GraphTyper: A pangenome method for identifying sequence variants at a population scale

Hannes Pétur Eggertsson

Dissertation submitted in partial fulfillment of a
Philosophiae Doctor degree in Computer Science

Advisor
Bjarni Vilhjálmur Halldórsson
Páll Melsted

PhD Committee
Bjarni Vilhjálmur Halldórsson
Páll Melsted
Daníel Fannar Guðbjartsson

Opponents
Knut Reinert
Zamin Iqbal

Faculty of Industrial Engineering, Mechanical Engineering and
Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, June 2019

GraphTyper: A pangenome method for identifying sequence variants at a population scale

Dissertation submitted in partial fulfillment of a *Philosophiae Doctor* degree in Computer Science

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Sæmundargata 2
101, Reykjavik
Iceland

Telephone: 525-4000

Bibliographic information:
Hannes Pétur Eggertsson, 2019, *GraphTyper: A pangenome method for identifying sequence variants at a population scale*, PhD dissertation, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, 154 pp.

ISBN 978-9935-9473-2-1

Printing: Háskólaprent
Reykjavik, Iceland, June 2019

Abstract

A fundamental requisite for genetic studies is an accurate determination of sequence variation. While human genome sequence diversity is increasingly well characterized, there is a need for efficient ways to utilize this knowledge in sequence analysis. Here we present GraphTyper, a publicly available novel algorithm and software for genotyping sequence variants. GraphTyper can discover small variants directly from sequence data and is able to encode and accurately genotype all sizes of variants. GraphTyper realigns short-read sequence data to a pangenome, a variation-aware graph structure that encodes sequence variation within a population by representing possible haplotypes as graph paths. Our results show that GraphTyper is fast, highly scalable, and provides sensitive and accurate genotype calls. GraphTyper genotyped 60 million sequence variants in whole-genomes of 49,962 Icelanders, including half a million structural variants, which is to our knowledge the largest such sequence analysis to date. We compare GraphTyper to previous methods and show that it is a valuable tool in characterizing sequence variation in both small and population-scale sequencing studies.

Útdráttur

Nauðsynleg krafa fyrir erfðafræðirannsóknir eru áreiðanlegar aðferðir til að finna arfgerðir einstaklinga með raðgreiningargögnum. Miklum upplýsingum um erfðabreytileika hefur nú þegar verið safnað, sem kallar á nýjar aðferðir til að nýta þessar upplýsingar. Hér kynnum við GraphTyper, frjáls og frír hugbúnaður sem finnur erfðabreytileika í raðgreiningargögnum. GraphTyper býr til stærðfræðilegt net sem inniheldur þekkta erfðabreytileika, þar sem að hver leið í netinu skilgreinir mögulegar erfðaraðir. GraphTyper ber saman raðgreiningargögn við netið til að bera kennsl á arfgerð einstaklings. Niðurstöður okkar sýna að GraphTyper skalast vel með fjölda einstaklinga og veitir bæði næm og nákvæm köll á arfgerðum í samanburði við aðra samskonar hugbúnaða. GraphTyper kallaði 60 milljón breytileika í 49,962 Íslendingum, þar á meðal hálfa milljón breytileika sem eru stærri en 50 basapör, og er það stærsta slík köllun sinnar tegundar. Við trúum að GraphTyper sé framför fyrir svið erfðafræðirannsókna og muni nýtast í að tengja erfðafræðiupplýsingar við sjúkdóma og aðrar svipgerðir.

Table of Contents

List of Figures

List of Tables

List of Original Papers

Paper I: Graphtyper enables population-scale genotyping using pangenome graphs

Paper II: GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs

Abbreviations

bp Base-pair
BAM Binary Alignment/Map
BWA Burrows-Wheeler Aligner
DAG Directed Acyclic Graph
DNA DeoxyriboNucleic Acid
DNM *de novo* Mutation
GATK Genome Analysis ToolKit
GWAS Genome-Wide Association Study
HLA Human Leukocyte Antigen
MS Multiple Sclerosis
RNA RiboNucleic Acid
SAM Sequence Alignment/Map
SNP Single-Nucleotide Polymorphism
SNV Single-Nucleotide Variant
SV Structural Variant

Acknowledgments

I would like to start by thanking my family for all their support. Especially Bryndís, who has loved and supported me in all the ups and downs during my PhD. I also would like to thank my 7 months old daughter, Sigurdís. Most of this thesis has been written while I was staying home with you. Without your calmness and patience it would not have been possible to write it.

I thank deCODE genetics / Amgen Inc. for the financial support during my PhD. I am grateful to my colleagues from deCODE and elsewhere for their help and contributions.

Finally, I also wish to thank all research participants who provided biological samples to deCODE.

1 Introduction

1.1 Population genetics

Human genomes have now been routinely sequenced for more than a decade. With recent advancements in sequencing technologies, the cost of sequencing genomes has decreased, which has resulted in a substantial growth of the number of individuals with whole-genome sequence data available. Analysis of the generated sequences has improved our understanding of the genotypes, the variation in the genetic material, and how they may affect the phenotype, the observable traits (Figure 1.1).

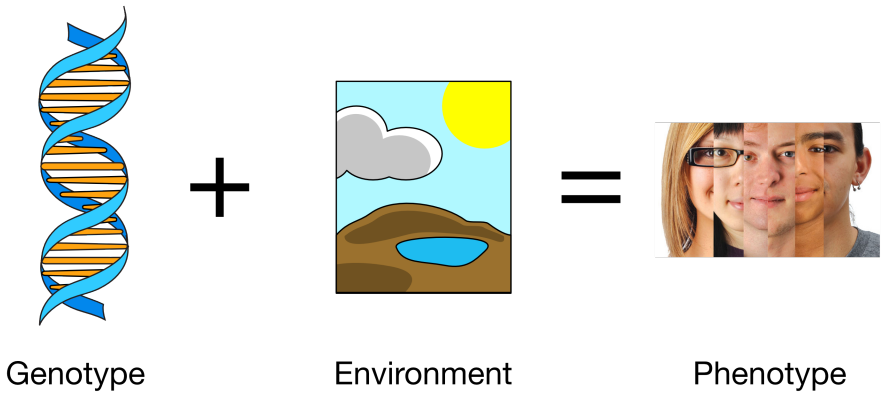


Figure 1.1. The underlying genetic material of a person (genotype) and the environment affect observable traits (phenotype) of that person.

The genotypes are called at sequence variant sites in a process called variant calling. Genome-wide association studies (GWAS) investigate how sequence variants in a population associate with a trait. Those studies may give insights into the impact of a sequence variant on a trait. GWAS of disease phenotypes are of particular interest.

Researchers have built various pipelines for analyzing sequence data. The pipelines need to be composed of efficient and scalable methods, in order to handle the vast amount of sequencing data available. A high level overview of such a pipeline is shown in Figure 1.2, where the sequence data from individuals are collected, processed and

their variation studied.

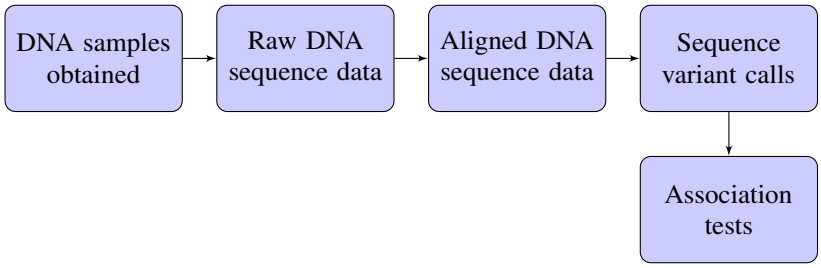


Figure 1.2. Simplified pipeline for association testing genetic variants.

In the first step of the pipeline, a DNA sample is obtained from each of the individuals. Second, the DNA sample is whole-genome sequenced with Illumina short-read sequencing machines or other sequencing methods. The raw sequences are called reads, a short substring of the samples' DNA sequence that may contain errors. Third, the reads are aligned against a human reference genome. This alignment step is required as the raw sequences contain no information where in the genome the sequence originated. For some reads, the read aligner may not be able to align a read and in which case it reports it as unaligned. The read aligner may also align a read to an incorrect location because the reads are short, contain errors, or contain variation which is not present in the reference. Fourth, the sequences variants are discovered and genotypes are called based on the read sequences. Finally, the sequence variant genotypes are tested for association against phenotypes of the individuals participating in the study. Such association tests are routinely applied to reveal insights how the human genome impacts diseases and other traits.

1.2 Sequence variant calling

The variant calling process in the pipeline above is a common task in analysis of sequence data. The most common type of variants are small sequence variants, such as single nucleotide polymorphisms (SNPs), where a single nucleotide is changed, and indels, where a short sequence is either deleted or inserted. In general, we define small sequence variants as those that modify the sequence by less than 50 bp. Larger sequence variants are referred to as structural variants (SVs), which may modify hundreds of thousands of nucleotides.

Population-scale variant calling refers to the process of calling variants for many samples from the same population, either one at a time or jointly. Joint-calling is typically favored as it generates a set of genotype calls which are comparable across the samples in the population and can be used directly in GWAS. Variant calling can be split into discovery and genotyping. In the discovery step, potential variation sites are

detected and in the genotyping step, genotypes are called at those sites.

Small variants are typically called based on the discordances between the read alignments and a reference genome (Figure 1.3). This approach is used in many widely used variant callers for short-read data, such as Genome Analysis ToolKit (GATK) Unified Genotyper (UG) (McKenna et al., 2010) and samtools (Li et al., 2009), and is very effective for calling SNPs, but often misses indels and other complex variants due to incorrect read alignments for those variants. To improve upon this, variant callers with a local assembly and/or haplotype awareness were implemented, such as Genome Analysis ToolKit (GATK) Haplotyper Caller (HC) (McKenna et al., 2010), Platypus (Rimmer et al., 2014), and FreeBayes (Garrison and Marth, 2012). These software are widely used for calling small variants.

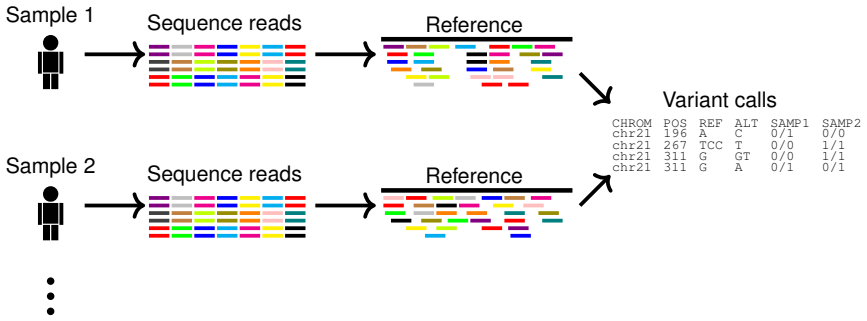


Figure 1.3. Sequence variant calling. Sequence reads are aligned to a reference genome and variants are commonly called from discordances between reads and reference.

SVs are larger variants and due to their sizes, they often cannot be directly discovered from short-read data. Instead, they are discovered from read *de novo* assemblies, split-read alignments, read alignment coverage, read-pair insert sizes, or other indirect inferences. Various methods exist for calling SVs from short-read data, including Manta (Chen et al., 2016), Delly (Rausch et al., 2012), PopIns (Kehr et al., 2016), Lumpy (Layer et al., 2014), SVTyper (Chiang et al., 2015), and more. Since SV discovery relies on indirect inference, they typically have both a higher false positive and a higher false negative rate compared to methods aimed at smaller variants.

Most previous sequence variant calling methods use reads aligned to the reference genome, which can cause bias toward the reference genome and misalignments around indels (DePristo et al., 2011; Shao et al., 2013). Approaches that find sequence variants in reference-free assemblies have been developed to avoid these limitations (Iqbal et al., 2012), however, they are computationally more expensive and have less sensitivity (Rimmer et al., 2014). Moreover, it is challenging to compare their results to available genome annotation data since they depend on data structures with complex coordinate system.

Rather than utilizing reference-free methods, another alternative is to encode multiple sequences within a reference instead of only one. The shared parts of the sequence

do not need to be defined repeatedly, but rather the data structure should only need to be aware of the variations between the sequences. For this purpose, so called variation-aware data structures (Garrison et al., 2018; Rakocevic et al., 2019; Sibbesen et al., 2018; Biederstedt et al., 2018) have been proposed to alleviate some of the limitations of previous methods (Computational Pan-Genomics Consortium, 2016). They incorporate prior information about variation and encode it using a mathematical graph data structure. Each node in the graph contains a sequence and haplotype sequences can be generated by traversing the graph and concatenating the node's sequences. A lot of effort has recently been put into creating read aligners (Garrison et al., 2018) and variant callers (Sibbesen et al., 2018; Rakocevic et al., 2019) that utilize variation-aware data structures. These methods aim to pave a way for the creation of a high quality human pangenome, a representation of all human genomic content.

1.3 GraphTyper

Here in this thesis I present GraphTyper, a pangenome graph-based method for genotyping sequence variants jointly at a population-scale. GraphTyper is a free and open-source software written in C++11, and it is available online at:
`http://github.com/DecodeGenetics/graphtyper`

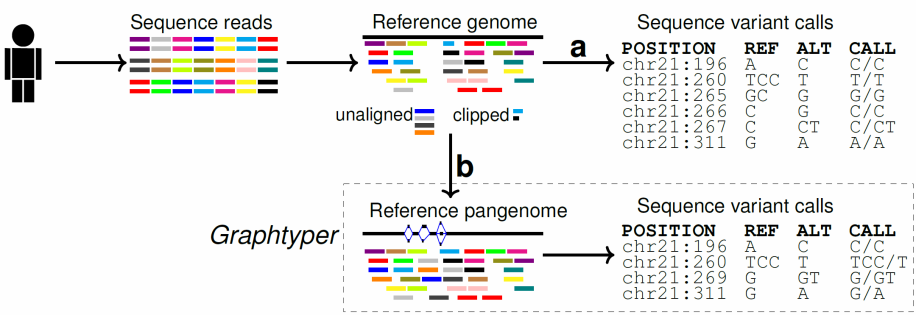


Figure 1.4. Comparison of sequence variant callers. (a) Variant calls are made based on discordances between reads and the reference. Unmapped and clipped sequences are typically ignored. (b) GraphTyper's variant calling. Sequence reads, including unaligned and clipped reads, are realigned to a pangenome graph and the genotype calls are determined based on the realignments.

Briefly, our method is motivated by the fact that a wide variety of sequence diversity has already been well characterized, but previous methods generally do not take known variants into account when calling variants (Li et al., 2009; McKenna et al., 2010; Rimmer et al., 2014). GraphTyper encodes prior variation information into a directed acyclic graph (DAG), where each node corresponds to a sequence and paths through the graph represent possible haplotypes. Sequence reads are realigned to the DAG and

variants are genotyped based on the graph alignments, in the same step (Figure 1.4). Even reads that were previously unaligned or clipped with an aligned mate can be realigned to the graph, whereas these reads are ignored by most previous alignment based methods. By clipped reads we refer to reads that are only partially aligned. A wide variety of sequence variants can be represented with a DAG, including both small sequence variants (SNPs and indels) and SVs.

By incorporating the prior variant information into the method we save time by not re-discovering known variants for each individual, but by putting more effort into discovering previously unobserved variants and simultaneously improving the genotype calling of the variants. Furthermore, we show that the graph realignment fixes many artifacts that arise due to read misalignments to the reference.

The sizes of the sequencing datasets are growing since the price of sequencing has dropped dramatically in the last decade. It was therefore our design decision that our method scaled well with the number of samples, such that it can genotype tens of thousands of genomes jointly. The compute time requirement of genotyping using GraphTyper scales well with the number of samples compared to previous methods, which makes it a suitable option for genotyping large cohorts.

GraphTyper can be applied to many different datasets. Most current methods are designed for specific variant types, while our method can accurately genotype nearly any type of sequence variant. Rather than running several different software to genotype sequence variants of a population, it is possible to run only GraphTyper and get accurate genotype calls across the sequence variation spectrum. With the release of version 2, our method is also not restricted to human genomes and can genotype any diploid organism that has a reference genome. We believe our method is thus valuable for population sequence analysis and can assist in understanding how the genetic variants affects disease and other phenotypes.

1.4 Publications

This thesis is composed in a cumulative style. The major contributions are presented as peer-reviewed journal paper and in a pending paper submission, and can be found in the Appendix. Summaries of the papers are provided in Chapter 2. Publications to which I only contributed to a lesser extent or are unrelated to the thesis, are deliberately excluded. The following publications are referenced throughout the thesis.

1.4.1 Appendend papers

Paper I:
Hannes P. Eggertsson, Hakon Jonsson, Snaedis Kristmundsdottir, Eirikur Hjartarson, Birte Kehr, Gisli Masson, Florian Zink, Kristjan E. Hjorleifsson, Aslaug Jonasdottir, Adalbjorg Jonasdottir, Ingileif Jonsdottir, Daniel F. Gudbjartsson, Pall Melsted, Kari Stefansson and Bjarni V. Halldorsson. "Graphtyper enables population-scale genotyping using pangenome graphs", in *Nature Genetics*, volume 49, pages 1654–1660 (2017).

Paper II:
Hannes P. Eggertsson, Snaedis Kristmundsdottir, Doruk Beyter, Hakon Jonsson, Astros Skuladottir, Marteinn T. Hardarson, Daniel F. Gudbjartsson, Pall Melsted, Bjarni V. Halldorsson, Kari Stefansson. "GraphTyper2 enables population-scale genotyping across the variation spectrum using pangenome graphs" (manuscript under consideration).

1.4.2 Contributions to other papers

Paper A:
Hákon Jónsson, Patrick Sulem, Birte Kehr, Snaedis Kristmundsdottir, Florian Zink, Eirikur Hjartarson, Marteinn T. Hardarson, Kristjan E. Hjorleifsson, **Hannes P. Eggertsson**, Sigurjon Axel Gudjonsson, Lucas D. Ward, Gudny A. Arnadottir, Einar A. Helgason, Hannes Helgason, Arnaldur Gylfason, Adalbjorg Jonasdottir, Aslaug Jonasdottir, Thorunn Rafnar, Mike Frigge, Simon N. Stacey, Olafur Th. Magnusson, Unnur Thorsteinsdottir, Gisli Masson, Augustine Kong, Bjarni V. Halldorsson, Agnar Helgason, Daniel F. Gudbjartsson and Kari Stefansson. "Parental influence on human germline *de novo* mutations in 1,548 trios from Iceland", in *Nature*, volume 549, pages 519-522 (2017).

Abstract: The characterization of mutational processes that generate sequence diversity in the human genome is of paramount importance both to medical genetics and to evolutionary studies. To understand how the age and sex of transmitting parents affect de novo mutations, here we sequence 1,548 Icelanders, their parents, and, for a subset of 225, at least one child, to 35× genome-wide coverage. We find 108,778 de novo mutations, both single nucleotide polymorphisms and indels, and determine the parent of origin of 42,961. The number of de novo mutations from mothers increases by 0.37 per year of age (95% CI 0.32-0.43), a quarter of the 1.51 per year from fathers (95% CI 1.45-1.57). The number of clustered mutations increases faster with the mother's age than with the father's, and the genomic span of maternal de novo mutation clusters is greater than that of paternal ones. The types of de novo mutation from mothers change substantially with age, with a 0.26% (95% CI 0.19-0.33%) decrease in cytosine-phosphate-guanine to thymine-phosphate-guanine (CpG>TpG) de novo mutations and a 0.33% (95% CI 0.28-0.38%) increase in C>G de novo mutations per year, respectively. Remarkably, these age-related changes are not distributed uniformly across the genome. A striking

example is a 20 megabase region on chromosome 8p, with a maternal C>G mutation rate that is up to 50-fold greater than the rest of the genome. The age-related accumulation of maternal non-crossover gene conversions also mostly occurs within these regions. Increased sequence diversity and linkage disequilibrium of C>G variants within regions affected by excess maternal mutations indicate that the underlying mutational process has persisted in humans for thousands of years. Moreover, the regional excess of C>G variation in humans is largely shared by chimpanzees, less by gorillas, and is almost absent from orangutans. This demonstrates that sequence diversity in humans results from evolving interactions between age, sex, mutation type, and genomic location.

In this paper I participated in creating methods for analyzing the data. In particular I assisted in developing the bamShrink method (`https://github.com/DecodeGenetics/bamShrink`), which was run prior to variant calling with Genome Analysis ToolKit (GATK).

Paper B:
Hákon Jónsson, Patrick Sulem, Gudny A. Arnadottir, Gunnar Pálsson, **Hannes P. Eggertsson**, Snaedis Kristmundsdottir, Florian Zink, Birte Kehr, Kristjan E. Hjorleifsson, Brynjar Ö. Jensson, Ingileif Jonsdottir, Sigurdur Einar Marelsson, Sigurjon Axel Gudjonsson, Arnaldur Gylfason, Adalbjorg Jonasdottir, Aslaug Jonasdottir, Simon N. Stacey, Olafur Th. Magnusson, Unnur Thorsteinsdottir, Gisli Masson, Augustine Kong, Bjarni V. Halldorsson, Agnar Helgason, Daniel F. Gudbjartsson and Kari Stefansson. "Multiple transmissions of de novo mutations in families." in *Nature Genetics*, volume 50, pages 1674-1680 (2018).

Abstract: De novo mutations (DNMs) cause a large proportion of severe rare diseases of childhood. DNMs that occur early may result in mosaicism of both somatic and germ cells. Such early mutations can cause recurrence of disease. We scanned 1,007 sibling pairs from 251 families and identified 878 DNMs shared by siblings (ssDNMs) at 448 genomic sites. We estimated DNM recurrence probability based on parental mosaicism, sharing of DNMs among siblings, parent-of-origin, mutation type and genomic position. We detected 57.2% of ssDNMs in the parental blood. The recurrence probability of a DNM decreases by 2.27% per year for paternal DNMs and 1.78% per year for maternal DNMs. Maternal ssDNMs are more likely to be T>C mutations than paternal ssDNMs, and less likely to be C>T mutations. Depending on the properties of the DNM, the recurrence probability ranges from 0.011% to 28.5%. We have launched an online calculator to allow estimation of DNM recurrence probability for research purposes.

In this paper, GraphTyper was used to genotype DNM candidates on samples which had been resequenced using targeted sequencing. GraphTyper enabled an accurate estimate of the variant allele frequency of the DNMs.

Paper C:
Lara Kular, Yun Liu, Sabrina Ruhrmann, Galina Zheleznyakova, Francesco Marabita, David Gomez-Cabrero, Tojo James, Ewoud Ewing, Magdalena Lindén, Bartosz Górnikiewicz, Shahin Aeinehband, Pernilla Stridh, Jenny Link, Till F. M. Andlauer, Christiane Gasperi, Heinz Wiendl, Frauke Zipp, Ralf Gold, Björn Tackenberg, Frank Weber, Bernhard Hemmer, Konstantin Strauch, Stefanie Heilmann-Heimbach, Rajesh Rawal, Ulf Schminke, Carsten O. Schmidt, Tim Kacprowski, Andre Franke, Matthias Laudes, Alexander T. Dilthey, Elisabeth G. Celius, Helle B. Søndergaard, Jesper Tegnér, Hanne F. Harbo, Annette B. Oturai, Sigurgeir Olafsson, **Hannes P. Eggertsson**, Bjarni V. Halldorsson, Haukur Hjaltason, Elias Olafsson, Ingileif Jonsdottir, Kari Stefansson, Tomas Olsson, Fredrik Piehl, Tomas J. Ekström, Ingrid Kockum, Andrew P. Feinberg and Maja Jagodic. "DNA methylation as a mediator of HLA-DRB1*15:01 and a protective variant in multiple sclerosis" in *Nature Communications*, volume 9, article number: 2397 (2018).

Abstract: The human leukocyte antigen (HLA) haplotype DRB1*15:01 is the major risk factor for multiple sclerosis (MS). Here, we find that DRB1*15:01 is hypomethylated and predominantly expressed in monocytes among carriers of DRB1*15:01. A differentially methylated region (DMR) encompassing HLA-DRB1 exon 2 is particularly affected and displays methylation-sensitive regulatory properties in vitro. Causal inference and Mendelian randomization provide evidence that HLA variants mediate risk for MS via changes in the HLA-DRB1 DMR that modify HLA-DRB1 expression. Meta-analysis of 14,259 cases and 171,347 controls confirms that these variants confer risk from DRB1*15:01 and also identifies a protective variant (rs9267649, $p < 3.32 \times 10^{-8}$, odds ratio = 0.86) after conditioning for all MS-associated variants in the region. rs9267649 is associated with increased DNA methylation at the HLA-DRB1 DMR and reduced expression of HLA-DRB1, suggesting a modulation of the DRB1*15:01 effect. Our integrative approach provides insights into the molecular mechanisms of MS susceptibility and suggests putative therapeutic strategies targeting a methylation-mediated regulation of the major risk gene.

*In this paper GraphTyper's HLA-DRB1 genotyping results were analyzed. One of the main results of the paper is that HLA-DRB1*15:01 associates with to multiple sclerosis, partially based on the HLA genotypes of 28,075 sequenced Icelanders.*

Paper D:
Bjarni V. Halldorsson, Gunnar Palsson, Olafur A. Stefansson, Hakon Jonsson, Marteinn T. Hardarson, **Hannes P. Eggertsson**, Bjarni Gunnarsson, Asmundur Oddsson, Gisli H. Halldorsson, Florian Zink, Sigurjon A. Gudjonsson, Michael L. Frigge, Gudmar Thorleifsson, Asgeir Sigurdsson, Simon N. Stacey, Patrick Sulem, Gisli Masson, Agnar Helgason, Daniel F. Gudbjartsson, Unnur Thorsteinsdottir and Kari Stefansson. "Characterizing mutagenic effects of recombination through a sequence-level genetic map" in *Science*, volume 363, issue 6425 (2019).

Abstract: Genetic diversity arises from recombination and de novo mutation (DNM).

Using a combination of microarray genotype and whole-genome sequence data on parent-child pairs, we identified 4,531,535 crossover recombinations and 200,435 DNMs. The resulting genetic map has a resolution of 682 base pairs. Crossovers exhibit a mutagenic effect, with overrepresentation of DNMs within 1 kilobase of crossovers in males and females. In females, a higher mutation rate is observed up to 40 kilobases from crossovers, particularly for complex crossovers, which increase with maternal age. We identified 35 loci associated with the recombination rate or the location of crossovers, demonstrating extensive genetic control of meiotic recombination, and our results highlight genes linked to the formation of the synaptonemal complex as determinants of crossovers.

In this paper, the SNP and indel GraphTyper genotyping results were used that were described in paper I.

2 Summary of Publications

2.1 Paper I

The first paper describes the initial version of GraphTyper, an open-source variant caller. The initial version supported discovering and genotyping single-nucleotide polymorphisms (SNPs) and indels (small insertions and deletions), the most common types of genetic variants. The paper shows that GraphTyper is sensitive and scales better with the number of samples, compared to previous methods. In our largest genotyping run described in the paper, we genotyped 28,075 whole-genome Icelanders using less than 100 CPU hours per genome.

My contributions to the paper include implementing the GraphTyper software and writing the initial version of the paper. Furthermore, I participated in designing the algorithms and data structures used in the paper, running the experiments and analyzing the results.

2.2 Paper II

The second paper describes the second version of GraphTyper, which features several improvements to GraphTyper. Most notably, in GraphTyper2 we have added the possibility of genotyping structural variants (SVs). GraphTyper relies on external software to discover SVs but can encode them into its graph structure and genotype them accurately. The update enables GraphTyper to genotype across the sequence variation spectrum. The paper shows that GraphTyper is sensitive and highly accurate compared to previous SV genotyping methods. We demonstrated the effectiveness of our method by genotyping SNPs, indels and SVs simultaneously in 49,962 Icelanders.

My contributions to the paper include implementing the updates to GraphTyper and writing the initial version of the paper. Similarly to paper I, I also participated in designing the algorithms and data structures used in the paper, running the experiments and analyzing the results.

3 Variant calling using pangenome graphs

In this chapter, we further define the motivation of using pangenome graphs and describe the data structure and algorithms in GraphTyper.

3.1 Background

Genetic material plays a fundamental role in the composition of living organisms. The hereditary material is a molecule called deoxyribonucleic acid (DNA) is in the nucleus of eukaryotic cells. It is composed of two long chains that form a double-helix structure (Watson and Crick, 1953) (Figure 3.1) that carries the genetic instructions for growth, development, functioning, and reproduction of its organism. These instructions are encoded as long sequences of nucleotides that form chromosomes.

The DNA nucleotides are composed of one of four different chemical bases: cytosine (C), guanine (G), adenine (A) or thymine (T). The bases are interconnected between strands with hydrogen bonds, forming a base pair (bp).

The central dogma of molecular biology describes how genetic information flows within a biological system (Crick, 1970). It describes how ribonucleic acid (RNA) strands are created using DNA strands as a template in a process called transcription. In another process, translation, these RNA strands specify the sequence of amino acids within functional proteins.

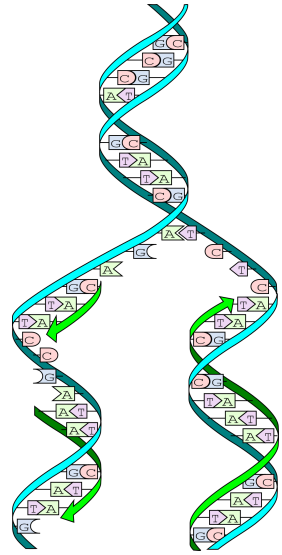


Figure 3.1. Double helix structure of DNA.

A landmark for the field of genetics was achieved when the first drafts of the human reference genome were created (International Human Genome Sequencing Consortium, 2001; Venter et al., 2001). The human reference genome is a single consensus sequence of the human genome and has undergone multiple improvements over the years. In the most recent version of the human reference genome (GRCh38) there are 3 gigabase-

pairs (Gbp) of sequence with less than a thousand gaps. A reference sequence is a consensus of human genomes, where a single sequence is derived from assemblies of a single or multiple genomes.

Sequence variation refers to any modification of the genome sequence. Such variations may have direct implications for the transcription of RNA and the translation into proteins; the structure of the protein could change or they may cause the protein not to be translated at all. Detecting these sequence variations and understanding their effect is in ongoing research.

In the current human reference sequence (GRCh38), there are several alternative loci that represent regions that are highly divergent compared to their counterpart on the reference sequence. Incorporating these loci enables aligners to be aware of the variation they represent. While these alternative loci are useful, representing any divergent sequence as a new alternative loci is extremely verbose as characterization of sequence diversity of the human genome continues. While the alternative sequences may be highly divergent from the reference, a shared sequence between them is repeatedly defined. Moreover, it is difficult to define clearly which sequences should be represented as an alternative locus and which should not. For example, the most polymorphic gene in the human genome, *HLA-B*, has more than 5 thousand known version of its sequence (Robinson et al., 2015). Optimally, all of those sequences should be considered in sequence analysis.

We believe there is a clear need to extend the linear reference genome such that it could represent multiple sequences compactly.

3.2 Pangenome graphs

Pangenome graphs (Computational Pan-Genomics Consortium, 2016) (also called genome graphs (Rakocevic et al., 2019; Biederstedt et al., 2018), population reference graphs (Dilthey et al., 2015), variation graphs (Garrison et al., 2018), and more) extend the linear reference using a mathematical graph data structure. Pangenomes incorporate prior information about variation (Figure 3.2a), allowing read aligners to be variation-aware and thus distinguish better between sequencing errors in reads and true sequence variation. A haplotype sequence is the sequence of a single chromosome. Variation is encoded into the reference such that each haplotype sequence is represented by a path in the graph. This way, the graph requires less space than storing every haplotype as a separate sequence.

The human reference sequence represents a consensus of genomes and is partially derived from only a single haploid genome. Sequence analysis that utilize the reference sequence can be biased towards the reference, as sequence reads that overlap variation are more likely to map incorrectly. This is particularly problematic for non-European

populations that are highly different than the reference sequence (Wang et al., 2008; Seo et al., 2016). A read alignment to a variation-aware data structure can alleviate the problem and reduce the bias towards the reference (Garrison et al., 2018; Rakocevic et al., 2019).

A variation-aware alignment may also refine alignments in proximity of variation. We demonstrated an example of this in Paper I (Figure 3.2b) when we compared Mendelian error rate of variant calling methods on a genomic region that contains a deletion. Due to the deletion, the read alignments that overlapped the deletion allele were commonly misaligned, which resulted in many artifact variants being called near the deletion (Figure 3.3). Every variant caller we assessed called these artifacts except GraphTyper, since it realigned the reads and realized that their alignment was incorrect. Furthermore, GraphTyper was the only method that had no Mendelian errors in this region. We therefore argue that a variation-aware realignment is necessary to be able to accurately genotype variants in complex regions such as this one.

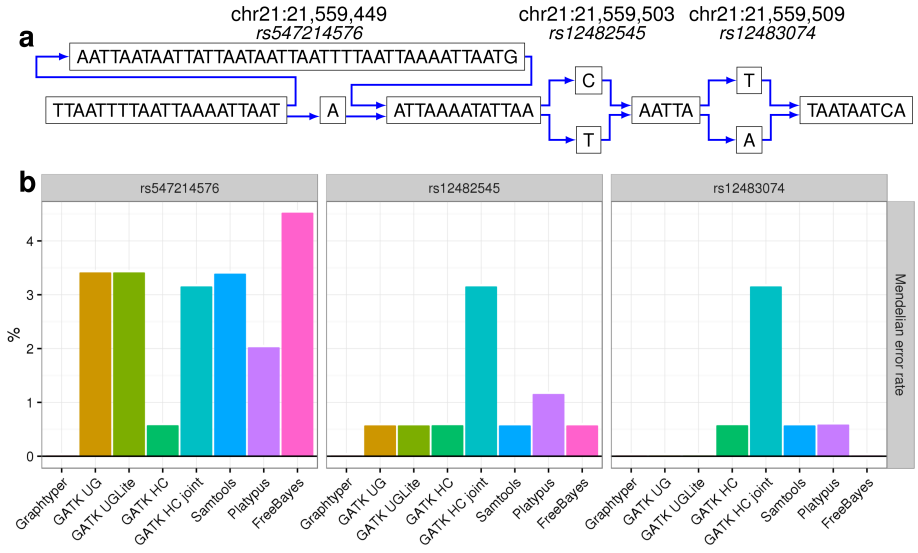


Figure 3.2. An example of how a pangenome graph realignment can reduce genotyping error rates. ***a.*** *The genomic region chr21:21,559,430 - 21,559,518 (GRCh38) and three previously reported sequence variants represented with a pangenome graph.* ***b.*** *Mendelian error rates of the three previously reported sequence variants called by eight variant callers. The Mendelian error rate is measured in 230 Icelandic parent-offspring trios.*

An argument against variation-aware realignment is the additional computational requirements. However, we showed in Paper I that variation-aware data structures, such as pangenomes, also allow read realignment and genotype calling to be performed in a single step. Therefore, while it is computationally more expensive to align each sequence read to a variation-aware data structure, they have a potential to scale well

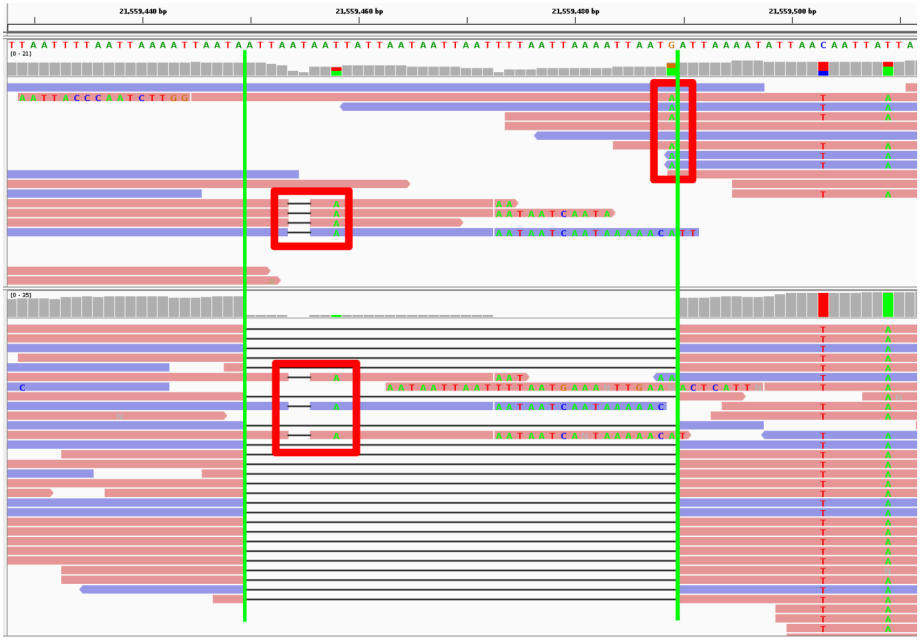


Figure 3.3. IGV (Robinson et al., 2011) visualization of sequence reads in two samples that were aligned to the assessed region in Figure 3.2. In red boxes are artifact variants due to read misalignments. The top sample is a heterozygous carrier of the deletion which only GraphTyper could correctly identify. The below sample is a homozygous carrier of the deletion.

with increasing number of samples. Thus, a variation-aware variant calling method which is slower when genotyping a single sample might be faster when genotyping a whole population.

We therefore believed there was a merit to implement GraphTyper, a population-scale variant caller that uses variation-aware realignments onto a pangenome graph.

3.3 GraphTyper algorithm design

3.3.1 Graph construction and indexing

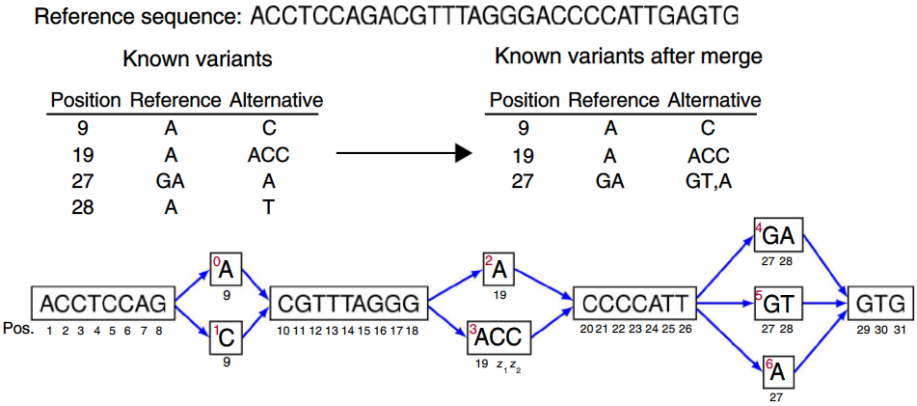


Figure 3.4. An example pangenome graph that was constructed from a reference sequence and several known variants. The path of the reference sequence is drawn as the topmost path of the graph. The red indexes in the variant nodes indicate their ID. The position of each base in the graph is shown below them. z_1 and z_2 are called special positions because they do not correspond to any genomic position.

The variation-aware data structure we use in GraphTyper is a directed acyclic graph (DAG). The nodes (or vertices) of the graph contain the DNA sequences, and the edges are directed and indicate paths in the graph (Figure 3.4). Prior to construction, all known variants are merged such that no two variants have overlapping reference alleles. GraphTyper constructs a graph from the reference sequence from a FASTA file and variants from a variant-call format (VCF) file, respectively.

A node is created for every allele sequence and for the sequences in-between variants. The allele sequences are associated to their corresponding nodes of the graph, which are called variant nodes. The reference sequences are similarly associated to nodes of the graphs, which are called reference nodes. The nodes are then connected with directed edges such that the paths of the graph encode potential haplotypes given the known

variants of the loci (Figure 3.4).

The graph and its construction defined in more detail in the Supplementary of Paper I.

In Paper II we further extended the graph construction algorithm such that it can also encode structural variants (SVs) into the graph structure by inserting their breakpoint sequences (Figure 3.5). To limit the size of the graph, we only insert up to 152 bp of the breakpoint sequences, which is determined by the short-read size. As a result, compute times are reduced and robust SV genotyping across SV lengths is allowed, as the mapping is not biased towards larger SVs. Further, the SV sequence is often only partially characterized.

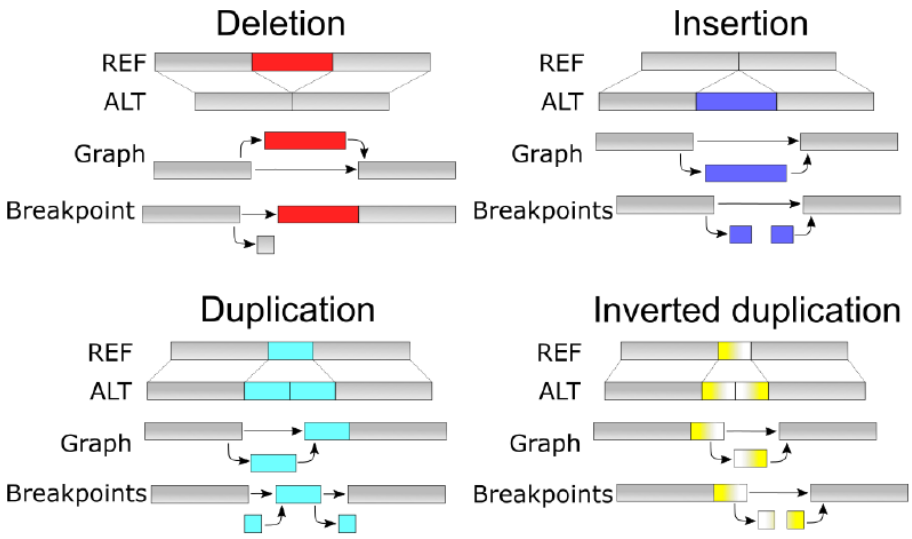


Figure 3.5. Example structural variants and their encoding in an acyclic graph structure. In GraphTyper, only the breakpoint sequences of the variants are inserted in the graph.

As mentioned before, the GraphTyper graph is required to be acyclic. Some types of sequence variations could be represented more compactly in a cyclic graph structure, and a few types cannot even be be represented using an acyclic graph (Garrison et al., 2018). For instance, inversions can be represented more compactly using a cyclic graph and repeat expansions of any length cannot be represented using an acyclic graph. However, the benefits of acyclic graphs include trivial conversion from FASTA and VCF to DAGs and vice versa, and a greater selection of efficient algorithms that require an acyclic graph (Sirén, 2016). We believe that the benefits of having using an acyclic graph in GraphTyper outweighs its limitations.

For GraphTyper, we designed an index data structure for speeding up read alignments against our pangenome graph structure. The index data is a key-value storage that maps

k-mers (sequence of length k, by default $k = 32$ is used in GraphTyper) to a list of all their location in the graph. More specifically, the location includes the starting position, the end position and any variant alleles the k-mers might overlap.

We store both the begin and end positions because that allows us to quickly check if two k-mers overlap each other by exactly 1 bp. Further, we store the variant allele such that in the subsequent alignment step we can determine which variant alleles a k-mer overlaps from the graph index.

5-mer	Start pos.	End pos.	Variant ID	Start pos.	End pos.	Variant ID
ACCCC	19	23	2	19	21	3
ACCTC	1	5	NA			
ACGTT	9	13	0			
AGACG	7	11	0			
AGCCG	7	11	1			
AGGGA	15	19	2	15	19	3
...						
CCCAT	21	25	NA			
CCCCA	20	24	NA			
CCCCC	z_1	22	3	z_2	23	3
...						
GGACC	17	z_2	3	17	21	2
GGGAC	16	z_1	3	16	20	2
...						
TTGTG	25	29	5			
...						

Figure 3.6. A graph index data structure constructed based on the pangenome graph example from Figure 3.4. A k-mer is associated to a list of unique start position, end position and variant ID of overlapping variant allele, if any. Here, $k = 5$ is used for demonstration.

The space usage of the index is 8 bytes for every k-mer, 4 bytes for the begin position, 4 bytes for the end position, and 2 bytes for the variant ID. Storing 32-mer is achieved by representing the four nucleotide bases with 2 bits: $A = 00$, $C = 01$, $G = 10$, and $T = 11$.

3.3.2 Read alignment

Our method for read alignment is a seed-extend algorithm. GraphTyper extracts a set of k-mers from the sequence read, which overlap by one DNA base in the read (Figure 3.7a), and looks them up in the graph using the graph index structure (Figure 3.7b). Seeds are generated from matches in the index look-up. Each seed has a graph begin position,

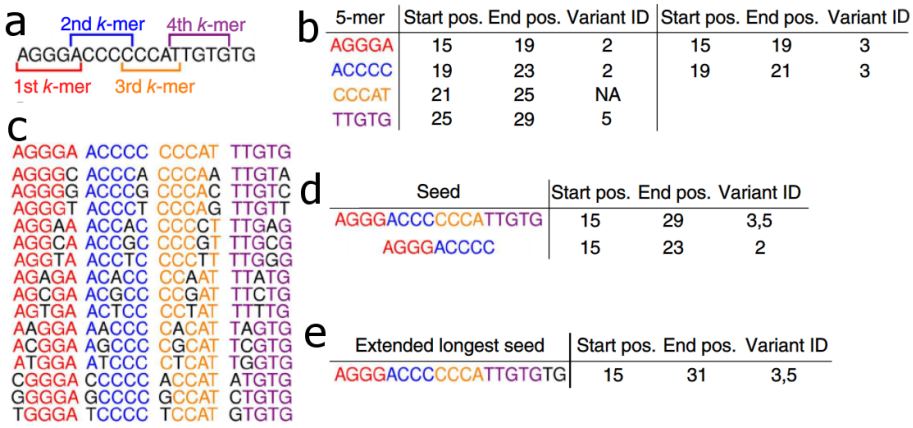


Figure 3.7. GraphTyper graph alignment against the pangenome graph from Figure 3.4 using the index from Figure 3.6. ***a.*** *An example sequence read. k-mers, here* 5*-mers, are extracted from the read such that 1 base is overlapping each adjacent* 5*-mer.* ***b.*** *Lookup of the extracted* 5*-mers in the graph index.* ***c.*** *All* 5*-mers in hamming distance 1 of the* 5*-mers in the sequence read.* ***d.*** *Seeds are founds by checking if the index lookup matches are a direct continuation of the previous* 5*-mer.* ***e.*** *Longest seed after it has been extended.*

graph end position, read begin position and read end position associated with it. If the seed alignments of two adjacent k-mers overlap by exactly one base, GraphTyper joins their matches into larger seeds (Figure 3.7d). This criterion holds when the graph end position and read end position of one seed matches the graph begin position and read begin position of another seed.

The longest seeds are then extended (Figure 3.7e) by finding a path in the graph with the fewest mismatches using a breadth first search. If no seeds are extended with 10 mismatches or fewer, GraphTyper again extracts a set of k-mers from the read which overlap by one base in a read, but now includes k-mers with one mismatch (Figure 3.7c). Only k-mers with mismatches are included since nearly all sequence errors in short-read sequence data are mismatches.

The graph alignment process is applied both to a read and its reverse complement, since we do not known how the read is oriented compared to the graph. If both orientations of a read align to the graph, we select the longer alignment or, if they are equally long, the alignment with fewer mismatches. Our read alignment algorithm does not guarantee that we find the optimal alignment to the graph. However, it will always find a match if the read aligns with no mismatches to the graph and very often when the read aligns with only a few mismatches. Furthermore, if a read matches with mismatches the alignment step is fast as it requires only a few index look-ups and an alignment extension to find the optimal graph alignment.

3.3.3 Sequence variant discovery

Once sequence reads are aligned to the graph, it is possible to discover sequence variants. For each read uniquely aligned to the graph, GraphTyper determines the position in the reference genome of its first and last aligned position in the graph and extracts the reference sequence between these two positions. Then on each side of the reference sequence, the read is extended by an additional 50 bases plus the number of soft clipped bases on the given side. Next, the read is locally aligned to the extracted reference sequence (Figure 3.8).

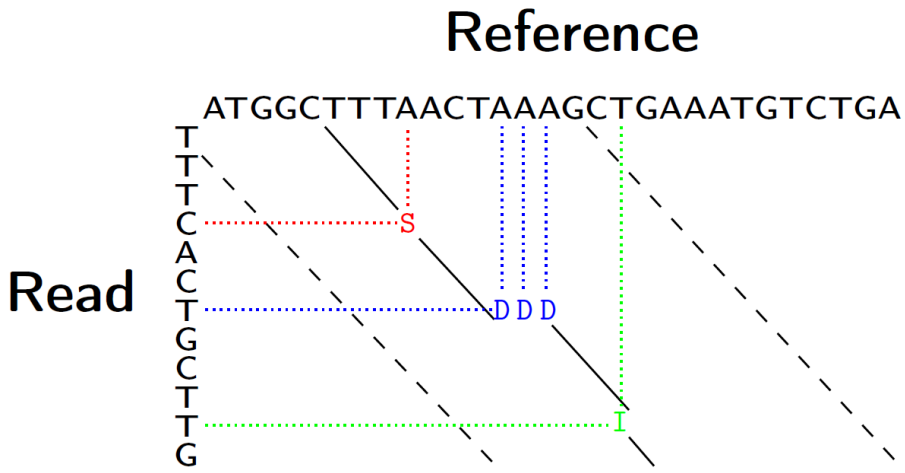


Figure 3.8. Banded alignment between a sequence read and the reference sequence. In this example, we observe a A>C SNP (red), a 3-bp deletion of AAA (blue), and an insertion of T (green).

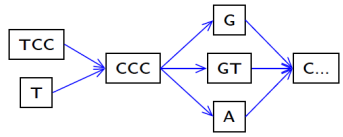
The alignment is banded with a 60 bp band on either side and the implementation is from the SeqAn library (Döring et al., 2008; Reinert et al., 2017). Differences in the local alignments are treated as observations of variants. Variants that are observed frequently enough in the sample individual are added to the graph structure in the following iterations (Section 3.4).

3.3.4 Sequence variant calling

We implemented a genotyping model in GraphTyper to call sequence variants in the graph based on the graph alignments. The graph alignments are treated as independent observations of each sample's underlying genotype. The genotyping models genotype sequence variants in the graph by considering nearby variants together.

Given graph-aligned sequence reads of a population, the likelihood that the reads were sampled from a pair of haplotypes is estimated for each sample and the haplotypes

with the highest likelihood are determined. To greatly reduce the number of haplotypes considered, all sequence variants located 5 bp or less from each other are grouped and each variant group is genotyped independently. Let $H_i = \{h_{i,1}, h_{i,2}\}$ be a multiset of the unknown haplotypes of sample i in a variant group, v, and let $R_i = \{r_{i,1}, r_{i,2}, ..., r_{i,|R_i|}\}$ be the sample's multiset of sequence reads aligned by GraphTyper to the variant group v.



Haplotypes

```
0: TCCCCCG-C
1: TCCCCCGTC
2: TCCCCCA-C
3: T--CCCG-C
4: T--CCCGTC
5: T--CCCA-C
```

Stacked reads

```
r  READ       (explained by haplotype #)
0: TCC        (0,1,2,3,4,5)
1: TCCC       (0,1,2,3,4,5)
2: TCCCC      (0,1,2)
3: TCCCCCG    (0,1)
4: TCCCCCGT   (1)
5: TCCCAC     (5)
6: TCCCA      (5)
```

Example

$$\mathcal{L}(H_1, H_1 | r_0, ..., r_6) = 1^5 \epsilon_{r_5, H_1, H_1} \epsilon_{r_6, H_1, H_1}$$
$$\mathcal{L}(H_1, H_5 | r_0, ..., r_6) = 1^2 \cdot (1/2)^5$$
$$\mathcal{L}(H_5, H_5 | r_0, ..., r_6) = 1^4 \epsilon_{r_2, H_5, H_5} \epsilon_{r_3, H_5, H_5} \epsilon_{r_4, H_5, H_5}$$

$\mathcal{L}(H_1, H_5 | r_0, ..., r_6)$ is selected

```
REF ALT   GT
TCC T     0/1
G   GT,A  1/2
```

Figure 3.9. An example of sequence variant calling. In the example, the graph represents six haplotypes that are to be genotyped. Given several sequence reads we can estimate the genotype likelihood of each pair of possible underlying haplotypes. Based on the sequence reads we can expect that haplotypes 1 and 5 are the most likely pair of haplotypes.

For each pair of haplotypes in the graph, a relative likelihood of the observed reads given the haplotypes $\mathcal{L}(R_i|H_i)$ computed (Equation 1). We assume that the reads from one individual are independent of other individuals' reads. GraphTyper computes the relative likelihood as:

$$\mathcal{L}(R_i|H_i) = \prod_{r_{ij} \in R_i} L(r_{ij}|H_i) \tag{1}$$

where the relative likelihood of observing a read r_{ij} given the pair of underlying haplotypes is set as:

$$L(r_{i,j}|H_i) = \begin{cases} 1 & \text{Both haplotypes in } H_i \text{ support the read.} \\ 1/2 & \text{One haplotype in } H_i \text{ supports the read.} \\ \varepsilon_{r_{i,j},H_i} & \text{Otherwise.} \end{cases} \quad (2)$$

where ε_{r_{ij},H_i} is chosen based on base quality, number of mismatches in read, mapping quality, alignment uniqueness, and read similarity to H_i. Possible values of ε_{r_{ij},H_i} are in the set $\left\{\frac{1}{2^4}, \frac{1}{2^5}, \frac{1}{2^6}, ..., \frac{1}{2^{13}}\right\}$. Restricting the selection to that set allows storing only the integer exponents, minimizing storage requirements and avoiding floating point precision problems.

Genotyping SVs is done similarly. However, deletion and duplications that are 50 bp or larger are also separately genotyped using another genotyping model, a coverage model. We expect that the coverage model is more effective for larger SVs than small ones. Relative likelihoods are estimated from coverage by mapping each graph alignment back to the reference haplotype and the alignment coverage is stored at each reference base-pair. To measure the coverage drop or increase, we look-up the alignment coverage every 20 bp and determine the median coverage in two 1,000 bp windows flanking the SV, c_{out} and median coverage inside the SV, c_{in}. We selected 1,000 bp since it gave us a good estimate of the alignment coverage in a window while being unlikely to overlap other SVs.

For deletions, we say that the coverage decrease, $\max(0, c_{\text{out}} - c_{\text{in}})$, is the number of reads supporting the deletion while c_{in} s the number of reads supporting the reference. The coverage model uses $\varepsilon_{r_{i,j},H_i} = 1/2^4$. For duplications the genotype likelihoods are calculated similarly but with coverage increase instead of decrease.

3.4 GraphTyper workflow

We have now described the main algorithms in GraphTyper. The full workflow of GraphTyper applies these algorithms to populate its graph with variants and later genotype them (Figure 3.10).

GraphTyper is run in several iterations. In the recommended pipeline, there are two discovery iterations, where new variants are added to the graph. Then there are three genotyping iterations where the graph is cleaned by removing likely false positive variants. Furthermore, with the introduction of genotyping SVs in Paper II, a fourth genotyping iteration is used for genotyping SVs.

In our small variant workflow for genotyping SNPs and indels, we partition the genome into 50 kbp regions (by default). We construct a graph, index it and genotype variants in each region separately. By partitioning the genome, we reduce the total time and memory requirement of the workflow. However, when genotyping SVs we partition

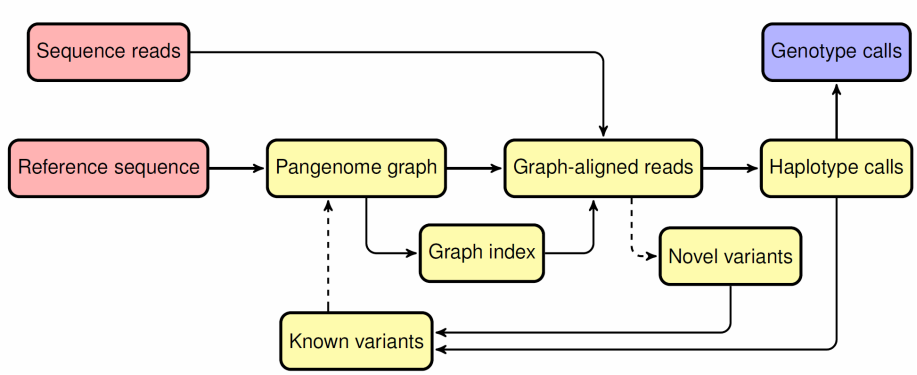


Figure 3.10. Diagram of GraphTyper's workflow. GraphTyper uses iterative genotyping processes. Dashed paths are optional. As input, GraphTyper requires a reference genome sequence and sequence reads (red) and outputs genotype calls (blue).

the genome into larger regions of 1.2 Mbp and make the partition overlap by 200 kbp. This is needed such that both breakpoints of the same SVs are in the same region/graph for as many variants as feasible.

3.5 GraphTyper applications

3.5.1 de novo mutation calling

De novo mutations (DNMs) are novel mutations occurring in the germ cells (egg or sperm) of one of the parents. These mutations can be absent from other cell lines of the parent but are present in their offspring (Figure 3.11). DNMs are known to be predominantly paternal and their rate increases with paternal age (Jónsson et al., 2017) (Paper A). They are important to study as they have often been implicated with diseases (Veltman and Brunner, 2012), in particular in children.

We have used GraphTyper to both discover and genotype these mutations, as well as re-genotype DNM sites that were discovered from other methods (Jónsson et al., 2018) (Paper B). Genotyping DNMs is challenging because not only does the offspring need to be correctly genotyped as a carrier, also both parents need to be correctly genotyped as non-carriers. An error in any of the three individuals causes the DNM to be missed by the analysis.

In Paper B the discovered DNMs were resequenced using targeted sequencing technologies to an average coverage depth of 500x. At such a high coverage, it is possible to estimate accurately the allele balance or variant allele fraction, which is the

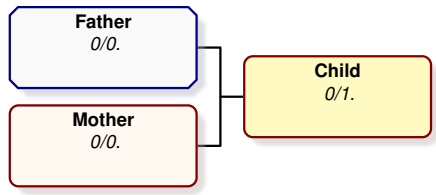


Figure 3.11. DNM genotypes in a parent-offspring trio. The mutation is novel and thus only observed in the child but not in either parent (only in one of their germ cells).

fraction of reads supporting each allele in the sample at the targeted locus. If the variant allele fraction is close to 50% the mutation is germline, while lower fractions indicate that the mutation is somatic.

3.5.2 HLA allele genotyping

The IPD-IMGT/HLA database (Robinson et al., 2015) contains known human leukocyte antigen (HLA) allele sequences. The sequences are identified HLA gene and a hierarchical field identifying system. The fields are colon separated. For example, an identifier might be: *HLA-A**01:02:01 for an allele of the *HLA-A* gene. The first field denotes the HLA allele family, the second field denotes the subtype within the family, the third field denotes groups with synonymous substitutions within the subtype, and the fourth field denotes allele differences in non-coding regions. Therefore, changes in the first fields are most likely to have functional impact.

In Paper I we showed that with GraphTyper it is possible to represent previously characterized HLA alleles using a pangenome graph. The graph is constructed from the allele sequences of the IPD-IMGT/HLA database. GraphTyper was able to genotype 27 HLA genes using this strategy. Six HLA genes were typed using PCR amplification, 2 of which to a 2-digit resolution and 4 of which to a 4-digit resolution. Previously, the deCODE Genetics laboratory performed HLA typing of the six genes with a PCR-based method at two-digit ($n = 647$) and four-digit ($n = 368$) resolution. GraphTyper's HLA calls are concordant with those previous typings (Table 3.1 and Table 3.2).

Table 3.1. 4-digit comparison of GraphTyper's HLA allele genotyping to PCR verified HLA genotypes.

HLA gene	n	Correct	1 error	2 errors	Accuracy
HLA-A	54	52	2	0	98.15%
HLA-DQA1	42	42	0	0	100.00%
HLA-DQB1	82	80	2	0	98.78%
HLA-DRB1	190	163	22	5	91.58%

The results of our analysis were used as a part of a meta-analysis that is described in Paper C (Kular et al., 2018). Prior to the paper, it had been identified that *HLA-*

Table 3.2. 2-digit comparison of GraphTyper's HLA allele genotyping to PCR verified HLA genotypes.

HLA gene	*n*	Correct	1 error	2 errors	Accuracy
HLA-A	54	52	2	0	98.15%
HLA-B	332	314	15	3	96.84%
HLA-C	315	290	19	6	95.08%
HLA-DQA1	42	42	0	0	100.00%
HLA-DQB1	82	81	1	0	99.39%
HLA-DRB1	190	189	1	0	99.74%

*DRB1*15:01* is a risk allele for multiple sclerosis (MS). Their paper found that *HLA-DRB1*15:01* is hypomethylated and predominantly expressed in monocytes among carriers of *HLA-DRB1*15:01*. Their data strongly suggest that DNA methylation in exon 2 of the *HLA-DRB1* gene mediates the effect of previously identified *HLA-DRB1*15:01*.

Our contribution to the study helped analyzing the effect and significance of the risk allele. In the Icelandic data there are 735 multiple sclerosis cases and 148,571 controls, which accounted for 86.7% of the controls used in the study. Thus, GraphTyper's genotyping of the HLA alleles was an important part of the study.

3.5.3 Genetic recombination maps

Crossover is the result of two homologous chromosomes crossing over during meiosis. It is initiated from double-strand breaks. Double-strand breaks do not occur at a uniform rate across the human genome, rather they occur more frequently in certain regions, which are called recombination hot-spots. Crossovers lead to offspring having different combinations of genes from those of their parents.

In Paper D (Halldorsson et al., 2019), 9,305,070 high-confidence small variants genotyped by GraphTyper were analyzed to detect crossovers in Icelanders. The median distance between the variants in the analysis was 178 bp. A total of 28,075 Icelanders were genotyping in the study. The locations of the crossovers were determined from haplotype phase transitions at chip-typed SNP variants. The phase transitions of a proband are detected at sites where either parent is heterozygous. The crossover locations are then refined using the GraphTyper variants, when available. As a result, the recombination genetic map generated described in the paper is at an extremely high resolution. Moreover, the paper assessed DNMs genotyped using GraphTyper in nearly 2,976 parent-offspring Icelandic trios and compared them to the crossover locations. Analysis of these data allowed assessing the contribution of crossovers to mutagenesis. The results of the study showed that mutation rate is substantially increased near crossovers. GraphTyper's genotyping of the Icelandic population was therefore an essential part of the study.

4 Discussion

4.1 Conclusions

Previous variant callers genotype based on read alignments to linear reference genomes, which limits their performance in polymorphic regions. To better characterize sequence diversity, we implemented a novel variation-aware data structure and developed efficient algorithms in a software called GraphTyper. GraphTyper is free and its source code is available on Github. GraphTyper locally realigns sequence reads from a genomic region to a pangenome graph, and concomitantly genotypes sequence variants.

We assessed GraphTyper's performance and compared it to previous variant callers. Compared to previous small variant callers, we showed in Paper I that GraphTyper had similar or lower computational requirements and a higher recall rate. We demonstrated that GraphTyper's compute times per sample were between 80 and 100 CPU hours for samples whole-genome sequenced at 35x average coverage. The compute time per sample scaled near-constantly with the number of samples, while it increased substantially with the number of samples when using GATK UG (McKenna et al., 2010). Therefore, we believe that GraphTyper is a valuable method for population-scale genotyping small variants. With the addition of SV genotyping, we further extended GraphTyper such that it can genotype across the variation spectrum. We demonstrated in Paper II that our SV genotyping method is both sensitive and accurate compared to previous methods by measuring transmission error rates in parent-offspring trios and validating our SVs with long-read sequence data. We showed that GraphTyper alleviates some of the problems of previous genotypers.

Our results further show the importance of replacing the linear reference with richer data structures to improve our understanding of how sequence diversity impacts diseases and traits. By developing GraphTyper, we believe we are contributing a method that can assist in creating a high quality human pangenome reference. It is an ambitious goal but an important one, since sequence analysis will always suffer in quality while it is biased towards a single reference allele.

4.2 Future work

Based on the work presented in this thesis, there are several aspects of that might be improved in the future.

4.2.1 Incorporation of global variation-aware alignment

Our current pipeline still relies on the linear reference sequence and BWA-MEM (Li and Durbin, 2009) for global read alignments in order to assign reads to a region. To completely remove bias towards the reference genome and fully utilize the promise of pangenome analysis, robust graph alignment methods are required. A notable project implementing a global variation-aware alignment is vg (Garrison et al., 2018).

In the future, we would like to extend our pipeline such that it could incorporate vg or another similar method into GraphTyper. However, such an extension is not easy to implement. Our current design expects sequence reads in SAM-formatted files (Li et al., 2009) but if graph read alignments cannot currently be converted into the format without some loss of information. Despite this, we have assessed the performance of using vg prior to GraphTyper (Garrison et al., 2018) and saw that there is a marginal improvement in variant sensitivity (0.02% for SNPs and 0.06% for indels) compared to using BWA-MEM with GraphTyper.

These results show that further improvements can be made by incorporating a global variation-aware read aligner into our pipeline.

4.2.2 Extend GraphTyper support for other species

GraphTyper was initially intended only for human reference genomes. Therefore, it was only possible to run GraphTyper using human builds such as hg19 or GRCh38. However, in GraphTyper version 2 we added the possibility of using other reference genomes as well, including non-human, but the genotyping models all assume that the sequences originate from a diploid genome. There are already sequence analysis study of cattle (Crysnanto et al., 2019) that use GraphTyper and studies of more species are currently in progress, including great apes and chimpanzees. The study compared GraphTyper to GATK and SAMTools and showed GraphTyper discovered the more polymorphic variant sites than the other variant callers, while also outperforming the other tools in terms of genotype concordance, non-reference sensitivity, and non-reference discrepancy.

While GraphTyper's results on other species are promising, there are still some limitations of our method. One limitation is that the total genome size cannot exceed 4,294,967,296 $\left(2^{32}\right)$ bp, as the positions of the genome are stored in 4-byte integers.

Alternative loci are counted as part of the total genome size. Some genome references exceed that maximum support genome size and thus the positions would not fit into 4 bytes. In the future we would like to add the possibility of storing the positions in 8-byte integers when the user inputs a very large reference genome.

Another limitation is that the genotype model of GraphTyper assumes that the organism is diploid. It is easy to convert diploid calls to haploid calls by changing the calls to the most probable homozygous genotype, but converting diploid calls to higher ploidy levels is typically not possible. Either extending the current genotyping model or creating a new model for higher ploidy levels is therefore needed. In the future we would like to make such a model, as it would make GraphTyper an applicable solution for even more sequence analysis datasets.

4.2.3 Local assembly

As previously discussed in Chapter 1, some variant calling methods perform a *de novo* assembly to improve discovery sensitivity of indels (McKenna et al., 2010; Iqbal et al., 2012; Rimmer et al., 2014) and structural variants (Chen et al., 2016). Current *de novo* assembly algorithms can either yield an overlap graph (Myers, 2005) a De Bruijn graph structure (Nurk et al., 2013; Zerbino and Birney, 2008; Iqbal et al., 2012).

The indel discovery sensitivity in GraphTyper is lower than both GATK HC and Platypus, but both of those programs use local assembly for discovering indels. It is thus likely that adding an iteration into the GraphTyper workflow, where a local assembly step is performed, could be beneficial. We would thus like to consider adding that into GraphTyper in the future.

4.2.4 de novo SV analysis

We have used GraphTyper to genotype *de novo* SNPs and indel variants in the Icelandic population (Paper B), but approximately 70 such mutations are expected on average per individual. Analysis of these mutations is important for improving understanding of novel mutations and because they are frequently implicated in rare diseases (Veltman and Brunner, 2012), in particular in children.

There has been some analyses of *de novo* SVs (Brandler et al., 2016), which have suggested that they are implicated in autism. But still a lot of questions regarding *de novo* SVs are unanswered, mostly because they are extremely rare. Their exact rate is unknown, however estimates are typically between 1 *de novo* SV per 6-12 individuals (Kloosterman et al., 2015; Brandler et al., 2016). Therefore, the analysis of *de novo* SVs requires an accurate SV genotype method that can handle genotyping a very large sample set.

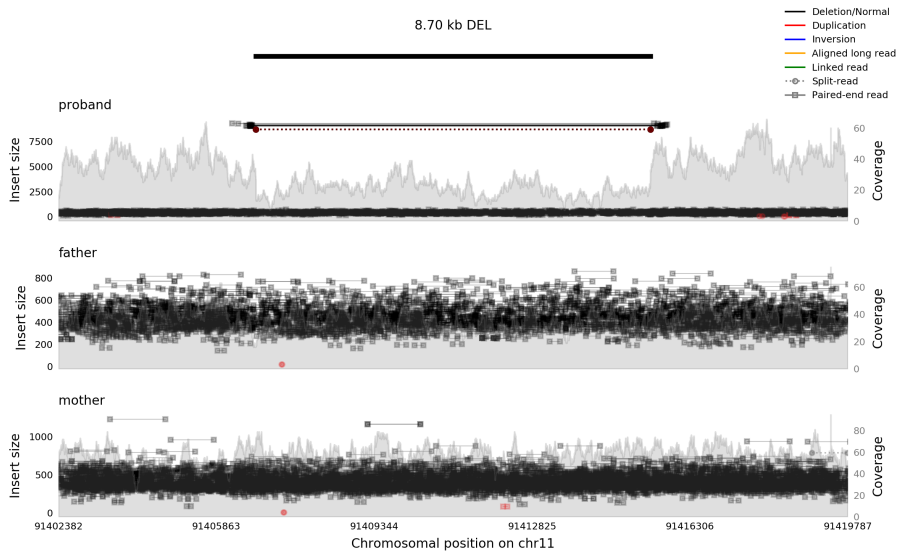


Figure 4.1. An example of a de novo 8.7 kb deletion on human chromosome 11 visualized with samplot. The top sample is the proband, which has many sequence reads mapping with insert size matching with the deletion, clipped reads with the breakpoint and a drop in coverage. The middle and bottom samples are the parents, which do not have any reads that support the deletion.

As we have previously described, we have added genotyping of SVs in GraphTyper2. The results in Paper II show that the accuracy of our method is very high compared to previous SV genotyping methods. Thus, we believe GraphTyper2 has paved a way to analyze de novo SVs. An example of a *de novo* SV that was genotyped using GraphTyper is shown in Figure 4.1. Further analyses are needed, as they may give insights into the rate, mechanisms, and origins of SVs.

4.2.5 Read-based phasing

The output of GraphTyper consists of unphased genotype calls and thus contain no information whether the variants are paternal or maternal. Additionally, it does not provide any information whether any given two variants are likely to be on the same haplotype or not. However, since the two reads in a read pair are sequenced of the same haplotype, we can phase the variants we are genotyping.

Phasing is the process of inferring the correct relationship (*cis* or *trans*) between alleles at multiple variant loci. For example, phasing a heterozygous deletion ACC/A and a heterozygous SNP T/G may reveal that the correct haplotypes are ACC-G and A-T. Phasing can be applied using chip and pedigree data (long-range phasing) (Kong et al., 2008) or directly from the sequence reads (read-phasing) (Lancia et al., 2001; Halldórsson et al., 2002; Martin et al., 2016). While read-phasing using short-read data results in a highly fragmented phase blocks, it is sometimes required to infer the correct phase. For example, it is only possible to phase *de novo* variants by read-phasing to determine their parent-of-origin.

GraphTyper realigns both reads in the read pair together, so for cases that both reads overlap two variants the necessary read-phasing information between those variant alleles is available, although the information is not currently used. The realignment and the genotyping steps are performed simultaneously, however it is possible to also perform read-phasing in the same step. Using previously developed methods for read-phasing (Martin et al., 2016) would require re-reading all sequence data again. It is therefore expected to be substantially faster to use the read-phasing information in GraphTyper to read-phase the variants at the same time as they are genotyped.

References

Biederstedt, E., Oliver, J. C., Hansen, N. F., Jajoo, A., Dunn, N., Olson, A., Busby, B., and Dilthey, A. T. (2018). NovoGraph: Human genome graph construction from multiple long-read de novo assemblies. *F1000Research*, 7:1391.

Brandler, W., Antaki, D., Gujral, M., Noor, A., Rosanio, G., Chapman, T., Barrera, D., Lin, G., Malhotra, D., Watts, A., Wong, L., Estabillo, J., Gadomski, T., Hong, O., Fajardo, K., Bhandari, A., Owen, R., Baughn, M., Yuan, J., Solomon, T., Moyzis, A., Maile, M., Sanders, S., Reiner, G., Vaux, K., Strom, C., Zhang, K., Muotri, A., Akshoomoff, N., Leal, S., Pierce, K., Courchesne, E., Iakoucheva, L., Corsello, C., and Sebat, J. (2016). Frequency and Complexity of De Novo Structural Mutation in Autism. *The American Journal of Human Genetics*, 98(4):667–679.

Chen, X., Schulz-Trieglaff, O., Shaw, R., Barnes, B., Schlesinger, F., Källberg, M., Cox, A. J., Kruglyak, S., and Saunders, C. T. (2016). Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, 32(8):1220–1222.

Chiang, C., Layer, R. M., Faust, G. G., Lindberg, M. R., Rose, D. B., Garrison, E. P., Marth, G. T., Quinlan, A. R., and Hall, I. M. (2015). SpeedSeq: ultra-fast personal genome analysis and interpretation. *Nature Methods*, 12(10):966–968.

Computational Pan-Genomics Consortium (2016). Computational pan-genomics: status, promises and challenges. *Briefings in bioinformatics*, page bbw089.

Crick, F. (1970). Central dogma of molecular biology. *Nature*, 227:561–563.

Crysnanto, D., Wurmser, C., and Pausch, H. (2019). Accurate sequence variant genotyping in cattle using variation-aware genome graphs. *bioRxiv*, page 460345.

DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., Philippakis, A. A., del Angel, G., Rivas, M. A., Hanna, M., McKenna, A., Fennell, T. J., Kernytsky, A. M., Sivachenko, A. Y., Cibulskis, K., Gabriel, S. B., Altshuler, D., and Daly, M. J. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, 43(5):491–498.

Dilthey, A., Cox, C., Iqbal, Z., Nelson, M. R., and McVean, G. (2015). Improved genome inference in the MHC using a population reference graph. *Nature Genetics*, 47(6):682–688.

Döring, A., Weese, D., Rausch, T., and Reinert, K. (2008). SeqAn an efficient, generic C++ library for sequence analysis. *BMC bioinformatics*, 9:11.

Garrison, E. and Marth, G. (2012). Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, page 9.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., Paten, B., and Durbin, R. (2018).

Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, 36(9):875–879.

Halldórsson, B. V., Bafna, V., Edwards, N., Lippert, R., Yooseph, S., and Istrail, S. (2002). A Survey of Computational Methods for Determining Haplotypes. In *Computational Methods for SNPs and Haplotype Inference*, pages 26–47. Springer, Berlin, Heidelberg.

Halldorsson, B. V., Palsson, G., Stefansson, O. A., Jonsson, H., Hardarson, M. T., Eggertsson, H. P., Gunnarsson, B., Oddsson, A., Halldorsson, G. H., Zink, F., Gudjonsson, S. A., Frigge, M. L., Thorleifsson, G., Sigurdsson, A., Stacey, S. N., Sulem, P., Masson, G., Helgason, A., Gudbjartsson, D. F., Thorsteinsdottir, U., and Stefansson, K. (2019). Characterizing mutagenic effects of recombination through a sequence-level genetic map. *Science (New York, N.Y.)*, 363(6425):eaau1043.

International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921.

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nature Genetics*, 44(2):226–232.

Jónsson, H., Sulem, P., Arnadottir, G., Pálsson, G., Eggertsson, H., Kristmundsdottir, S., Zink, F., Kehr, B., Hjorleifsson, K., Jensson, B., Jonsdottir, I., Marelsson, S., Gudjonsson, S., Gylfason, A., Jonasdottir, A., Jonasdottir, A., Stacey, S., Magnusson, O., Thorsteinsdottir, U., Masson, G., Kong, A., Halldorsson, B., Helgason, A., Gudbjartsson, D., and Stefansson, K. (2018). Multiple transmissions of de novo mutations in families. *Nature Genetics*.

Jónsson, H., Sulem, P., Kehr, B., Kristmundsdottir, S., Zink, F., Hjartarson, E., Hardarson, M., Hjorleifsson, K., Eggertsson, H., Gudjonsson, S., Ward, L., Arnadottir, G., Helgason, E., Helgason, H., Gylfason, A., Jonasdottir, A., Jonasdottir, A., Rafnar, T., Frigge, M., Stacey, S., Th Magnusson, O., Thorsteinsdottir, U., Masson, G., Kong, A., Halldorsson, B., Helgason, A., Gudbjartsson, D., and Stefansson, K. (2017). Parental influence on human germline de novo mutations in 1,548 trios from Iceland. *Nature*, 549(7673).

Kehr, B., Melsted, P., and Halldórsson, B. V. (2016). PopIns: population-scale detection of novel sequence insertions. *Bioinformatics*, 32(7):961–967.

Kloosterman, W. P., Francioli, L. C., Hormozdiari, F., Marschall, T., Hehir-Kwa, J. Y., Abdellaoui, A., Lameijer, E.-W., Moed, M. H., Koval, V., Renkens, I., van Roosmalen, M. J., Arp, P., Karssen, L. C., Coe, B. P., Handsaker, R. E., Suchiman, E. D., Cuppen, E., Thung, D. T., McVey, M., Wendl, M. C., Genome of Netherlands Consortium, G. o. t. N., Uitterlinden, A., van Duijn, C. M., Swertz, M. A., Wijmenga, C., van Ommen, G. B., Slagboom, P. E., Boomsma, D. I., Schönhuth, A., Eichler, E. E., de Bakker, P. I. W., Ye, K., Guryev, V., Consortium, G. o. t. N., Uitterlinden, A., van Duijn, C. M., Swertz, M. A., Wijmenga, C., van Ommen, G. B., Slagboom, P. E., Boomsma, D. I., Schönhuth, A., Eichler, E. E., de Bakker, P. I., Ye, K., and Guryev, V. (2015). Characteristics of de novo structural changes in the human genome. *Genome research*, 25(6):792–801.

Kong, A., Masson, G., Frigge, M. L., Gylfason, A., Zusmanovich, P., Thorleifsson, G., Olason, P. I., Ingason, A., Steinberg, S., Rafnar, T., Sulem, P., Mouy, M., Jonsson, F., Thorsteinsdottir, U., Gudbjartsson, D. F., Stefansson, H., and Stefansson, K. (2008).

Detection of sharing by descent, long-range phasing and haplotype imputation. *Nature Genetics*, 40(9):1068–1075.

Kular, L., Liu, Y., Ruhrmann, S., Zheleznyakova, G., Marabita, F., Gomez-Cabrero, D., James, T., Ewing, E., Lindén, M., Górnikiewicz, B., Aeinehband, S., Stridh, P., Link, J., Andlauer, T., Gasperi, C., Wiendl, H., Zipp, F., Gold, R., Tackenberg, B., Weber, F., Hemmer, B., Strauch, K., Heilmann-Heimbach, S., Rawal, R., Schminke, U., Schmidt, C., Kacprowski, T., Franke, A., Laudes, M., Dilthey, A., Celius, E., Søndergaard, H., Tegnér, J., Harbo, H., Oturai, A., Olafsson, S., Eggertsson, H., Halldorsson, B., Hjaltason, H., Olafsson, E., Jonsdottir, I., Stefansson, K., Olsson, T., Piehl, F., Ekström, T., Kockum, I., Feinberg, A., and Jagodic, M. (2018). DNA methylation as a mediator of HLA-DRB1 15:01 and a protective variant in multiple sclerosis. *Nature Communications*, 9(1).

Lancia, G., Bafna, V., Istrail, S., Lippert, R., and Schwartz, R. (2001). SNPs Problems, Complexity, and Algorithms. In *Algorithms - ESA 2001*, pages 182–193. Springer, Berlin, Heidelberg.

Layer, R. M., Chiang, C., Quinlan, A. R., and Hall, I. M. (2014). LUMPY: a probabilistic framework for structural variant discovery. *Genome Biology*, 15(6):R84.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079.

Martin, M., Patterson, M., Garg, S., Fischer, S. O., Pisanti, N., Klau, G. W., Schöenhuth, A., and Marschall, T. (2016). WhatsHap: fast and accurate read-based phasing. *bioRxiv*, page 085050.

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., and DePristo, M. A. (2010). The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9):1297–1303.

Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics*, 21(Suppl 2):ii79–ii85.

Nurk, S., Bankevich, A., Antipov, D., Gurevich, A., Korobeynikov, A., Lapidus, A., Prjibelsky, A., Pyshkin, A., Sirotkin, A., Sirotkin, Y., Stepanauskas, R., McLean, J., Lasken, R., Clingenpeel, S. R., Woyke, T., Tesler, G., Alekseyev, M. A., and Pevzner, P. A. (2013). Assembling Genomes and Mini-metagenomes from Highly Chimeric Reads. *RECOMB*, pages 158–170.

Rakocevic, G., Semenyuk, V., Lee, W.-P., Spencer, J., Browning, J., Johnson, I. J., Arsenijevic, V., Nadj, J., Ghose, K., Suciu, M. C., Ji, S.-G., Demir, G., Li, L., Toptaş, B. Ç., Dolgoborodov, A., Pollex, B., Spulber, I., Glotova, I., Kómár, P., Stachyra, A. L., Li, Y., Popovic, M., Källberg, M., Jain, A., and Kural, D. (2019). Fast and accurate genomic analyses using genome graphs. *Nature Genetics*, page 1.

Rausch, T., Zichner, T., Schlattl, A., Stutz, A. M., Benes, V., and Korbel, J. O. (2012). DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, 28(18):i333–i339.

Reinert, K., Dadi, T. H., Ehrhardt, M., Hauswedell, H., Mehringer, S., Rahn, R., Kim, J., Pockrandt, C., Winkler, J., Siragusa, E., Urgese, G., and Weese, D. (2017). The SeqAn

C++ template library for efficient sequence analysis: A resource for programmers. *Journal of Biotechnology*, 261:157–168.

Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S. R. F., Wilkie, A. O. M., McVean, G., and Lunter, G. (2014). Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature genetics*, 46(November 2013):1–9.

Robinson, J., Halliwell, J. A., Hayhurst, J. D., Flicek, P., Parham, P., and Marsh, S. G. E. (2015). The IPD and IMGT/HLA database: Allele variant databases. *Nucleic Acids Research*, 43(D1):D423–D431.

Robinson, J. T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E. S., Getz, G., and Mesirov, J. P. (2011). Integrative genomics viewer. *Nature biotechnology*, 29(1):24–6.

Seo, J.-S., Rhie, A., Kim, J., Lee, S., Sohn, M.-H., Kim, C.-U., Hastie, A., Cao, H., Yun, J.-Y., Kim, J., Kuk, J., Park, G. H., Kim, J., Ryu, H., Kim, J., Roh, M., Baek, J., Hunkapiller, M. W., Korlach, J., Shin, J.-Y., and Kim, C. (2016). De novo assembly and phasing of a Korean human genome. *Nature*, 538(7624):243–247.

Shao, H., Bellos, E., Yin, H., Liu, X., Zou, J., Li, Y., Wang, J., and Coin, L. J. M. (2013). A population model for genotyping indels from next-generation sequence data. *Nucleic Acids Research*, 41(3):e46–e46.

Sibbesen, J. A., Maretty, L., and Krogh, A. (2018). Accurate genotyping across variant classes and lengths using variant graphs. *Nature Genetics*, 50(7):1054–1059.

Sirén, J. (2016). Indexing Variation Graphs. *arxiv*.

Veltman, J. A. and Brunner, H. G. (2012). De novo mutations in human genetic disease. *Nature Reviews Genetics*, 13(8):565–575.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., Skupski, M., Subramanian, G., Thomas, P. D., Zhang, J., Gabor Miklos, G. L., Nelson, C., Broder, S., Clark, A. G., Nadeau, J., McKusick, V. A., Zinder, N., Levine, A. J., Roberts, R. J., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I., Charlab, R., Chaturvedi, K., Deng, Z., Di Francesco, V., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A. E., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T. J., Higgins, M. E., Ji, R. R., Ke, Z., Ketchum, K. A., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G. V., Milshina, N., Moore, H. M., Naik, A. K., Narayan, V. A., Neelam, B., Nusskern, D., Rusch, D. B., Salzberg, S., Shao, W., Shue, B., Sun, J., Wang, Z., Wang, A., Wang, X., Wang, J., Wei, M., Wides, R., Xiao, C., Yan, C., Yao, A., Ye, J., Zhan, M., Zhang, W., Zhang, H., Zhao, Q., Zheng, L., Zhong, F., Zhong, W., Zhu, S., Zhao, S., Gilbert, D., Baumhueter, S., Spier, G., Carter, C., Cravchik, A., Woodage, T., Ali, F., An, H., Awe, A., Baldwin, D., Baden, H., Barnstead, M., Barrow, I., Beeson, K., Busam, D., Carver, A., Center, A., Cheng, M. L., Curry, L., Danaher, S., Davenport, L., Desilets, R., Dietz, S., Dodson, K., Doup, L., Ferriera, S., Garg, N., Gluecksmann, A., Hart, B., Haynes, J., Haynes, C., Heiner, C., Hladun, S., Hostin, D., Houck, J., Howland, T., Ibegwam, C., Johnson, J., Kalush, F., Kline, L.,

Koduru, S., Love, A., Mann, F., May, D., McCawley, S., McIntosh, T., McMullen, I., Moy, M., Moy, L., Murphy, B., Nelson, K., Pfannkoch, C., Pratts, E., Puri, V., Qureshi, H., Reardon, M., Rodriguez, R., Rogers, Y. H., Romblad, D., Ruhfel, B., Scott, R., Sitter, C., Smallwood, M., Stewart, E., Strong, R., Suh, E., Thomas, R., Tint, N. N., Tse, S., Vech, C., Wang, G., Wetter, J., Williams, S., Williams, M., Windsor, S., Winn-Deen, E., Wolfe, K., Zaveri, J., Zaveri, K., Abril, J. F., Guigó, R., Campbell, M. J., Sjolander, K. V., Karlak, B., Kejariwal, A., Mi, H., Lazareva, B., Hatton, T., Narechania, A., Diemer, K., Muruganujan, A., Guo, N., Sato, S., Bafna, V., Istrail, S., Lippert, R., Schwartz, R., Walenz, B., Yooseph, S., Allen, D., Basu, A., Baxendale, J., Blick, L., Caminha, M., Carnes-Stine, J., Caulk, P., Chiang, Y. H., Coyne, M., Dahlke, C., Mays, A., Dombroski, M., Donnelly, M., Ely, D., Esparham, S., Fosler, C., Gire, H., Glanowski, S., Glasser, K., Glodek, A., Gorokhov, M., Graham, K., Gropman, B., Harris, M., Heil, J., Henderson, S., Hoover, J., Jennings, D., Jordan, C., Jordan, J., Kasha, J., Kagan, L., Kraft, C., Levitsky, A., Lewis, M., Liu, X., Lopez, J., Ma, D., Majoros, W., McDaniel, J., Murphy, S., Newman, M., Nguyen, T., Nguyen, N., Nodell, M., Pan, S., Peck, J., Peterson, M., Rowe, W., Sanders, R., Scott, J., Simpson, M., Smith, T., Sprague, A., Stockwell, T., Turner, R., Venter, E., Wang, M., Wen, M., Wu, D., Wu, M., Xia, A., Zandieh, A., and Zhu, X. (2001). The sequence of the human genome. *Science (New York, N.Y.)*, 291(5507):1304–51.

Wang, J., Wang, W., Li, R., Li, Y., Tian, G., Goodman, L., Fan, W., Zhang, J., Li, J., Zhang, J., Guo, Y., Feng, B., Li, H., Lu, Y., Fang, X., Liang, H., Du, Z., Li, D., Zhao, Y., Hu, Y., Yang, Z., Zheng, H., Hellmann, I., Inouye, M., Pool, J., Yi, X., Zhao, J., Duan, J., Zhou, Y., Qin, J., Ma, L., Li, G., Yang, Z., Zhang, G., Yang, B., Yu, C., Liang, F., Li, W., Li, S., Li, D., Ni, P., Ruan, J., Li, Q., Zhu, H., Liu, D., Lu, Z., Li, N., Guo, G., Zhang, J., Ye, J., Fang, L., Hao, Q., Chen, Q., Liang, Y., Su, Y., San, A., Ping, C., Yang, S., Chen, F., Li, L., Zhou, K., Zheng, H., Ren, Y., Yang, L., Gao, Y., Yang, G., Li, Z., Feng, X., Kristiansen, K., Wong, G. K.-S., Nielsen, R., Durbin, R., Bolund, L., Zhang, X., Li, S., Yang, H., and Wang, J. (2008). The diploid genome sequence of an Asian individual. *Nature*, 456(7218):60–65.

Watson, J. D. and Crick, F. H. (1953). A structure for deoxyribose nucleic acid. *Nature*, 171:737–738.

Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, 18(5):821–9.

Appendix

The two original papers are displayed in the following pages.

Paper I

Paper I

Graphtyper enables population-scale genotyping using pangenome graphs

Hannes P. Eggertsson, Hakon Jonsson, Snaedis Kristmundsdottir, Eirikur Hjartarson, Birte Kehr, Gisli Masson, Florian Zink, Kristjan E. Hjorleifsson, Aslaug Jonasdottir, Adalbjorg Jonasdottir, Ingileif Jonsdottir, Daniel F. Gudbjartsson, Pall Melsted, Kari Stefansson, Bjarni V. Halldorsson

Nature Genetics, volume 49, pages 1654-1660 (2017)

Title

Graphtyper enables population-scale genotyping using pangenome graphs

Authors

Hannes P. Eggertsson[1,2], Hakon Jonsson[1], Snaedis Kristmundsdottir[1,3], Eirikur Hjartarson[1], Birte Kehr[1,4], Gisli Masson[1], Florian Zink[1], Kristjan E. Hjorleifsson[1], Aslaug Jonasdottir[1], Adalbjorg Jonasdottir[1], Ingileif Jonsdottir[1,5], Daniel F. Gudbjartsson[1,2], Pall Melsted[1,2], Kari Stefansson[1,5], Bjarni V. Halldorsson[1,3]

[1]deCODE genetics / Amgen Inc., Sturlugata 8, Reykjavik, Iceland

[2]School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland

[3]School of Science and Engineering, Reykjavik University, Reykjavik, Iceland

[4]Berlin Institute of Health (BIH), 10178 Berlin, Germany

[5]Faculty of Medicine, School of Health Sciences, University of Iceland, Reykjavik, Iceland

Corresponding authors: Hannes P. Eggertsson (hannese@decode.is), Bjarni V. Halldorsson (bjarnih@decode.is)

Abstract

A fundamental requisite for genetic studies is an accurate determination of sequence variation. While human genome sequence diversity is increasingly well characterized, there is a need for efficient ways to utilize this knowledge in sequence analysis. Here we present Graphtyper, a publicly available novel algorithm and software for discovering and genotyping sequence variants. Graphtyper realigns short-read sequence data to a pangenome, a variation-aware graph structure that encodes sequence variation within a population by representing possible haplotypes as graph paths. Our results show that Graphtyper is fast, highly scalable, and provides sensitive and accurate genotype calls. Graphtyper genotyped 89.4 million sequence variants in whole-genomes of 28,075 Icelanders using less than 100,000 CPU days, including detailed genotyping of six human leukocyte antigen (HLA) genes. We show that Graphtyper is a valuable tool in characterizing sequence variation in both small and population-scale sequencing studies.

Introduction

Advances in DNA sequencing technology have improved characterization of sequence diversity in the human genome and have resulted in refinements of the reference sequence[1–4]. The human reference sequence is extremely useful, but it represents a consensus of genomes and therefore it does not capture sequence variation within or between populations[5,6].

In the latest version of the human reference genome (GRCh38), there are several alternate loci where the sequence variation is too complex to be represented with a single sequence. These loci are generally highly polymorphic, and many are known to co-segregate with disease and are therefore of great interest in population genetics. The most prominent example, the human leukocyte antigen (HLA) region, is known to associate with a number of human diseases[7]. Given the importance of this region, it has been further characterized in the IPD-IMGT/HLA database[8], which contains a large collection of known HLA allele sequences. Such variation should be included in genome diversity analyzes[9].

Short-read sequencing is the standard in genome-wide sequence analysis. Most common approaches for discovering sequence variants involve aligning sequence reads to a reference genome[10] and searching for variants as alternative sequences in read alignments (Fig. 1a i). However, some reads cannot be aligned to a reference genome, particularly those originating from highly polymorphic regions and regions absent from the reference genome. Reference genome alignments are also generally done without awareness of variation, causing mapping bias towards the reference allele and misalignments around indels[11,12].

Richer data structures that utilize the large amount of available sequence variation data promise to alleviate some of the limitations of previous methods[13–16]. Although approaches that find polymorphisms in reference-free assemblies have been developed to avoid these limitations[17,18], *de novo* assembly algorithms remain computationally expensive, have less sensitivity[18], and use data structures that have a complex coordinate system.

Pangenomes[13,19,20] have recently been proposed to counter weaknesses of both reference alignments and *de novo* assemblies by extending the linear reference alignments with variation-aware alignments[21]. Pangenomes incorporate prior information about variation, allowing read aligners to better distinguish between sequencing errors in reads and true sequence variation. Unlike *de novo* assembly algorithms, pangenomes represent sequence variation with respect to the reference genome, enabling a direct access to its annotated biological features. Variation-aware data structures, such as pangenomes, also allow read mapping and genotype calling to be performed in a single step[13].

Graph-like data structures with directed edges have commonly been used to represent pangenomes[20,22–25]. In an idealized pangenome graph, nodes represent sequences and the sequence of every genotyped individual genome is a path in the graph, but not necessarily vice versa. A number of algorithms have recently been developed that tackle the problems of graph construction, indexing and alignment of sequence reads to graphs[20,22,26–28], Paten *et al*.[25] provide a recent survey of current efforts. However, there is no method that combines these operations and uses the resulting alignments to update the graph with novel variation for the purpose of variant calling[13].

Here we present Graphtyper, a method and software for discovering and genotyping sequence variants in large populations using pangenome graphs. Graphtyper realigns all

sequence reads of a genomic region, including unaligned and clipped sequences, to a variation-aware graph (Fig. 1a ii). Concomitantly, it aligns sequence reads and genotypes sequence variants present in its graph. Furthermore, Graphtyper discovers novel single nucleotide polymorphisms (SNPs) and short sequence insertion or deletion variants (indels), which can be used to update the pangenome graph (Methods).

An important benefit of Graphtyper's realignment step is to improve read alignments near indels. Figure 2a shows how Graphtyper represents three common sequence variants, a 40-bp deletion and two SNPs. Using variation-aware realignment, Graphtyper is capable of better characterization of the region's variation than previous methods, with no Mendelian errors (Fig. 2b) and no falsely reported additional sequence variants around the indel (Supplementary Table 1) due to misaligned sequence reads (Supplementary Fig. 1).

Results

Data structures and genotyping pipeline Graphtyper uses a reference sequence and optionally all known sequence variants as input to construct pangenome graphs. Sequence reads mapped to a genomic region of the reference sequence, including unaligned and trimmed reads, are realigned to the pangenome graph. Using these graph alignments, Graphtyper discovers variants within the genomic region. This process is iterated several times (Supplementary Note), i.e., a pangenome graph is constructed, indexed and aligned with sequence reads, from which novel variants are discovered and previously discovered variants are genotyped (Fig. 1b).

The underlying pangenome data structure is a directed acyclic graph (DAG) where edges connect nodes that contain a DNA sequence (Supplementary Note). Graphtyper takes as input a reference genome and a list of known variants. Each known variant is a record of a chromosomal position, a reference allele, and one or more alternative alleles. First, variant records with overlapping reference alleles are merged into a single record (Fig. 3a). Second, *allele nodes* are constructed, containing the sequence and start position of each allele of the variant records. Third, *reference nodes* are constructed between two adjacent variant records, storing the corresponding reference sequence and its start position. Finally, nodes at adjacent positions are connected. Paths in the graph alternate between *reference* and *allele* nodes and nodes that share a start position are parallel to each other. Each character in an allele node sequence is given a position equal to the first position of the node plus the character's offset from that position (Fig. 3b). Allele node positions longer than the reference allele are assigned new unique positions (z_1 and z_2 in Fig. 3b) to avoid conflicts

with the following positions. The final graph represents the reference sequence and all haplotypes in the population as paths.

Aligning sequence reads by traversing the graph is time consuming. To expedite graph alignments, the graph structure is preprocessed by creating an index that maps k-mers to their start and end positions in the reference genome and to overlapping allele nodes (if any) (Fig. 3c, Methods). Read alignment then follows the seed-and-extend paradigm (Fig. 3d-3h, Methods, and Supplementary Note).

The output of each iteration is a file in variant-call format (VCF) including both newly and previously discovered variants, which Graphtyper uses to update the graph in the next iteration (Methods).

Population-scale genotyping We compared Graphtyper to seven widely used genotyping pipelines on human chromosome 21 in a set of 691 whole-genome sequenced Icelanders (Table 1). Of these, 404 individuals were contained in 230 trios (parent-offspring trio families). The genotypers used were Genome Analysis ToolKit UnifiedGenotyper (GATK UG)[29], GATK-Lite UnifiedGenotyper (UGLite), GATK HaplotypeCaller (HC), GATK HC GVCF joint genotyping (HC joint), Samtools[30], Platypus[18], and FreeBayes[31] (Supplementary Note). To ensure a fair comparison between genotyping pipelines, no known sequence variants were given to Graphtyper as input and all pipelines were given the same BAM files and reference sequence (GRCh38).

Our results show that GATK UG, Graphtyper and Samtools all had comparable compute times and completed the genotyping in between 576 and 594 hours (Table 1). The other five genotypers required considerably greater compute times (1,030-12,964 hours).

We assessed the raw output of all eight genotyping pipelines to compare them independent of filtering technique and to include analysis of all germline variation, somatic variation, and wrongfully reported variation due to sequencing or alignment errors. Compared to other genotypers, Graphtyper called a large number of SNPs (406,087) with a reasonably high ratio of transitions (Ti) to transversions (Tv) (1.49). We observed that all eight genotypers had a large excess of alternative alleles with a transmission rate below 50% (Supplementary Fig. 2). We also observed higher Ti/Tv ratios among alleles with higher transmission rates (Supplementary Fig. 3). Motivated by these realizations, we estimated the number of germline alternative alleles based on the transmission rate of the alternative alleles in the 230 trios (Methods). Graphtyper detected the largest number of estimated germline alternative alleles in the trios (267,057), followed by GATK UGLite (264,753) and GATK UG (264,447) (Table 1).

We found 105,302 SNPs and 7,694 indels that were called by all eight genotypers and have been reported as common (minor allele frequency > 1% in any population) in dbSNP build 149. In the 230 trios, Graphtyper called these sequence variants with a mean alternative allele transmission rate of 49.98%, very close to the expected 50%. Graphtyper had the highest Mendelian accuracy (99.52%) and the lowest number of missing genotype calls (0.201%) (Table 1). We also compared SNP calls to 3,284,976 in-house microarray genotypes (Methods). For each call set, we measured array site recall rate and precision at recalled sites, and counted how many genotypes and alleles were concordantly inferred over all array sites. If an array site was not recalled we interpreted it as a homozygous reference call ("0/0").

From our comparison of genotypers, we concluded that Graphtyper and GATK UG were the two best genotypers for population-scale genotyping in terms of performance, accuracy and sensitivity. We assessed a call set of highly confident Graphtyper sequence variants using our own filtering criteria and filtered the GATK call sets (UG, HC and HC joint) using their available 'best practices' filtering criteria (Supplementary Note). Graphtyper achieved a substantially lower estimate of false discovery rate (FDR) (2.19%) than the other call sets (10.26-31.22%), but also had a lower estimated number of germline alternative alleles (200,984) than the other call sets (214,801-240,020) (Supplementary Table 2).

We measured scalability by genotyping chromosome 21 on a dataset of 15,220 Icelanders[32,33], in which there are 1,729 trios (3,863 unique individuals). Our results show that Graphtyper scales much better than GATK UG (Fig. 4), with GATK UG using approximately 2.5x more time for computations than Graphtyper (Table 2). The compute time used by Graphtyper per sample did not increase substantially when the sample size increased from 691 to 15,220 (changed from 0.842 hr/sample to 0.867 hr/sample), while GATK UG used 2.65x more compute time per sample (changed from 0.834 hr/sample to 2.206 hr/sample).

Based on the transmission of alternative alleles the 1,729 trios, we observed that the FDR increased for Graphtyper and GATK UG compared to the 230 trio dataset in both raw and filtered call sets. We estimated that Graphtyper detected more germline alternative alleles (308,204) with a significantly lower FDR (8.89%) than GATK UG (305,404 and 22.62%, respectively) in the filtered call sets (Table 2).

Single sample genotyping We assessed the single sample genotyping performance of Graphtyper on a well-studied parent-offspring trio (NA12878, NA12891 and NA12892).

Whole-genome sequence data (50x 101-bp paired-end Illumina HiSeq 2000) of these samples are publicly available through the Platinum Genome project[34]. We genotyped each sample independently using the same genotyping pipelines as in our population-scale experiment. We ran Graphtyper with and without initializing its graph structure with publicly available common (minor allele frequency > 1% in any population) sequence variants (dbSNP build 150). Our experiments showed that GATK does not benefit from incorporating known dbSNP variants as part of its input (Supplementary Note).

We assessed sequence variant call sets of the offspring (NA12878) by comparing it to the set of publicly available high-confidence variant calls[34] to measure variant recall rate and precision. Based on the genotyping of the parents (NA12891 and NA12892), we estimated FDR and the number of transmitted germline alternative alleles in the trio (Methods).

Our results show that even without the knowledge of known variation, Graphtyper has a considerably higher recall rate (98.14%) than the other genotypers (90.24-95.91%), high precision (99.774%), and overall the highest number of validated calls (4,081,193) (Table 3). Incorporating common dbSNP variants increased Graphtyper's recall rate (to 98.46%), in particular at non-SNP sites where it increased from 91.23% to 93.38%. Consistent with its measured high recall rate, we also estimated that Graphtyper called the highest number of germline alternative alleles in the trio (5,991,012 and 5,874,556 with and without dbSNP variants, respectively), substantially more than the other genotypers (5,190,838-5,562,776). However, Graphtyper had the longest compute time (154.1 hours) as the time of constructing and indexing a graph is relatively long for only a single sample.

We also filtered the Graphtyper call sets (Supplementary Note) and compared it with GATK's call sets filtered according their 'best practices' guidelines. After filtering, Graphtyper's recall rate was reduced to 96.47% and its estimated FDR reduced from 6.06% to 4.69% (Table 3).

28,075 Icelandic whole-genome samples We used Graphtyper to genotype the autosomes and chromosome X of 28,075 whole-genome sequenced Icelandic samples. The samples have a mean sequencing depth of 35.3x (s.d. 7.9x; range 2-200x) stored in a total of 2.12 PB of BAM files. The overall compute time for genotyping was 97,917 CPU days or 83.7 CPU hours per sample on average. Graphtyper genotyped 89.4 million sequence variants: 1.1 million complex variants, 6.4 million indels, and 81.9 million SNPs with a Ti/Tv ratio of 1.04.

The compute time of genotyping chromosome 21 in 28,075 Icelandic samples was 27,853 CPU hours or 0.99 CPU hours per sample on average. Compared to Graphtyper's chromosome 21 genotyping of 691 samples, the sample size 40-folded, the number of sequence variants increased by 220%, but the compute time per sample only increased by 17.6%.

HLA typing The IPD-IMGT/HLA database[8] contains known HLA allele sequences identified with a field (usually two digits) hierarchical colon separated identifier. The first field denotes the HLA allele family, the second field denotes the subtype within the family, the third field denotes groups with synonymous substitutions within the subtype, and the fourth field denotes allele differences in non-coding regions.

Based on known HLA allele sequences, we created graphs for six important HLA genes: *HLA-A*, *HLA-B*, *HLA-C*, *HLA-DRB1*, *HLA-DQA1*, and *HLA-DQB1* (Methods). Using these graphs, we were able to HLA type the same dataset of 28,075 Icelanders in a single genotyping-only

iteration. Our results show high diversity of HLA allele families in the Icelandic population (Supplementary Table 3).

The total compute time of the HLA genotyping of the six genes was 2,609 hours, or 5.6 minutes per sample. The compute time of Graphtyper for the HLA region was orders of magnitudes lower than other genotypers[14,35] (Supplementary Note). Previously, deCODE genetics laboratory performed HLA typing of the six genes with a PCR based method at 2-digit ($n = 647$) and 4-digit ($n = 368$) resolutions. These previous typings are in good concordance (95.1-100% 2-digit; 91.6-100% 4-digit) with Graphtyper's HLA genotype calls (Table 4). Upon manual inspection, we concluded that a large fraction of the discrepancy between the two methods are most likely explained by sample mix-up (Supplementary Note).

Discussion

Previous genotypers use read alignments to linear reference genomes, which limits their performance in polymorphic regions. To better characterize sequence diversity we implemented a novel variation-aware data structure and developed efficient algorithms in a software called Graphtyper. Graphtyper locally realigns sequence reads from a genomic region to a pangenome graph, and concomitantly genotypes sequence variants. We show that combining these two steps is not only practical, but improves sensitivity and is more scalable than other genotyping methods. Our results show that Graphtyper has the highest Mendelian accuracy at previously reported variant sites among the genotypers in our comparison.

Graphtyper can use known variants as input, further improving sensitivity. When using dbSNP as part of the input, Graphtyper fails to recall only 0.73% of SNP variants in the Platinum genome dataset, a rate 5 times lower than the 3.61% missed by the best competitor. Additionally, the graph representation allows us to construct graphs with known sequence variation in the HLA region and accurately genotype known alleles of six HLA genes. Our HLA types are in good concordance to previously PCR verified HLA types. Graphtyper's ability to determine genotype calls for more sequence variants, including those that have complex representation, such as the HLA region may help geneticists in characterizing genomes and their impact. Despite these successes, additional work is required, for example, currently Graphtyper cannot call structural variants.

All of the experiments presented here were run on a high-performance computing cluster, but none of the pipelines are limited to such environments. Even with a large computing cluster, the computational requirements are so large that it is infeasible to effectively apply

them to population-sized data sets. For large datasets, the computational requirements of Graphtyper are significantly lower than previous methods, requiring full utilization of a 10,000 core computer cluster for 10 days to genotype the 28,075 whole-genome sequenced Icelanders, compared to an estimated minimum of 25 days for GATK UG.

It is important to note that our current pipeline still relies on the linear reference sequence and BWA for global read alignments in order to assign reads to a region. To completely remove bias towards the reference genome and fully utilize the promise of pangenome analysis requires developing robust methods for graph alignment, some of which are on the horizon[25,26,28]; one such notable project is vg (https://github.com/vgteam/vg). Our results further show the importance of replacing the linear reference with richer data structures to improve our understanding of how sequence diversity impacts diseases and other phenotypes.

Acknowledgments We are grateful to our colleagues from deCODE genetics / Amgen Inc. for their contributions. We also wish to thank all research participants who provided biological samples to deCODE genetics.

Author contributions HPE implemented the Graphtyper software. HPE, PM and BVH designed the Graphtyper algorithm. HPE, DFG, PM, BVH and KS designed the experiments. HPE, EH, GM and FZ ran all evaluated genotypers. HPE, HJ and KEH analyzed the call sets. Aslaug Jonasdottir, Adalbjorg Jonasdottir and IJ were responsible for PCR validation. HJ and SK contributed software for the project. HPE wrote the initial version of the manuscript, HJ, SK, BK, PM, BVH and KS contributed to subsequent versions. All authors reviewed and approved the final version of the manuscript.

Competing financial interests All authors are employees of deCODE Genetics/Amgen, Inc.

References

1. Francioli, L. C. *et al.* Whole-genome sequence variation, population structure and demographic history of the Dutch population. *Nat. Genet.* **46,** 818–825 (2014).

2. Gudbjartsson, D. F. *et al.* Large-scale whole-genome sequencing of the Icelandic population. *Nat. Genet.* **47,** 435–444 (2015).

3. Sudmant, P. H. *et al.* An integrated map of structural variation in 2,504 human genomes. *Nature* **526,** 75–81 (2015).

4. Auton, A. *et al.* A global reference for human genetic variation. *Nature* **526,** 68–74 (2015).

5. Seo, J.-S. *et al.* De novo assembly and phasing of a Korean human genome. *Nature* **538,** 243–247 (2016).

6. Wang, J. *et al.* The diploid genome sequence of an Asian individual. *Nature* **456,** 60–65 (2008).

7. Tiwari, J. L. & Terasaki, P. I. *HLA and Disease Associations*. (Springer New York, 1985).

8. Robinson, J. *et al.* The IPD and IMGT/HLA database: Allele variant databases. *Nucleic Acids Res.* **43,** D423–D431 (2015).

9. Dilthey, A., Cox, C., Iqbal, Z., Nelson, M. R. & McVean, G. Improved genome inference in the MHC using a population reference graph. *Nat. Genet.* **47,** 682–688 (2015).

10. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25,** 1754–1760 (2009).

11. DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43,** 491–498 (2011).

12. Shao, H. *et al.* A population model for genotyping indels from next-generation sequence data. *Nucleic Acids Res.* **41,** e46–e46 (2013).

13. Computational Pan-Genomics Consortium, T. C. P.-G. Computational pan-genomics: status, promises and challenges. *Brief. Bioinform.* bbw089 (2016). doi:10.1093/bib/bbw089

14. Dilthey, A. T. *et al.* High-Accuracy HLA Type Inference from Whole-Genome Sequencing Data Using Population Reference Graphs. *PLOS Comput. Biol.* **12,** e1005151 (2016).

15. Paten, B., Novak, A. & Haussler, D. Mapping to a Reference Genome Structure. (2014).

16. Huang, L., Popic, V. & Batzoglou, S. Short read alignment with populations of genomes. *Bioinformatics* **29,** i361–i370 (2013).

17. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. & McVean, G. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat. Genet.* **44,** 226–232 (2012).

18. Rimmer, A. *et al.* Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat. Genet.* **46,** 1–9 (2014).

19. Li, R. *et al.* Building the sequence map of the human pan-genome. *Nat. Biotechnol.* **28,** 57–63 (2010).

20. Sirén, J., Välimäki, N. & Mäkinen, V. Indexing graphs for path queries with applications in genome research. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* **11,** 375–388 (2014).

21. Schneeberger, K. *et al.* Simultaneous alignment of short reads against multiple genomes. *Genome Biol.* **10,** R98 (2009).

22. Zhao, M., Lee, W. P., Garrison, E. P. & Marth, G. T. SSW library: An SIMD Smith-Waterman C/C++ library for use in genomic applications. *PLoS One* **8,** (2013).

23. Sirén, J. Indexing Variation Graphs. (2016). doi:10.1137/1.9781611974768.2

24. Church, D. M. *et al.* Extending reference assembly models. *Genome Biol.* **16,** 13 (2015).

25. Paten, B., Novak, A. M., Eizenga, J. M. & Garrison, E. Genome graphs and the evolution of genome inference. *Genome Res.* **27,** 665–676 (2017).

26. Sirén, J. Indexing Variation Graphs. in *2017 Proceedings of the Ninteenth Workshop on Algorithm Engineering and Experiments (ALENEX)* 13–27 (Society for Industrial and Applied Mathematics, 2017). doi:10.1137/1.9781611974768.2

27. Kehr, B., Trappe, K., Holtgrewe, M. & Reinert, K. Genome alignment with graph data structures: a comparison. *BMC Bioinformatics* **15,** 99 (2014).

28. Maciuca, S., Elias, C. D. O., McVean, G. & Iqbal, Z. A natural encoding of genetic variation in a burrows-wheeler transform to enable mapping and genome inference. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9838,** 222–233 (2016).

29. McKenna, A. *et al.* The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20,** 1297–1303 (2010).

30. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).

31. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. *arXiv Prepr. arXiv1207.3907* 9 (2012). doi:arXiv:1207.3907 [q-bio.GN]

32. Jónsson, H. *et al.* Parental influence on human germline de novo mutations in 1,548 trios from Iceland. *Nature* **Accepted,** (2017).

33. Jónsson, H. *et al.* Whole genome characterization of sequence diversity of 15,220 Icelanders. *Sci. data* **Accepted,** (2017).

34. Eberle, M. A. *et al.* A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27,** 157–164 (2017).

35. Szolek, A. *et al.* OptiType: precision HLA typing from next-generation sequencing data. *Bioinformatics* **30,** 3310–6 (2014).

Figures

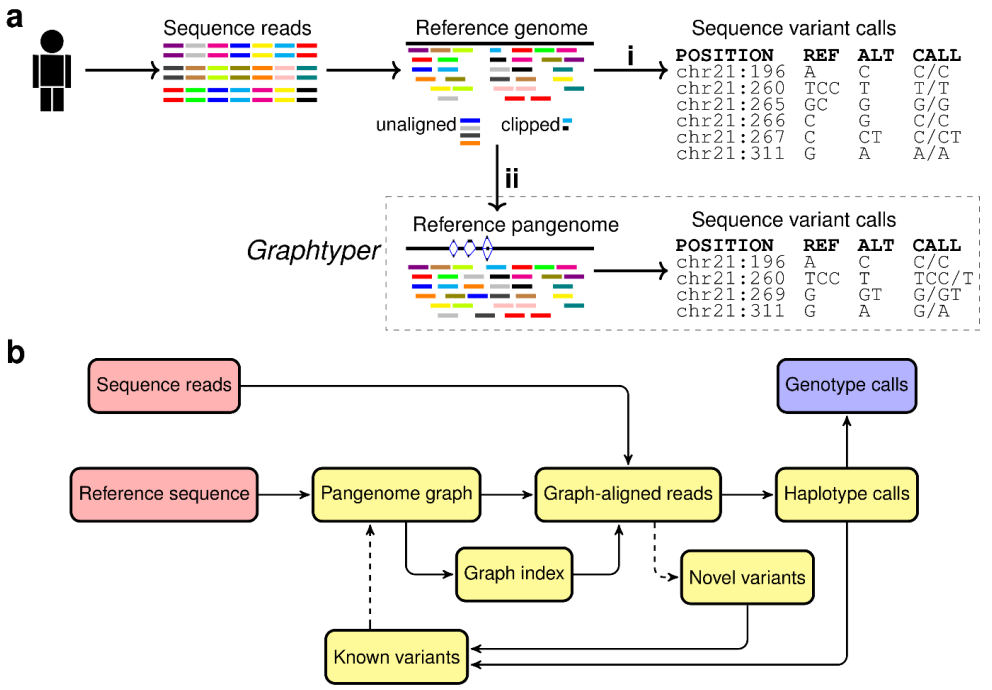


Figure 1: Genotyping pipeline designs. ***(a)*** *Overview of two genotyping pipeline designs.* ***(i)*** *A commonly used genotyping pipeline, where sequence reads are aligned to a reference genome sequence and sequence variant are called from discordances between the reads and the reference.* ***(ii)*** *Graphtyper's genotyping pipeline. Sequence reads are realigned to a variant-aware pangenome graph and variants are called based on which path the reads align to.* ***(b)*** *Graphtyper's iterative genotyping process. Dashed paths are optional. As input, Graphtyper requires a reference genome sequence and sequence reads (red) and outputs genotype calls (blue) of variants.*

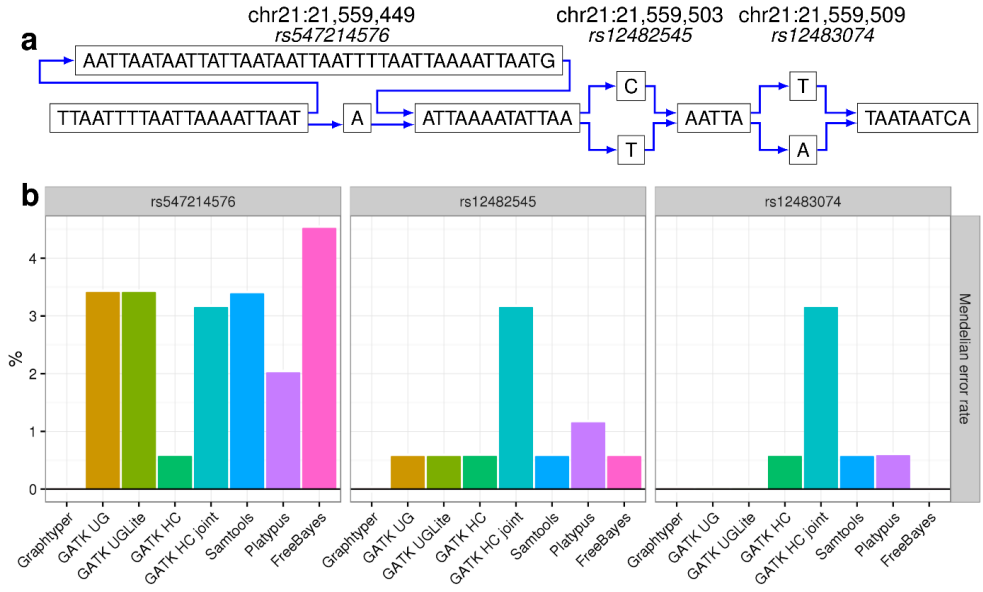


Figure 2: Importance of variation-aware alignment. ***(a)*** *The genomic region chr21:21,559,430-21,559,518 (GRCh38) and three previously reported sequence variants represented with a pangenome graph.* ***(b)*** *Mendelian error rates of the three previously reported sequence variants called by eight genotypers. The Mendelian error rate is measured in 230 Icelandic parent-offspring trios.*

a

Reference sequence: ACCTCCAGACGTTTAGGGACCCCATTGAGTG

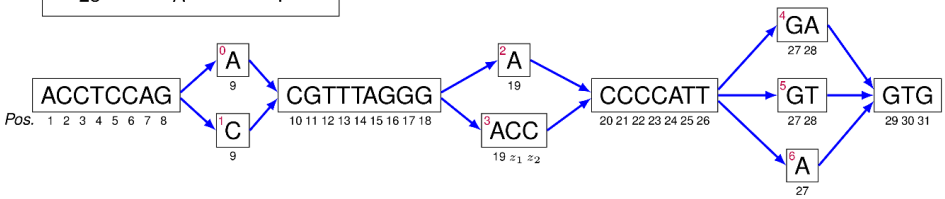
Known variants

Position	Reference	Alternative
9	A	C
19	A	ACC
27	GA	A
28	A	T

Known variants after merge

Position	Reference	Alternative
9	A	C
19	A	ACC
27	GA	GT,A

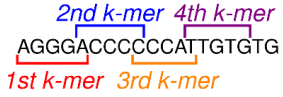
b



c

5-mer	Start pos.	End pos.	Variant ID	Start pos.	End pos.	Variant ID
ACCCC	19	23	2	19	21	3
ACCTC	1	5	NA			
ACGTT	9	13	0			
AGACG	7	11	0			
AGCCG	7	11	1			
AGGGA	15	19	2	15	19	3
...						
CCCAT	21	25	NA			
CCCCA	20	24	NA			
CCCCC	z_1	22	3	z_2	23	3
...						
GGACC	17	z_2	3	17	21	2
GGGAC	16	z_1	3	16	20	2
...						
TTGTG	25	29	5			
...						

d



e

5-mer	Start pos.	End pos.	Variant ID	Start pos.	End pos.	Variant ID
AGGGA	15	19	2	15	19	3
ACCCC	19	23	2	19	21	3
CCCAT	21	25	NA			
TTGTG	25	29	5			

f

AGGGA	ACCCC	CCCAT	TTGTG
AGGGC	ACCCA	CCCAA	TTGTA
AGGGG	ACCCG	CCCAC	TTGTC
AGGGT	ACCCT	CCCAG	TTGTT
AGGAA	ACCAC	CCCCT	TTGAG
AGGCA	ACCGC	CCCGT	TTGCG
AGGTA	ACCTC	CCCTT	TTGGG
AGAGA	ACACC	CCAAT	TTATG
AGCGA	ACGCC	CCGAT	TTCTG
AGTGA	ACTCC	CCTAT	TTTTG
AAGGA	AACCC	CACAT	TAGTG
ACGGA	AGCCC	CGCAT	TCGTG
ATGGA	ATCCC	CTCAT	TGGTG
CGGGA	CCCCC	ACCAT	ATGTG
GGGGA	GCCCC	GCCAT	CTGTG
TGGGA	TCCCC	TCCAT	GTGTG

g

Seed	Start pos.	End pos.	Variant IDs
AGGGACCCCCCATTGTG	15	29	3,5
AGGGACCCC	15	23	2

h

Extended longest seed	Start pos.	End pos.	Variant IDs
AGGGACCCCCCATTGTGTG	15	31	3,5

Figure 3: Graphtyper's sequence alignment algorithm. ***(a)*** *An example reference sequence and its known variation. All overlapping variants are merged.* ***(b)*** *Constructed pangenome reference graph. We draw the path of the reference sequence as the topmost path.* ***(c)*** *The index data structure with k = 5. 5-mers in the graph are mapped to a list of its start position, end position, and a variant ID which it overlaps, if any.* ***(d)*** *Four k-mers are extracted from a sequence read. Each k-mer*

overlaps its neighbor k-mer by one character. ***(e)*** *An example look-up of the k-mers from the index data structure from c).* ***(f)*** *All extracted k-mers with a single substitution.* ***(g)*** *Seeds are generated from matches in the index look-up.* ***(h)*** *Final graph alignment after extending the longest seed.*

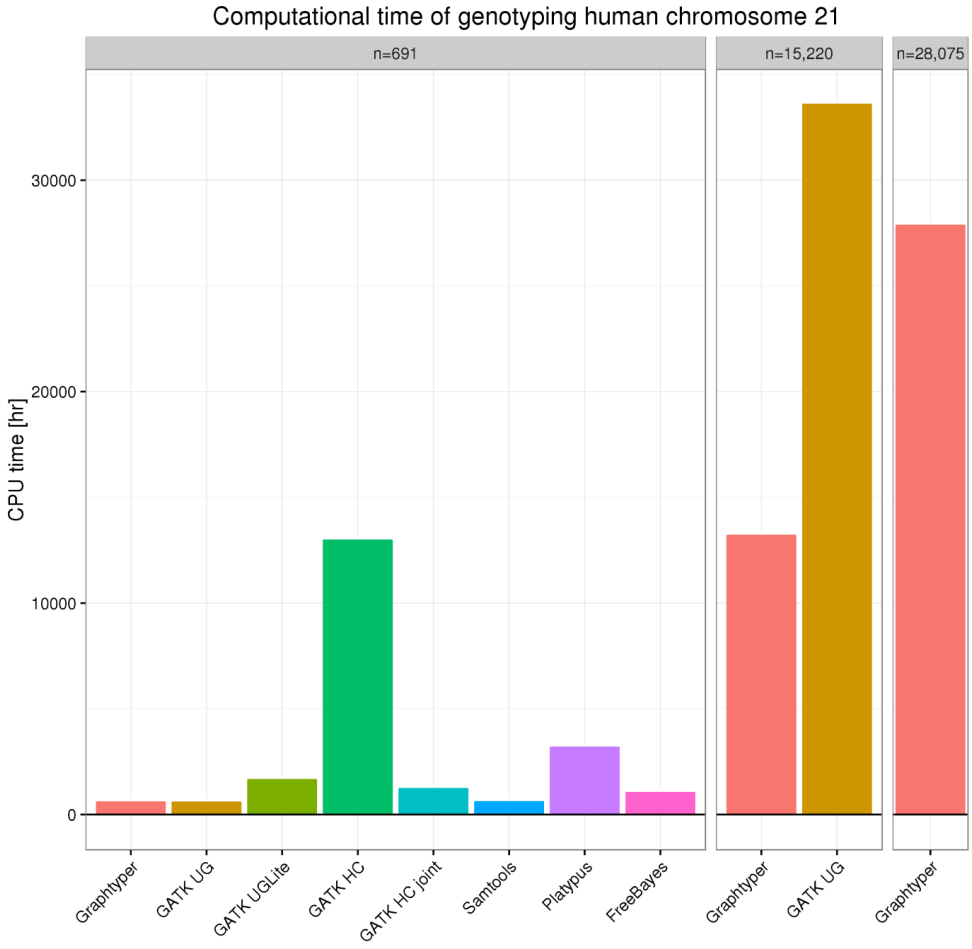


Figure 4: Genotyping time summary. Compute times required to genotype chromosome 21 on three whole-genome sequence datasets. All genotyping pipelines were run once.

Tables

Table 1: Raw sequence variant calls comparison of 691 whole-genome sequenced Icelanders of chromosome 21.

Genotyping pipeline	Graphtyper	GATK UG	GATK UGLite	GATK HC	GATK HCjoint	Samtools	Platypus	FreeBayes
Sequence variant records	453,288	451,131	451,415	311,731	418,949	411,907	424,000	596,499
SNPs	406,087	397,821	397,890	267,949	352,293	336,544	301,066	562,319
Transitions/Transversions	1.49	1.46	1.46	**1.75**	1.56	1.5	1.38	0.7
Indels	47,866	53,310	53,525	46,779	73,934	75,363	110,347	33,086
MNPs	1,002	0	0	0	0	0	26,086	21,044
Complex	3,682	0	0	0	34,592	0	0	4,532
Common (dbSNP b149)	157,288	**158,700**	158,590	153,543	158,411	157,998	156,280	136,882
SNPs	145,143	145,723	**145,724**	140,533	144,858	145,135	142,417	126,653
Indels	12,145	12,977	12,866	13,010	13,553	12,863	**13,863**	10,229
Alternative alleles called in trios	454,157	447,144	450,241	312,275	435,511	392,960	408,648	448,429
$Germline_{estimated}$	**267,057**	264,447	264,753	237,978	254,427	255,630	228,646	200,776
$FDR_{estimated}$	41.20%	40.86%	41.20%	**23.79%**	41.58%	34.95%	44.05%	55.23%
SNPs	371,214	366,068	366,019	243,815	307,024	295,707	255,775	364,942
$Germline_{estimated}$	**232,256**	227,858	227,872	206,084	216,448	215,042	183,375	172,226
Non-SNPs	82,943	81,076	84,222	68,460	128,487	97,253	152,873	83,487
$Germline_{estimated}$	34,801	36,589	36,881	31,894	37,979	40,588	**45,271**	28,550
Common dbSNP calls								
Mean alt. transmission rate	49.98%	50.08%	50.08%	50.01%	50.01%	50.11%	49.47%	50.17%
Mean missing call rate in trios	**0.20%**	0.29%	0.29%	0.33%	0.25%	0.38%	0.45%	0.26%
Mendelian accuracy	**99.52%**	99.48%	99.48%	99.37%	99.41%	99.38%	99.11%	99.44%
Microarray SNP comparison								
Correctly inferred genotypes	3,267,641	3,273,959	3,273,959	3,270,243	3,270,590	**3,274,628**	3,177,098	2,967,527
Correctly inferred alleles	6,547,170	6,555,670	6,555,670	6,550,301	6,550,875	**6,557,029**	6,426,358	6,127,836
Site recall rate	97.06%	97.33%	97.33%	97.09%	97.22%	**97.43%**	93.02%	80.38%
Precision at recalled sites	99.79%	99.80%	99.80%	99.78%	99.76%	99.78%	99.20%	**99.90%**
Only ref/ref array calls	99.92%	99.92%	99.92%	99.93%	99.93%	99.93%	99.90%	**99.96%**
Only ref/alt array calls	99.65%	99.63%	99.63%	99.54%	99.52%	99.58%	99.01%	**99.83%**
Only alt/alt array calls	99.71%	99.81%	99.81%	99.80%	99.74%	99.76%	97.94%	**99.85%**
CPU time [hr]	582	**576**	1,640	12,964	1,216 (87*)	594	3,173	1,030
Time per sample [hr]	0.842	**0.834**	2.373	18.761	1.76 (0.13*)	0.86	4.592	1.491
Mean memory [GB]	10.68	50.17	40.55	65.22	51.98	**1.97**	6.31	6.77
Maximum memory [GB]	45.4	52.72	45.86	307.47	53.58	**2.69**	50.15	196.03

**CPU time of the joint calling step.*

Table 2: Comparison of Graphtyper and GATK UG genotyping chromosome 21 of 15,220 sequenced Icelanders.

	Raw		Filtered	
Genotyping pipeline	Graphtyper	GATK UG	Graphtyper	GATK UG
Sequence variant records	1,101,540	1,160,333	473,813	493,620
SNPs	1,024,677	1,035,206	437,844	423,407
Transitions/Transversions	**1.14**	1.06	2.24	2.27
Indels	81,848	125,127	36,086	70,213
MNPs	3,487	0	133	0
Complex	10,707	0	888	0
Alternative alleles called in trios	979,451	1,032,839	338,266	394,679
$Germline_{estimated}$	383,998	**397,283**	**308,204**	305,404
$FDR_{estimated}$	**60.79%**	61.53%	**8.89%**	22.62%
SNPs	821,098	850,761	304,881	294,004
Transitions/Transversions	**1.01**	0.92	2.18	2.19
$Germline_{estimated}$	340,313	**349,878**	**281,972**	264,441
$FDR_{estimated}$	**58.55%**	58.87%	**7.51%**	10.06%
Non-SNPs	158,353	182,078	33,385	100,675
$Germline_{estimated}$	43,685	**47,405**	26,232	**40,963**
$FDR_{estimated}$	**72.41%**	73.96%	**21.43%**	59.31%
CPU time [hr]	**13,192**	33,573	-	-
Time per sample [hr]	**0.867**	2.206	-	-

Table 3: Comparison of whole-genome sequence variant calls of NA12878. Graphtyper was run with and without given the knowledge of common dbSNP variation.

	No sequence variants given										Common dbSNP given	
	Raw							Filtered			Raw	Filtered
Genotyping pipeline	Graphtyper	GATK UG	GATK UGLite	GATK HC	Samtools	Platypus	FreeBayes	Graphtyper	GATK UG	GATK HC	Graphtyper	Graphtyper
SNPs	4,210,841	3,913,454	3,912,894	3,774,031	3,729,409	3,511,646	3,760,288	3,821,418	3,585,462	3,569,701	4,230,056	3,817,459
Transitions/Transversions	1.91	1.97	1.97	1.99	2.02	2.02	1.98	1.99	2.04	2.04	1.9	1.99
Indels	726,382	649,301	649,477	781,960	735,279	823,257	617,530	703,251	646,057	771,134	761,794	730,566
MNPs	1,146	0	0	0	0	176,269	96,809	940	0	0	1,199	974
Complex	7,538	0	0	0	0	0	35,463	6,625	0	0	7,626	6,693
Recalled platinum variants	**4,090,418**	3,967,739	3,967,654	3,997,455	3,874,091	3,760,978	3,813,506	4,020,670	3,862,484	3,918,216	4,103,693	4,030,504
Recall rate	**98.14%**	95.20%	95.20%	95.91%	92.95%	90.24%	91.50%	96.47%	92.67%	94.01%	98.46%	96.70%
Validated variant calls	**4,081,193**	3,963,186	3,963,134	3,994,476	3,861,985	3,757,577	3,798,996	4,011,769	3,857,999	3,915,296	4,094,264	4,021,641
Precision	99.774%	99.885%	99.886%	**99.925%**	99.688%	99.910%	99.620%	99.779%	99.884%	**99.925%**	99.770%	99.780%
Validated SNP calls	**3,567,543**	3,465,168	3,465,145	3,457,324	3,422,248	3,221,031	3,327,170	**3,502,636**	3,360,971	3,380,200	3,568,374	3,501,379
Recall rate	**99.24%**	96.39%	96.39%	96.17%	95.20%	89.60%	92.55%	**97.43%**	93.49%	94.02%	99.27%	97.40%
Precision	99.990%	99.991%	99.991%	**99.998%**	99.993%	99.996%	**99.998%**	99.992%	99.993%	**99.998%**	99.986%	99.990%
Validated non-SNP calls	513,650	498,018	497,989	**537,152**	439,737	536,546	471,826	509,133	497,028	**535,096**	525,890	520,262
Recall rate	91.23%	87.70%	87.69%	**94.29%**	78.85%	94.25%	84.90%	90.40%	87.52%	**93.93%**	93.38%	92.33%
Precision	98.304%	99.153%	99.159%	**99.464%**	97.371%	99.393%	97.032%	98.333%	99.154%	**99.469%**	98.330%	98.389%
Peak memory usage [GB]	7.68	43.97	40.48	44	**1.35**	3.93	2.23	-	-	-	9.15	-
CPU time [hr]	154.1	31.1	41.7	71	35.2	**9.4**	22.3	-	-	-	166.5	-
Alt. alleles called in trio	6,253,839	5,754,093	5,757,400	5,736,575	5,439,047	5,826,828	5,596,394	5,529,778	5,272,137	5,434,920	6,374,281	5,589,820
$FDR_{estimated}$	6.06%	3.34%	3.38%	**3.32%**	4.56%	4.90%	4.67%	4.69%	**2.62%**	2.86%	6.01%	4.57%
$Germline_{estimated}$	**5,874,556**	5,562,132	5,562,776	5,546,352	5,190,838	5,541,586	5,335,096	5,270,514	5,133,770	**5,279,402**	5,991,012	5,334,150
SNP alt. alleles	5,322,813	4,948,488	4,948,129	4,684,879	4,554,216	4,350,270	4,662,174	4,642,251	4,473,460	4,405,919	5,366,101	4,643,158
$FDR_{estimated}$	4.69%	2.55%	2.55%	2.16%	**1.61%**	2.61%	3.63%	2.99%	1.65%	**1.60%**	4.68%	2.94%
$Germline_{estimated}$	**5,073,098**	4,822,380	4,821,792	4,583,794	4,480,936	4,236,524	4,493,068	**4,503,294**	4,399,652	4,335,432	5,115,034	4,506,482
Non-SNP alt. alleles	931,026	805,605	809,271	1,051,696	884,831	1,476,558	934,220	887,527	798,677	1,029,001	1,008,180	946,662
$FDR_{estimated}$	13.92%	**8.17%**	8.44%	8.48%	19.77%	11.61%	9.87%	13.56%	**8.08%**	8.26%	13.11%	12.57%
$Germline_{estimated}$	801,458	739,752	740,984	962,558	709,902	**1,305,062**	842,028	767,220	734,118	**943,970**	875,978	827,668

Table 4: Comparison of Graphtyper's HLA typings to PCR verified HLA types.

		4 digit resolution				2 digit resolution			
HLA gene	*n*	Correct	1 error	2 errors	Accuracy	Correct	1 error	2 errors	Accuracy
HLA-A	54	52	2	0	98.15%	52	2	0	98.15%
HLA-B	332	-	-	-	-	314	15	3	96.84%
HLA-C	315	-	-	-	-	290	19	6	95.08%
HLA-DQA1	42	42	0	0	100.00%	42	0	0	100.00%
HLA-DQB1	82	80	2	0	98.78%	81	1	0	99.39%
HLA-DRB1	190	163	22	5	91.58%	189	1	0	99.74%

Online Methods

Icelandic DNA data The Icelandic samples were whole-genome sequenced at deCODE genetics[2,32,33] using Illumina HiSeq and HiSeqX sequencing machines[36] and aligned to the GRCh38 human reference genome using the BWA MEM algorithm[10]. All sequenced individuals were also SNP chip typed using Illumina Human Hap or Omni chip arrays. DNA was isolated from both blood and buccal samples.

All participating subjects signed informed consent. Personal identities of the participants and biological samples were encrypted by a third party system approved and monitored by the Data Protection Authority. The National Bioethics Committee and the Data Protection Authority in Iceland approved these studies.

Sequence read alignment In Graphtyper, sequence variation of small genomic regions (we used 50 kbp regions in this study) are represented with a pangenome graph structure. Sequence reads are realigned to the graph of a region if BWA reported them to be in the same region. First, Graphtyper extracts a set of k-mers from the sequence read, which overlap by one DNA base in the read (Fig. 3d), and determines if they are present in the graph using an index structure (Fig. 3e). Seeds are generated from matches in the index look-up. If the alignments of two adjacent k-mers overlap by exactly one base, Graphtyper joins their matches into larger seeds (Fig. 3g). The longest seeds are then extended (Fig. 3h) by finding a path in the graph with the fewest mismatches using a breadth first search algorithm. If no seeds are extended with 12 mismatches or fewer, Graphtyper again extracts a set of k-mers from the read which overlap by one base in a read, but now also k-mers with one mismatch are included (Fig. 3f). The process is applied both to a read and its reverse

complement. If both orientations of a read align to the graph, Graphtyper selects the longer alignment or, if they are equally long, the alignment with fewer mismatches.

Novel variant discovery Graphtyper post-processes graph alignments to discover novel small sequence variants. Novel sequence variants are classified as SNPs, indels (up to approx. 50 bp), and complex variation (e.g. multiple nucleotide polymorphisms and microsatellites). For each read uniquely aligned to the graph, Graphtyper starts by determining the position in the reference genome of its first and last aligned position in the graph and extracts the reference sequence between these two positions. Then on each side of the reference sequence, the read is extended by an additional 50 bases plus the number of soft clipped bases on the given side. The read is then locally aligned to the extracted reference sequence using a banded semi-global version of Gotoh's algorithm (Supplementary Fig. 4a). Differences in the local alignments are treated as observations of variants (Supplementary Fig. 4b).

Once all reads have been processed, Graphtyper outputs sequence variants where there exists a sample that has at least 5 observations of an alternative allele and its frequency is at least 20% (default values).

Genotyping Graphtyper genotype calls sequence variants in the graph by treating the graph alignments as independent observations of each sample's underlying genotype. It genotypes sequence variants in the graph by considering nearby variants together. Given graph-aligned sequence reads of a population, the likelihood that the reads were sampled from a pair of haplotypes is estimated for each sample and the haplotypes with the highest likelihood are determined. To greatly reduce the number of haplotypes considered, all sequence variants located 5 bp or less from each other are grouped (Supplementary Fig. 5a) and each variant

group is genotyped independently. Let $H_i = \{h_{i,1}, h_{i,2}\}$ be a multiset of the unknown haplotypes of sample i in a variant group, v, and let $R_i = \{r_{i,1}, r_{i,2}, \ldots, r_{i,|R_i|}\}$ be the sample's multiset of sequence reads aligned by Graphtyper to the variant group v.

For each pair of possible haplotypes, a relative likelihood of the observed reads given the haplotypes $\mathcal{L}(R_i|H_i)$ is computed. We assume that the reads from one individual are independent of other individuals' reads. Graphtyper computes the relative likelihood as:

$$\mathcal{L}(R_i|H_i) = \prod_{r_{ij} \in R_i} L(r_{ij}|H_i) \quad (1)$$

where the relative likelihood of observing a read $r_{i,j}$ given the pair of underlying haplotypes is set as:

$$L(r_{ij}|H_i) = \begin{cases} 1 & \text{, if both } h_{i,1} \text{ and } h_{i,2} \text{ support the read.} \\ 1/2 & \text{, if exactly one of } h_{i,1} \text{ and } h_{i,2} \text{ support the read.} \\ \varepsilon_{r_{ij},H_i} & \text{, if neither } h_{i,1} \text{ nor } h_{i,2} \text{ support the read.} \end{cases} \quad (2)$$

where $\varepsilon_{r_{i,j},H_i}$ is the relative likelihood of observing an error, given the underlying haplotypes H_i and the read $r_{i,j}$. These relative likelihoods are chosen from the set $\left\{\frac{1}{2^5}, \frac{1}{2^6}, \ldots, \frac{1}{2^{13}}\right\}$ based on how similar the read is to the haplotypes H_i, the base pair quality, mapping quality of the read, and if the read is soft clipped (Supplementary Note). Restricting relative likelihoods to this set allows storing only the integer exponents, minimizing storage requirements and avoiding floating point precision problems.

As sequence variants are genotyped in groups, Graphtyper can identify the haplotypes in the population within each group (Supplementary Fig. 5b) and remove unobserved haplotypes from the graph (Supplementary Fig. 5c). In complex regions, this process can greatly reduce the number of haplotype paths in the graph.

Sequence variant quality assessment For each sequence variant, we estimated the Mendelian error rate as the fraction of incorrectly inferred offspring in trios with two homozygous parents (Supplementary Fig. 6a). We defined Mendelian inaccuracy as the estimated Mendelian error rate plus the fraction of trios with a missing genotype call, which are genotypes reported as "." or "./." in the VCF output.

While Mendelian error rate is effective for assessing common alternative alleles, the majority of them are rare and they often have no homozygous carriers. When either parent is heterozygous we cannot deterministically infer the genotype of the offspring (Supplementary Fig. 6b). For those trios we instead calculated the transmission rate of each alternative allele from a parent to its offspring. We used the difference of alternative allele transmission rates above and below 50% to estimate the false discovery rate (FDR) using:

$$FDR_{\text{estimated}} = \max\left(\frac{\#(AA_{TMR<50\%}) - \#(AA_{TMR>50\%})}{\#(AA)}, 0\right) \quad (3)$$

Here, $\#(AA)$ is the number of called alternative alleles, and $\#(AA_{TMR>50\%})$ and $\#(AA_{TMR<50\%})$ are the number of alternative alleles with a transmission rate above and below 50%, respectively. The Mendelian laws of inheritance dictate that each allele is equally likely to be transmitted from a parent to its offspring. Therefore, in a given variant call set that contains only true germline alternative alleles ($\text{FDR} = 0\%$) then we would expect $\#(AA_{TMR>50\%}) = \#(AA_{TMR<50\%})$ and $FDR_{\text{estimated}} = 0\%$. We also made the assumption that reported non-germline discovered alleles, e.g. due to sequencing errors or somatic mutations, are not transmitted. In a call set with no germline alternative alleles (FDR = 100%), we would not expect that alternative alleles are transmitted, $\#(AA_{TMR>50\%}) = 0$ and $FDR_{\text{estimated}} = 100\%$.

Based on the above assumptions, we can estimate the number of germline alternative alleles using:

$$\#(\text{Germline } AA)_{\text{estimated}} = \#(AA)(1 - FDR_{\text{estimated}}) \quad (4)$$

HLA typing pre-processing We retrieved HLA allele sequences from the IPD-IMGT/HLA database (version 3.23.0, see URLs). We extracted the differences to a VCF file that we used to create the pangenome graphs for HLA typing. A more detailed description of our HLA typing method as well as comparisons to other methods have been published in our previous work[37] and are described in Supplementary Note.

URLs IPD-IMGT/HLA (http://www.ebi.ac.uk/ipd/imgt/hla/, Github page: https://github.com/ANHIG/IMGTHLA)

Data availability Access to the raw Icelandic sequence data that support the findings of this study is available on request from KS. The data are not publicly available because of Icelandic state law.

Code availability Graphtyper is available at https://github.com/DecodeGenetics/graphtyper (GNU GPLv3 license).

Methods-only References

36. Bentley, D. R. *et al.* Accurate whole human genome sequencing using reversible terminator chemistry. *Nature* **456,** 53–59 (2008).

37. Eggertsson, H. P. Gyper: A graph-based HLA genotyper using aligned DNA sequences. (2015)

Supplementary data for "Graphtyper enables population-scale genotyping using pangenome graphs"

Hannes P. Eggertsson[1,2], Hakon Jonsson[1], Snaedis Kristmundsdottir[1,3], Eirikur Hjartarson[1], Birte Kehr[1,4], Gisli Masson[1], Florian Zink[1], Kristjan E. Hjorleifsson[1], Aslaug Jonasdottir[1], Adalbjorg Jonasdottir[1], Ingileif Jonsdottir[1,5], Daniel F. Gudbjartsson[1,2], Pall Melsted[1,2], Kari Stefansson[1,5], Bjarni V. Halldorsson[1,3]

[1] *deCODE genetics/Amgen Inc., Sturlugata 8, Reykjavik, Iceland*

[2] *School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland*

[3] *School of Science and Engineering, Reykjavik University, Reykjavik, Iceland*

[4] *Berlin Institute of Health (BIH), 10178 Berlin, Germany*

[5] *Faculty of Medicine, School of Health Sciences, University of Iceland, Reykjavik, Iceland*

Supplementary Note

Contents

1 Genotyping details

Genotyping computations were submitted to deCODE's inhouse computer cluster. In our smallest genotyping run of 691 Icelanders we ran all genotyping pipelines on a single thread but in our other runs we ran Graphtyper and GATK UG in multi-threaded mode (up to 24 threads, depending on memory usage) to better utilize system resources. All programs were run in pooled calling mode except GATK HC joint caller.

Versions and parameters of genotyping pipelines In our comparisons of genotypers we used GATK UnifiedGenotyper (UG) 2015.1, GATK-Lite UG 2.3-9 (UGLite), GATK HaplotypeCaller (HC) 2015.1, and Samtools v1.3. As for Platypus and FreeBayes, we used the latest versions of Platypus and FreeBayes as of 13th of December 2016 from their respective GitHub page. These versions include modifications from their release of Platypus v0.8.1 and FreeBayes v1.1.0.

Below are the commands used for each genotyping pipeline to call variants from input BAM files. Here, $GENOME_REF is the GRCh38 reference genome, $BAMS is a file containing a list of all BAM files, $TMP is a temporary directory on a local disk, $MAX_MEM is the maximum of allocated memory in megabytes, $REGION is target genomic region, and $OUT is the output VCF file.

Graphtyper

Graphtyper was run with one and two discovery iterations for single and multi-sample genotyping, respectively. The discovery iterations were followed by two genotyping-only iterations. The purpose of the first genotyping iteration is to "clean" complex variation in the graph by removing all unobserved haplotypes. The final iteration is to generate the final genotype calls from the clean graph.

```
graphtyper call $GRAPH --sams $BAMS --threads 1 --output $TMP/results [--
    no_new_variants]
```

The `--no_new_variants` parameter was supplied in genotyping-only iterations. In Graphtyper's single sample calling, we also added novel sequence variants more permissively using the parameters `--minimum_variant_support=3 --minimum_variant_support_ratio=0.15`. Further instruction on how to run Graphtyper are available on Graphtyper's Github page.

GATK UG

```
java -Djava.io.tmpdir='$TMP' -Xmx$MAX_MEMm -jar '$GATK_JAR' -T UnifiedGenotyper -R '
    $GENOME_REF' -o '$OUT' -U ALL -I $BAMS -glm BOTM -L \$REGION
```

GATK UGLite

```
java -Djava.io.tmpdir='$TMP' -Xmx$MAX_MEMm -jar '$GATKLITE_JAR' -T UnifiedGenotyper -
    R '\$GENOME_REF' -o '$OUT' -U ALL -I $BAMS -glm BOTM -L $REGION
```

GATK HC

```
java -Djava.io.tmpdir='$TMP' -Xmx$MAX_MEMm -jar '$GATK_JAR' -T HaplotypeCaller -R '
    $GENOME_REF' -o '$OUT' -U ALL -I $BAMS -L $REGION
```

GATK HC joint

```
java -Djava.io.tmpdir='$TMP' -Xmx$MAX_MEMm -jar '$GATK_JAR' -T CombineGVCFs -R '
    $GENOME_REF' -o '$OUT' -U ALL (--variant $GVCF)+
```

Samtools

```
samtools mpileup -u -g -f '$GENOME_REF' -r $REGION -b $BAMS | bcftools call -vmO z -o
    $OUT
```

Platypus

```
platypus callVariants --bamFiles=$BAMS --output='-' --logFileName='$ID.log' --nCPU=1
   --refFile='$GENOME_REF' --maxReads 1000000000 --bufferSize 1000 | bgzip -c > $OUT
```

FreeBayes

```
freebayes --bam-lists $BAMS --fasta-reference $GENOME_REF --use-best-n-alleles=0 --
   haplotype-length=3 | bgzip -c > $OUT
```

Sequence variant filtering Some of the calls made by Graphtyper are more likely to be false positives than others. We obtained a filtered set of variants using `vcffilter` v1.0.0-rc0, which is a part of the vcflib (`https://github.com/vcflib/vcflib`). For single sample call sets, we filtered using the following command:

```
vcffilter -f "ABHet < 0.0 | ABHet > 0.30" -f "MQ > 30" -f "QD > 6.0" <VCF>
```

where `VCF` is the VCF file to filter. For the multi sample call sets we used:

```
vcffilter -f "ABHet < 0.0 | ABHet > 0.33" -f "ABHom < 0.0 | ABHom > 0.97" -f "MaxAASR
    > 0.4" -f "MQ > 30" <VCF>
```

Graphtyper's best practices of sequence variant filtering are described on its Github page.

We filtered the GATK call sets using their best practices guidelines for SNP and indel discovery (`https://software.broadinstitute.org/gatk/best-practices/`).

2 Graphtyper data structures

Pangenome graph structure Given an alphabet Σ we define a sequence to be a string of characters $S = s_0, ..., s_{n-1}$, where $s_a \in \Sigma$ for all a. $|S| = n$ is the length of a string S. We denote a substring of S as $S_{i,j} = s_i, ..., s_{j-1}$ for any $0 \leq i, j \leq n$. If $j \leq i$ the substring is empty. All strings have a substring equal to itself, $S_{0,n} = S$. The concatenation of two strings $S^1 = s^1_0, ..., s^1_{n_1-1}$ and $S^2 = s^2_0, ..., s^2_{n_2-1}$ is denoted by $S^1S^2 = s^1_0, ..., s^1_{n_1-1}, s^2_0, ..., s^2_{n_2-1}$.

Let graph $G = (N, E)$ consist of a set of nodes $N = \{n_1, ..., n_{|N|}\}$ and edges $E \subseteq N^2$. Given $u, v \in N$, we say that $(u, v) \in E$ is an edge from node u to node v. If such an edge exists, we say that u and v are connected. Each node u is labelled with a sequence, S^u. A path in the graph is a non-empty sequence of nodes $P = u_1, ..., u_{|P|}$ where there is an edge $(u_k, u_{k+1}) \in E$ for all $k < |P|$. The sequence of a path P is $S^{u_1} \cdots S^{u_{|P|}}$. We say a graph contains the sequence S if there exists a path in the graph with sequence S.

Let R be the reference genome sequence and V be a set of sequence variants $V = \{V_1, ..., V_{|V|}\}$. A sequence variant V_k has one or more *alternative allele sequences* $S^k \in V^k$ which replace a non-empty substring R_{i_k,j_k} of the reference, called the *reference allele sequence*. The sequence variants are ordered in ascending order by their start position i_k. If two sequence variants, $V_a, V_b \in V$, replace reference substrings R_{i_a,j_a} and R_{i_b,j_b} where $i_a \leq i_b$, we say that V_a, V_b overlap if and only if $i_b < j_a$. In other words, two variant sequences overlap if they replace the same substring of R.

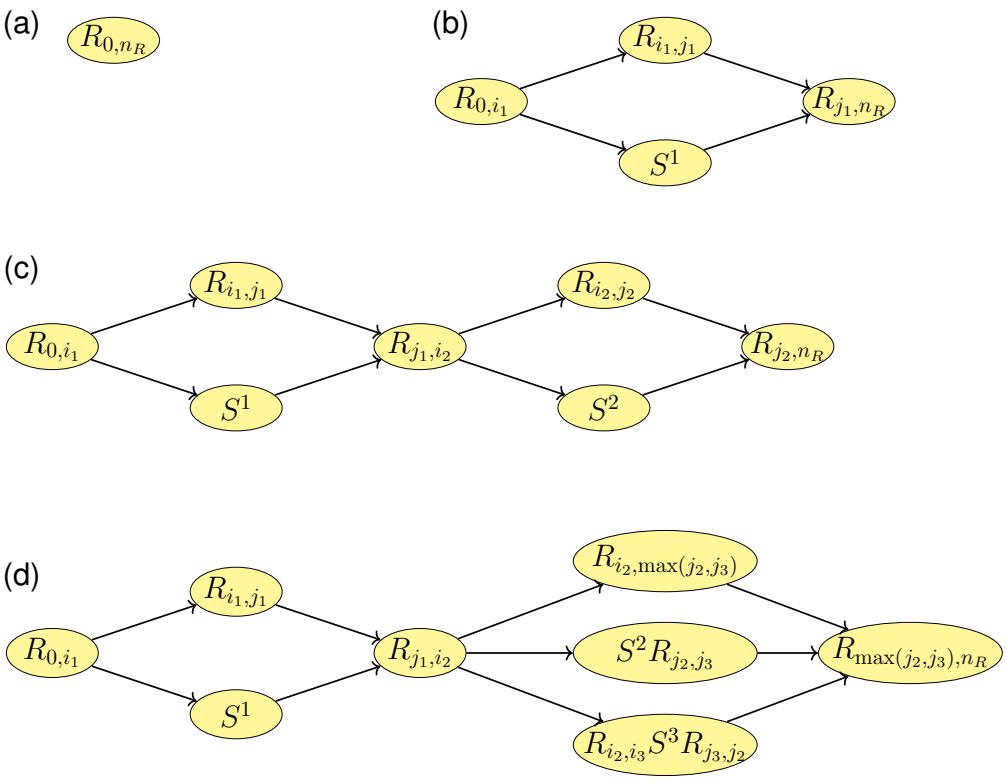
Graph construction Given a sequence variant set V and a reference sequence R, we initialize the graph with a single node labelled with R and no edges (Supplementary Note Figure 1a). We then iteratively add variants in order of their start position in the reference. We add the first variant

sequence S^1, replacing reference sequence R_{i_1,j_1}, to the graph by first splitting the node into three connected nodes with sequences R_{0,i_1}, R_{i_1,j_1}, and R_{j_1,n_R}, respectively. Then, we add a new node with sequence S^1 and edges from R_{0,i_1} to S^1 and from S^1 to R_{j_1,n_R} (Supplementary Note Figure 1b).

Graphtyper adds subsequent variants V_k to the graph by first determining whether its reference sequence overlaps one of the existing variants. If an overlap is found with a variant that has sequence R_{i_m,j_m}, we expand the variant to $R_{i_k,\max j_m,j_k}$ and add S^k as a new path in it (Supplementary Note Figure 1d). If no overlap is found we split a reference node into three nodes as described above (Supplementary Note Figure 1c).

Once the graph has been constructed we label all nodes with a *first position*. We start by creating a single numerical ordering of all chromosomes, by concatinating the chromomsomes in numerical order, letting chromosomes X and Y be the 23rd and 24th chromosomes, respectively. For example, using the GRCh38 human genome as reference, position 248,956,422 corresponds to the last base of the first chromosome (chr1:248,956,422) and 248,956,423 corresponds to the first base of the second chromosome (chr2:1). For reference nodes the first position is the position of the first character in the reference. The first position of allele nodes is the first position of the reference allele in the same variant. Consequently, allele nodes in the same variant have the same first position.

The position of any character in a reference node sequence can be derived from the character's distance from the first base. This also holds for alternative allele node sequences which are shorter than or equal to the variant's reference node sequence. When the alternative allele node sequence is longer than the reference node sequence, we assign a special position to characters that reach further than the reference node sequence, denoted as z_i where $i \in \mathbb{N}$.



Supplementary Note Figure 1: (a) A graph with a single node labeled with the reference sequence R. (b) A variant with reference sequence R_{i_1,j_1} and variant sequence S^1. (c) A non-overlapping variant sequence S^2 is added to the graph. (d) The reference sequence of V_3 overlaps V_2. The variant sequence of V_2 is extended and S^3 added to the variant.

The final result is a directed acyclic graph (DAG) with some special characteristics:

- There is a single start node and a single terminal node.
- If the DAG contains any variant sequence, all reference nodes are connected to two or more allele nodes in a variant.
- All allele nodes in a variant are labeled with the same first position and are connected to the same two reference nodes, but none of them have exactly the same sequence.
- One allele node of a variant contains a substring of the reference sequence and is called the

reference allele node. Other allele nodes in the variant are referred to as alternative allele nodes.

- The ordering of nodes by their start position is a valid topological ordering of the graph.

Index data structure Finding optimal alignments of a read to a linear reference genome is very time-consuming for large genomes, such as the human genome. The widely used Smith-Waterman algorithm is an optimal alignment algorithm which can align two sequences in quadratic time[1]. The Smith-Waterman algorithm has been extended to optimal variant-aware alignments[2,3]. It is however not practical to align reads to the human pangenome optimally.

Instead, our read alignment strategy follows a seed-and-extend paradigm. First, using a hash table, we identify a series of exact, or near-exact, matching k-mers (string of length k) in the graph. A k-mer, B, extends the match found by a k-mer A if the start position of B is the final position of A. We refer to a series of matching k-mers as a *seed*, noting that seeds have in previous literature more commonly been used for a single k-mer. The length of the seed is the number of k-mers in this extension and we assume that the optimal alignment of a read will pass through one of the longest seeds. Once the set of longest seeds has been identified, all longest seeds are extended using a local alignment algorithm more tolerant of approximate matches.

To identify the alignment seeds, we index the graph using a hash table, mapping a k-mer to one or more positions in the graph. Both the first and last location need to be known to efficiently determine whether a k-mer location is an extension of another. At each hash value we store, the k-mer hashed (8-bytes) and location information. The location information stored is a list of starting position of the k-mer (4 bytes), the offset position of the k-mer (2 bytes), and a variant of the graph that the k-mer overlaps (4 bytes), if any. If the k-mer overlaps multiple variants, it will be stored multiple times in the list. If the k-mer does not overlap any variant at a particular location

we store a known special value to indicate such events. The offset position is the last position of the k-mer subtracted by its start position. The memory used for each k-mer (excluding overhead) is $8 + 10t$ bytes, where t is the sum over all locations the k-mer maps to of the maximum of 1 and the number of variants the k-mer overlaps.

To limit the size of the index structure it is not practical to add every possible k-mer of the graph to the hash table. In particular, regions where there is a large number of consecutive variants the number of k-mers starting at a particular location might become very large. For example, if a single k-mer overlapped 20 biallelic SNPs then we would need to add $2^{20} \approx 1,000,000$ k-mers to the index at that location. To counter this problem, we set limit of the number of alternative sequences each k-mer can overlap (default: 4). This drastically reduces the number of k-mers stored in regions with high degree of polymorphism.

Implementation notes We store Graphtyper's index in RocksDB, which is a key-value persistent storage. By default, the index is loaded into memory to a SparseHash hash table. Reading the data into memory has an initial cost of time, but querying the in-memory index is much faster. For very large datasets, it is possible to use the RocksDB index directly from disk, e.g., if the index cannot fit into memory.

External libraries Graphtyper has the following library dependencies:

- args (`https://github.com/Taywee/args`): Argument parser.
- Catch (`https://github.com/philsquared/Catch`): Framework for unit tests.
- htslib (`https://github.com/samtools/htslib`): Library for HTS data formats.
- RocksDB (`https://github.com/facebook/rocksdb`): Key-value storage.
- SeqAn[4] (forked version, `https://github.com/hannespetur/seqanhts`): Library for sequence analysis.

- Snappy (`https://github.com/google/snappy`): Compression library.
- SparseHash (`https://github.com/sparsehash/sparsehash`): Hash map containers.
- StatGen (`https://github.com/statgen/libStatGen`): Statistical genetic library.
- Stations (`https://github.com/hannespetur/stations`): Multi-threading wrapper library.
- zlib (`http://www.zlib.net/`): Compression library.

External programs We used for our experiments the following programs:

- bamShrink (`https://github.com/DecodeGenetics/bamShrink`): Description below.
- chopBai[5]: Partitions bam index files.
- samtools[6]: Manipulates SAM formatted files.
- vcflib (`https://github.com/vcflib/vcflib`): Manipulates VCF files.
- vt (`http://genome.sph.umich.edu/wiki/Vt`): Manipulates VCF files.

bamShrink bamShrink extracts sequence reads of a region and reduces the output file size by binarizing base qualities values, removing unused BAM tags, removing unaligned reads, duplicate reads and reads that have fewer than 40 matching bases in their alignment. In addition to this, bamShrink performs coverage filtering in regions where the coverage is more than 3 times the average coverage, removes Ns if present on either end of a read and removes hard clipped entries from CIGAR strings. Lastly, bamShrink performs adapter removal by clipping overhanging ends of read pairs where the reverse read has been aligned in front of the forward read and their alignments overlap.

3 k-mer length

Our choice of using $k = 32$ as our k-mer length was justified based on three factors:

First, we want to be able to efficiently use the k-mers in a hash table. DNA sequences are represented with the alphabet $\Sigma = \{A, C, G, T\}$ and can be represented using two binary digits, $A = 00,\ C = 01,\ G = 10,\ T = 11$. Our software's target computer machines have a 64 bit word size, allowing us to store at most a 32-mer in a single word.

Second, to counter substitution sequencing errors we want to choose k small enough so that any k-mer rarely has more than one mismatch, with mismatches being the primary type of error in Illumina sequencing data and a reported error rate of as low as 0.1%[7]. Assuming the errors are independent of each other, the probability of seeing m errors with error rate e have binomial distribution

$$P(\text{\#errors} = m) = \binom{k}{m} e^m (1-e)^{k-m} \tag{1}$$

The probabilty of observing more than one error, and therefore failing to retrieve a k-mer, is

$$P(\text{\#errors} > 1) = 1 - P(\text{\#errors} = 0) - P(\text{\#errors} = 1) \tag{2}$$

Assuming an error rate of 0.1% the probability of observing more than one substitution error in a k-mer is 0.012%, 0.048%, and 0.193% with $k = 16, 32$ and 64, respectively. All of these values should be well acceptable, especially since our read alignment algorithm tries to retrieve multiple k-mers from each read as described in the following section.

Third, the k-mers need to be long enough to map to the graph as uniquely as possible. Analysis of the human reference genome reveals that 85.7% of 32-mers are unique in the genome and 79.3% are unique up to a Hamming distance 1 (the number of substitution differences between two sequences). On the other hand, only 21.8% of 16-mers can be uniquely identified and 0.000786%

of 16-mers are unique up to a Hamming distance of 1[8].

In Graphtyper we have implemented methods to construct graphs using a reference sequence and sequence variants which are read from FASTA and VCF files, respectively. In future releases we expect to support other sequencing and variation data file formats. In particular we are interested in using graph file formats, such as GFA (`https://github.com/GFA-spec/GFA-spec`).

4 GATK genotyping with common dbSNP sequence variants

Traditionally, GATK is only run with no external knowledge of sequence variants to be found in a sample. Because Graphtyper can incorporate such knowledge into its pipeline, we attempted to do so with GATK using common dbSNP variants (build 150) in genotyping NA12878. First, we ran GATK in discovery mode on NA12878 and created a union set of common dbSNP and discovered variants. Next, we used the parameters `--alleles <VCF>` and `--genotyping_mode GENOTYPE_GIVEN_ALLELES` to run GATK in genotyping mode for genotyping variants in the union set. Finally, we removed all sites with no alternative allele calls (`AC=0`) and evaluated the remaining sequence variants.

The `--alleles` parameter is noted to be not well tested. When we ran the experiment using GATK HaplotypeCaller the genotyping always failed due to a runtime error caused by IndexOutOfBoundsException. We are able to run the experiment using GATK UnifiedGenotyper (Supplementary Note Table 1). Our experiments show that GATK UnifiedGenotyper does not benefit from incorporating known dbSNP sequence variants.

Supplementary Note Table 1: GATK UG genoptyping of NA12878 with and without given the a set of common dbSNPs.

	No variants given		Given dbSNPs	
	Raw	Filtered	Raw	Filtered
Genotyping pipeline	GATK UG	GATK UG	GATK UG	GATK UG
SNPs	3,913,454	3,585,462	3,802,468	3,480,426
Transitions/Transversions	1.97	2.04	2.02	2.10
Indels	649,301	646,057	685,330	666,888
MNPs	0	0	0	0
Complex	0	0	0	0
Recalled platinum variants	3,967,739	3,862,484	3,868,264	3,759,398
Recall rate	95.20%	92.67%	92.81%	90.20%
Validated variant calls	3,963,186	3,857,999	3,837,421	3,730,899
Precision	99.885%	99.884%	99.203%	99.242%
Validated SNP calls	3,465,168	3,360,971	3,364,295	3,261,211
Recall rate	96.39%	93.49%	93.61%	90.74%
Precision	99.991%	99.993%	99.961%	99.965%
Validated non-SNP calls	498,018	497,028	473,126	469,688
Recall rate	87.70%	87.52%	87.76%	86.79%
Precision	99.153%	99.154%	94.126%	94.495%
Peak memory usage [GB]	43.97	–	43.97	–
CPU time [hr]	31.1	–	65.6	–
Alt. alleles called in trio	5,754,093	5,272,137	5,589,563	5,089,341
$FDR_{estimated}$	3.34%	2.62%	2.71%	2.01%
$Germline_{estimated}$	5,562,132	5,133,770	5,437,952	4,987,148
SNP alt. alleles	4,948,488	4,473,460	4,794,335	4,331,012
$FDR_{estimated}$	2.55%	1.65%	2.56%	1.67%
$Germline_{estimated}$	4,822,380	4,399,652	4,671,782	4,258,844
Non-SNP alt. alleles	805,605	798,677	795,228	758,329
$FDR_{estimated}$	8.17%	8.08%	3.65%	3.96%
$Germline_{estimated}$	739,752	734,118	766,170	728,304

5 HLA genotyping methods

Here we will describe our HLA genotyping methods. A preliminary implementation of these methods have previously been described in a MSc thesis under a software package called Gyper[9]. Graphtyper generalizes methods of Gyper, which was only designed to genotype the HLA region. Graphtyper includes Gyper's code base with a number of improvements to the code, both in terms of speed and accuracy. Several important features, which are only in Graphtyper, include:

- Support for graph construction using a reference sequence and a VCF file.
- Serialization of the graph and graph index.
- k-mer query with hamming distance 1 (versus exact k-mer matches only).
- Discovery model.
- Extended genotype likelihood model.

In summary, the HLA reference alleles were fetched from the IPD-IMGT/HLA database[10] in XML format. Each exon and intron sequence were multiple sequence aligned separately using MUSCLE[11]. Then, their differences are reported in a VCF file used to construct a graph with HLA allele variation. Each HLA allele corresponds to a path in Graphtyper's graph. A graph is constructed for each gene and the alleles are called based on the paths the reads align to.

The HLA gene cluster is only a very small part of the human genome, and thus only a very small portion of whole genome sequenced reads are relevant to HLA genotyping. Our method makes use of the fact the reads are usually stored aligned in alignment files (SAM/BAM files). All read pairs belong in one of three categories:

- Both reads are mapped to the reference genome at locations l_1 and l_2.
- One read is unmapped, but the other read in the pair is mapped to location l. By convention,

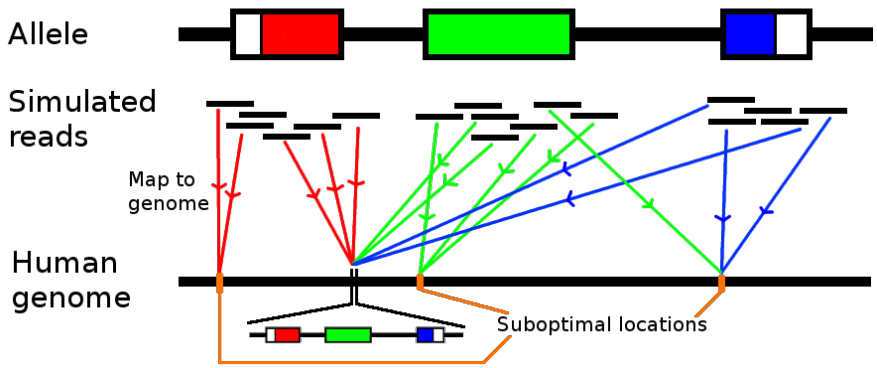
both reads are marked to be located at l.

- Both reads are unmapped. The reads are both marked as unmapped.

Our goal is to find regions of the genome which are likely to have HLA relevant reads mapped to them. The following three steps explain our process:

1. Simulate reads that overlap the alleles' exons.
2. Map the simulated reads to the human reference genome.
3. Check where the simulated reads map.

Supplementary Note Figure 2 shows the process for a single allele. The aligner will map to a correct location, a suboptimal location outside HLA, or not map the read to any position of the genome. We are only interested in reads overlapping the exons, since the exons determine the first 3 fields of the HLA genotype. All aligned positions are extracted and used as the regions of interest. When genotyping, we only used reads located inside the regions of interest.



Supplementary Note Figure 2: The process of finding relevant positions of the human genome of one allele. Reads overlapping the exons are simulated and mapped to the human reference genome.

Computional results 180 exome BAM files were fetched from the 1000 Genomes FTP site and genotyped for the three main HLA class I genes, which the samples had been verified[12]. The samples were taken from individuals from with ancestry from all over the world.

We compared Gyper to OptiType. Gyper's genotype call accuracy was 97.9% at a 4 digit resolution (Supplementary Note Table 2), marginally better than the 97.8% accuracy of OptiType. We also

Supplementary Note Table 2: 4 digit exome accuracy.

	Gyper					OptiType
HLA gene	0 errors	1 error	2 errors	Correct alleles	Accuracy	Accuracy
HLA-A	171	9	0	351 of 360	**97.5%**	96.9%
HLA-B	168	12	0	348 of 360	96.7%	**98.6%**
HLA-C	178	2	0	358 of 360	**99.4%**	97.7%
All genes	517	23	0	1057 of 1080	**97.9%**	97.8%

verified Gyper using 20 low coverage WGS alignment files obtained from the 1000 Genomes project. These files have at least 3x non duplicated aligned coverage. Gyper and OptiType had the same 95.0% accuracy at a 4 digit resolution (Supplementary Note Table 3).

Supplementary Note Table 3: 4 digit low coverage WGS accuracy.

	Gyper					OptiType
HLA gene	0 errors	1 error	2 errors	Correct alleles	Accuracy	Accuracy
HLA-A	18	2	0	38 of 40	95.0%	95.0%
HLA-B	17	3	0	37 of 40	92.5%	92.5%
HLA-C	19	1	0	39 of 40	97.5%	97.5%
All genes	54	6	0	114 of 120	95.0%	95.0%

Several other HLA typing methods have been released[13–16] that use second-generation sequencing data. We did not compare Gyper or Graphtyper to those as they require long running time and OptiType had reported that they type with better or similar accuracy compared to the other methods.

As we had determined that Gyper was a competitive method, we ran our comparison on Icelandic samples to Gyper. The overall accuracy of Grathtyper was slightly higher than Gyper (Supplementary Note Table 4 and Supplementary Note Table 5).

Supplementary Note Table 4: 2 digit call accuracy compared to deCODE's PCR verification.

	Gyper					Graphtyper
HLA gene	0 errors	1 error	2 errors	Correct alleles	Accuracy	Accuracy
HLA-A	33	2	0	68 of 70	97.1%	**98.2%**
HLA-B	167	10	2	344 of 358	96.1%	**96.8%**
HLA-C	157	7	3	321 of 334	**96.1%**	95.1%
HLA-DQA1	45	0	0	90 of 90	100.0%	100%
HLA-DQB1	77	2	0	156 of 158	98.7%	**99.4%**
HLA-DRB1	183	2	0	368 of 370	99.5%	**99.8%**

Supplementary Note Table 5: 4 digit call accuracy compared to deCODE's PCR verification.

	Gyper					Graphtyper
HLA gene	0 errors	1 error	2 errors	Correct alleles	Accuracy	Accuracy
HLA-A	33	2	0	68 of 70	97.1%	**98.2%**
HLA-DQA1	45	0	0	90 of 90	100.0%	100.0%
HLA-DQB1	70	9	0	149 of 158	94.3%	**98.8%**
HLA-DRB1	162	21	2	345 of 370	**93.2%**	91.6%

6 Investigation of Graphtyper's incorrect HLA calls

Our investigation of the incorrect HLA-B and HLA-C calls revealed that most of the verified types are highly inconsistent with the sequence reads from the BAM files. We are unsure what caused this, but it could be that the PCR verified sample and sequenced sample is not the same in these cases. We also investigated the incorrect HLA-DRB1 calls at the 4-digit resolution, which showed that 26 of 27 samples were verified with at least one allele in the HLA-DRB1*4 family. HLA-DRB1*4 carriers also carry the HLA-DRB4 gene, which has a high homology to HLA-DRB1 but is not a part of the primary reference genome, as the presence of DRB4 is linked with allelic variants of DRB1, which differs between ethnic groups[17]. We suspect that many of our HLA-DRB1 incorrect typing are due to misaligned sequence reads from the HLA-DRB4 gene.

7 Definition of epsilon

We defined ϵ, the relative likelihood of observing a read given that it was sequenced from an individual with a pair of haplotypes h_1, h_2 that do not support the read. We restricted ϵ to be in the set $\{1/2^5, 1/2^6, .., 1/2^{13}\}$ and defined it as

$$\epsilon = \min\left(1, \frac{2^{\text{LQS}} 2^{\text{LMQ}} 2^{\text{MQ0}} 2^{\text{CPD}}}{2^4}\right) \frac{2^{\text{SML}}}{2^9} \tag{3}$$

where

$$\text{LQS} = \begin{cases} 1 & \text{, if the base pair quality of a base that aligned to a SNP is below 25} \\ 0 & \text{, otherwise} \end{cases}$$

$$\text{LMQ} = \begin{cases} 2 & \text{, if we did not map the read uniquely or BWA reported mapping quality } < 25 \\ 0 & \text{, otherwise} \end{cases}$$

$$\text{MQ0} = \begin{cases} 1 & \text{, if BWA reported mapping quality } = 0 \\ 0 & \text{, otherwise} \end{cases}$$

$$\text{CLP} = \begin{cases} 3 & \text{, if we soft clipped some part of the read} \\ 0 & \text{, otherwise} \end{cases}$$

$$\text{SML} = \begin{cases} 4 & \text{, } \textit{haplotype\_dissimilarity}(h_1,\text{read}) = 1 \text{ and } \textit{haplotype\_dissimilarity}(h_2,\text{read}) = 1 \\ 3 & \text{, } \textit{haplotype\_dissimilarity}(h_1,\text{read}) = 1 \text{ xor } \textit{haplotype\_dissimilarity}(h_2,\text{read}) = 1 \\ 2 & \text{, } \textit{haplotype\_dissimilarity}(h_1,\text{read}) = 2 \text{ and } \textit{haplotype\_dissimilarity}(h_2,\text{read}) = 2 \\ 1 & \text{, } \textit{haplotype\_dissimilarity}(h_1,\text{read}) = 2 \text{ xor } \textit{haplotype\_dissimilarity}(h_2,\text{read}) = 2 \\ 0 & \text{, otherwise} \end{cases}$$

Here, LQS denotes low quality SNP, LMQ denotes low mapping quality, MQ0 denotes mapping quality zero, CLP denotes soft clipped read, and SML denotes read-to-haplotype similarity.

haplotype_dissimilarity$(h,$read$)$ measures the dissimilarity between the read alignment path and the haplotype path h by counting how many different allele sequence the read does not match of the haplotype path.

Supplementary Tables

Supplementary Table 1: Mendelian error rate of sequenced variants in chr21:21,559,430-21,559,518 (GRCh38). We determined these sequence variants to be artifacts yielded by misalignments around a common 40-bp deletion. A dash means that the genotyping pipeline did not call the misalignment artifact.

Sequence variant	Graphtyper	GATK UG	GATK UGLite	GATK HC	GATK HC joint	Samtools	Platypus	FreeBayes
chr21:21559431:A/AT	–	–	–	–	–	–	–	0.0%
chr21:21559453:AAT/A	–	4.9%	3.4%	–	–	6.0%	4.5%	4.1%
chr21:21559454:AT/A	–	–	–	0.7%	3.6%	–	–	–
chr21:21559457:AT/A	–	–	–	0.6%	3.6%	–	–	–
chr21:21559458:T/A	–	3.0%	3.0%	–	–	–	4.4%	–
chr21:21559489:G/A	–	1.1%	1.1%	–	–	1.7%	1.2%	–
chr21:21559489:G/T	–	–	–	–	–	0.5%	–	–

Supplementary Table 2: Filtered sequence variant calls comparison of 691 whole-genome sequenced Icelanders of human chromosome 21.

Genotyping pipeline	Graphtyper	GATK UG	GATK HC	GATK HC joint
Sequence variant records	221,221	260,164	238,594	325,355
SNPs	200,728	219,648	203,966	266,619
Transitions/Transversions	2.15	2.06	2.05	1.81
Indels	20,431	40,516	36,711	65,027
MNPs	88	0	0	0
Complex	439	0	0	29,454
Alternative alleles in trios	205,492	260,166	239,371	348,961
$\text{Germline}_{\text{estimated}}$	200,984	227,056	214,801	**240,020**
$\text{FDR}_{\text{estimated}}$	**2.19**%	12.73%	10.26%	31.22%
SNPs	182,097	197,124	184,803	233,753
Transitions/Transversions	2.15	2.07	2.09	1.87
$\text{Germline}_{\text{estimated}}$	181,420	192,790	183,957	**203,233**
$\text{FDR}_{\text{estimated}}$	**0.37**%	2.20%	0.46%	13.06%
Non-SNPs	23,395	63,042	54,568	115,208
$\text{Germline}_{\text{estimated}}$	19,564	34,266	30,844	**36,787**
$\text{FDR}_{\text{estimated}}$	**16.38**%	45.65%	43.48%	68.07%

Supplementary Table 3: Frequencies of HLA allele families in the Icelandic population ($n = 28,075$).

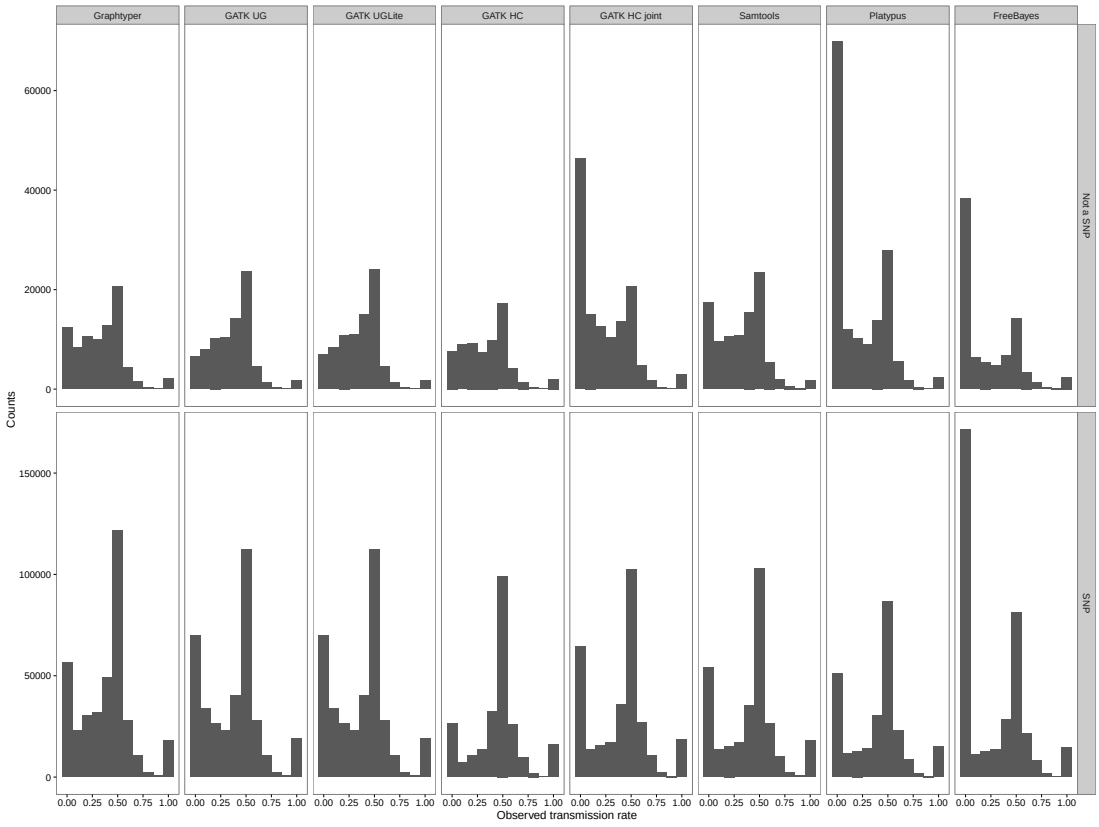
HLA-A		HLA-B		HLA-C		HLA-DQA1		HLA-DQB1		HLA-DRB1	
Allele family	Freq.	Allele family	Freq.	Allele family	Freq.	Allele family	Freq.	Allele family	Freq.	Allele family	Freq.
*A*01*	12.20%	*B*07*	18.89%	*C*01*	4.67%	*DQA1*01*	51.47%	*DQB1*02*	14.29%	*DRB1*01*	9.86%
*A*02*	30.23%	*B*08*	8.79%	*C*02*	5.15%	*DQA1*02*	9.02%	*DQB1*03*	29.83%	*DRB1*03*	9.20%
*A*03*	14.27%	*B*13*	2.07%	*C*03*	20.84%	*DQA1*03*	19.68%	*DQB1*04*	4.48%	*DRB1*04*	17.89%
*A*11*	7.23%	*B*14*	3.24%	*C*04*	7.13%	*DQA1*04*	4.39%	*DQB1*05*	12.35%	*DRB1*07*	9.02%
*A*23*	1.72%	*B*15*	10.67%	*C*05*	9.81%	*DQA1*05*	14.81%	*DQB1*06*	39.05%	*DRB1*08*	4.92%
*A*24*	10.56%	*B*18*	3.53%	*C*06*	8.76%	*DQA1*06*	0.64%			*DRB1*09*	1.79%
*A*25*	2.09%	*B*27*	6.88%	*C*07*	34.11%					*DRB1*10*	1.57%
*A*26*	1.25%	*B*35*	5.43%	*C*08*	2.22%					*DRB1*11*	2.47%
*A*29*	2.99%	*B*37*	2.08%	*C*12*	3.39%					*DRB1*12*	2.24%
*A*30*	1.52%	*B*38*	0.56%	*C*14*	0.36%					*DRB1*13*	18.29%
*A*31*	4.78%	*B*39*	2.02%	*C*15*	1.27%					*DRB1*14*	0.54%
*A*32*	3.83%	*B*40*	11.78%	*C*16*	2.16%					*DRB1*15*	21.96%
*A*33*	0.12%	*B*44*	13.15%	*C*17*	0.14%					*DRB1*16*	0.24%
*A*68*	7.22%	*B*45*	0.23%								
		*B*49*	1.23%								
		*B*51*	2.21%								
		*B*52*	0.20%								
		*B*53*	0.53%								
		*B*55*	1.40%								
		*B*56*	0.46%								
		*B*57*	4.34%								
		*B*58*	0.31%								

Supplementary Figures

Supplementary Figure 1

Supplementary Figure 1: IGV[18] visualization of mapped sequence reads of two Icelanders carrying a 40-bp deletion. The genomic region shown is chr21:21,559,430-21,559,518 (GRCh38) and the deleted sequence is between the two vertical green lines. **(a)** A heterozygous carrier of the deletion. Graphtyper was the only genotyping pipeline that correctly recognized the individual as a carrier. The other pipelines called the false sequence variants due to misalignments around the indel (red boxes). **(b)** A homozygous carrier of the deletion. In this case most of the reads are correctly mapped as carrying the deletion, but again some misalignment artifacts are observed (red box).

Supplementary Figure 2

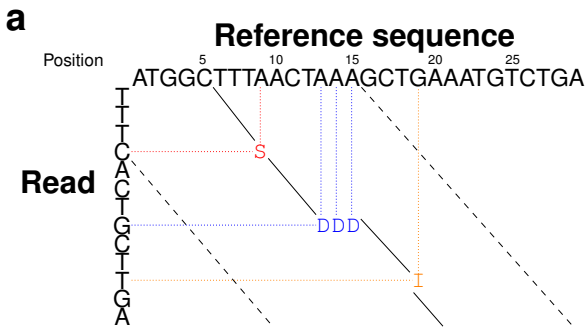


Supplementary Figure 2: Alternative allele transmission rate in 230 Icelandic parent-offspring trios. All genotyping pipelines have an excess of sequence variants that are never transmitted from parent to offspring, which may be calls due to sequencing error or non-germline variation. Bin width is 0.1.

Supplementary Figure 3

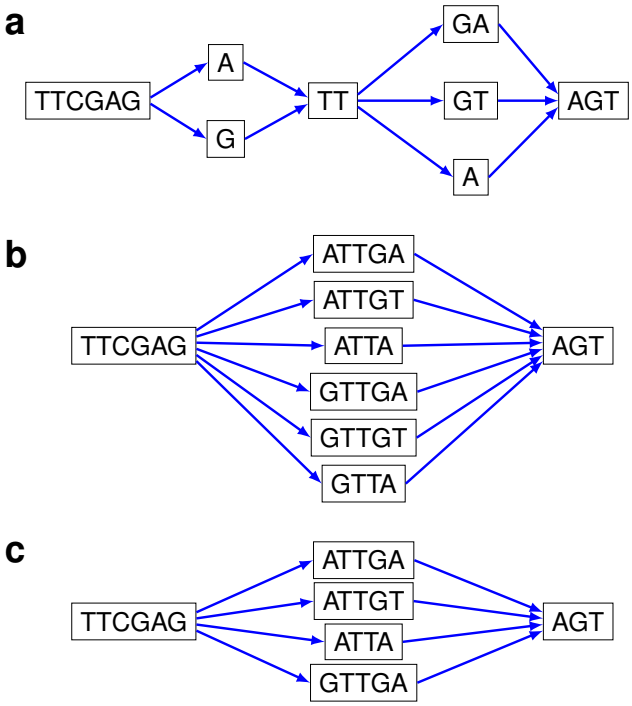
Supplementary Figure 3: Alternative allele transmission rate in 230 Icelandic parent-offspring trios by SNP mutation type. The mutation ratio of transitions and transversions is estimated to be around two in the human autosomal genome. We observed that the transition/transversion ratio improved at higher transmission rates, indicating that transmission rate is measuring quality. There was also a large excess of transversions that are not transmitted.

Supplementary Figure 4



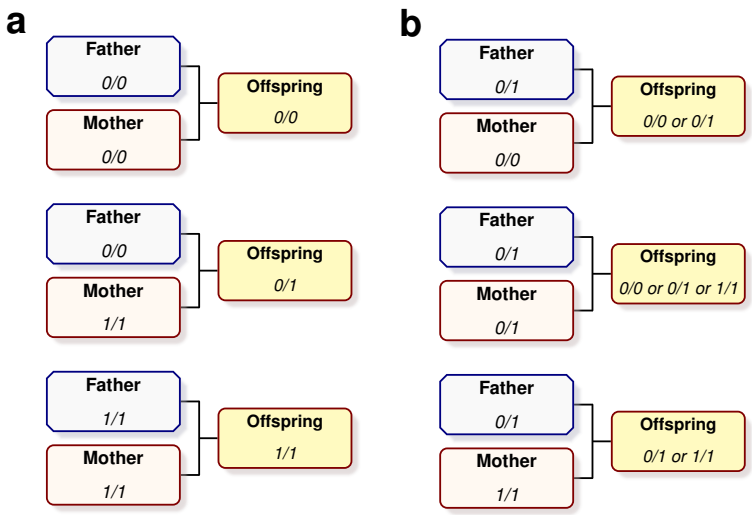
Supplementary Figure 4: Detection of novel alleles. **(a)** Semi-global banded alignment of a sequenced read to an extracted reference sequence. **(b)** Observed variation with respect to the reference sequence.

Supplementary Figure 5



Supplementary Figure 5: Merging sequence variants can reduce the number of haplotypes in the graph. **(a)** An example of a graph with two sequence variants. The graph has a total of six haplotypes. **(b)** Two sequence variants are closer than 5 bp from each other and are grouped together. **(c)** If we only observed four out of six haplotypes in a population, we can reduce the number of haplotypes in the graph to four as shown here.

Supplementary Figure 6



Supplementary Figure 6: Mendelian inheritance of alleles in parent-offspring trios. **(a)** Both parents are homozygous and the offspring's genotype can be inferred. **(b)** At least one parent is heterozygous and we cannot uniquely infer the offspring's genotype. We can measure the transmission rate of an allele in these types of trios.

Supplementary References

1. Smith, T. F. & Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology* **147**, 195–197 (1981).

2. Zhao, M., Lee, W.-P., Garrison, E. P. & Marth, G. T. SSW library: An SIMD smith-waterman C/C++ library for use in genomic applications. *PLoS One* **8**, 12 (2013).

3. Lee, C., Grasso, C. & Sharlow, M. F. Multiple sequence alignment using partial order graphs. *Bioinformatics* **18**, 452–464 (2002).

4. Döring, A., Weese, D., Rausch, T. & Reinert, K. SeqAn an efficient, generic C++ library for sequence analysis. *BMC bioinformatics* **9**, 11 (2008).

5. Kehr, B. & Melsted, P. chopBAI: BAM index reduction solves I/O bottlenecks in the joint analysis of large sequencing cohorts. *Bioinformatics* **32**, 2202–2204 (2016).

6. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27(21)**, 2987–2993 (2011).

7. Hoffmann, S. *et al.* Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput Biol* **5(9)**, e1000502 (2009).

8. Shajii, A., Yorukoglu, D., Yu, Y. W. & Berger, B. Fast genotyping of known snps through approximate k-mer matching. *Bioinformatics* **32**, 17 (2016).

9. Eggertsson, H. P. Gyper: A graph-based HLA genotyper using aligned DNA sequences (2015).

10. Robinson, J. *et al.* The IPD and IMGT/HLA database: allele variant databases. *Nucleic Acids Research* **43**, D423–431 (2015).

11. Edgar, R. C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* **32(5)**, 1792–1797 (2004).

12. Erlich, R. L. *et al.* Next-generation sequencing for HLA typing of class I loci. *BMC Genomics* **12**, 42 (2011).

13. Warren, R. L. *et al.* Derivation of HLA types from shotgun sequence datasets. *Genome Medicine* **4**, 95 (2012).

14. Liu, C. *et al.* ATHLATES: accurate typing of human leukocyte antigen through exome sequencing. *Nucleic Acids Research* **41**, e142–e142 (2013).

15. Major, E., Rigó, K., Hague, T., Bérces, A. & Juhos, S. HLA Typing from 1000 Genomes Whole Genome and Whole Exome Illumina Data. *PLoS ONE* **8**, e78410 (2013).

16. Dilthey, A. T. *et al.* High-Accuracy HLA Type Inference from Whole-Genome Sequencing Data Using Population Reference Graphs. *PLOS Computational Biology* **12**, e1005151 (2016).

17. Naruse, T. K. *et al.* HLA-DRB4 genotyping by PCR-RFLP: diversity in the associations between HLA-DRB4 and DRB1 alleles. *Tissue antigens* **49**, 152–9 (1997).

18. Thorvaldsdóttir, H., Robinson, J. T. & Mesirov, J. P. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. *Briefings in Bioinformatics* **14**, 178–192 (2013).

Paper II

Paper II

GraphTyper 2 enables population-scale genotyping of structural variation using pangenome graphs

Hannes P. Eggertsson, Snaedis Kristmundsdottir, Doruk Beyter, Hakon Jonsson, Astros Skuladottir, Marteinn T. Hardarson, Daniel F. Gudbjartsson, Pall Melsted, Bjarni V. Halldorsson, Kari Stefansson

Submission is under consideration

Title

GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs

Authors

Hannes P. Eggertsson[1,2], Snaedis Kristmundsdottir[1,3], Doruk Beyter[1], Hakon Jonsson[1], Astros Skuladottir[1], Marteinn T. Hardarson[1], Daniel F. Gudbjartsson[1,2], Kari Stefansson[1,4], Bjarni V. Halldorsson[1,3], Pall Melsted[1,2]

[1]deCODE genetics / Amgen Inc., Sturlugata 8, Reykjavik, Iceland

[2]School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland

[3]School of Science and Engineering, Reykjavik University, Reykjavik, Iceland

[4]Faculty of Medicine, School of Health Sciences, University of Iceland, Reykjavik, Iceland

Corresponding authors: Hannes P. Eggertsson (hannese@decode.is), Bjarni V. Halldorsson (bjarnih@decode.is), Pall Melsted (pmelsted@decode.is)

Abstract

Analysis of sequence diversity in the human genome is fundamental for genetic studies. Structural variants (SVs) are frequently omitted in sequence analysis studies although each has relatively great impact on the genome. We present GraphTyper2, which uses pangenomes to genotype SVs along with small variants using short-reads. We use public datasets to show that our SV genotyping is sensitive and variant segregation in families demonstrates the accuracy of our approach. We generated long-read sequences from 41 Icelanders to assess the quality our short-read SV calls. Using the long-reads, we validated 67.7% of our 8,568 high-confidence SVs on average per genome. We show that GraphTyper2 can simultaneously genotype tens of thousands of whole-genomes by characterizing 60 million small variants and half a million SVs in 49,962 Icelanders, thereof 80 thousand SVs with high-confidence. GraphTyper2 is a valuable tool for characterizing sequence variation in large-scale sequencing studies.

Introduction

Characterization of sequence variants in the human genome has greatly improved[1–5] with lower sequencing costs and improvements in sequencing technologies. Particularly sequence variants that modify fewer than 50 nucleotides, which are usually detected by finding discordances in short-read alignments compared to a reference genome[6–8]. Larger sequence variants are known as structural variants (SVs) and include insertions, deletions, duplications, and inversions of 50 base-pairs or more. While SVs are only a small portion of all sequence variation, analyses suggest they have a high impact on gene expression[9] and they have been implicated in many rare diseases[10–12]. Recent studies suggest that each individual has on average over 27 thousand SVs[13], but some of them can only be discovered using specific library preparation or sequencing technologies, such as long-read sequencing.

Short-read whole-genome sequencing (WGS) technologies are widely used in population-scale genotyping. They are readily available and have low error rates. However, due to their read length limitations, SVs need to be discovered from read assemblies, split-read alignments, read alignment coverage, read-pair insert sizes or other indirect inferences. These SV discovery methods have lower sensitivity and specificity than methods aimed at smaller variants. Further, breakpoints of the detected SVs are often imprecise and the SV sequence is often only partially characterized[4].

Population-scale genotyping refers to when samples from the same population are genotyped together, either one at a time or jointly. Joint-calling is typically favored for population-scale genotyping as it generates a set of genotype calls which are comparable across the samples in the population and can be used directly in genome-wide association studies (GWAS). In the widely used Genome Analysis ToolKit (GATK) HaplotypeCaller[7],

genotyping small variation in a population is performed by joint-calling from intermediate files (gVCF), which contain support (or lack of support) for a variation at every position of the genome. The data are then combined across all samples to generate a variation map for the population. Applying a similar strategy for SVs is difficult because the exact boundaries of SVs are often imprecise and thus may be represented slightly differently between samples.

In our previous publication we presented GraphTyper[6], a method for population genotyping sequence variants using pangenome graphs[14–16] (or genome graphs). Pangenome graphs are an extension to the linear reference genome such that sequence variants can be encoded in a graph. Each path in the pangenome graph encodes a potential haplotype. Reads are realigned to the variation-aware pangenome, which can refine alignments near variation and reduce bias towards the reference sequence.

In summary, GraphTyper extracts reference aligned reads from a small genomic region, locally realigns them to a pangenome graph and, concurrently, genotypes variants in the graph. We showed that combining these two operations makes our method easily parallelizable and scale better than methods that use a linear reference. This strategy allowed us to jointly-call more than 28 thousand samples. Our previous version of the method was, however, limited to small sequence variants (single nucleotide variants; insertions and deletions shorter than 50bp).

Here we present a second version of GraphTyper (GraphTyper2) that enables efficient encoding of structural variation into the pangenome graph (Figure 1a) and genotyping of those variants. Our method can now simultaneously genotype small variants and SVs in tens of thousands of samples. The problem of variant calling can be split into discovery and genotyping. In the discovery step, potential variation sites are detected and in the

genotyping step genotypes are called at those sites. A number of methods exist for discovering SVs[17–20] and we used these methods to discover SV sites on a per sample basis. We then merge these SVs across all samples and encode them into a population graph structure (Figure 1b) that we subsequently use to jointly genotype the population. Our new GraphTyper version also features several other enhancements that are summarized in the Supplementary Note, including support for using non-human reference genomes, refined indel detection, and improved variant discovery filters to remove systematic false positive variant sites.

Results

SV genotyping pipeline The main data structure in GraphTyper is a directed acyclic graph (DAG), where a path in the DAG represents a possible haplotype. Structural variants can be encoded in the DAG and can coexist in the graph with small variants. Two breakpoints are typically added for each SV, representing the start and end locations of the SV with respect to the reference. It is also possible that only one of these locations was discovered, in which case the SV is represented by a single breakpoint. Each breakpoint defines two alleles in the graph, a reference allele and an alternative allele, represented by nodes in the graph.

To limit the size of the graph, GraphTyper inserts only the breakpoint sequences (up to 152 bp – determined by the short-read length) into the graph (Figure 1a). This limits compute times and allows robust SV genotyping across SV lengths, as the mapping is not biased towards larger SVs. Another advantage is that the SV sequence is often only partially characterized. GraphTyper realigns all sequence reads of a genomic region, including unaligned and clipped sequences, to the graph structure and genotypes the variants encoded in that graph. GraphTyper genotypes SVs in a graph along with previously discovered SNPs and indels (Figure 1b).

In our pipeline for genotyping SNPs and indels, we partition the genome into 50 kbp regions (by default) and genotype each region separately. We use larger graphs to genotype SVs (default 1.2 Mbp and overlap by 200 kbp) such that the breakpoints of each SV are typically in the same graph. Of the SVs generated in the 1000G project[4], only 36 (0.052%) would have had breakpoints on different graphs if they were binned using our scheme. For those larger SVs we can expect less accurate results from GraphTyper.

Prior to genotyping, we extract all reads aligned to the region and realigned them to the pangenome graph. Misalignment near SV breakpoints often leads to false positive variant calls (Supplementary Figure 1), a problem that can be alleviated by realigning the reads onto a variation-aware data structure[6,16].

GraphTyper has two models for genotyping SVs, one is based on read realignments to the SV breakpoints and the other is based on alignment coverage (Methods). Briefly, breakpoints are genotyped by counting the number of reads aligning through the different paths of a breakpoint. Each breakpoint is genotyped separately for variants with two breakpoints. Deletions and duplications are also genotyped from decrease and increase in alignment coverage, respectively. The aggregated genotype call is made by selecting the call among all genotype models that has the highest genotyping quality (GQ) (Supplementary Note).

The graph construction, indexing and alignment otherwise follows what we described previously[6].

Genotyping public data of parent-offspring trio We evaluated the genotyping performance of GraphTyper on a well-studied parent-offspring trio (NA12878, NA12891, and NA12892). Whole-genome sequence data of these samples are publicly available from the Platinum Genome project[21]. We ran Manta[18] on all three samples independently and then merged all SV sites (Methods). We also ran GraphTyper's small variant pipeline to construct a graph with SNPs and indels and added the SV sites discovered by Manta to the graph and genotyped the entire set of variants.

We compared the results from both of GraphTyper's deletion genotyping models to 2,612 high-confidence deletions found in the autosomes of NA12878, originally discovered using multiple sequencing technologies[22]. We would expect to find fewer deletions with only

short-read data. Our results showed that the breakpoint model had higher sensitivity than the coverage model when coverage is 10x or more (Figure 2a). As expected, the aggregated model had the highest sensitivity at all coverages with more than 90% of the deletions in the truth set recalled. In what follows, all GraphTyper evaluations were performed only on the aggregated model.

We also performed SV genotyping on the same parent-offspring trio using widely-used methods: Manta[18], Delly[17], BayesTyper[23], and Lumpy[19] (discovery) with SVTyper[24] (genotyping) using appropriate filters (Supplementary Note). In our experiment, Manta+GraphTyper had the highest SV deletion sensitivity at all tested breakpoint precision thresholds (67.8%-90.7%) (Figure 2b). Manta+BayesTyper had the highest precision when testing with a strict breakpoint precision threshold (14 bp or less) but Manta+GraphTyper had only a slightly lower precision (Figure 2c). Lumpy+SVTyper had the highest precision when allowing a more lenient threshold. We also calculated the F_1-score for each method (Figure 2d) which ranks Manta+GraphTyper highest at lower breakpoint precision thresholds (26 bp or lower). Based on the above observations, we concluded that Manta+Graphtyper is a sensitive method to detect SVs.

Genotyping four Icelandic families The public dataset contains only a single parent-offspring trio and thus we further assessed our method by genotyping chromosome 20 of 56 individuals in four Icelandic families (Figure 3a). The families consist of 8 parents and 48 offsprings: two families have 10 offsprings, one family 11, and one 17 offsprings. We merged all SVs discovered by Manta and genotyped them using Manta+GraphTyper (Methods) and compared the results to joint calling all 56 individuals using Manta. Parent-offspring trio analysis shows high-confidence Manta+GraphTyper genotypes have a Mendelian inheritance

error rate of 0.27%, while the high-confidence Manta genotypes have more than 10-fold higher error rate (Figure 3b). Since all individuals in the large families have at least 10 close relatives, we would expect that almost every SV is carried by multiple individuals. In the Manta joint-calls we surprisingly saw that 22.2% of the SVs had only one carrier (Figure 3c), while Manta+GraphTyper did not genotype any SVs in only one carrier.

Next, we measured the transmission rate of SV alleles in the 48 parent-offspring trios (Methods) and investigated their distribution (Figure 3d). Assuming Mendelian inheritance, germline alleles transmit from parent to offspring with 50% probability. We thus expect that high quality variants have transmission rate distribution symmetric around 50%. The distribution of Manta+GraphTyper calls were close to being symmetric around the 50%, while the Manta joint calls were heavily biased towards lower transmission rates, indicating erroneous genotyping (Figure 3d).

Based on the above observations, we conclude that Manta is useful for discovering SVs but underestimates their frequencies when joint calling. By merging Manta variants from multiple samples and using them as input for GraphTyper, we can alleviate that problem.

Long-read validation While variant segregation in families can reveal genotyping inaccuracies, they do not verify that the SVs are of the correct size and type. To tackle this we sequenced the genomes of 41 Icelanders using long-read sequencing (Methods) and compared the short-read SV calls to those of Sniffles'[25]. The two methods are orthogonal as our pipeline only used Illumina short-reads (median 38.3x, range 25.1x-164.7x) while Sniffles used Oxford Nanopore long-reads (median 13.4x, range 8.3x-33.9x). We expect to detect more true SVs in long-reads than in short-reads, although with a lower breakpoint accuracy. We required two long-reads to support an SV of the same type (deletion or insertion) for it

to be considered validated and used a maximum breakpoint threshold of 50 bp (Supplementary Note).

In the 41 Icelanders, we genotyped 17,787 high-confidence SVs using Manta+GraphTyper. On average per genome, 8,568 high-confidence were genotyped and, thereof, we validated with long-reads 5,798 SVs (67.7%) (Supplementary Table 1). We validated more deletions (3,273 on average) than insertions and duplications (2,526 on average), which is consistent with previous short-read SV studies[2,4] and suggests that deletions are easier to discover than insertions in short-read sequence data. Using a more lenient maximum breakpoint threshold of 100 bp and 200 bp, we validate 74.8% and 76.9% of the high-confidence SVs, respectively, which indicates that many SVs are not validated because they may have inaccurate breakpoint positions in either the short-read or long-read SV calls.

Population-scale SV genotyping We assessed the scalability of our method by genotyping a large cohort of 49,962 whole-genome sequenced (median 36.9x, range 17.8x-307.3x) Icelandic genomes (Methods). As before, we merged SVs discovered by Manta which resulted in 543,939 SVs discovered in the population. These SVs were added to the SNP and indel graphs that were previously created with GraphTyper. Subsequently, we genotyped all samples using GraphTyper which resulted in 486,158 SVs that were called in at least one sample. After filtering the genotyped SVs, we retained 79,318 high-confidence SVs (Supplementary Note).

We analyzed how many SVs in previously published datasets[2,4,26] overlapped with our SVs (Methods) (Figure 4). As expected, we found greater overlap with common variants and more with deletions than insertions. In the same genotyping run, GraphTyper genotyped 59.5 million small variants: 44.3 million SNPs, 4.0 million indels and 11.2 million other small

variants. Merging the SVs required 2,223 CPU hours of compute time and genotyping using GraphTyper took 4.15 million CPU hours or 83 CPU hours per sample on average. Our results show that GraphTyper is a practical solution for high quality genotyping across the sequence variation spectrum on few and many samples.

Discussion

We have extended our previously published variant caller to enable SV genotyping in large scale datasets. With our extension, GraphTyper can be used to genotype variants across the full variation spectrum. Our experiments suggest that GraphTyper is sensitive in comparison to other widely-used structural variant callers. We also show that it can be applied to genotype nearly 50 thousand genomes, which is to our knowledge the largest WGS-based SV set in terms of number of genomes. We believe our approach of realigning reads to a variation-aware graph is important for decreasing mapping bias towards the reference genome and improving genotype quality. Our analysis emphasizes the importance of also applying joint calling when genotyping SVs as it has clear advantages, resulting both in greater sensitivity and lower inheritance error rate.

GraphTyper relies on alignments to the linear reference genome, and thus does not completely fulfill the promise of a pangenome reference. A previous study suggested that using vg's graph alignment prior to GraphTyper could marginally increase the number of true positive SNP and indel calls[27]. Applying a similar strategy would be possible for SV graphs but first we would require high-quality SV population graphs. Unfortunately, these graphs do not exist yet and they would be difficult to construct since SV false discovery rate still remains high. This may change with improvements in sequencing technologies, methods for representing sequence variants with graphs and discovering SVs.

Our method has several other limitations. For instance, GraphTyper's breakpoint model genotypes SVs based on realignments to a SV breakpoint sequence but some SVs can have sequences at the breakpoints that are completely homologous and longer than the read length. For those SVs the reads align ambiguously to both the reference and the SV allele

and the model fails to identify the genotype. Further improvements are needed to be able to genotype those SVs reliably, for example by incorporating read-pair information. Our method also only supports the simple SV types, several types of complex SVs currently cannot be encoded in our graph structure.

We believe that with our method, SVs can be more easily incorporated into large sequencing studies. Here we have demonstrated that our method is sensitive, genotypes SVs accurately and can be robustly applied to very large WGS data sets. Our next goal is to further investigate our Icelandic population SV results and find phenotypic implications of the genotyped SVs. Further, our method has a very low error rate compared to previous methods and thus we have paved a way for high quality *de novo* SVs analysis to further study the origin and mechanics of SVs.

Acknowledgments We are grateful to our colleagues from deCODE genetics / Amgen Inc. for their contributions. We also wish to thank all research participants who provided a biological sample to deCODE genetics.

Author contributions: H.P.E. implemented the GraphTyper software. H.P.E. and S.K. implemented the SV merging software. H.P.E., S.K., P.M. and B.V.H. designed the algorithms. H.P.E., P.M., B.V.H., K.S. designed the experiments. S.K., H.J. and M.T.H. contributed software for the study. H.P.E. wrote the initial version of the manuscript and S.K., D.B., H.J., A.S., M.T.H., D.F.G., P.M., B.V.H. and K.S. contributed to the subsequent versions. All authors reviewed and approved the final version of the manuscript.

Competing financial interests All authors are employees of deCODE Genetics/Amgen, Inc.

References

1. Gudbjartsson, D. F. *et al.* Large-scale whole-genome sequencing of the Icelandic population. *Nat. Genet.* **47,** 435–444 (2015).

2. Francioli, L. C. *et al.* Whole-genome sequence variation, population structure and demographic history of the Dutch population. *Nat. Genet.* **46,** 818–825 (2014).

3. Auton, A. *et al.* A global reference for human genetic variation. *Nature* **526,** 68–74 (2015).

4. Sudmant, P. H. *et al.* An integrated map of structural variation in 2,504 human genomes. *Nature* **526,** 75–81 (2015).

5. Maretty, L. *et al.* Sequencing and de novo assembly of 150 genomes from Denmark as a population reference. *Nature* **548,** 87–91 (2017).

6. Eggertsson, H. P. *et al.* Graphtyper enables population-scale genotyping using pangenome graphs. *Nat. Genet.* **49,** (2017).

7. McKenna, A. *et al.* The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20,** 1297–1303 (2010).

8. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).

9. Chiang, C. *et al.* The impact of structural variation on human gene expression. *Nat. Genet.* **49,** 692–699 (2017).

10. Brandler, W. M. *et al.* Frequency and Complexity of De Novo Structural Mutation in Autism. *Am. J. Hum. Genet.* **98,** 667–679 (2016).

11. Weischenfeldt, J., Symmons, O., Spitz, F. & Korbel, J. O. Phenotypic impact of genomic structural variation: insights from and for human disease. *Nat. Rev. Genet.* **14,** 125–138 (2013).

12. Benonisdottir, S. *et al.* Sequence variants associating with urinary biomarkers. *Hum. Mol. Genet.* (2018). doi:10.1093/hmg/ddy409

13. Chaisson, M. J. P. *et al.* Multi-platform discovery of haplotype-resolved structural variation in human genomes. *bioRxiv* 193144 (2018). doi:10.1101/193144

14. Garrison, E. *et al.* Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.* **36,** 875–879 (2018).

15. Paten, B., Novak, A. M., Eizenga, J. M. & Garrison, E. Genome graphs and the evolution of genome inference. *Genome Res.* **27,** 665–676 (2017).

16. Rakocevic, G. *et al.* Fast and accurate genomic analyses using genome graphs. *Nat. Genet.* 1 (2019). doi:10.1038/s41588-018-0316-4

17. Rausch, T. *et al.* DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* **28,** i333–i339 (2012).

18. Chen, X. *et al.* Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* **32,** 1220–1222 (2016).

19. Layer, R. M., Chiang, C., Quinlan, A. R. & Hall, I. M. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol.* **15,** R84 (2014).

20. Kehr, B., Melsted, P. & Halldórsson, B. V. PopIns: population-scale detection of novel sequence insertions. *Bioinformatics* **32,** 961–967 (2016).

21. Eberle, M. A. *et al.* A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. *Genome Res.* **27,** 157–164 (2017).

22. Parikh, H. *et al.* svclassify: a method to establish benchmark structural variant calls. *BMC Genomics* **17,** 64 (2016).

23. Sibbesen, J. A., Maretty, L. & Krogh, A. Accurate genotyping across variant classes and lengths using variant graphs. *Nat. Genet.* **50,** 1054–1059 (2018).

24. Chiang, C. *et al.* SpeedSeq: ultra-fast personal genome analysis and interpretation. *Nat. Methods* **12,** 966–968 (2015).

25. Sedlazeck, F. J. *et al.* Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods* **15,** 461–468 (2018).

26. Abel, H. J. *et al.* Mapping and characterization of structural variation in 17,795 deeply sequenced human genomes. *bioRxiv* 508515 (2018). doi:10.1101/508515

27. Garrison, E. *et al.* Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nat. Biotechnol.* **36,** 875–879 (2018).

28. Consortium, I. H. G. S. Initial sequencing and analysis of the human genome. *Nature* **409,** 860–921 (2001).

29. Venter, J. C. *et al.* The sequence of the human genome. *Science* **291,** 1304–51 (2001).

30. Jónsson, H. *et al.* Whole genome characterization of sequence diversity of 15,220 Icelanders. *Sci. data* **Accepted,** (2017).

31. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler

transform. *Bioinformatics* **25,** 1754–1760 (2009).

32. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34,** 3094–3100 (2018).

33. Jeffares, D. C. *et al.* Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat. Commun.* **8,** 14061 (2017).

Figure legends

Figure 1: Overview of data structure and workflow. ***a****. Example structural variants and their encoding in an acyclic graph structure.* ***b****. Workflow for constructing a GraphTyper graph with SNPs, indels and SVs. SVs are detected from each sample independently and then merged across all the samples, such that SV sites of the same type and similar position and size are reported only once. SNPs and indels that are given as input into the graph construction can be detected using GraphTyper or obtained from a database.*

Figure 2: Comparisons to svclassify's deletion truth set. The breakpoint precision threshold is the maximum number of allowed difference of begin and end positions between the query and svclassify's high-confidence set. ***a****. Deletion sensitivity of GraphTyper's genotyping models in the Manta+GraphTyper pipeline. Comparison of each SV deletion genotyping model was compared against the svclassify high-confidence deletion set for NA12878. Sequence reads were subsampled to test how coverage affected sensitivity of each genotyping model.* ***b****. Deletion sensitivity comparison between Delly, Manta, Manta+BayesTyper, Manta+GraphTyper, and Lumpy+SVTyper using svclassify's high-confidence deletion set for NA12878 as a truth set.* ***c****. Deletion precision comparison.* ***d****. Deletion F_1-scores comparison.*

Figure 3: High-confidence SV genotypes in four Icelandic families. ***a****. Family tree of the four families. Shown are genotypes of a 313 bp deletion starting at chr20:19,080,772 (GRCh38).* ***b****. Distribution of offspring genotypes (rows) given the genotype of the parents (columns). n is the count of genotypes in each column.* ***c****. Frequency distribution of SVs called on chromosome 20.* ***d****. The allele transmission rate of an SV from parent to offspring. For germline variants, the distribution is expected to be symmetric around 50%.*

Figure 4: Overlap of previously published SV datasets and SVs we find in Iceland. ***a****. Fraction of SVs in an external SV dataset that are also found in Iceland.* ***b****. Distribution of the number of insertions, deletions and breakends of an external dataset that is found in Iceland. Maximum distance threshold used was 50 bp.*

Online Methods

Icelandic DNA data The Icelandic samples were whole-genome sequenced at deCODE Genetics using Illumina GAIIx, HiSeq, HiSeqX and NovaSeq sequencing machines, and sequences were aligned to the human reference genome[28–30] (GRCh38) using BWA-MEM[31].

The 41 Icelandic samples with long-read data were also sequenced using Oxford Nanopore Technologies sequencing machines and basecalled using Albacore (version 2.1.3). The reads were mapped using minimap2[32] (version 2.14). The average alignment coverage was calculated using samtools'[8] depth (version 1.9). Structural variants with at least two supporting reads were discovered using Sniffles[25] (version 1.0.10).

DNA was isolated from both blood and buccal samples. All participating subjects signed informed consent. The personal identities of the participants and biological samples were encrypted by a third-party system approved and monitored by the Data Protection Authority. The National Bioethics Committee and the Data Protection Authority in Iceland approved these studies.

Public DNA data Whole-genome sequence data of NA12878, NA12891 and NA12892 are publicly available from the Platinum Genome project[21]. The data is 101-bp paired-end Illumina HiSeq 2000 reads sequenced to 50x average coverage depth. We aligned the sequence reads to the human reference genome (hg19) using BWA-MEM[31].

Comparison to public truth data The high-confidence svclassify[22] deletion set contains only begin and end coordinates of the deletions but no genotype information. We therefore only tested if the deletion allele was called (genotype 0/1 or 1/1) in the query sets. We allowed an offset of the breakpoint locations as they are not always accurately reported by the SV

discovery tools. The truth set contains only a small fraction of the expected number of deletions in a genome[13] and is likely missing deletions that are harder to discover, e.g. deletions in repetitive regions. The number of deletions we consider true is therefore an under-estimate. However, we believe the truth set serves a purpose in comparing different genotyping methods.

We required both begin and end positions to be within the selected offset for a deletion to be considered recalled. Rather than arbitrarily selecting one offset threshold we compared the methods using different thresholds. We also subsampled the reads using samtools[8] and repeated the experiment using 3x and 10x average coverage.

Merging of SV sites across samples Many SV discovery methods do not joint-call SVs but discover SVs on each sample independently, or on a small number of samples simultaneously. In each sample, the same SV may be reported slightly differently due to imprecise breakpoint resolution. To avoid populating the graph structures with multiple versions of the same SV we created a method for merging SV sites from many single samples VCFs into a single VCF file with all predicted SVs in the population. Our SV merging method is similar to the one used in SURVIVOR[33]. The main difference is that the INFO field from the original VCF is included in our output. Also, our merging method ignores the samples' genotype information to reduce compute time and memory, and only SV site information is needed for the graph construction.

We group all SVs that are of the same type, strand (only applies to single breakpoint SVs and inversions reported as single breakpoints), have a size difference within 100 bp (default value), and where both begin and end position are within 200 bp (default value) of each other. If an SV that fulfills these criteria with any other SV in a group then it is merged into

the group. To prohibit any group from getting extremely large, we disallowed an SV to be added in a group if either its begin or end position is further than 10,000 bp apart from an SV of that group.

When all SVs have been merged into groups, we find the most common pair of begin and end positions and select an SV to be a representative for the group. While merging SVs discovered by Manta we merged all reported SVs, including those that Manta did not set filter to "PASS".

Relative genotype likelihoods All SVs are genotyped independently of each other by comparing how many reads support a given SV breakpoint compared to the reference allele (Supplementary Figure 2). We consider a read to support an allele if its best graph alignment is in a path that overlaps the allele. The input can include multi-allelic SV sites and they are represented as such in the graph. A read might have equally good alignments to more than one allele if a sequence is shared between the alleles. We handle those cases by saying that all those alleles are considered supported because we do not expect to frequently observe two SV events occurring at the same position in the same sample. Therefore, GraphTyper genotypes multi-allelic sites as two or more biallelic sites (reference allele vs. alternative allele).

Given a biallelic SV breakpoint with alleles *x* and *y*, let G_{xy} be the unphased SV genotype of a sample and let *R* be the multiset of the sample's reads that have a graph alignment that overlaps the SV breakpoint. Here, allele 0 denotes the reference allele and allele 1 denotes the alternative allele. As we call each SV as a biallelic variant, the unphased SV genotype can only be G_{00}, G_{01}, or G_{11}.

The relative genotype likelihood of each of those genotypes are:

$$L(R|G_{xy}) = \prod_{r_i \in R} L(r_i|G_{xy}) \quad (1)$$

Where the relative likelihood of observing read r_i given the genotype is:

$$L(r_i|G_{xy}) = \begin{cases} 1 & \text{, if both alleles } x \text{ and } y \text{ support the read} \\ 1/2 & \text{, if exactly one of } x \text{ and } y \text{ support the read} \\ \varepsilon_{r_i} & \text{, if neither alleles } x \text{ nor } y \text{ support the read} \end{cases} \quad (2)$$

Where we arbitrarily chose that ε_{r_i} is $1/2^8$ if the read is paired and its mate mapped onto the graph and $1/2^4$ otherwise. GraphTyper selects the genotype that has the highest relative likelihood for each sample.

We created a genotyping model to estimate genotypes of SV deletions and duplications (including inverted duplications) based on the drop and increase of alignment coverage in the graph, respectively. Each graph alignment is aligned back to the reference haplotype and the alignment coverage is stored at each reference base-pair. To measure the coverage drop or increase, we look-up the alignment coverage every 20 bp and determine the median coverage in two 1,000 bp windows flanking the SV, c_{out}, and median coverage inside the SV, c_{in}. We selected 1,000 bp since it gave us a good estimate of the alignment coverage in a window while being unlikely to overlap other SVs.

For deletions, we say that the coverage decrease, $\max(0, c_{\text{out}} - c_{\text{in}})$, is the number of reads supporting the deletion while c_{in} is the number of reads supporting the reference (i.e. no deletion). We calculate relative genotype likelihoods using Equation 2 with $\varepsilon = 1/2^4$. For duplications the genotype likelihoods are calculated similarly but with coverage increase instead of decrease.

Parent-offspring trio transmission rate We defined the SV transmission rate to be the rate at which an SV is transmitted from a heterozygous parent to his/her offspring. The rate can

be measured for every SV that has at least one heterozygous parent since then his/her SV allele is expected to transmit to the offspring with a probability of 50%, assuming that the allele is present in the germline. The observed mean transmission rate is often below 50% though, due to false positive variants, somatic variants, reference bias and more.

Data availability Access to the raw Icelandic sequence data, that support the findings of this study, is available on request from KS. The data are not publicly available because of Icelandic state law. The short-read sequences for NA12878, NA12891 and NA12892 were obtained from the Platinum Genome project[21] and the deletion truth set for NA12878 was obtained from the Supplementary Information of svclassify's article[22].

Code availability GraphTyper is available at: https://github.com/DecodeGenetics/graphtyper (GNU GPLv3 license). The SV merging software is available at: https://github.com/DecodeGenetics/svimmer (GNU GPLv3 license).

Supplementary data for "GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs"

Hannes P. Eggertsson[1,2], Snaedis Kristmundsdottir[1,3], Doruk Beyter[1], Hakon Jonsson[1], Astros Skuladottir[1], Marteinn T. Hardarson[1], Daniel F. Gudbjartsson[1,2], Kari Stefansson[1,4], Bjarni V. Halldorsson[1,3], Pall Melsted[1,2]

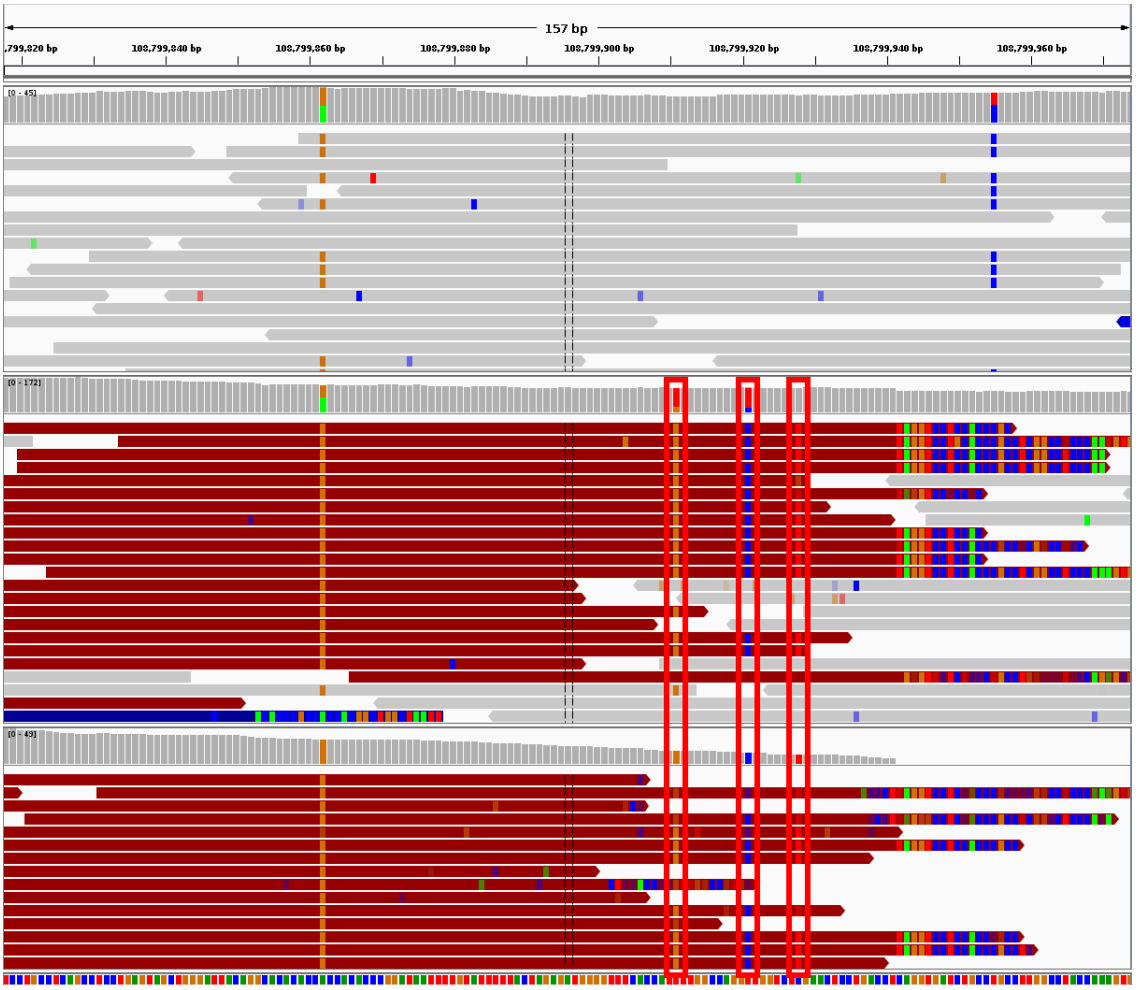
[1] *deCODE genetics/Amgen Inc., Sturlugata 8, Reykjavik, Iceland*

[2] *School of Engineering and Natural Sciences, University of Iceland, Reykjavik, Iceland*

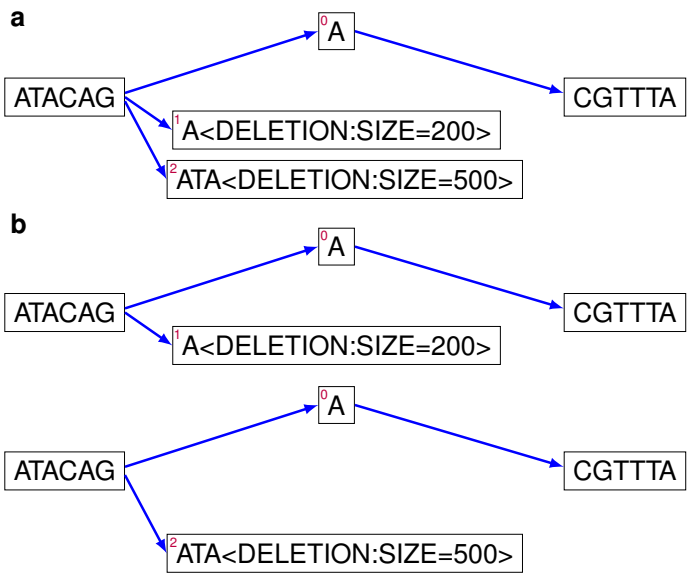
[3] *School of Science and Engineering, Reykjavik University, Reykjavik, Iceland*

[4] *Faculty of Medicine, School of Health Sciences, University of Iceland, Reykjavik, Iceland*

Supplementary Figures



Supplementary Figure 1: Alignments of sequence reads of three Icelanders. A common 1163 bp deletion is at position chr4:108,799,896 (GRCh38). The top sample is not a carrier of the deletion, the sample in the middle is a heterozygous carrier, and the bottom sample is a homozygous carrier. The sequence reads overlapping the deletion breakpoint are mistakenly aligned into the deleted part of the reference such that three SNPs might be falsely genotyped due to misalignments (shown in red squares). The read alignments are visualized using IGV.



Supplementary Figure 2: GraphTyper graph with multi-allelic SV site. **a**. Graph with two SV deletion sites. The two SV alleles are tips in the graph where reads can align into the variant node, but not out of them since the variant node have no outgoing edges. **b**. Each alternative SV allele is compared against the reference allele separately, as if the sites were biallelic.

Supplementary Tables

Supplementary Table 1: Number of validated SVs in Icelanders using Manta+GraphTyper. An SV is considered validated if it passed all filters and has at least two supporting long-reads.

	Validated SVs			High-confidence SVs		
Sample	Deletions	Insertions	Total	Deletions	Insertions	Total
1	3131	2546	5677	4822	3520	8342
2	3290	2713	6003	5072	3468	8540
3	3419	2154	5573	6048	2744	8792
4	3307	2639	5946	5053	3417	8470
5	2939	2393	5332	4781	3428	8209
6	3338	2508	5846	5436	3282	8718
7	3121	2456	5577	5058	3310	8368
8	3449	2761	6210	5113	3445	8558
9	3204	2651	5855	4982	3643	8625
10	3156	2091	5247	5860	2881	8741
11	3319	2700	6019	4779	3421	8200
12	3228	2624	5852	5216	3435	8651
13	3314	2172	5486	5969	2851	8820
14	3318	2798	6116	4723	3568	8291
15	3260	2548	5808	5138	3409	8547
16	3041	2326	5367	5403	3308	8711
17	3295	2627	5922	5555	3470	9025
18	3170	2489	5659	5588	3532	9120
19	3414	2807	6221	4776	3446	8222
20	3271	2782	6053	4716	3541	8257
21	3340	2738	6078	5093	3386	8479
22	3157	2550	5707	5091	3441	8532
23	3126	2556	5682	5100	3445	8545
24	3534	2245	5779	5958	2798	8756
25	3227	2668	5895	4828	3494	8322
26	3308	2726	6034	4822	3468	8290
27	3208	2609	5817	4774	3485	8259
28	3253	2108	5361	5856	2800	8656
29	3329	2801	6130	4698	3497	8195
30	3323	2543	5866	5377	3304	8681
31	3450	2813	6263	5525	3582	9107
32	3181	2666	5847	4817	3520	8337
33	3270	2676	5946	4809	3492	8301
34	3290	2224	5514	5695	2846	8541
35	3357	2710	6067	5370	3398	8768
36	3272	2602	5874	5105	3373	8478
37	3422	2213	5635	5951	2817	8768
38	3218	2584	5802	5352	3428	8780
39	3523	2243	5766	5993	2826	8819
40	3329	2181	5510	5844	2884	8728
41	3080	2313	5393	5522	3233	8755
Average	3272.7	2525.7	5798.4	5260.2	3308.2	8568.4

Supplementary Note

Contents

1 Original GraphTyper publication

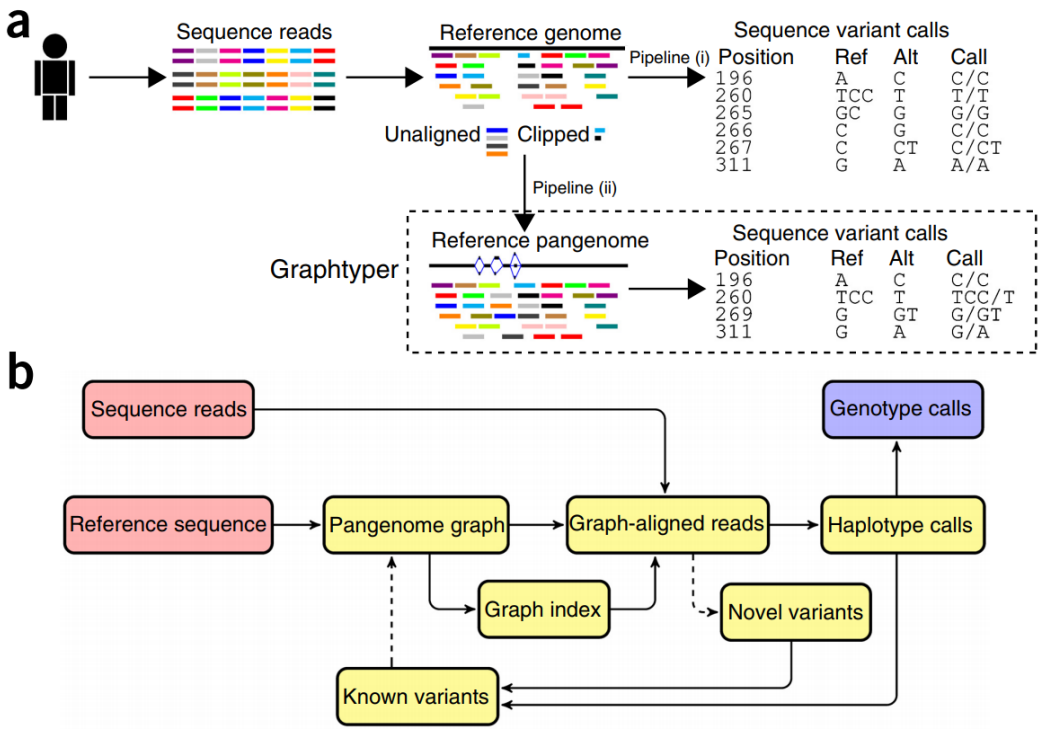
The current publication is an extension of GraphTyper [1], we review its main data structures below.

GraphTyper's main data structure is a directed acyclic graph that encodes a reference pangenome graph, an extension of the traditional linear reference genome (Supplementary Note Figure 1a). In an idealized pangenome graph there exists a path that contains the genome of every sequenced individual. The data structure incorporates known variation to improve sequence alignments and may be able to align unaligned and clipped sequences. In GraphTyper combined sequence alignment and variant calling in a single step, which makes the method scale well with the number of samples. For example when genotyping 15,220 whole-genome samples we measured GraphTyper to take 60% less time than GATK Unified Genotyper [2] on the same samples.

GraphTyper performs variant calling in several iterations (Supplementary Note Figure 1b). During each iteration, GraphTyper updates the pangenome graph structure, and makes final variant calls in the last iteration. In the first iteration the reference sequence is given as input and, optionally, a set of known variants (Supplementary Note Figure 2a). The sequence reads are aligned to the graph and the variation within the graph is genotyped for each sample individually. Optionally, GraphTyper discovers novel SNPs and indels based on the read alignments and adds them to the graph structure in the following iteration.

The graph data structure consists of nodes, which are connected by directed edges. Sequences are associated with the nodes and the edges describe how a possible haplotype could be reconstructed (Supplementary Note Figure 2b). The index data structures con-

tain a key-value store where k-mers are keys and a list of their start and end positions are values, along with any variants they might overlap (Supplementary Note Figure 2c). In the figure we demonstrate how the algorithm works using $k = 5$, however we use $k = 32$ in GraphTyper.



Supplementary Note Figure 1: Genotyping pipeline designs. **a**. Overview of two genotyping pipeline designs. Pipeline (i): a commonly used genotyping pipeline, where sequence reads are aligned to a reference genome sequence and sequence variants are called from discordances between the reads and the reference. Pipeline (ii): Graphtyper's genotyping pipeline. Sequence reads are realigned to a variants-aware pangenome graph and variants are called on the basis of which path the reads align to. **b**. Graphtyper's iterative genotyping process. Dashed paths are optional. As input, Graphtyper requires a reference genome sequence and sequence reads (red) and outputs genotype calls (blue) of variants.

a

Reference sequence: ACCTCCAGACGTTTAGGGACCCCATTGAGTG

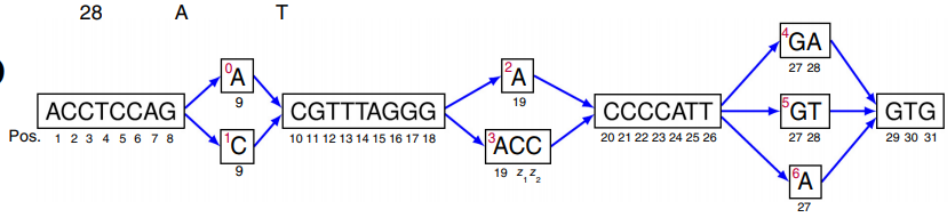
Known variants

Position	Reference	Alternative
9	A	C
19	A	ACC
27	GA	A
28	A	T

Known variants after merge

Position	Reference	Alternative
9	A	C
19	A	ACC
27	GA	GT,A

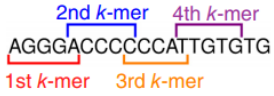
b



c

5-mer	Start pos.	End pos.	Variant ID	Start pos.	End pos.	Variant ID
ACCCC	19	23	2	19	21	3
ACCTC	1	5	NA			
ACGTT	9	13	0			
AGACG	7	11	0			
AGCCG	7	11	1			
AGGGA	15	19	2	15	19	3
...						
CCCAT	21	25	NA			
CCCCA	20	24	NA			
CCCCC	z_1	22	3	z_2	23	3
...						
GGACC	17	z_2	3	17	21	2
GGGAC	16	z_1	3	16	20	2
...						
TTGTG	25	29	5			
...						

d



e

5-mer	Start pos.	End pos.	Variant ID	Start pos.	End pos.	Variant ID
AGGGA	15	19	2	15	19	3
ACCCC	19	23	2	19	21	3
CCCAT	21	25	NA			
TTGTG	25	29	5			

f

AGGGA ACCCC CCCAT TTGTG
AGGGC ACCCA CCCAA TTGTA
AGGGG ACCCG CCCAC TTGTC
AGGGT ACCCT CCCAG TTGTT
AGGAA ACCAC CCCCT TTGAG
AGGCA ACCGC CCCGT TTGCG
AGGTA ACCTC CCCTT TTGGG
AGAGA ACACC CCAAT TTATG
AGCGA ACGCC CCGAT TTCTG
AGTGA ACTCC CCTAT TTTTG
AAGGA AACCC CACAT TAGTG
ACGGA AGCCC CGCAT TCGTG
ATGGA ATCCC CTCAT TGGTG
CGGGA CCCCC ACCAT ATGTG
GGGGA GCCCC GCCAT CTGTG
TGGGA TCCCC TCCAT GTGTG

g

Seed	Start pos.	End pos.	Variant ID
AGGGACCCCCCATTGTG	15	29	3,5
AGGGACCCC	15	23	2

h

Extended longest seed	Start pos.	End pos.	Variant ID
AGGGACCCCCCATTGTGTG	15	31	3,5

Supplementary Note Figure 2: Graphtyper's graph and index data structures and sequence alignment algorithm. **a**. An example reference sequence and its known variation. All overlapping variants are merged. **b** Constructed pangenome reference graph. We draw the path of the reference sequence as the topmost path. **c**. The index data structure with $k = 5$. 5-mers in the graph are mapped to a list of its start position, end position, and a variant ID that it overlaps, if any. **d**. Four k-mers are extracted from a sequence read. Each k-mer overlaps its neighbor k-mer by one character. **e**. An example lookup of the k-mers from the index data structure from c. **f**. All extracted k-mers with a single substitution. **g**. Seeds are generated from matches in the index lookup. **h**. Final graph alignment after extending the longest seed.

2 New features in GraphTyper since its original publication

SV genotyping The main paper describes the addition of SV genotyping in GraphTyper. A few tweaks have been made to the base GraphTyper code to improve the quality when genotyping SVs. Most notably, we have changed the default parameters for read alignments to be considered when genotyping. We now only allow reads to have at most 3% error rate compared to the graph (previously it was 5%), we do not allow reads to a clipped graph alignment at either end, and we do not consider any reads shorter than 90 bp (which may happen if the reads have been shortened by removing adapters).

We have also changed the alignment algorithm slightly, such that if a read partially overlaps an allele, we only count it as supporting if there are at least 5 bp overlapping the allele. Reads that fail on this criterion are considered ambiguous, i.e. they have no effect on the called genotyped using the breakpoint model but may still used in the coverage model.

Other features In addition to enabling population-scale SV genotyping, we have added several new features to GraphTyper since its original publication. The major improvements are noted below.

We have added a new subcommand in GraphTyper called `discover` that will discover variants directly from read alignments of the global read aligner, i.e. BWA-MEM [3]. The subcommand is useful in the very first GraphTyper iteration when the pangenome graph contains only the reference haplotype and thus a graph realignment is not expected to improve the alignment. Since no realignment to a graph is done, the operation is much faster compared to discovering variants with the `call` command. We have also measured

that using `discover` subcommand results in similar or better recall so using instead of the `call` subcommand in the first iteration is highly recommended. We have updated our recommended GraphTyper pipelines (`https://github.com/DecodeGenetics/graphtyper-pipelines`) with the new subcommand.

We have also updated GraphTyper's variant discovery filters to reduce systematic false positive calls. In addition to checking for the number and fraction of reads supporting an alternative allele at a site, we require that for each alternative allele: (1) There must be some support from both read strands (forward and reverse). (2) At least one of the supporting reads must have a base-pair quality of 25 or more in the base-pairs that overlap the alternative allele. (3) There must be at least 2 unique read positions that overlap the variant. (4) There must be support by both first-in-pair reads and second-in-pair reads. (5) There must be at least 3 supporting reads that have a mate that maps to the same graph. We measured that these new criteria removed approximately 67% of non-germline calls (mostly false positive) in our population-scale genotyping, while having almost no effect (<0.1%) on germline recall.

We have also added support for working with any reference genome in GraphTyper. Originally GraphTyper had some hard-coded values that only worked for either the hg19 or the GRCh38 human references, but now we gather these values at graph construction and store them along with the graph. With the new update, contigs are accepted with any name and length. This features makes it possible to use GraphTyper for variant calling on other species than human, as long it has a reference genome available.

3 High-confidence SV filter

We used the following filters on the set of aggregated SVs in the Manta+GraphTyper dataset. For other datasets we used all variant that are flagged "PASS" in the FILTER field of the VCF file. We applied the filters using vcffilter from vcffib (`https://github.com/vcflib/vcflib`).

Breakends filter:

```
QD > 8 & ( SB > 0.4 & SB < 0.6 ) & ( AN < 40 | NHet < 0.55 * AN ) & ( AC /
    NUM_MERGED_SVS < 10 ) & MaxAASR > 0.4 & MaxAAS > 10 & MaxAltPP > 5 & (
    ABHet > 0.3 | ABHet < 0 ) & ABHom > 0.9
```

Deletions filter:

```
QD > 8 & ( ABHet > 0.4 | ABHet < 0 ) & ( AN < 40 | NHet < 0.55 * AN ) & ( AC /
     NUM_MERGED_SVS ) < 50 & SB > 0.2 & SB < 0.8 & ABHom > 0.85 & MaxAASR >
    0.4 & MaxAAS > 10 & MaxAltPP > 5
```

Duplications filter:

```
QD > 8 & ( AN < 40 | NHet < 0.55 * AN ) & ( AC / NUM_MERGED_SVS ) < 50 & SB >
    0.2 & SB < 0.8 & MaxAASR > 0.4 & MaxAAS > 15 & ( ABHet > 0.25 | ABHet < 0
    ) & ABHom > 0.95
```

Insertions filter:

```
( AN < 40 | NHet < 0.55 * AN ) & ( AC / NUM_MERGED_SVS ) < 50 & SB > 0.2 & SB
    < 0.8 & MaxAltPP > 1 & ABHom > 0.95
```

Inversions filter:

```
QD > 8 & ( AN < 40 | NHet < 0.55 * AN ) & ( AC / NUM_MERGED_SVS ) < 50 & SB >
    0.4 & SB < 0.6 & MaxAASR > 0.40 & MaxAAS > 15 & MaxAltPP > 0 & ABHom >
    0.95 & ( ABHet > 0.35 | ABHet < 0 )
```

Command The full command used to filter the VCF containing the aggregated calls was:

```
vcffilter -f "( SVTYPE = BND & QD > 8 & ( SB > 0.4 & SB < 0.6 ) & ( AN < 40 |
    NHet < 0.55 * AN ) & MaxAASR > 0.4 & MaxAAS > 10 & MaxAltPP > 5 & ( ABHet
    > 0.3 | ABHet < 0 ) & ABHom > 0.9 ) | ( SVTYPE = DEL & QD > 8 & ( ABHet >
    0.4 | ABHet < 0 ) & ( AN < 40 | NHet < 0.55 * AN ) & SB > 0.2 & SB < 0.8 &
     ABHom > 0.85 & MaxAASR > 0.4 & MaxAAS > 10 & MaxAltPP > 5 ) | ( SVTYPE =
    DUP & QD > 8 & ( AN < 40 | NHet < 0.55 * AN ) & SB > 0.2 & SB < 0.8 &
    MaxAASR > 0.4 & MaxAAS > 15 & ( ABHet > 0.25 | ABHet < 0 ) & ABHom > 0.95
    ) | ( SVTYPE = INS & ( AN < 40 | NHet < 0.55 * AN ) & SB > 0.2 & SB < 0.8
    & MaxAltPP > 1 & ABHom > 0.95 ) | ( SVTYPE = INV & QD > 8 & ( AN < 40 |
    NHet < 0.55 * AN ) & SB > 0.4 & SB < 0.6 & MaxAASR > 0.40 & MaxAAS > 15 &
    MaxAltPP > 0 & ABHom > 0.95 & ( ABHet > 0.35 | ABHet < 0 ) )" $(
    SV_AGGREGATED_VCF)
```

Filtering SV genotype calls In addition to filtering SV sites, we also created a filter that removes low quality SV genotype calls on a per sample basis at high-confidence SV sites. Both Manta and Manta+GraphTyper genotype calls were filtered using the FT (filter) field of the VCF:

```
vcffilter -g "FT = PASS" $(VCF)
```

The following criteria are used in GraphTyper for each sample:

- All genotyping models must have at least 10 unique reads (reads that do not support

more than one allele).

- If all genotyping models agree on the genotype, the lowest GQ in all models must be above 10.
- If the genotyping models do not agree which genotype to call, the highest GQ must be above 40 and no genotyping model can have a PHRED value above 20 for that genotype.

All criteria must pass such that the genotype call is passed in GraphTyper's filter.

4 Experimental setups

In our experiments we used Manta version 1.4. We compared our method to Delly version 0.7.8, BayesTyper version 1.3.1 and lumpyexpress version v0.2.13 with SVTyper version 0.7.0. All experiments were run on deCODE's computer cluster.

We evaluated the set of SVs that these tools set as "PASS" in the VCF filter field, if available. The exact commands we used are shown below.

Manta + GraphTyper

```
bin/configManta.py --referenceFasta=$(GENOME) --runDir NA12878 --bam NA12878.
   bam
bin/configManta.py --referenceFasta=$(GENOME) --runDir NA12891 --bam NA12891.
   bam
bin/configManta.py --referenceFasta=$(GENOME) --runDir NA12892 --bam NA12892.
   bam
NA12878/runWorkflow.py --mode=local --memGb=110 --jobs=24
NA12891/runWorkflow.py --mode=local --memGb=110 --jobs=24
NA12892/runWorkflow.py --mode=local --memGb=110 --jobs=24
ls NA128*/results/variants/diploidSV.vcf.gz > input_vcfs
python -O merge_sv_vcfs input_vcfs `seq 1 22` | bgzip -c > merged.vcf.gz
tabix merged.vcf.gz
```

We then used the published GraphTyper SV pipeline (`https://github.com/DecodeGenetics/graphtyper-pipelines`) with `merged.vcf.gz` as the `SV_VCF` in the config.

Delly

```
delly_v0.7.8_parallel_linux_x86_64bit call --genome=$(GENOME) --outfile=1_78.
   bcf --exclude human.hg19.excl.tsv NA12878.bam
delly_v0.7.8_parallel_linux_x86_64bit call --genome=$(GENOME) --outfile=1_91.
   bcf --exclude human.hg19.excl.tsv NA12891.bam
delly_v0.7.8_parallel_linux_x86_64bit call --genome=$(GENOME) --outfile=1_92.
   bcf --exclude human.hg19.excl.tsv NA12892.bam
delly_v0.7.8_parallel_linux_x86_64bit merge -o sites.bcf 1_78.bcf 1_91.bcf 1
   _92.bcf
delly_v0.7.8_parallel_linux_x86_64bit call --vcffile=sites.bcf --genome=$(
   GENOME) --outfile=2_78.bcf --exclude human.hg19.excl.tsv NA12878.bam
delly_v0.7.8_parallel_linux_x86_64bit call --vcffile=sites.bcf --genome=$(
   GENOME) --outfile=2_91.bcf --exclude human.hg19.excl.tsv NA12891.bam
delly_v0.7.8_parallel_linux_x86_64bit call --vcffile=sites.bcf --genome=$(
   GENOME) --outfile=2_92.bcf --exclude human.hg19.excl.tsv NA12892.bam
bcftools merge -m id -O z -o final_results.vcf.gz 2_78.bcf 2_91.bcf 2_92.bcf
tabix final_results.vcf.gz
```

Manta

```
bin/configManta.py --referenceFasta=$(GENOME) --runDir=joint --bam=NA12878.bam
    --bam=NA12891.bam --bam=NA12892.bam
joint/runWorkflow.py --mode=local --memGb=110 --jobs=24
```

BayesTyper

We used the BayesTyper Snakemake workflow with the same three BAM files. The workflow was configured with Genome Analysis ToolKit version 3.6, Platypus (commit: cbbd9146183a2aba5f4884df36fbd58988133150), Manta version 1.4.0, bcftools 1.5 and

KMC version 3.1.0.

Lumpy + SVTyper

```
lumpyexpress -B NA12878.bam,NA12891.bam,NA12892.bam -o final.vcf
svtyper -i final.vcf -B NA12878.bam,NA12891.bam,NA12892.bam -T $(GENOME) -o
    final.gt.vcf
```

Sniffles

```
sniffles --report_seq --ignore_sd -l 30 -d 1000 -s 2 -m $(INPUT_BAM) -v $(
    OUTPUT_VCF) -t 24 --num_reads_report 30 --genotype
```

We then post-filtered SVs that had breakpoints of different chromosomes and were smaller than 50 bp.

5 Evaluations

Checking overlap with external SV datasets We used our merging program for joining a truth SV set to a query SV callset using the `--join-mode` option. The 1000G SVs were obtained from: `http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/integrated_sv_map/supporting/GRCh38_positions/ALL.wgs.mergedSV.v8.20130502.svs.genotypes.GRCh38.vcf.gz`. The file contains sites after their positions had been lifted over to GRCh38. The GoNL SVs were obtained from: `https://molgenis26.target.rug.nl/downloads/gonl_public/variants/release6.1/20161013_GoNL_AF_genotyped_SVs.vcf.gz` and lifted over from build 37 to 38. The Abel *et al.* SVs were obtained from the follow zip file: `https://www.biorxiv.org/content/biorxiv/early/2018/12/31/508515/DC1/embed/media-1.zip?download=true`.

The following command was used to check overlap between our SV dataset and the external datasets:

```
merge_sv_vcf <(echo $(EXTERNAL_VCF); echo $(GraphTyper_VCF)) --max_distance $(
    DISTANCE) --max_size_difference -1 --ignore-types --join-mode | bgzip -c >
     $(OUTPUT)
tabix $(OUTPUT)
```

Variants that have `NUM_JOINED_SVS` greater than 1 are SVs in external dataset but are also found in Iceland. The following values for `$(DISTANCE)` were tested in our analysis: 1, 3, 5, 7, 9, 11, 13, 15, 17, 20, 23, 26, 30, 35, 40, 50, 70, and 100. We needed to use the --ignore-types option since there were many inconsistencies in the classifications of SV types between datasets. For example, in the 1000G SV set there are many deletion-

s/duplications classified as "CNV" while the other datasets separate these. When filtering based on allele frequency we used the `EUR_AF` in the 1000G dataset and AF in the other datasets.

Long-read validation We also used our merging program for joining GraphTyper SVs with Sniffles SVs.

```
merge_sv_vcf <(echo $(GraphTyper_VCF); echo $(SNIFFLES_VCF)) --max_distance 50
    --max_size_difference -1 --join-mode | bgzip -c > $(OUTPUT)
tabix $(OUTPUT)
```

Variants that have `NUM_JOINED_SVS` greater than 1 are considered validated SVs. In this analysis we made sure that the type of SV matched between datasets, i.e. the `--ignore-types` option was not used.

6 External tools and dependencies

External libraries GraphTyper has the following library dependencies:

- args (`https://github.com/Taywee/args`): Argument parser.
- Boost (`https://www.boost.org/`).
- Catch (`https://github.com/philsquared/Catch`): Framework for unit tests.
- htslib (`https://github.com/samtools/htslib`): Library for HTS data formats.
- paw::Station (`https://github.com/hannespetur/paw`): Multi-threading wrapper library.
- RocksDB (`https://github.com/facebook/rocksdb`): Key-value storage.
- SeqAn[4] (forked version, `https://github.com/hannespetur/seqanhts`): Library for sequence analysis.
- Snappy (`https://github.com/google/snappy`): Compression library.
- SparseHash (`https://github.com/sparsehash/sparsehash`): Hash map containers.
- StatGen (`https://github.com/statgen/libStatGen`): Statistical genetic library.
- zlib (`http://www.zlib.net/`): Compression library.

External programs In addition to the tools we evaluated, we used the following tools in our experiments:

- bamShrink (`https://github.com/DecodeGenetics/bamShrink`): Description below.
- chopBai[5]: Partitions bam index files.
- samtools[6]: Manipulates SAM formatted files.

- vcflib (`https://github.com/vcflib/vcflib`): Manipulates VCF files.
- vt (`http://genome.sph.umich.edu/wiki/Vt`): Manipulates VCF files.

bamShrink bamShrink was run before genotyping with GraphTyper. bamShrink extracts sequence reads of a region and reduces the output file size by binarizing base qualities values, removing unused BAM tags, removing unaligned reads, duplicate reads and reads that have fewer than 40 matching bases in their alignment. In addition to this, bamShrink performs coverage filtering in regions where the coverage is more than 3 times the average coverage, removes Ns if present on either end of a read and removes hard clipped entries from CIGAR strings. Lastly, bamShrink performs adapter removal by clipping overhanging ends of read pairs where the reverse read has been aligned in front of the forward read and their alignments overlap.

Supplementary References

1. Eggertsson, H. *et al.* Graphtyper enables population-scale genotyping using pangenome graphs. *Nature Genetics* **49** (2017).

2. McKenna, A. *et al.* The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research* **20**, 1297–1303 (2010). URL `http://genome.cshlp.org/cgi/doi/10.1101/gr.107524.110.arXiv:1011.1669v3`.

3. Li, H. & Durbin, R. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).

4. Döring, A., Weese, D., Rausch, T. & Reinert, K. SeqAn an efficient, generic C++ library for sequence analysis. *BMC bioinformatics* **9**, 11 (2008).

5. Kehr, B. & Melsted, P. chopBAI: BAM index reduction solves I/O bottlenecks in the joint analysis of large sequencing cohorts. *Bioinformatics* **32**, 2202–2204 (2016).

6. Li, H. *et al.* The sequence alignment/map format and SAMtools. *Bioinformatics* **25(16)**, 2078–2079 (2009).