method for genotyping arbitrarily complex variants from high-throughput sequencing data that we expect will improve the calling of complex variation from both existing and future genome sequencing data.
Online Methods

Querying reads. The input variants and reference sequence are represented as a graph in which nodes represent sequences (reference and alternative allele) and edges correspond to sequence junctions. To query the reads for variant information we use exact matching of k-mers between variant alleles in the graph and the sequencing reads and we therefore need to obtain k-mer profiles of both variants and reads.

For the sequencing reads, we simply count k-mers for each sample using the \textit{KMC2} program\textsuperscript{12} with default parameters except for \(k\), which was set to 55. We define the k-mer profile of a variant allele to be the multiset of all k-mers overlapping that allele (e.g. for a biallelic SNV, the reference and alternative allele profiles would each contain 55 k-mers of which some may occur more than once). The k-mer profiles of variants less than \(k\) nucleotides apart are dependent and we therefore need to genotype them together. More specifically, rather than doing genotyping in the space of single alleles, we instead cluster variants less than \(k\) nucleotides apart and work in the space of combinations of their constituent alleles (i.e. haplotypes). The objective is then to estimate the individual’s diplotype (i.e. haplotype pair) from which the genotypes for each variant can be directly obtained. Optimally, we would consider all possible haplotypes corresponding to all paths in the variant graph within a cluster, but this is not possible for larger clusters as the number of possible haplotypes grows exponentially in the number of variants. Instead, we will use a heuristic based on k-mer occurrences in the reads to select a set of candidate haplotypes, but as we also want to use information from the variant graph to limit the number of read k-mers considered, we propose an iterative approach.

We first construct a set of variant graphs (one for each cluster) from the variant and reference input (Figure 1a). Using the graphs, we then build a table of all k-mers (small-mers) overlapping variant alleles across the genome using a value of \(k\) small enough (currently 18) to enable us to simply enumerate all possible small-mers within a cluster without resorting to the heuristic method described below. We then compile a table with read k-mer counts from the \textit{KMC2} output, but only include k-mers that are completely covered by small-mers as this is necessary but not sufficient for a k-mer to be informative (Figure 1b). This requirement dramatically decreases the number of k-mers included without adding significant computation time at this stage.

A set of haplotype candidates is then generated from the variant graph using a heuristic n-best algorithm (Figure 1c). In this algorithm, only the n-best paths, as defined by a two level score, are kept when traversing the graph. First, the paths are ranked by the fraction of allele k-mers that are observed in a sample averaged across all alleles covered by the path. Second, for paths with equal score, a second minimum set cover approach is used, ranking
paths by the number of alleles covered by the path that are not covered by a path with a higher ranking. Using this scoring scheme, the 24 best paths are kept for each sample during graph transversal. The eight best paths from each sample are then combined across samples to create a final set of haplotype candidates used for inference. Only paths with a unique haplotype sequence are kept during graph transversal and the final merging step; the path covering the highest number of variants is selected when collapsing redundant paths. Finally, to ensure that almost all alleles are covered by at least one path, an additional dummy sample with 24 paths is introduced during the transversal, however this dummy sample ranks paths only by the minimum set cover (prioritising reference alleles over alternative alleles). All dummy paths that cover alleles not covered by the sample paths are included in the final candidate set.

Candidate k-mer profiles are then filtered before inference. K-mers that satisfies one of the following criteria are excluded:

- Is observed in one of the supplied decoy sequences (sequences with unknown ploidy such as the mitochondrial sequence should be provided as decoys).
- Is observed more than 127 times in the reference and the variant graphs combined.
- Is observed in more than one variant cluster group. Variant clusters are grouped together if they are part of the deleted or copied downstream sequence of a copy number variation.
- Has the same multiplicity in all haplotype candidates.

The output from the query step is then a set of haplotypes (represented as k-mer profiles) for each cluster and a table with sample k-mer counts for every k-mer observed in the haplotypes.

**Estimating genotypes.** The objective of the inference problem is to determine the most likely diplotypes (i.e. pairs of haplotypes) for the population given the sample k-mer profiles. To solve this we created a model for how k-mer counts are generated from an individual's diplotype and a method for doing inference in this model.

**Model**

The model is illustrated in Figure 1d. To allow for sharing of information between individuals, we let each haplotype $h$ in the set of haplotypes $H$ be associated with a random variable $f$ that models its population frequency\(^\text{16}\). As the input variant set will contain alleles that are not present in the population and some linkage between variants within a cluster is likely, we expect only a subset of the haplotypes to actually be present in the population. We therefore need to model sparsity in the haplotype frequencies (i.e. that only some of the frequencies will be non-zero) and introduce a sparse-Dirichlet distribution with sparsity parameter $\pi$ as a
prior on these frequencies. We then model the generation of each individual's diplotype $d \in D = \{(a, b) \mid a \in H \land b \in H \land a \leq b\}$ as an independent draw of two haplotypes from the categorical distribution specified by the frequencies.

For each individual, the vector of observed cluster k-mer counts $C_o$ is then modelled as generated from two sources. First, a vector of counts from the individual's diplotype $C_d$ is generated by independently drawing a count for each k-mer in the diplotype from the negative binomial distribution

$$P(c|p, r) = \binom{c + r - 1}{c} p^r (1 - p)^c$$

where the parameters $p$ and $r$ are specific to each individual, but shared across clusters. The negative binomial distribution is used, as sequencing read - and thus k-mer - counts are overdispersed relative to the Poisson distribution. Moreover, we truncate the negative binomial distribution at 255 to handle that the k-mers counts from KMC2 are capped at this value. Counts from k-mers with multiplicity (number of occurrences) higher than one are modelled by scaling the size parameter $r$ with the multiplicity.

Second, a vector of noise counts $C_n$ is drawn from a noise model to account for counts originating from sequencing errors. The noise model is build on the assumption that sequencing errors are rare and therefore only affect a subset of all k-mers. For each k-mer, we therefore draw a noise count from a zero-inflated Poisson distribution with parameters $z$ and $o$ to produce the vector of noise counts $C_n$. The zero-inflation parameter $z$ and noise rate parameter $o$ is specific to each individual. We use a shared beta-prior distribution on the zero-inflation parameters and a shared gamma prior on the noise Poisson parameters.

Finally, provided with counts from the diplotype and noise for each individual, the cluster k-mer count vector is then given by

$$C_o = C_d + C_n$$

Inference

The inference objective is to estimate the posterior distribution over diplotypes, haplotype frequencies and count distribution parameters given a vector of k-mer counts for each individual in a population. As sample k-mers not observed in any variant graph carry abundant information about the genomic k-mer count distribution, we use counts from these k-mers to pre-estimate the negative binomial distribution parameters for each individual. The parameters are estimated using method-of-moments such that
\[ p = \frac{\bar{p}}{\sigma^2} \quad \text{and} \quad r = \frac{\bar{p}^2}{\sigma^2 - \bar{p}} \]

where \( \bar{p} \) and \( \sigma^2 \) denote the sample mean and variance, respectively\(^\text{18}\). We further pre-
estimate the haplotype sparsity parameter \( \pi \) as the number of haplotypes required to explain all unique k-mers in a cluster divided by the total number of haplotypes as previously described by Maretty et al.\(^\text{17}\). The posterior distribution over diplotypes, frequencies and noise parameters is then inferred using the following collapsed Gibbs sampling scheme.

The sampler is initialised by drawing the noise parameters from their prior distributions with parameters of 1 and the haplotype frequencies for each cluster from the \(|H|\)-dimensional, symmetric Dirichlet distribution with parameter 1 (i.e. non-sparse). The algorithm proceeds as follows.

**Step 1: Sampling of diplotypes.** Independently for each cluster and for each individual, a diplotype \( d \) is drawn conditioned on the cluster k-mer counts, frequencies and count distribution parameters from the conditional posterior distribution given by

\[
P(d|C_o, f, p, r, z, o) = \frac{P(C_o|d, p, r, z, o)P(d|f)}{\sum_{e \in D} P(C_o|e, p, r, z, o)P(e|f)}
\]

where \( P(d|f) \) denotes the multinomial distribution with probability parameters given by the frequencies and \( n = 2 \), and

\[
p(C_o|d, p, r, z, o) = \prod_{c_o \in C_o} \sum_{c_o} P(c_o|d, p, r)P(c_o - c_o|z, o)
\]

where the product runs over all k-mers in the cluster and the first and last terms in the convolution are the probability mass functions for the negative binomial distribution with pre-
estimated parameters and the noise process defined above, respectively.

**Step 2: Sampling of frequencies.** Independently for each cluster, a frequency vector \( F \) is drawn conditioned on the set of diplotypes \( D \) and the sparsity parameter \( \pi \) using the conditional posterior distribution previously described by Maretty et al.\(^\text{17}\).

**Step 3: Sampling of noise parameters.** To make sampling of the noise parameters tractable, we first deconvolve the diplotype and noise contributions to the observed count for each k-
mer by sampling the noise contribution \( c_n \) from
\[ P(c_n|c_o, d, p, r, z, o) = \frac{P(c_o - c_n|d, p, r)P(c_n|d, z, p)}{\sum_{m=0}^{c_o} P(c_o - m|d, p, r)P(m|d, z, p)} \]

Noise parameters is then estimated by iteratively sampling the following two steps. First, the number of k-mers with a zero count \( c_{n0} \) contributed by the Poisson component \( c_{p0} \) as opposed to the zero-inflation component is sampled from a binomial distribution conditioned on the current noise parameters and the allocated noise counts from

\[ P(c_{p0}|c_{n0}, z, o) = Binom\left(c_{n0}, \frac{(1 - z)e^{-o}}{z + (1 - z)e^{-o}}\right) \]

Second, the zero-inflation parameters \( z \) and the noise rate parameters \( o \) are sampled from their beta-posterior and gamma-posterior, respectively, conditioned on \( c_{p0} \) and the allocated noise counts. This procedure is iterated 10,000 times to ensure convergence.

To reduce computation time, we first estimate the noise model parameters by running 200 iterations of the Gibbs sampler on 1,000,000 randomly chosen autosomal single nucleotide polymorphisms containing no ambiguous bases, excluded k-mers and k-mers with a multiplicity above one. We then fix the noise parameters to the mean of the last 100 collected samples for all clusters and estimates their posterior distribution over genotypes by iteratively sampling a diplotype (step 1) and haplotype frequencies (step 2) (i.e. without sampling the noise parameters). To handle mixing problems due to multi-occurring k-mers (see below), we run ten independent Gibbs sampling chains. In each of these, we first iterate a 100 burn-in samples, followed by a 100 samples used for the estimation of the posterior distribution over diplotypes. Finally, the genotype posterior estimates are obtained by summing the posterior probabilities of diplotypes that specify the same genotype.

**Nested variants.** Variants that remove or substitute larger parts of the reference sequence (e.g. deletions, inversions) and contains other variants introduce dependencies that can vastly increase the number of possible haplotypes. For instance, a long deletion may contain hundreds of SNPs, many of which may be further than k apart, implying that the number of possible haplotypes will not only be very large, but that we will also lose phase information which in turn will degrade the performance of our haplotype enumeration algorithm.

We solve this by defining the outer variant to be independent of the variants (clusters) contained within it if these are located more than k nucleotides away from the variant breakpoint. We will refer to the latter class of variants as nested variants and note that a nested variant can itself contain other nested variants (e.g. due to a deletion within a
deletion). Note that, both the outer and nested variants will belong to the same variant cluster group. The genotype of such nested variants depend on the genotype of the outer variant and we can represent this dependency structure as a tree as shown in Supplementary Figure 2. Provided with such a tree structured variant arrangement, we can then do sampling sequentially by first sampling the diplotype of the root variant and then move down the tree each time estimating the diplotype of a variant dependent on the diplotype estimated in the outer variant (i.e. at the parent node in the tree). Finally, while the genotyping of outer variants is done independently of the nested variants, we still use any reference k-mers contained within a structural variant in the genotype estimation provided that they do not overlap any of the nested variants.

**Multi-occurring k-mers.** Many k-mers occur at multiple places in the genome e.g. due to repeat content, duplications or inversions. Depending on the type of multi-occurrence, we handle such k-mers differently. k-mers that are observed more than once in the same variant cluster are handled as mentioned above by just scaling the negative binomial distribution with its multiplicity. The same approach is used for k-mers that are observed both in a variant cluster and one or more non-variable genomic regions (i.e. between variant clusters). However, k-mers that are observed in more than one variant cluster within the same group is handled differently as they imply a dependency between the clusters (k-mers observed in different groups are excluded). To handle these, we use the combined multiplicity across variant clusters and updates a cluster's contribution to the combined multiplicity based on the sampled diplotype in each Gibbs iteration. To initialize the multiplicity, multi-cluster k-mers are not used for inference in the first iteration. A maximum of 10,000 multi-cluster k-mers are used during inference for each cluster to decrease computational complexity. This set is sampled randomly and independently for each gibbs-chain.

**Filtering.** By default, BayesTyper applies two hard filters to the data after inference to handle errors arising from data properties not completely accounted for by the model such as the sparsity of non-excluded k-mers in highly repetitive regions:

- Genotypes without a unique maximum posterior probability (GPP)
- Sampled alleles (partial filtering) with less than three informative k-mers observed in a sample (NOK), where a k-mer is considered informative if it has unique multiplicity in the allele compared to the other alleles in the variant.

We do not recommend a value below three for the NOK filter, however if a more conservative call-set is of interest this threshold can be increased. Moreover, for applications where only the maximum a posteriori genotype estimate, rather than the entire posterior distribution over genotypes, is of interest we recommend filtering genotypes with a GPP below 0.9.
Implementation. The method is implemented in C++ and supports multithreading. It comes with extensive facilities for merging input VCF files from different sources and for post-processing a call-set to produce a VCF file ready for release. Source code and static builds are available from https://github.com/bioinformatics-centre/BayesTyper under the MIT license.

Simulated data. Reads were simulated from ten randomly chosen female individuals from the CEU population in the 1000 genomes project\textsuperscript{4,5}. More specifically, symbolic alleles in 1000 genomes phase 3 call-set were first converted to sequences (excluding mitochondrial mobile insertions) and all variants left-aligned and normalised using \textit{bcftools (v1.2-123)}\textsuperscript{19}. Next, haplotype sequences from the ten individuals were constructed using the phased genotype calls and four batches of 5x paired-end 100 bp reads were simulated from each haplotype using \textit{pIRS (v2.0.0)}\textsuperscript{20} with default settings (mean insert size of 180 and a standard deviation of 18).

To construct a set of candidate variants to use in the re-genotyping benchmark, the reads were first mapped to the reference genome (hs37d5) and sorted using \textit{bwa-mem (v0.7.12)}\textsuperscript{1} and \textit{samtools (v0.1.19)}\textsuperscript{21}, respectively, followed by duplication marking using the \textit{Picard toolkit (v2.0.1)}. The complete set of mapped reads (i.e. 40x) were used to predict a set of candidate variants using six different methods all capable of discovering deletions and insertions with breakpoint accuracy (table 1). All methods, except \textit{Platypus}, were run with default settings. For \textit{Platypus --assembly} and \textit{--assembleBrokenPairs} were enabled. The predicted variants from these six methods were then filtered to remove variants without breakpoint resolution and insertions without sequence content. In addition, variants larger than 3M bases were also filtered. Finally, the filtered call-sets were combined with all variants from the 1000 genomes phase 3 call-set that were called in at least one individual in the CEU population (99 individuals), creating a final candidate set containing 14,760,269 variants.
Table 1: Candidate variants. Summary statistics of candidate variant call-sets.

The re-genotyping accuracy of FreeBayes, Platypus, HaplotypeCaller was then assessed on this combined set using the mapped sequencing reads. For the HaplotypeCaller and FreeBayes, all variants larger than 100 bases were first removed as the methods could not run on these, whereas for Platypus all variants larger than 10,000 were removed. Default settings for all methods were used besides specifying that all variants should be written to output regardless of quality. Instead of the mapped sequencing reads the BayesTyper were given a k-mer database (k=55) of the sequencing reads created using KMC2 (v2.3.0).

As post-filtering of the output calls is recommended by the developers of each used tool, we filtered all call-sets using bcftools (v1.2-123) and a set of custom hard-filters (table 2). The same filters were used for both the 10x and the 40x benchmark. For the BayesTyper, we used the default filters described above together with a genotype posterior probability (GPP) threshold of 0.9; nested genotypes were filtered if the outer genotyped allele was filtered.

Table 2: Call-set filters. Hard-filters used on each genotyped call-set. *Same version used as in table 1. MNV: Multiple nucleotide variant. INV: Inversion.
The filtered call-sets were then compared to the CEU calls used to generate the simulated sequencing reads. Since the same variant can be encoded in numerous different ways it is not possible to simply just compare the genotypes in the CEU call-set with the predicted genotypes at the same position. To alleviate this, all possible haplotypes were instead generated from the predicted genotypes and compared to the CEU haplotypes around the allele/genotype to be evaluated. More specifically, the haplotypes were initialised at the variant position and extended in both directions until either the haplotype did not align perfectly to a CEU haplotype in the same region or had extended 1000 bases in both directions outside the variants. Allele sensitivity was defined as the fraction of called alleles in the CEU call-set that could be fully recovered by a haplotype generated from the predicted genotypes. Allele precision was defined as the fraction of predicted genotyped alleles, or haplotypes extended from these, that matched the same region of a CEU haplotype (not allowing both genotyped alleles to match the same haplotype). Both measures were calculated independently for each of the 10 individuals. The edit distance measure was calculated by counting the number of SNVs and SVs in the CEU haplotypes in a window extending 99 bases (read length - 1) upstream and downstream of the allele of interest. When calculating the edit distance of the predicted call-sets (allele precision), we only used those where the position of the variant of interest was not located inside a structural variant in the CEU call-set. This was required in order to be able to anchor the edit distance window correctly on the CEU haplotypes. For computing the allele precision as a function of edit distance, the mean distance of the two haplotypes was used.

Real data from GenomeDenmark. BayesTyper was used to generate the main call-set in the GenomeDenmark project (Sibbesen et al. manuscript submitted). The data consists of approximately 78X genome sequencing data distributed across a range of paired-end libraries (180bp, 500bp and 800bp) and mate-pair libraries (2kb, 5kb, 10kb and 20k) from 50 parent-offspring trios. A detailed description of the data processing pipeline is described in Sibbesen et al. (manuscript submitted). In brief, one set of candidate variants was generated by mapping all reads to the reference genome followed by variant calling using the HaplotypeCaller\(^2\). Another set of variants was generated by doing de novo assembly of all 150 individuals using AllPaths-LG\(^25\) followed by alignment and variant calling using AsmVar\(^26\). These call-sets were then permissively filtered to reduce computational complexity by taking the 99.9% sensitivity tranche of the HaplotypeCaller call-set and discarding SNVs from the AsmVar call-set. The filtered call-sets were then merged into a single set. Sample k-mers (k=55) were counted using KMC2 (v2.2.0)\(^12\). We omitted singletons in the counting to reduce the size of the k-mer set as the high sequencing depth implies that most non-noise k-mers will have a count larger than one. The k-mer database and the merged call-set were then provided as input to the BayesTyper, which was run with default parameters. Please see Sibbesen et al. (manuscript submitted) for a detailed description of the pipeline including genotype filtering.
Trio concordance defined as the fraction of parent-offspring trios where the child genotype is explained by the parental genotypes was calculated for both the BayesTyper call-set, requiring a genotype posterior (GPP) of at least 0.9, and the HaplotypeCaller call-set, which was filtered using a tranche sensitivity threshold of 99.0%. Finally, trios not satisfying all of the following criteria were filtered prior to calculation of the concordance rate:

- Variant is located on an autosome
- At least one genotype in the trio is not a reference call
- None of the genotypes in the trio are filtered
- The trio is informative (bi-allelic variants, where both parents are heterozygotes are uninformative)

Concordance rates were computed on both call-sets independently. To also evaluate BayesTyper and HaplotypeCaller on exactly the same set of variants, a set containing only the variants shared between the two call-sets was constructed requiring that the same variants satisfied the above criteria in both call-sets.

Note that an older version of BayesTyper was used to create the GenomeDenmark call-set. The main improvements in the current version over the GenomeDenmark version is: A simplified and improved noise model, handling of multi-cluster k-mers within the same group, running multiple gibbs sampling chains with different starting values and partial filtering.

**Acknowledgement**

This work was supported by grants from the Novo Nordisk Foundation and the Danish National Advanced Technical Foundation.

**References**


Chapter 2: Unbiased genotyping of complex variation...
Figure 1: BayesTyper. (a) A variant graph is constructed from the input set of variants and the reference genome. All possible 18-mers (small-mers) in the graph are then tabulated (colors represent reference and alternative variant alleles, circles represent k-mers - only a single k-mer from each allele combination is shown). (b) All k-mers in the sequencing reads are then tabulated for each sample using KMC2 and k-mers that are fully covered by small-mers kept and used in the downstream analysis. (c) Haplotype candidates and their corresponding k-mer profiles are generated by traversing the variant graph for each sample using an N-best algorithm, where the score is based on the sample k-mer counts. In the end, all unique sample paths including a set of dummy paths, which cover alleles not included by the sample paths, are combined to create the final set of haplotype candidates. (d) Inference is based on a generative model. First, the individual’s haplotype pair is drawn conditioned on their population haplotype frequencies. Then, conditioned on the haplotype pair, each individual’s vector of observed k-mer counts is modelled as generated by combining counts from the individual’s haplotypes with noise counts (only k-mers overlapping both variants are shown for simplicity).
Figure 2: Allele sensitivity and precision (40x). (a,b) Allele sensitivity and precision for SNVs and inversions (minimum length of 100). Allele sensitivity was defined as the fraction of alternative alleles in the CEU genotypes that were genotyped in the call-set. Allele precision was defined as the fraction of alternative alleles in the call-set genotypes that matched an allele in the corresponding CEU genotype. (c,d) Allele sensitivity and precision as a function of structural variant length for both insertions and deletions. (e,f) Allele sensitivity and precision as a function of the flanking region (99 nts in both directions) edit distance for both SNVs and short structural variants (< 16 nts), where the edit distance is defined as the number of differences between the CEU haplotype and the reference genome (a structural variant is counted as a single edit). MNV: Multiple nucleotide variant.
Figure 3: Trio concordance. (a) Trio concordance (1 - mendelian error rate) of the 50 trios for each variant type. (b) Trio concordance as a function of structural variant length for bi-allelic insertions and deletions. (c, d) Same as (a) and (b) on the subset of variants that were evaluated by both BayesTyper and HaplotypeCaller.
Supplementary Figure 1: Allele sensitivity and precision (10x). (a,b) Allele sensitivity and precision for SNVs and inversions (minimum length of 100). Allele sensitivity was defined as the fraction of alternative alleles in the CEU genotypes that were genotyped in the call-set. Allele precision was defined as the fraction of alternative alleles in the call-set genotypes that matched an allele in the corresponding CEU genotype. (c,d) Allele sensitivity and precision as a function of structural variant length for both insertions and deletions. (e,f) Allele sensitivity and precision as a function of the flanking region (99 nts in both directions) edit distance for both SNVs and short structural variants (< 16 nts), where the edit distance is defined as the number of differences between the CEU haplotype and the reference genome (a structural variant is counted as a single edit). MNV: Multiple nucleotide variant.
Supplementary Figure 2: Nested variants. Variants that remove or substitute larger parts of the reference sequence (e.g. deletions, inversions) and contain other variants introduce dependencies between their genotypes. We refer to these variants as nested and represent their genotype dependency structure as a tree. Gibbs sampling is performed sequentially by traversing the tree from the root, sampling a diplotype at each node conditioned on the parent node.
Chapter 3

De novo assembly of 150 Danish genomes reveals rich structural complexity
De novo assembly of 150 Danish genomes reveals rich structural complexity

Jonas Andreas Sibbesen§1, Jacob Malte Jensen§2,3, Bent Petersen§4, Lasse Maretty§1, Siyang Liu§1,5, Palle Villesen§2,3,6, Laurits Skov§2,3, Kirstine Belling§4, Christian Theil Have7, Marie Grosjean4, Jose Izarzugaza4, Jette Bork-Jensen7, Jakob Grove3,8, Thomas D. Als3,8, Shujia Huang5,9, Yuqi Chang10, Ruiqi Xu5, Weijian Ye5, Junhua Rao5, Xiaosen Guo10, Jihua Sun5,7, Hongzhi Cao10, Chen Ye10, Johan v Beusekom4, Thomas Espeseth11, Esben Flindt4, Rune M Friborg2,3, Stephanie Le Hellard11,12, Christian M Hultman13, Francesco Lescai3,8, Shengting Li3,8, Ole Lund4, Peter Løngren4, Thomas Mailund2,3, Maria Luisa Matey-Hernandez4, Ole Mors3,6, Christian NS Pedersen2,3, Thomas Sicheritz-Pontén4, Patrick Sullivan13,14, Ali Syed4, David Westergaard4, Rachita Yadav4, Ning Li5, Xun Xu10, Torben Hansen7, Anders Krogh1, Lars Bolund8,10, Thorkild IA Sørensen7,15,16,17, Oluf Pedersen7, Ramneek Gupta4, Simon Rasmussen4, Søren Besenbacher2,6, Anders D. Børglum3,8, Jun Wang*3,10, Hans Eiberg*18, Karsten Kristiansen*10,19, Søren Brunak*4, Mikkel Heide Schierup*2,3,20

§ contributed equally

* Correspondence. wangjun30@gmail.com, he@sund.ku.dk, kk@bio.ku.dk, brunak@cbs.dtu.dk, mheide@birc.au.dk
Chapter 3: De novo assembly of 150 Danish genomes . . .

22 Affiliations

23 1. Bioinformatics Centre, Department of Biology, University of Copenhagen, 2200
Copenhagen N, Denmark

24 2. Bioinformatics Research Centre, Aarhus University, 8000 Aarhus C, Denmark

25 3. iSEQ, Centre for Integrative Sequencing, Aarhus University, 8000 Aarhus C, Denmark

26 4. Center for Biological Sequence Analysis, Department of Systems Biology, Technical
University of Denmark, Kemitorvet, 2800 Kongens Lyngby, Denmark

27 5. BGI-Europe, Ole Maaløes Vej 3, 2200 Copenhagen N, Denmark

28 6. Department of Clinical Medicine, Aarhus University, 8000 Aarhus C, Denmark

29 7. Novo Nordisk Foundation Center for Basic Metabolic Research, Section of Metabolic
Genetics, University of Copenhagen, 2100 Copenhagen Ø, Denmark

30 8. Department of Biomedicine, Aarhus University, 8000 Aarhus C, Denmark

31 9. School of Bioscience and Biotechnology, South China University of Technology,
Guangzhou 510006, China

32 10. BGI-Shenzhen, Shenzhen 518083, China

33 11. NORMENT, KG Jebsen Centre for Psychosis Research, Department of Clinical Science,
University of Bergen, Bergen, 5021, Norway

34 12. Dr E. Martens Research Group of Biological Psychiatry, Center for Medical Genetics
and Molecular Medicine, Haukeland University Hospital, Bergen, 5021, Norway

35 13. Department of Medical Epidemiology and Biostatistics, Karolinska Institutet,
Stockholm, 17177, Sweden

36 14. Department of Genetics, University of North Carolina, Chapel Hill, NC 27599-7264,
USA

37 15. Institute of Preventive Medicine, Bispebjerg and Frederiksberg Hospital, The Capital
Region, Copenhagen, Denmark
Chapter 3: De novo assembly of 150 Danish genomes ...
Most known genetic variation in human genomes has been called from comparison of short reads to the reference genome, but this approach is biased against finding complex variation. We sequenced 150 individuals from 50 parent-offspring trios to very high coverage including multiple insert-size libraries. We show that each genome could be independently de novo assembled into a small number of high-quality scaffolds (N50>21 Mb). By calling variants from comparison of these assemblies with the reference, we show that the resulting variant call set is far more complete in terms of complex variation than previous studies. Importantly, even the complex 4Mb HLA region was assembled and resolved into haplotypes, and major parts of the Y chromosome including some palindromes were assembled with high accuracy. Finally, we show that our variant call-set allows for the genotyping of many more complex variants when used as a reference-panel for imputation into SNP-chip data.

Several thousand complete human genomes have now been published\(^1\)-\(^3\) and projects aiming at sequencing hundreds of thousands of genomes have been initiated\(^4\). Central goals of these projects have been to characterize as much of the genetic variation as possible and to use this information to augment association mapping studies of complex disorders and other phenotypic traits and to identify mutations causing monogenic diseases.

Single nucleotide and short indel variants have been called by mapping short sequence reads to the reference genome, often combined with realignment and local reassembly of regions with poor mapping quality. Larger structural variants are more challenging and a number of approaches based on mapping patterns or local reassembly have been
employed, see e.g. refs 2-5-7. While a substantial part of the variation can be confidently
typed by these approaches, many regions of the genome are either too complex or too
repetitive and are often left out from analyses. These include important regions such as
the HLA region, the Y chromosome and complex gene families. Although methods have
been developed to get as much information out of short reads as possible8-10, high
quality de novo assemblies of genomes for many individuals are needed. This has not
been possible in previous large-scale studies, where insufficient sequencing coverage
and read lengths have precluded assembly of regions with large repeats or large
structural variants. Recent developments in sequencing technology with much longer
reads do provide a potential solution11-15 as does the addition of chromatin interaction
data for scaffolding16, but these approaches are still prohibitively expensive for more
than a few genomes.

Our aim was to produce a Danish pan-genome based on de novo assembly of 150
individual genomes. We chose an experimental design based on parent-offspring trios
that combines high coverage sequencing with high quality mate pair libraries with insert
sizes up to 20 kb. We report 150 individual high quality de novo assemblies that we
compared with the reference genome to obtain a rich catalogue of novel complex
variation and which we use to obtain complete extended HLA haplotypes and
assemblies of major parts of the Y chromosome.

Results

The 50 trios were selected from the Copenhagen Family Bank, a database of DNA
samples from approximately 1,000 large Danish families established in 197217. The final
set was selected from a representative set of 60 candidate trios based on PCA analysis of
SNP-chip data in order to obtain an ancestry specific reference genome of Denmark
(Online methods). The sequence coverage after mapping was approximately uniform and averaged 78X for the 150 individuals, with ~50X from paired end libraries (180bp, 500bp and 800bp insert sizes) and ~28X from mate pair libraries (2kb, 5kb, 10kb and 20k insert sizes) (see Supplementary figure 1, 2 of Supplement)).

**De novo assembly**

Using Allpaths-LG\(^{18}\), we were able to create *de novo* assemblies for each of the 150 genomes of very high quality. Specifically, the *de novo* assemblies have a median scaffold N50 of ~21 Mb (maximum ~30Mb) and only around 3% missing base pairs, which are typically located in a few larger gaps. The 100 largest scaffolds in each assembly typically cover more than 80% of the genome with the largest scaffolds exceeding 110Mb in size, longer than any previously published *de novo* assembly (Figure 1, bottom insert and Supplement). To evaluate the accuracy of the assemblies, we subsequently aligned the scaffolds for each individual to the human reference genome (hg38) using colinearity as a measure of scaffold quality. Figure 1 displays an example individual where the ten largest scaffolds for each chromosome are shown to align to one chromosome and to cover almost all of the euchromatic genome (summary of all 150 assemblies, Figure 1, top insert). In several cases, scaffolds cover almost entire chromosome arms. Only in rare cases do we find that large scaffolds break and align to more than one chromosome (Figure 1, top insert) suggesting that even the largest scaffolds are rarely chimeric. We conclude that the contiguity of each assembly is superior to all previously published *de novo* assembled human genomes except for the reference genome and that of the hydatidiform haploid mole\(^{12,19}\) (Figure 1, bottom insert).
We compared the 150 assemblies in order to identify regions that generally resist assembly into scaffolds (bright red lines in ideograms, Figure 1). These regions are mainly located around centromeres or associated with known large structural variants found in hg38, that are not shared with any of the 100 independent genomes of Danish ancestry. De novo assemblies allow identification of specific genomic events such as viral integration; for instance, we observe a complete telomeric HHV6b integration in one family shared by mother and daughter with complete assembly of the viral genome (Supplementary figure 3,4). In addition to Allpaths-LG we compared to SoapDenovo and SGA and found these algorithms to have lower N50 measures for almost all individuals (see Extended Figure 1 and Supplement for a detailed comparison).

**Variant calling and genotyping**

We devised a hybrid variant calling strategy that first employs both mapping- and assembly-based approaches to discover candidate variants and then genotypes the study population on these variants using a probabilistic method (Figure 2a). The mapping-based approach was based on GATK using a permissive filter on the recalibrated variant quality scores (99.9% tranche) and identified 21,234,298 candidate variants. In the assembly-based approach, *de novo* assemblies were aligned to the hg38 reference genome and candidate variants were identified using the AsmVar pipeline producing a total of 11,372,493 non-SNV candidates. We genotyped the study population on the combined set of variants by re-interrogating the raw sequencing data to get a consistently merged and scored set (see methods and Sibbesen et al., *submitted* and attached). To estimate the true positive rate (TPR) of the variant calls, 200 variants were selected randomly across length and allele frequency bins and subjected to experimental validation by cloning and sequencing; variants longer than 50 bp were
further subjected to validation by gel electrophoresis of the PCR products. 79% of the
variants validated corresponding to an estimated TPR of 0.90 after taking into account
the number of variants in each bin. The validation rate was highest for shorter and non-
repetitive variants (Supplementary Figure 5). The high TPR is further supported by an
overall trio concordance rate of 0.98.

A summary of the candidate and final call-set is shown in Figure 2b. Of the candidate
SNVs 84.5% were called (i.e. genotyped in at least one individual); 14.8% of the called
SNVs were observed neither in dbSNP142 nor in the 1000 genomes (1000G) project
phase 3 call-set (novel). A markedly different picture was observed for structural
variants, especially longer ones, for which significantly less candidates were called, but
more were novel (e.g. for long insertions 24.7% were called, but 76.2% of the called
insertions were novel, increasing to almost all variants as the size increases (Figure 2c,
Extended Figure 2a)). The distribution of insertion and deletion calls was fairly balanced
with most longer variants (especially insertions) provided by the assembly-based
approach (Figure 2c and Extended Figure 2b) and thus not biased towards deletions as
observed in previous studies such as the genome of the Netherlands (goNL)23.
Importantly, even for deletions, the number of large variants called markedly exceeded
those reported for previous large-scale sequencing efforts like the 1000G project. For
instance, we identified 61,678 larger deletions (>= 50 bp), whereas 1000G identified
42,279 of such deletions despite 25 times more individuals from diverse human
populations being sequenced 7. The allele frequency was generally observed to decrease
with variant length (Figure 2d) suggesting purifying selection against longer variants.
When conditioning on variants being novel we observe a much higher frequency of
longer variants suggesting that many were missed in previous studies (Figure 2e).
To further characterize the variants, we searched for repetitive content in the variant sequences. Many of the called structural variants were classified as repetitive with 20,262 of them containing either whole or partial ALU elements with 93.3% of these being novel (Extended Figure 3a). Shorter repetitive variants (<= 50 bp) were mainly classified as simple repeats that exhibited a strong tendency towards even numbered allele lengths (Extended Figure 3b and 3c), whereas longer variants, especially deletions, contained significant amounts of ALU derived sequence (Extended Figure 3b). Indeed, the observed peak in the number of structural variants with a length between 300 and 350 bases (Figure 2c and Extended Figure 3b) can be explained by ALU polymorphisms.

Evaluating loss-of-function (LOF) mutations we found, using the same filters as in Besenbacher et al.\(^2\), that indels account for \(\sim 2/3\) of LOF mutations compared to \(\sim 1/2\) in Besenbacher et al.\(^2\). Each individual carries on average 58 homozygous LOF mutations compared to previous estimates of \(\sim 30\) (Extended Figure 4)

### De novo variation

We identified 3,205 de novo SNVs and 322 de novo indels from interrogation of 2.5 Gb of the genome. Eleven of twelve de novo indels passed experimental validation by Sanger sequencing (92%). For an average paternal generation time of 27.7 years (using a fully probabilistic approach to establish the callable fraction of the genome) we find a mutation rate per generation for SNVs of \(1.28 \times 10^{-8}\) and for short indels of \(1.3 \times 10^{-9}\). These rates are higher than in other recent studies \(^{14,24,25}\), suggesting that previous studies have less access to complex, more mutable parts of the genome. We find many more de novo deletions than insertions and an overrepresentation of even-sized variants (Extended Figure 5). We confirm a strong correlation with paternal age for both the mutation rate of SNVs and of indels (Figure 3a). The high sequencing coverage and
Chapter 3: De novo assembly of 150 Danish genomes... 

different insert size libraries allows us to determine the parent of origin by read backed phasing for 49% of the SNVs and 48% of the indels and thus disentangling paternal and maternal mutation processes (Figure 3b and Supplement). We find that significantly more SNVs than indels are of paternal origin (78% vs 66%, P=0.002), in agreement with indels being less dependent on replication errors. We find a direct and highly significant effect of maternal age on the number of SNV mutations coming from the mother (Figure 3b). This is consistent with a recent maternal age analysis that did not establish the parent of origin26 and it shows that de novo mutations accumulate over time in the female germline even though there are no further cell divisions. The CpG mutations have a smaller (and not significant) correlation to paternal age than non-CpG mutations and as a consequence the fraction of CpG mutations are negatively correlated with the paternal age (Figure 3c). Transitions are known to occur frequently at methylated CpG sites due to cytosine deamination damage and we observe that the mutation rate is significantly higher at CpG sites that tend to be methylated in humans (Figure 3d).

Resolving HLA and the Y chromosome

The high quality de novo assemblies allowed us to interrogate variation in the HLA and the Y chromosome, which is typically ignored in resequencing studies due to their structural complexity. The full 4Mb HLA region is covered by 1-7 scaffolds in our individual assemblies (see Supplementary figure 7). We extracted the assembly graph from the scaffolds from each trio and used a combination of alignment within the trio, transmission, remapping against the scaffolds and read-backed phasing to determine the full HLA haplotypes in 25 of the 50 trios for a total of 100 complete HLA haplotypes where >92% of all the variants are phased. These 100 HLA haplotypes complement the pgf haplotype of hg38 and the seven previously published haplotypes where only one
(cox) is complete (the other six have 7-49% gaps), bringing the number of complete HLA haplotypes to 102. The accuracy of the haplotypes was reflected in our validation experiments (Supplemental Table 1), which validated the existence and correct phase of 86.2% of the interrogated variants. SNV variation and indel variation is extensive (Figure 4a). Figure 4b shows an example haplotype dot plot of one HLA haplotype against the hg38 reference with repetitive structures and regions of very high diversity corresponding to class I and class II classical HLA loci. We find strong linkage disequilibrium within several regions of the HLA loci (Figure 4c).

We also fully assembled ~20Mb of the Y chromosome in long scaffolds (N50 scaffold size over 50 fathers of 1.5 Mb, Figure 5a). It is mainly the very long palindromes and the X-transposed region that resist assembly (Figure 5a). The average concordance between father and son Y-chromosomes is 99.97% for the X-degenerate region (6,327,706 bp) and 99.95% for ampliconic regions (4,697,107 bp). We identify 4,692 SNVs, 513 deletions (range 1-4,119 bp), 349 insertions (range 1-10004 bp) and 29 complex variants. Variants had an experimental validation rate of 91.6% (Supplemental Table 1). The assembled Y-chromosomes belong to haplogroups that are common in Europa and Scandinavia with R1b1 (30 individuals) and I1a1 (26 individuals) being the most common. Palindromes 4-8 assemblies were initially collapsed to one palindrome, but read backed phasing allowed almost complete resolution of the palindrome arms (77.8% could be phased), which in turn allowed for gene conversions to be identified (Figure 5b).

Discussion

We have shown that it is possible to assemble human genomes into very few accurate scaffolds (>80% of the genome in <100 scaffolds) from short sequencing reads if the
experimental design includes sufficient sequence coverage and paired-end information ranging different insert sizes. Our design is economically feasible even for hundreds of individuals. Avoiding the use of a single reference genome offers an unbiased approach to variant calling across the full range of human genetic variation, and our results show that most previous studies have been biased against finding complex variants. High quality assemblies, however, is only the first step, since variant calling requires assemblies to be aligned. Comparing fully assembled genomes remains challenging since it depends on full genome alignment, which is computationally complex and prone to false variation calling due to alignment errors. To alleviate this, we applied a probabilistic method that effectively and accurately combines candidate variants from both standard mapping-based calls and aligned contigs, and genotypes the study population on this set, while removing false positives. This strategy not only significantly reduced the bias normally observed towards deletions, but also allowed us to discover substantially more structural variation compared to both the 1000G and goNL projects. Indeed, we found 40% more large deletions (>=50 bp) than the 1000G project and more than twice as many deletions compared to the goNL project in much fewer individuals. Importantly, even when taking into account that some of the variation may be explained by less conservative variant calling as suggested by our validation experiments, we still identify much more structural variation per individual as compared with these previous studies. Furthermore, de novo assembly is the only way to interrogate specific complex regions of the genome such as the HLA. HLA is by far the most important region of the genome for risk alleles for autoimmune as well as other diseases but the specific location of these variants have remained elusive since the complex haplotype structure of the region was very scarcely known. We have now
added 100 full haplotypes to the eight known haplotypes and expect these to be of great value when revisiting association studies.

One of the primary goals of the project is to improve clinical genetics interpretation in Denmark by establishing a regional reference genome. Our identification of an unprecedented catalogue of structural variation from the Danish population suggests that a much larger set of variants can be imputed with greater precision than when using other reference panels alone. In the Danish GOYA obesity cohort (5,222 cases and controls) we found a higher number of imputed variants in key genomic regions (Extended Figure 6a). Combining our panel with the 1000G panel allows imputation of 1,223,384 more variants than the 1000G panel alone after quality control and leads to a higher accuracy independent of the Minor Allele Frequency (Extended Figure 6b and 6c). More than a fifth of the additionally imputed variants were insertions and deletions (Extended Figure 6c). We believe that high quality de novo assembled genomes like the ones presented here can be of use in precision medicine initiatives which now are being initiated in many countries including Denmark.
Chapter 3: De novo assembly of 150 Danish genomes ...
Figure 1. Assembly quality of 150 de novo assemblies. For each chromosome, alignment of the 10 largest scaffolds (yellow–blue) to the reference genome shows near perfect continuity, with very few chimeric scaffolds (pink). Bar height reflects alignment length, showing most secondary alignments are very small (<1kb). Regions without alignments across all assemblies (bright red) overlap primarily with reference gaps (dark red). The two holes in the largest scaffolds on chromosome 6 and 14 correspond to known alternative loci in the reference genome. Top insert: Most of the assemblies have very few breaks in the largest scaffolds comparable to the values for five other de novo assemblies: NA12878 assembled by Allpaths-LG (red), SGA (yellow) and SOAPdenovo (green), NA18507 assembled by SOAPdenovo (blue), YH (pink), CHM1 (red) and HuRef (yellow). Bottom insert: Most of the 150 assemblies cover more than 80% of the genome with the largest scaffolds and are superior to other de novo assemblies except for two genomes (overlapping triangles: HuRef (yellow) and the haploid CHM1 (red)).
Chapter 3: De novo assembly of 150 Danish genomes...
Figure 2. Variant calling and genotyping. (a) Candidate alleles were first generated using both the mapping-based (MB) and the assembly-based (AB) variant callers. After permissive filtering candidate alleles and raw sequencing data were given as input to BayesTyper (BT), a probabilistic genotyping method supporting arbitrarily complex variants. *LQ: Low Quality. WGS: Whole Genome Sequencing. (b) The fraction of candidate alleles that are called (i.e. genotyped in at least one individual) decreases with complexity reflecting the removal of assembly errors. Despite this, the AB input still constitutes an important source of novel (i.e. unannotated) structural variants (SV), especially insertions, in the called set. (c) The distribution of SV lengths for insertions and deletions shows that a higher degree of symmetry is observed for the BT called alleles relative to the raw alleles; this symmetry persists for even larger SVs (Extended Figure 2a). Note that the figure includes insertions with uncertain size due to ambiguous bases from inter-scaffold gaps (supplementary figure 6a). (d-e) The folded site frequency spectrum shows that frequency tends to decline with increasing length of the variant allele, whereas the opposite was true for novel SVs that were much more likely to be common in the population relative to novel SNVs.
Figure 3. De novo variation. (a) The mutation rate per generation as a function of paternal age for SNVs (upper panel) and for indels (lower panel). (b) Insert Table shows the number of mutations that can be assigned to parental origin for SNVs and indels. The number of mutations received from the father (blue symbols) are correlated to father’s age and the number of mutations from the mother (red symbols) are correlated to mother’s age. (c) The proportion of mutations that hits CpG sites as a function of paternal age. Insert figure shows the CpG mutation rate for CpG sites as a function of the methylation rate from H1 cells.
Chapter 3: De novo assembly of 150 Danish genomes
Figure 4. HLA haplotype assembly. (a) Dot plot of one complete haplotype against the reference, marked with major structural rearrangements and diversity along the haplotype. (b) Heatmap of binned pairwise linkage disequilibrium measured as $r^2$ for SNPs with MAF > 0.1. Bins are defined so each bin is >20Kb and contain >20 SNPs. (c) Histograms show the distribution of pairwise differences among the 100 new haplotypes and the decay of LD among the 46 thousand variants identified.
Chapter 3: De novo assembly of 150 Danish genomes...
Figure 5. De novo assembly of Y chromosomes. a) A dot plot with segments longer than 1000 bp of the reference (hg38) vs. one individual (844-01) covering the interval 2.7-3.1 and 6-27 Mb on the Y chromosome. To the left in the figure are the genes of the Y chromosome and the Azoospermia factor regions (AZF) and to the right the different type of regions on the Y chromosome and the palindromes are displayed. The legend shows the different colors associated with the regions. Next to it is a table that shows the region the average father-son concordance for all sites in the region and the number of sites that aligned. At the bottom of the legend is a color code that shows the length of the assembled scaffolds that can be seen below the dot plot. b) The histograms show the N50 and length of scaffolds greater than 100bp in length for 62 (6 were of very poor quality and excluded) Y chromosomes of this study. c) A zoom in on one resolved palindrome, we see an example of a gene conversion and a new mutation.
Methods

Cohort selection

The 50 trios (mother-father-child) were selected from the Copenhagen Family Bank. A candidate set of 60 trios was selected randomly from a pool of nearly 1,000 while maintaining the constraint of an average Danish male and female height and blood type distribution. The study protocol was reviewed and approved by The Danish National Committee on Health Research Ethics file no: 1210920, submission numbers 36615 and 38259.

The HumanCoreExome BeadChip v.1.0 was used to genotype the 60 trios (180 individuals) using the HiScan system (Illumina, San Diego, California). Genotypes were called using GenomeStudio software (version 2011.1; Illumina). All subjects had a high call rate (>98%), and the familial relationship and the sex of the subjects were confirmed. SNPs with a low call rate (<98%) or deviation from Hardy-Weinberg equilibrium (p<0.0001) were excluded. SNP genotype data from reference populations was obtained from previously published GWAS studies in Denmark and neighboring populations, i.e. Norway, Sweden and Germany. Standard Principle Component Analysis of the 120 trio parents combined with the GWAS reference dataset was conducted after merging of datasets and LD-based pruning of SNPs in order to assess the homogeneity of the trios and select a set of 50 trios that best represent the Danish population, thus removing trios with one or more members that appear as outliers (see Supplementary figure 8). From the 60 trios seven trios were removed because of admixed ancestry, both shown in the PCAs and further confirmed by telephone interview with the families and asking about their ancestry. One trio was removed due to lack of sufficient blood for sequencing. From the remaining 52 Danish trios, 50 were
chosen maintaining the constraint of an average Danish male and female height and blood type distribution.

**Sequencing and sequence QC**

DNA was extracted from fresh or frozen blood samples of the 150 donors. At least 278 ug was obtained for each individual and used to create approximately 7 libraries, with insert sizes 180-230 bp, 500-550 bp, 750-800 bp for paired-end libraries and 2 kb, 5kb 10kb and 20kb for mate pair libraries. Sequencing was conducted on Illumina HiSeq2000. SOAPfilter_v2.2 was used to preprocess the sequencing data by filtering reads with adapter contaminations, reads having more than 40% low quality bases (Q < 7) or >10% N bases. For mate pair insert size libraries, we filtered out reads with erroneous alignment orientations.

**Mapping**

All reads from the compendium of libraries were mapped to the human reference genome build 38 supplemented with unlocalized contigs and the decoy sequence. All paired-end libraries were mapped using BWA-MEM version 0.7.5a and refined using Stampy version 1.0.23, whereas mate pair libraries were mapped entirely using Stampy. SAMtools version 0.1.19 was used to process the alignment files and to remove duplicate reads.

**Mapping-based variant calling**

The Genome-Analysis-ToolKit (GATK) version 3.2-2 was used for variant calling from BAM files. Duplicate marking, base recalibration and local indel realignment were performed at lane level BAMs before merging BAMs by sample. We used
HaplotypeCaller in the ERC mode to generate the genotype likelihoods for each individual. We combined all variation calls from the 150 individuals and performed joint genotyping. We recalibrated the SNPs and indels separately using the known variant files from GATK bundle 3.2. For SNP recalibration, we used hapmap_3.3, 1000G_omni2.5, 1000G_phase1.snps as the positive training and true datasets and dbSNP_v141 as the known dataset. For indel recalibration, we used Mills_and_1000G_gold_standard.indels as the positive training and true dataset and dbSNP_v141 as the known dataset. The metrics "DP", "FS", "ReadPosRankSum", "MQRankSum" were used in the recalibration process. We decided on the recalibration threshold for both the SNPs and indels to be 99.0.

De novo assembly

All 150 individuals were individually de novo assembled using the three assemblers SOAPdenovo2, SGA and Allpaths-LG. Each of the assemblers have different approaches to the de novo assembly problem (SGA uses string graphs and SOAPdenovo2 and Allpaths-LG uses different implementations of the de Bruijn graph).

Raw reads from all 150 individuals were assembled using Allpaths-LG. Overlapping paired end reads with an insert size of 180nt were added as fragment libraries and all other libraries (>180nt insert size) were added as jumping libraries. Libraries with an insert size up to 800nt were set with an inward read orientation; all others were set to be oriented away from each other in an outward direction. For one sample a 5kb library was discarded as it was error prone and could not be processed by the Allpaths-LG PrepareAllPathsInputs.pl script. Allpaths-LG assemblies were run with default settings, except for ~45 samples that kept failing in a specific module.
(BuildUnipathLinkGraphsLG). These samples were run with the setting
BULG_TRANSITIVE_FILL_IN=False. Tests were performed on different non-failing
individuals in order to assess if disabling the module would affect the assemblies, but no
difference in the assemblies were observed.

The SOAPDenovo2 assembly of the 150 individuals was carried out as we did in
Besenbacher et al.\textsuperscript{2}

For SGA the 180bp paired end reads were initially collapsed using FLASH\textsuperscript{36} to single
end reads and hereafter all libraries were processed using the SGA pipeline. Filter was
run with --kmer-threshold 2, FM-merge with -m 75, overlap with -m 77 and assemble
using -m 77 -d 0.4 -g 0.1 -r 10. Hereafter contigs were scaffolded iteratively beginning
with the smallest library. Scaffolding was performed by mapping with BWA-MEM,
calculating astat and bam2de and finally using the scaffold command in SGA.

\textbf{Assembly-based variant calling}

We applied the LAST aligner\textsuperscript{37} to align the scaffolds to the human reference genome.
Split alignment was performed to allow for the existence of genome rearrangements.
The misalignment probabilities were computed to provide Phred-scaled confidence
measures of the correctness of genome-scale and base-scale alignments. In the final
assembly-vs-assembly alignments, every non-overlapping DNA pieces of the scaffold
were anchored to a unique position in the reference and we only kept alignments with
misalignment probabilities <0.01. Candidate variants were called using the module A in
AsmVar\textsuperscript{22}, which detects and characterizes structural variants from the alignments.
**Variant integration and genotyping**

Variants from mapping-based and assembly-based calling were then integrated, refined and genotyped by BayesTyper (manuscript submitted) using sample k-mer counts and candidate variants from the two call-sets as input. More specifically, the sample input was obtained by counting the number of occurrences of all 55-mers in the cleaned sequencing data for each individual using KMC2 (v2.2.0; 38) with removal of singleton k-mers enabled. The candidate variant input to BayesTyper consisted of permissively filtered GATK and AsmVar variants (including variants containing ambiguous bases) merged after being trimmed to their simplest representation using the combine tool from the BayesTyper package. More specifically, the 99.9 sensitivity tranche of the GATK SNVs and indels was merged with the AsmVar calls from which SNVs and variants that failed to be realigned using AGE were removed. Joint genotyping using the BayesTyper was done in ten batches of 15 individuals (five parent-offspring trios in each) with the complete variant set from all 150 samples provided as input to each run; all arguments were set to their default values. If not stated otherwise, all post-processing was done using tools and scripts that are part of the BayesTyper package.

The genotype calls from the ten batches were combined to create a joint call-set containing the genotypes of all 150 individuals. An allele was classified as called if the allele call probability, calculated using the genotype posterior probability from all samples, was greater than or equal to 0.9. Each alternative allele was further classified as SNV, deletion, insertion, inversion or complex. Inversions were defined as alternative alleles of equal length to the reference allele, that are at least ten nucleotides long and that match the reverse complement of the reference allele with no more than 5% mismatches.
Each alternative allele in the combined call-set was annotated using dbSNP142 (dbSNP) and the structural variants from the 1000G project phase 3 (1kgSV). Both sets of annotated variants were first normalized using bcftools norm\textsuperscript{39}. Mitochondrial mobile insertions without sequence content were removed from the 1kgSV set prior to normalization. Alternative alleles with a 50% or greater, contiguous sequence overlap with an allele in either dbSNP or 1kgSV were considered annotated. All alleles that could not be annotated were classified as novel.

Insertions and deletions were further classified based on repeat content using RepeatMasker. More specifically, the variant sequences (alternative allele sequence for insertions and reference sequence for deletions) were provided as input to RepeatMasker (v4.0.6) using dfam\textsuperscript{40} (v2.0 running HMMER v3.1b2) and annotations extracted; sequences annotated with more than one repeat family were labeled as “MultiRepeats”.

The annotated call-set was further post-processed using sample genotype filters, which were employed to handle errors arising from data properties not completely accounted for by the model such as the sparsity of unique k-mers in highly repetitive regions. In this study, a sample genotype was removed if any allele in the genotype was covered by less than three informative k-mers or if more than 80% of the informative k-mers had zero counts in the sequencing reads. The filter thresholds were determined empirically by comparing the number of filtered genotypes (sensitivity) to three different measures of specificity: fraction of called annotated alleles (the reference allele was defined as...
annotated) as well as the mean and variance of the inbreeding coefficient distribution.

Allele call probabilities were re-estimated using only the unfiltered sample genotypes.

To validate the structural variants in the integrated BayesTyper call-set, insertions and deletions that were called as bi-allelic, that contained no ambiguous bases, had no overlap with any other variants and that were amenable to PCR amplification were selected. For all such variants, a random, heterozygotic sample (with genotype posterior probability >= 0.9) was chosen among the 29 trios picked for validation. Primers were then designed for all variants passing this step and variants for which no primers could be designed discarded. To create a representative validation set, the variants passing the above criteria were first divided into insertions and deletions. These were then further divided into five different length bins (<5, 6-19, 20-49, 50-99, 100<=) and again into rare and common with rare alleles defined as having an estimated allele count lower than or equal to 15 (0.05%). Finally, ten variants were randomly selected from each bin providing a total of 200 variants for validation.

Validation was performed by sequencing five cloned PCR products for each variant. For each clone, the forward and reverse sequencing reads were trimmed and a consensus sequence assembled using SeqTrace\textsuperscript{41} (v0.9.0) with default parameters; reads for which no consensus sequence could be assembled were discarded. Consensus sequences were aligned to both the reference and alternative alleles including flanks using needle\textsuperscript{42} with default parameters. A clone was then assigned to an allele if the alignment contained no gaps; clones that could not be assigned to neither of the alleles were labelled as invalid. Alignments for all invalid clones were subsequently inspected manually and alleles assigned where possible. The PCR products for all variants longer than 50 nts were also
run on a gel and the fragment lengths for each band estimated using GelAnalyzer (v2010a). An allele was considered validated if it was observed in either the cloning and sequencing-based approach or if its expected size matched that of a band on the gel. The validation rate was computed by dividing the number of alternative alleles that validated with the number of variants for which either the reference or the alternative alleles was observed.

Phasing
We preprocessed the data by setting the trio genotypes with mendelian error rate as missing and filtered the variants with less than 95% genotyping rate. We subsequently integrated the family relatedness, linkage disequilibrium and read linkage information to phase the variations with parameters --assemble, --duo hmm and --input-map. The genetic map that we used was lifted over from the b37 version (http://www.shapeit.fr/files/genetic_map_b37.tar.gz).

Imputation
We imputed into the GOYA cohort dataset after lifting over the data to hg38 using the ENSEMBL assembly converter. This dataset contains 5,222 individuals and 514,705 single nucleotide polymorphisms. The imputation was performed using IMPUTE2 with 1KGP Phasel and PhaseIII lifted over, with the DanishPanGenome reference panel and merged panels using the merge option of IMPUTE2. Imputed variants were filtered on the info score generated by IMPUTE2 with a threshold of 0.882. This threshold corresponds to an $R^2$ of 0.8 defined by Aurer et al. (2012) as a quality cutoff. The $R^2$ score presented in Extended Figure 6 was computed by IMPUTE2.
De novo mutation and SNV calling

We used the approach of \(2\) with the following filtering criteria:

- \(GQ \geq 80\) (for the homozygote filter) and \(GQ \geq 250\) (heterozygote filter)
- \(DP \in [20;150]\) (for both the homozygote and heterozygote filter)
- \(AD2 = 4\) (for the homozygote filter)
- Allele balance \(\in [0.3; 1]\)

We also required that the new allele should be seen on both strands.

Parent of origin assignment of de novo mutations

For each variant, \(X\), we use \(o(X)\) to denote the parental origin of the alternative allele. The reads might provide conflicting evidence and to find the most likely parental origin we calculated a likelihood ratio comparing how likely it is that the alternative allele is on the paternal chromosome (\(o(X) = 1\)) to how likely it is that the alternative allele is on the maternal chromosome (\(o(X) = 0\)):

\[
LR_X = \frac{P(o(X) = 1|data)}{P(o(X) = 0|data)}
\]

\[
= \prod_{Y \in \text{variant less than } 30\text{kb from } X \text{ with PoO assigned}} \sum_{\beta \in (0,1)} P(o(Y) = \beta) \cdot P(data|o(X) = 1, o(Y) = \beta) / \sum_{\beta \in (0,1)} P(o(Y) = \beta) \cdot P(data|o(X) = 0, o(Y) = \beta)
\]

If \(LR_X\) is above one it indicates that the alternative allele of variant \(X\) is on the paternal chromosome and if \(LR_X\) is below one it indicates that it is on the maternal chromosome.

The data that is informative about the PoO are the reads that cover both \(X\) and \(Y\):
The probability that a read supports the true phasing is 1 if the read is mapped correctly and \( \frac{1}{2} \) if the read is not mapped correctly. We calculated the conditional probability of the read as:

\[
P(r_{XY} = (i,j)|o(X) = \alpha, o(Y) = \beta) = \begin{cases} 
P(r_{XY \text{ correct}}) + (1 - p(r_{XY \text{ correct}}))/2, & (i = j \land \alpha = \beta) \lor (i \neq j \land \alpha \neq \beta) \\ (1 - p(r_{XY \text{ correct}}))/2, & (i \neq j \land \alpha = \beta) \lor (i = j \land \alpha \neq \beta) \end{cases}
\]

where \( P(r_{XY \text{ correct}}) \) is the probability that \( r_{XY} \) is mapped correctly (estimated from the phred score in the bam file) and the values of \( i \) and \( j \) is either “ref” or “alt” depending on whether the read contains the reference allele or the alternative allele at position \( X \) and \( Y \). For inherited variants where the parental origin could be assigned by just looking at the genotypes of the family members, \( P(o(Y) = 1) \) is calculated using the phred-scaled genotype probabilities of the three family members. If the PoO of variant \( Y \) has been assigned using read information we calculate \( P(o(Y) = 1) \) from the estimated LR:

\[
P(o(Y) = 1) = LR_Y/(LR_Y + 1). \]

The assignment of parental origin is carried out iteratively until no additional variants can be assigned.

**HLA analysis**

Haplotypes of the whole HLA region were constructed using Allpaths-LG scaffolds as the starting point including variants in the fastG version of the scaffolds.
Scaffolds aligning to the HLA region in alignment blocks of at least 50kb were extracted from the assembly graphs. Strand information and median starting points of alignment blocks were used to determine orientation and order of scaffolds in order to concatenate scaffolds into full-length HLA scaffolds. Scaffolds were trimmed to 1mb telomeric of HLA-F and 1kb centromeric of KIFC1 determined in each case by BLAST$^\text{43}$ (blastn) of the gene sequences to the HLA scaffolds. Positions of variant sites from the graph were determined within the trio by exact matching of 40bp upstream of each variant. Variants were then phased by transmission. Consensus sequences were created for each parent-offspring haplotype using global alignment between all pairwise sets of phased variants. Haplotypes were refined by first mapping reads to the four haplotypes within each trio using BWA-MEM version 0.7.5a$^\text{32}$, then calling variants with Platypus, version 0.7.9.1$^\text{44}$ and finally phasing variants that passed quality control by determining the parent-of-origin of alternative alleles (see above). Gaps in the haplotypes were closed using the GapCloser module from SOAPdenovo through five iterations of adding one read library at a time. After gap closing all transmitted haplotypes were submitted to remapping, variant calling and phasing as described above. Variant positions in non-transmitted haplotypes were mapped by pairwise alignment to the transmitted haplotypes. All transmitted haplotypes were aligned to hg38 using LAST aligner$^\text{37}$. The AsmVar pipeline was used to create a candidate set of genotypes from the two haplotypes from each individual. BayesTyper was used to call variants from the candidate set of variants and phasing was restored by using the allele call origin INFO field from AsmVar and removing any variants discordant in respect to phasing and allele call origin. A dot plot was created using the nucmer module from MUMmer 3.0$^\text{45}$

**Y chromosome analysis**
From de novo assemblies scaffolds they were aligned to the reference (GRCh38) using LAST and scaffolds mapping to the Y chromosome were extracted.

For closing the gaps within the scaffolds we used GapCloser. The scaffolds were then repeatmasked.

Concordance between father and sons was estimated by aligning scaffolds using MAFFT and removing repeatmasked regions and regions 50 bp around alignment gaps. AsmVar and BayesTyper was used to identify structural variation. Haplotypes were called with respect to a minimal list of SNPs. Palindrome arms were resolved by first mapping to individual collapsed scaffolds using BWA-mem and filtered and sorted with Sambamba version 0.5.1 followed by read backed phasing using GATK.

The scaffolds that aligned to the Y chromosome in the LAST alignments were extracted. Only scaffolds where the majority of the bases mapped to the Y chromosome with a length > 1000nt were chosen to avoid scaffolds that mapped too ambiguously. The gap in the scaffolds was closed with a module from SOAPdenovo called GapCloser. All duplicated sequences within an individual were found by using LASTZ with the scaffolds as both target and query. The list of duplications was required to have 95% sequence similarity and sequence length > 100. All repeats were grouped together and searched for within all scaffolds from all individuals using BLAST. Repeats that had a sequence identity of 95% and 95% of the repeats length covered were kept.

Validation

We picked a set of 400 variants for experimental validation of 1) de novo calls (15 indels), 2) structural variants (200), HLA variants (66 regions with 202 variants) and Y chromosome (44 variants). In each case a region of 1 kb around the target variants was used to design primers for PCR followed by cloning and sequencing of five clones from
each assay. PCR, cloning and Sanger sequencing of 5 clones from each PCR reaction was done by EuroFins Genomics.
Chapter 3: De novo assembly of 150 Danish genomes ... 132

References


Chapter 3: De novo assembly of 150 Danish genomes . . .


Chapter 3: De novo assembly of 150 Danish genomes...
Chapter 3: De novo assembly of 150 Danish genomes . . .


Chapter 3: De novo assembly of 150 Danish genomes . . .


51 Harris, R. S. *Improved pairwise alignment of genomic DNA*, The Pennsylvania State University, (2007).
Acknowledgments

The study was funded by Innovation Fund Denmark. Support from the Novo Nordisk Foundation is acknowledged.

Author contribution

Designed the study and supervised analyses


Provided samples


Performed sequencing


Analysed data


Provided computational infrastructure

A.S, R.F., A.H., P.L.

Wrote the manuscript

Data availability

The raw data, the full de novo assemblies and the complete variant call set in the form of a phased VCF file is available under controlled access by request. The full variant call set can also be freely used for imputation through The International Haplotype Reference Consortium. The full set of de novo variants is found as Supplemental data set 4. The 100 HLA haplotypes will be incorporated into the next release of the human reference genome. For unrestricted use, we release a fully phased VCF file which is randomly sampled from a probabilistic graph-based data representation that retains most of the LD structure (see Extended Figure 7), but which renders it impossible to extract individual genomes or haplotypes in order to respect donor consent.
Extended data

Figure 1. Comparison between assemblers

Standard assembly statistics like the total size, N percent, maximum, minimum and mean length as well as the N10-N90 statistics were averaged for the 50 trios and were compared between the three assemblers (a) Allpaths-LG, (b) SGA, (c) SOAPdenovo2. Blue and purple bars refer to the scaffold and scaffig measurements, corresponding to the left and right y-axis respectively. Scaffig refers to contigs that are cut out from the scaffolds at Ns. After descending sorting of the scaffold/scaffig based on length, N10-N90 refer to the minimum length that the assembled sequences with the length equal and above that comprise 10% - 90% of total scaffold size.
Chapter 3: De novo assembly of 150 Danish genomes
Figure 2. Variant size spectra. (a) Extended variant size spectrum of all called and novel insertions and deletions. (b) Variant size spectrum showing the contribution of mapping-based and assembly-based callers to the set of called variant insertions and deletions.
Chapter 3: De novo assembly of 150 Danish genomes...
Figure 3. Repeat family classification of variant alleles. Repeat content of insertions and deletions was determined using RepeatMasker; variant alleles assigned to multiple repeat families were labelled as "MultiRepeats". Variants shorter than 17 nts were omitted from the analysis as these are not classified by RepeatMasker. (a) Number of variant alleles (left bars) and fraction of novel alleles (right bars) in each repeat family. (b,c) Distribution of repeat classes across the variant size spectrum; MultiRepeats were omitted from these figures.
Figure 4. Loss-of-function analysis. (a) The number of loss of function variants for each of the 100 parents divided into categories. (b) The number of heterozygous variants.
Figure 5. Size distribution of de novo indels. The size distribution of the identified de novo indels with respect to Hg38.
Chapter 3: De novo assembly of 150 Danish genomes...
Figure 6. Summary of the results of the imputation on clinical data. The GOYA genotype array data set after passing the quality control contains 5,222 individuals and 514,705 SNPs. (a) The number of additional variants imputed with the Genome Denmark reference panel (GDK) every 2000kb. These variants are imputed with the merged panel of the 1000 genomes reference panels (1KG) and the GDK but are not imputed with the 1KG alone. In green are the regions where the percentage of unknown bases on the reference genome is higher than 15% (>15% 'N'). (b) Median quality (r²) of the imputation per minor allele frequency, with the three reference panels. (c) Number of variants imputed and filtered, (info score above 0.882) selected variants and percentage of insertion or deletions (indels) for the Genome Denmark reference panel (GDK), the 1000 genomes reference panel (1KG), the two panels merges (GDK + 1KG) and the additional variants imputed by GDK, see (a) for explanations (GDK + 1KG due to GDK).
Chapter 3: De novo assembly of 150 Danish genomes ...
Figure 7. Comparison of the original and publicly available Danish pan genome. (a) Comparison of the correlation of minor allele frequency of the 7,500 first variants from chromosome 9 between the 150 real individuals and 150 individuals sampled from the genome-graph. (b) Comparison of the R-squared measure of linkage disequilibrium between pairs of variants in the real individuals versus the sampled for all pairs with an R-squared value above 0.2 within a sliding window of eight variants. Variant-pairs are represented as a dot which is colored and size-scaled by their proximity stratum (number of in-between variants) ranging from proximal (red/small) to distal (green/large). For each proximity stratum, a linear model is fitted and shown as a line of corresponding color, demonstrating a slight deterioration of the correlation as a function of variant pair proximity. The dotted line show the linear model fitted for all pairs regardless of distance.
Supplementary information for “De novo assembly of 150 Danish genomes reveals rich structural complexity”

Contents

Raw data ............................................................................................................................................................ 2
Sequence coverage of libraries per individual .................................................................................................. 2
De novo assembly ............................................................................................................................................... 4
Assembly statistics for all 150 individuals ...................................................................................................... 4
Comparison of assemblers ............................................................................................................................... 4
HHV6b integration ............................................................................................................................................ 4
Variant calling and genotyping ....................................................................................................................... 6
Validation of called variants ............................................................................................................................. 6
Ambiguous nucleotide distribution ................................................................................................................ 7
De novo variants ............................................................................................................................................... 8
Detailed information about de novo variants .................................................................................................. 8
HLA .................................................................................................................................................................... 8
Cohort selection ............................................................................................................................................... 8
Validation of HLA and Y chromosome variants ............................................................................................ 9
Raw data

Sequence coverage of libraries per individual

See excel file: Suppl.table.data.production.xlsx

Supplementary figure 1: Mean and the standard deviation of the raw base coverage of the sequencing data by different library type for each individual. Dots represent individual. Orange bar indicates mean coverage of the 50 trios. The depth is estimated by the total number of sequencing bases divided by the size of GRCh38 (GCA_000001405.15_GRCh38_full_analysis_set.fa, N = 3209457928)
Supplementary figure 2: Mean and the standard deviation of the clean base coverage of the sequencing data by different library type for each individual. Dots represent individual. Orange bar indicates mean coverage of the 50 trios. The depth is estimated by the total number of clean sequencing bases divided by the size of GRCh38 (GCA_000001405.15_GRCh38_full_analysis_set.fa, N = 3209457928). Cleaning was performed using SOAPfilter_v2.2 by filtering reads with adapter contaminations, reads having more than 40% low quality bases (Q < = 7) or >10% N bases. For mate pair insert size libraries, we filtered out reads with erroneous alignment orientations.
**De novo assembly**

**Assembly statistics for all 150 individuals**

See excel file: Suppl.table.AllpathsLG_statistics_150_genomes.xlsx

**Comparison of assemblers**

See excel file: Suppl.table.assembly.comparison.xlsx

**HHV6b integration**

We identified two individuals with a high number of reads mapping to the HHV6b genome (Supplementary figure 3). Blasting of the HHV6b genome against these assemblies revealed that a single scaffold from each individual aligned with the HHV6b reference genome and showed complete assembly of the viral genome (Supplementary figure 4).

**Supplementary figure 3**: Number of reads mapped to viral genomes.
Supplementary figure 4: Dot plot of two scaffolds vs HHV6b reference genome.
Variant calling and genotyping

Validation of called variants

Supplementary figure 5: BayesTyper validation rates. (a) Validation rate for common and rare (allele count <= 15) structural variants as a function of allele length. (b) The number of valid variants used to calculate the rates shown in (a). (c) Validation rate for structural variants with or without repeat content as a function of allele length. (d) The number of valid variants used to calculate the rates shown in (c).
Ambiguous nucleotide distribution

Supplementary figure 6: Ambiguous nucleotide distribution. (a) Fraction of candidate and called insertions with ambiguous nucleotides in the alternative allele. (b) Fraction of candidate and called complex alleles with ambiguous nucleotides in the alternative allele. Only insertions and complex alleles had ambiguous nucleotides.
De novo variants

Detailed information about de novo variants

See excel file: Supl.table.denovo_variants.xlsx

HLA

Supplementary figure 7: Number of scaffolds aligning to the MHC. Frequency distribution of the number of scaffolds aligning to the MHC region (pgf) or any of the alternative reference haplotypes (apd, cox, dbb, mann, mcf, qbl, ssto) in alignment blocks of at least 50 kb.

Cohort selection
**Supplementary figure 8:** PCA plot of the two first principle components of the analysis conducted using *SNPRelate*. Black circles refer to Danish, red circles to German, blue circles to Norwegians and green circles to Swedish reference samples, while filled dots are the 120 parents of the 60 trios of this study.

**Validation of HLA and Y chromosome variants**

Supplemental Table 1, see excel file: Suppl.table.HLA&Y.validation.xlsx