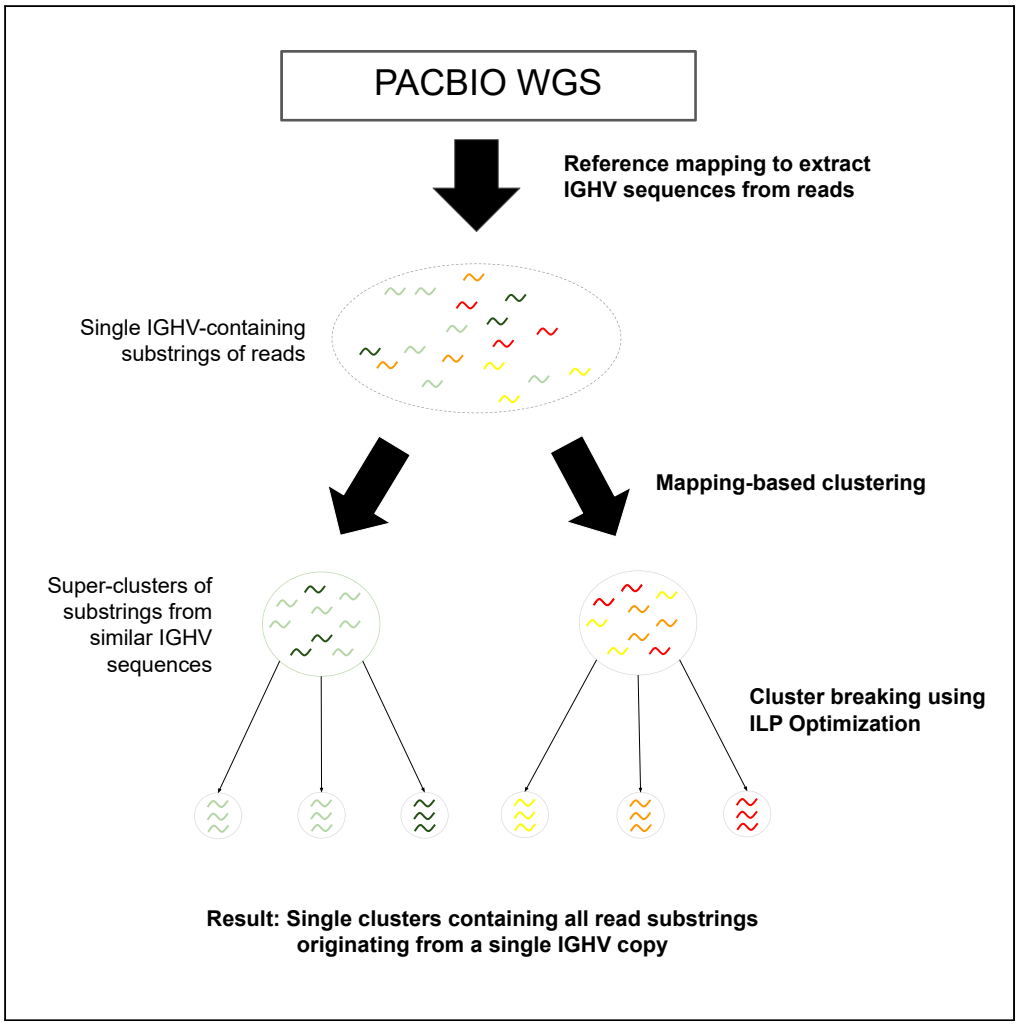
iScience

CellPress

Article

Genotyping and Copy Number Analysis of Immunoglobin Heavy Chain Variable Genes Using Long Reads



Michael Ford, Ehsan Haghshenas, Corey T. Watson, S. Cenk Sahinalp

cenk.sahinalp@nih.gov

HIGHLIGHTS

We describe ImmunoTyper, a WGS Immunoglobulin Heavy Chain Variable Genotyping tool

Immunotyper is the first such tool to use long reads and call alleles for pseudogenes

We demonstrate high allele call accuracy using simulated and real WGS data

Ford et al., iScience *23*, 100883
March 27, 2020
https://doi.org/10.1016/j.isci.2020.100883

iScience

CellPress

Article

Genotyping and Copy Number Analysis of Immunoglobin Heavy Chain Variable Genes Using Long Reads

Michael Ford,[1] Ehsan Haghshenas,[1] Corey T. Watson,[2] and S. Cenk Sahinalp[3,4,*]

SUMMARY

One of the remaining challenges to describing an individual's genetic variation lies in the highly heterogeneous and complex genomic regions that impede the use of classical reference-guided mapping and assembly approaches. Once such region is the Immunoglobulin heavy chain locus (IGH), which is critical for the development of antibodies and the adaptive immune system. We describe ImmunoTyper, the first PacBio-based genotyping and copy number calling tool specifically designed for IGH V genes (IGHV). We demonstrate that ImmunoTyper's multi-stage clustering and combinatorial optimization approach represents the most comprehensive IGHV genotyping approach published to date, through validation using gold-standard IGH reference sequence. This preliminary work establishes the feasibility of fine-grained genotype and copy number analysis using error-prone long reads in complex multi-gene loci and opens the door for in-depth investigation into IGHV heterogeneity using accessible and increasingly common whole-genome sequence.

INTRODUCTION

With the advent of modern, high-speed bioinformatics tools and high-throughput sequencing, reconstructing a human genome has gone from being one of the big challenges in genomics to standard protocol. Despite being a routine step in modern bioinformatics pipelines, there remains parts of the genome that are difficult to reconstruct using standard techniques. One such region is the immunoglobulin heavy chain locus (IGH), whose genes encode the foundation to the structure and development of antibodies. Although IGH genes are critical to the structure and function of the adaptive immune system of vertebrates, performing genotyping and copy number analysis of IGH genes remains challenging owing to the complexity of the region, which is one of the most dynamic regions of the human genome (Watson and Breden, 2012).

Of the four classes of coding gene segments present in the IGH region, the Variable genes class (IGHV) plays a critical role in defining epitope binding affinity, as it completely contains two and partially contains the last of the three complementary-determining regions. However, many of the IGHV alleles are highly similar (see Figure 1), which in combination with their short length of between 165 and 305 bp (mean of 291 bp) and the high number in an individual (can be greater than 50 functional genes [Watson et al., 2013; Matsuda et al., 1998]), makes the problem of IGHV genotyping challenging. To further complicate the problem, the IGH region has been shown to contain many large structural variants (SVs), including segmental duplications, large insertions and deletions, and other copy number variants (CNVs) (Watson et al., 2013). Finally, there are two non-functional orphons of IGH (on chromosomes 15 and 16) that have similar sequence to IGH (Lefranc, 2001a). As a result, classical reference-based mapping approaches to IGH analysis typically perform poorly (see Figure 2).

To date there have been two attempts at IGHV genotyping using high-throughput sequence from germline DNA-sourced materials, both focused exclusively on functional genes. For clarity, we consider a successful IGHV genotyping result to report all the IGHV genes present in a given sample and report the allele for every copy of every IGHV gene. Work by Yu et al. (2017) created a whole-genome sequencing (WGS) Illumina short read analysis pipeline for identification of IGHV and T cell receptor sequence using a reference mapping-based variant calling and frequency thresholding. Although the results of their paper are initially impressive, with 8,750 novel IGHV sequences having been found, there have been doubts raised regarding the accuracy of the findings by others in the field (Watson et al., 2017; Boyd et al., 2010; Kidd et al., 2012; Gidoni et al., 2019). One of the main criticisms is the reliance on a genome reference. The high degree of

[1]School of Computing Science, Simon Fraser University, Burnaby V5A 1S6, Canada

[2]Department of Biochemistry and Molecular Genetics, University of Louisville, Louisville 40292, USA

[3]Cancer Data Science Laboratory, National Cancer Institute, Bethesda, MD 20892, USA

[4]Lead Contact

*Correspondence: cenk.sahinalp@nih.gov

https://doi.org/10.1016/j.isci.2020.100883

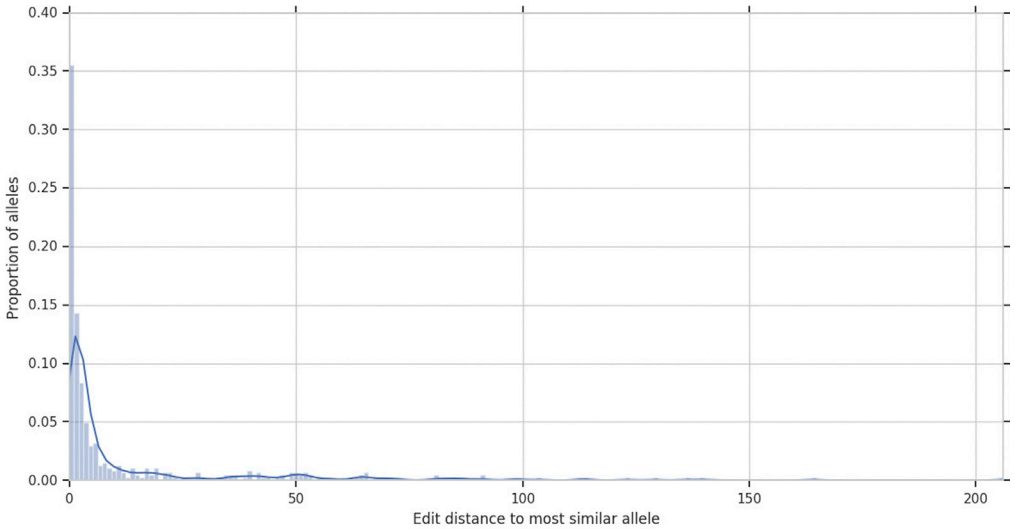


Figure 1. Histogram of the Edit Distance between Each Allele from the IGHV (Pseudo)Gene Database and its Most Similar Allele (with Respect to Edit Distance)

haplotype diversity mentioned above means that any reads that may originate from an insertion or novel sequence in the IGH region, relative to the mapping reference, will be missed from the pipeline.

The other work on IGHV genotyping using germline sequence data has been done by Luo et al. (2016, 2019), also using WGS Illumina short read data. Although their initial work also relied on whole reference genome mapping, without addressing possible novel insertion sequence, their later work avoided this pitfall by mapping short reads directly to IGHV reference sequences. This method focuses on gene identification and copy number calling. However, their method calls alleles only for 11 functional genes, as they identify these as only having a single copy per chromosome. Additionally, there are seven groups of genes, each of which is a set of genes they are not able to differentiate owing to high sequence similarity.

One increasingly popular approach to investigating the variations within the genes of the IGH region is through genotype and haplotype inference, using repertoire sequencing data. Although the analysis of germline sequencing data is challenging, gathering sequencing data on expressed IGH sequences, typically called Adaptive Immune Receptor Repertoire sequencing (AIRR-seq), is commonplace, has established protocols, and can easily be sequenced to a high depth (Vander Heiden et al., 2018). The availability and quality of these data make it an appealing source to infer and investigate the germline sequence; however, owing to the nature of IGH sequence expression this is not straightforward. An IGH mRNA sequence, as expressed by a B cell, is not only different from the germline sequence owing to VDJ recombination, but has potentially also undergone somatic hypermutation, which introduces new variants relative to the germline sequence. However, despite these challenges, there have been numerous published studies and tools that have investigated the IGHV germline sequence through repertoire sequencing inference and have been successful at identifying novel IGHV alleles and features (Gadala-Maria et al., 2015, 2019; Boyd et al., 2010; Corcoran et al., 2016; Ralph and Matsen, 2016; Thörnqvist and Ohlin, 2018). There has additionally been work done on haplotype inference through statistical learning frameworks, using the IGHJ genotype (Kirik et al., 2017; Kidd et al., 2012) and/or IGHD genotype (Gidoni et al., 2019) as an IGHV haplotype indicator.

However, it has been noted that there are challenges to performing IGHV germline analysis through repertoire inference. For example, recent work has demonstrated that inferring some IGHV variants can be nearly impossible because of the unpredictable removal of 3′ bases during VDJ recombination or be particularly hard to overcome at regions of "mutational hotspots" (Kirik et al., 2017). Additionally, it has been shown that the initial reference database used can affect the reliability of inference calls for alleles that are highly similar (Kirik et al., 2017).

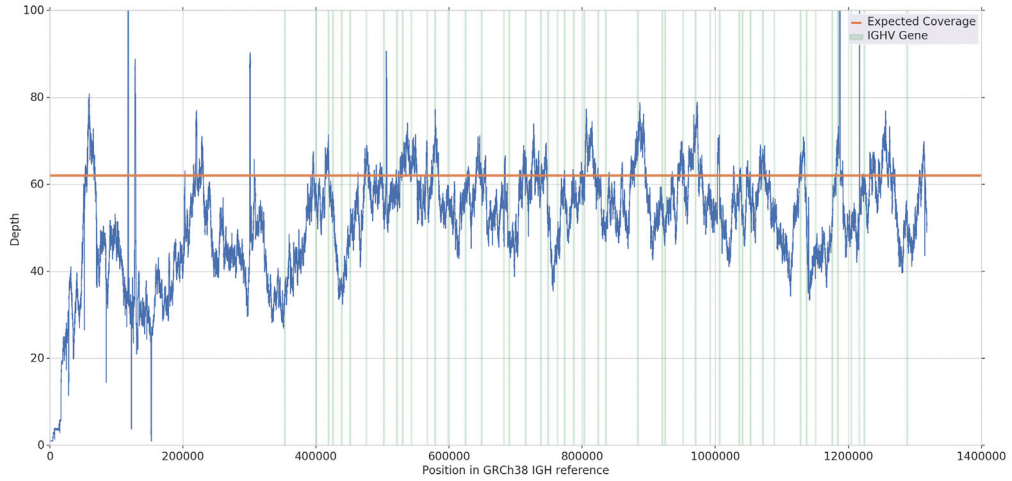


Figure 2. Read Depth of IGH Region for CHM1 WGS PacBio Reads Mapped to CHM1 Reference Using minimap2 with Default Parameters, Demonstrating Significant Deviation from the Expected Coverage, Including at Positions Containing IGHV Genes, Which Are Marked by Vertical Green Lines

Another inherent challenge to IGHV inference is the effect of non-uniform expression of certain VDJ configurations. This effect can be additionally complicated by the types and ratios of B cells that are sequenced. Fundamentally, since inferring the presence of some allele is dependent on the allele being expressed, the lack of some allele does not indicate its absence in the germline sequence. This means that, although inference may result in the identification of confident true positives, true negatives are impossible to differentiate from false negatives. Additionally, since the repertoire is adaptive and dynamic, some method to account for possible temporal biases to expression ratios is necessary to confidently make claims regarding the general functional significance of the presence or absence of any given allele. The effect of expression bias is also particularly relevant to haplotype inference, whose reliance on gene usage estimates can be directly confounded by expression bias (Gidoni et al., 2019).

Although inference techniques have made significant progress at genotyping despite the challenges, there has been little work done on the other major sources of IGH heterogeneity, namely, SVs and CNVs. These variants are expected to be common, as work by Watson et al. (2013) has discovered several large-scale insertions and deletions in the IGH region, each containing multiple IGHV genes. However, this work was done using Sanger sequencing of BAC and fosmid clones, which is prohibitively expensive and time consuming. Haplotype inference has had some success at CNV calling, deletion detection, and even phased haplotype calling (Gidoni et al., 2019; Kidd et al., 2012); however, it is limited by gene expression bias as noted above. The work by Luo et al. includes copy number calls but does not call alleles for genes with CNVs, thus missing a critical step in the path toward complete haplotype calling.

Another large gap in our knowledge about IGH heterogeneity are non-coding sequence variants. Non-coding sequence is already known to play a critical role in the antibody repertoire as it contains the recombination signal sequence, which is required for V(D)J recombination (Janeway et al., 2001). However, limitations in methodology have inhibited investigation into possible further effects through mechanisms such as enhancers and promoters.

Identification of novel IGH and IGHV sequences, genes, and alleles is an important problem, as it has been noted that the primary database for IGH gene reference sequences, hosted by the international ImMunoGeneTics information system (IMGT) (Lefranc et al., 2015), is incomplete (Ohlin et al., 2019), and the complexity of the IGH locus is likely to lead to high sequence heterogeneity across individuals and populations. However, there is still a need for fast IGHV genotyping of known alleles using common data types that are not specific to IGH research. Such tools can be integrated into standard precision medicine pipelines, allowing for investigations such as disease association studies to be done with larger sample sizes. Although the performance of IGHV genotyping tools may suffer initially depending on their degree of

Sample	# IGHV Occurrences in Reference	# IGHV Calls	Precision	Recall	True Positive	False Positive	False Negative
CHM1 (simulated)	117	111	94.6%	89.7%	105	6	12
GRCh37 (simulated)	112	109	97.2%	94.6%	106	3	6
CHM1 + GRCh37 (simulated)	229	227	94.3%	93.4%	214	13	15
CHM1 WGS	117	110	87.3%	82.1%	96	14	21

Table 1. Genotype Results for Simulated and CHM1 Real Data Samples

reliance on established IGHV reference databases, they will increase in accuracy as databases become more complete over time.

In this paper we present *ImmunoTyper*, an IGHV genotyping and CNV calling tool that is the first to be based on long read data. By using long read data we ensure that reads span the complete IGHV coding region, and they provide information from non-coding regions, at the cost of increased sequencing error rate over short read technologies. In order to avoid the gene expression biases found in inference-based methods, it utilizes WGS to provide a complete picture of the IGHV germline landscape. Although ImmunoTyper in its current implementation is solely for rapid genotyping of known IGHV alleles, several of its design features, such as allele identification using ambiguity instead of identity, can allow for implementation of novel allele discovery in future versions of the tool. Finally, ImmunoTyper is the first IGH-specific tool to report non-coding sequence by providing high-quality sequence for regions flanking IGHV genes, as well as the first to provide allele and CNV calling for the vast majority of IGHV pseudogenes.

RESULTS

Owing to the lack of published IGH germline sequences, our ability to validate allele calls and copy number variants is limited. As a result, we performed experiments using simulated data using both the GRCh37 and GRCh38 references, which are the only published complete IGH sequences. Since the GRCh38 IGH reference is derived from the CHM1 hydatidiform mole haploid genome (Watson et al., 2013), we were also able to perform tests with real data using publicly available WGS data for CHM1. For clarity, we used CHM1 instead of GRCh38 to reference this sample.

Simulated Data

Simulated data experiments were set up with the goal of testing the ImmunoTyper method, without the confounding effects of unavoidable noise inherent in WGS datasets.

For generating the simulated data, we first extracted the IGHV genes and pseudogenes, along with 1-kbp flanking regions, from the GRCh37 (NCBI NC_000014.8:106031614-107289051) and CHM1 (NCBI NC_000014.9:105586437-106880844) references using the NCBI GenBank annotations (Clark et al., 2015). Next, we discarded all sequences corresponding to alleles that are ignored (as described in Transparent Methods). We simulated the reads from the IGHV-containing sequences at 20x using Simlord (Stöcker et al., 2016) in single-pass configuration, resulting in a 15.8% mean total error rate. This resulted in 2,360 reads for the CHM1 sample and 2,236 for the GRCh37 sample. The reads are simulated so that their length matches the length of the extracted sequences (2,300 bp) to emulated extracted subreads from a WGS sample. The resulting sets of reads were then combined and provided as input to ImmunoTyper. The option "`--no-coverage-estimation`" was used to skip the subread coverage estimation step described in Transparent Methods, and use the user provided depth parameter of 20x. For the CHM1 and GRCh37 samples, 1,524 and 1,323 of the inputs reads, respectively, were identified as ambiguous and assigned in the second stage of the pipeline.

In addition to these simulated haploid runs, the subreads from both samples were combined to create a set of 4,596 reads that simulate a diploid sample. Of the input reads, 2,760 were identified as ambiguous.

Results are shown in Table 1, where ImmunoTyper demonstrates strong results in all simulated samples, with precision and recall above 94%, with the exception of 89% recall in the simulated CHM1 sample. Note that the results in Table 1 are for all functional IGHV genes and non-functional IGHV pseudogenes.

Sample	Expected Read Error	Median Mapping Error
CHM1 (simulated)	15.8%	2.0%
GRCh37 (simulated)	15.8%	2.0%
CHM1 + GRCh37 (simulated)	15.8%	2.2%
CHM1 WGS	16.19%[a]	2.3%

Table 2. Allele Sequence Error Reduction Results

[a]Taken from Laehnemann et al., 2015.

Additionally, in all cases except GRCh37 ImmunoTyper was able to successfully differentiate alleles that were distinguished by only a single SNP (see section Investigation into False-Positive Allele Calls and Figures S5–S7). Note that True Pos indicates the allele was called by ImmunoTyper and was present in the sample, False Pos indicates the allele was called by ImmunoTyper but was not in the sample, and False Neg indicates the allele was not called by ImmunoTyper but was present in the sample.

WGS Data with Validation

ImmunoTyper was tested on the publicly available CHM1 PacBio sequence (62x coverage; SRA: SRX1164774) (Chaisson et al., 2015), and the resulting allele calls were validated as with the simulated CHM1 data. A total of 7,772 reads were extracted from the WGS sample, 3,131 of which contained at least one complete IGHV gene with flanking sequences, resulting in 5,176 subreads; 1,431 were identified as ambiguous. Table 1 shows that ImmunoTyper successfully genotypes the WGS CHM1 sample with reasonable precision and recall values of 87% and 82%, respectively, and is able to successfully differentiate alleles that have as few as four distinguishing SNPs (see section Investigation into False-Positive Allele Calls and Figure S8).

Sequence Recovery and Reference Mapping

To further evaluate the performance of ImmunoTyper in subread error reduction, consensus sequences (including coding and non-coding flanking sequences) from all clusters were mapped back to their reference sequence using minimap2 (Li, 2018) with default parameters. As shown in Table 2, ImmunoTyper reduces the median sequence error rate by at least 86% from the raw read error rate. Visualizations of the distribution of error reduction can be found in Figures S1–S4. Note that the expected error rate for PacBio reads is taken from Laehnemann et al., (2015).

Investigation into False-Positive Allele Calls

In order to investigate whether sequence similarity is a major contributor to false-positive allele calls, for each sample we plot the number of false-positive alleles against the number of SNPs that distinguish them from their most similar allele in the sample. We also include true positives in the plot to provide context for the minimum number of variants ImmmunoTyper needs to successfully differentiate and call alleles. The plots can be found in Figures S5–S8.

Identification of Sequence Differences between GRCh37 and CHM1 References

The GRCh37 and CHM1 references have significant difference in sequence and IGHV gene composition. The two references together contain four of the six known IGH insertion sequences listed in IMGT and partially cover a fifth (Lefranc et al., 2015; Clark et al., 2015; Lefranc, 2001b, a). In Table 3, we provide the IGHV genes and pseudogenes contained in each insertion sequence, as well as list the source reference and an individual identifier.

The simulated diploid sample is the most suited to evaluate ImmunoTyper's ability to identify inserted sequence as it covers the most amount of insertions. Table 4 provides a summary of the gene and allele calls for IGHV genes and pseudogenes belonging to inserted sequence. ImmunoTyper was able to call the presence and correctly identify the alleles 12 of 14 genes and pseudogenes contained in the inserted sequences, demonstrating the ability to identify known insertion sequences in a sample. The missing allele calls were likely lost owing to high coding and flanking sequence similarity with other genes in the region (89% and 88% sequence identity for 3-69-*01 and 3-71*01; 1-8*01 and 1-69*06, respectively).

Insertion Identifier	Reference	Genes and Pseudogenes Present and Their Alleles
A	CHM1	1-69*06, 1-69-2*01, 2-70D*04, 3-69-1*01
B	GRCh37	4-31*02, (II)-31-1*01
C	CHM1	(II)-30-21*01, 4-30-2*01
D	CHM1	3-64D*06, 5-10-1*03
E	GRCh37	3-9*01, 2–10*01, 1–8*01
F	CHM1	7-4-1*01

Table 3. Sequence Differences between CHM1 and GRCh37 References

CNV Analysis

There are several IGHV genes in the GRCh37 and CHM1 references that are present with multiple copies. The greatest number of CNVs are present in the GRCh37 + CHM1 diploid sample, and ImmunoTyper's results for calling all CNV genes in the sample are summarized in Table 5. ImmunoTyper accurately calls the copies and alleles for the CNV genes in the sample in all cases except for 1-69, where the incorrect calls are likely a result of the extreme challenge of differentiating the *01 and *06 alleles as they differ by a single base pair. The 4-31 gene is included despite having a copy number of 2, because the second copy (4-30-2) is due to a duplication in the B insertion sequence in GRCh37, rather than diploidy.

DISCUSSION

ImmunoTyper represents a generalizable approach to multigene genotyping and copy number analysis. The results described above, although limited in sample size, provide robust validation of the methodology against publicly available genotype calls that have been produced through gold-standard approaches.

In addition to accurate genotyping results with high precision and recall, the low mapping error rates described in section Sequence Recovery and Reference Mapping demonstrate the success of our clustering approach, especially considering the high error rates of the source reads and moderate sequencing depth. However, it is clear that complete IGHV genotyping using long reads is especially difficult. ImmunoTyper under-reported the number of IGHV genes present in the CHM1 WGS sample, likely because of variation in the sequencing depth or IGHV-containing subread dropout due to subreads not being identified as a result of high sequence error. Subread dropout and potential noise from mistakenly including subreads from elsewhere in the genome, such as the 2 IGH orphons, are also likely explanations of the difference seen in the results of the CHM1 WGS and CHM1 simulated samples, in addition to the unavoidable shortcomings of simulating sequencing data. There also remain a few outlying cases in all samples where the allele call was incorrect and/or the sequence recovery had a high number of errors. Given the proportion of IGHV alleles that have a high degree of sequence similarity, it may be exceedingly difficult, if not impossible, to achieve perfect genotyping and CNV calls using error-prone long reads without reducing the sequence error rate through a method such as CCS reads or increasing the sequencing depth.

Insertion	Reference	Number of Genes and Pseudogenes	Number of Matching Genes in Result	Number of Correct Allele Calls	Missing Allele Calls
A	CHM1	4	3	3	3-69-1*01
B	GRCh37	2	2	2	
C	CHM1	2	2	2	
D	CHM1	2	2	2	
E	GRCh37	3	2	2	1-8*01
F	CHM1	1	1	1	

Table 4. IGHV Identification in Insertion Sequences Between GRCh37 and CHM1 in Diploid Sample

Gene	Number of Copies in Sample	Number of Copies Copies Called	Correct Allele Calls	False-Positive Calls	False-Negative Calls
1-69	4	5	1-69-2*01, 1-69*06, 1-69*06	1-69*06, 1-69*06	1-69*01
2-70	3	3	2-70*01, 2-70D*04 2-70*13		
3-64	3	3	3-64*02, 3-64D*06 3-64*02		
4-31	2	2	4-30-2*01, 4-31*02		

Table 5. Calls for Known CNV Genes in the CHM1 + GRCh37 Sample

In addition to identifying known IGHV alleles, ImmunoTyper also provides an opportunity to discover novel sequences through the following features. First, the *Mapping-based clustering* step clusters reads based on ambiguity rather than on allele sequence similarity. This allows for reads originating from a novel allele to be clustered with the closest matching allele in the database. Super-clusters also account for novel alleles, as they are formed solely based on read-to-read sequence similarity and are therefore not dependent on the known allele database. Finally, the *non_code_cov_var* error function acts as a reference-free counterbalance to *code_var_cov* error function, as it is independent of allele references and influences clustering based on read-to-read similarity, under the constraints of variant depth. As a result, the user is able to call novel alleles using the output consensus sequence for each IGHV gene. However, owing to the challenge of calling novel alleles using long reads, especially if they differ significantly from known alleles, ImmunoTyper is focused on known allele calling.

In addition to IGH, there are other regions of the genome where ImmunoTyper could be applied with minimal modification. In particular, the immunoglobulin κ and λ light chain loci and the T cell receptor loci are related to IGH in that they all share a similar multi-gene segment construction and undergo V(D)J recombination (Janeway et al., 2001). Luo et al. (2019) have taken this approach by applying their tool to the T cell beta variable locus. Extending the protocol to these similar regions is an accessible opportunity to investigate lesser-studied regions of the genome, given the current configuration of ImmunoTyper.

Fundamentally, ImmunoTyper is the first IGHV genotyping tool to use error-prone long reads, the first to integrate pseudogene calls, and the first to provide data on non-coding sequence that flanks IGHV genes. Although it is developed specifically for IGHV analysis, the approach and the integer linear programming formulation for allele assignment is generalizable to any multi-gene genotyping and copy number analysis problem with known alleles.

Although this initial investigation was intentionally limited to samples that have published gold-standard references, the results make us confident that ImmunoTyper represents the closest attempt at complete IGHV genotyping using WGS data to date.

Limitations of the Study

By limiting our testing of ImmunoTyper to samples with published gold-standard references, we can be confident in the accuracy of our results; however, that comes at the cost of a certain degree of generalizability. We can speculate that there may exist IGH haplotypes that have combinations of IGHV alleles, either previously described or novel, which are challenging for ImmunoTyper to accurately identify. However, in the absence of further complete IGH haplotypes or alternative validation methods to compare ImmunoTyper with, we are limited in our ability to significantly test ImmunoTyper beyond what has been demonstrated in this paper.

METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

DATA AND CODE AVAILABILITY

The datasets used for obtaining the results of this article can be retrieved from the Sequence Read Archive (SRA) via accession number SRA: SRX1164774. These datasets have been published in an article by Chaisson et al. (2015). The instructions to generate simulated data used in this article can be found in Simulated Data.

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2020.100883.

ACKNOWLEDGMENTS

We would like to thank Felix Breden and Pavel Pevzner for introducing us to the problem and offering us encouragement and help during the development and testing of ImmunoTyper. This research was partially funded by NSF Grant CCF-1619081, NIH grant GM108348, and the Indiana University Grand Challenges Program, Precision Health Initiative to SCS.

AUTHOR CONTRIBUTIONS

M.F., S.C.S., and C.T.W. identified the problem and developed its mathematical formulation; M.F. and S.C.S. developed the theory underlying ImmunoTyper; M.F. implemented ImmunoTyper; M.F. and E.H. optimized and tested ImmunoTyper and evaluated it on various datasets; M.F. and S.C.S. wrote the manuscript with the help and feedback from E.H. and C.T.W.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: July 15, 2019
Revised: November 8, 2019
Accepted: January 29, 2020
Published: March 27, 2020

REFERENCES

Boyd, S.D., Gaëta, B.A., Jackson, K.J., Fire, A.Z., Marshall, E.L., Merker, J.D., Maniar, J.M., Zhang, L.N., Sahaf, B., Jones, C.D., et al. (2010). Individual variation in the germline ig gene repertoire inferred from variable region gene rearrangements. J. Immunol. *184*, 6986–6992.

Chaisson, M.J., Huddleston, J., Dennis, M.Y., Sudmant, P.H., Malig, M., Hormozdiari, F., Antonacci, F., Surti, U., Sandstrom, R., Boitano, M., et al. (2015). Resolving the complexity of the human genome using single-molecule sequencing. Nature *517*, 608.

Clark, K., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., and Sayers, E.W. (2015). Genbank. Nucleic Acids Res. *44*, D67–D72.

Corcoran, M.M., Phad, G.E., Bernat, N.V., Stahl-Hennig, C., Sumida, N., Persson, M.A., Martin, M., and Hedestam, G.B.K. (2016). Production of individualized v gene databases reveals high levels of immunoglobulin genetic diversity. Nat. Commun. *7*, 13642.

Gadala-Maria, D., Gidoni, M., Marquez, S., Vander Heiden, J.A., Kos, J.T., Watson, C.T., OConnor, K., Yaari, G., and Kleinstein, S.H. (2019). Identification of subject-specific immunoglobulin alleles from expressed repertoire sequencing data. Front. Immunol. *10*, 129.

Gadala-Maria, D., Yaari, G., Uduman, M., and Kleinstein, S.H. (2015). Automated analysis of high-throughput b-cell sequencing data reveals a high frequency of novel immunoglobulin v gene segment alleles. Proc. Natl. Acad. Sci. U S A *112*, E862–E870.

Gidoni, M., Snir, O., Peres, A., Polak, P., Lindeman, I., Mikocziova, I., Sarna, V.K., Lundin, K.E., Clouser, C., Vigneault, F., et al. (2019). Mosaic deletion patterns of the human antibody heavy chain gene locus shown by bayesian haplotyping. Nat. Commun. *10*, 628.

Janeway, C.A., Travers, P., Walport, M., and Shlomchik, M. (2001). Immunobiology: The Immune System in Health and Disease, Fifth Edition (Garland Publishing).

Kidd, M.J., Chen, Z., Wang, Y., Jackson, K.J., Zhang, L., Boyd, S.D., Fire, A.Z., Tanaka, M.M., Gaëta, B.A., and Collins, A.M. (2012). The inference of phased haplotypes for the immunoglobulin h chain v region gene loci by analysis of VDJ gene rearrangements. J. Immunol. *188*, 1333–1340.

Kirik, U., Greiff, L., Levander, F., and Ohlin, M. (2017). Parallel antibody germline gene and haplotype analyses support the validity of immunoglobulin germline gene inference and discovery. Mol. Immunol. *87*, 12–22.

Laehnemann, D., Borkhardt, A., and McHardy, A.C. (2015). Denoising DNA deep sequencing data high-throughput sequencing errors and their correction. Brief. Bioinform. *17*, 154–179.

Lefranc. (2001a). The Immunoglobulin Factsbook (Academic Press).

Lefranc, M.P. (2001b). Nomenclature of the human immunoglobulin heavy (IGH) genes. Exp. Clin. Immunogenet. *18*, 100–116.

Lefranc, M.P., Giudicelli, V., Duroux, P., Jabado-Michaloud, J., Folch, G., Aouinti, S., Carillon, E., Duvergey, H., Houles, A., Paysan-Lafosse, T., et al. (2015). IMGT(R), the international ImMunoGeneTics information system(R) 25 years on. Nucleic Acids Res. *43*, D413–D422.

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

Luo, S., Jane, A.Y., Li, H., and Song, Y.S. (2019). Worldwide genetic variation of the IGHV and TRBV immune receptor gene families in humans. Life Sci. Alliance *2*, e201800221.

Luo, S., Jane, A.Y., and Song, Y.S. (2016). Estimating copy number and allelic variation at the immunoglobulin heavy chain locus using short reads. PLoS Comput. Biol. *12*, e1005117.

Matsuda, F., Ishii, K., Bourvagnet, P., Kuma, K.i., Hayashida, H., Miyata, T., and Honjo, T. (1998). The complete nucleotide sequence of the human immunoglobulin heavy chain variable region locus. J. Exp. Med. *188*, 2151–2162.

Ohlin, M., Scheepers, C., Corcoran, M., Lees, W.D., Jackson, K.J.L., Ralph, D., Schramm, C.A., and Marthandan, N. (2019). Inferred allelic variants of immunoglobulin receptor genes: a system for their evaluation, documentation, and naming. Front. Immunol. *10*, 1–13.

Ralph, D.K., and Matsen, F.A., IV (2016). Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. PLoS Comput. Biol. *12*, e1004409.

Stöcker, B.K., Köster, J., and Rahmann, S. (2016). Simlord: simulation of long read data. Bioinformatics *32*, 2704–2706.

Thörnqvist, L., and Ohlin, M. (2018). The functional 3′-end of immunoglobulin heavy chain variable (IGHV) genes. Mol. Immunol. *96*, 61–68.

Vander Heiden, J.A., Marquez, S., Marthandan, N., Bukhari, S.A.C., Busse, C.E., Corrie, B., Hershberg, U., Kleinstein, S.H., Matsen, F.A., IV, et al. (2018). AIRR community standardized representations for annotated immune repertoires. Front. Immunol. *9*, 2206.

Watson, C., and Breden, F. (2012). The immunoglobulin heavy chain locus: genetic variation, missing data, and implications for human disease. Genes Immun. *13*, 363.

Watson, C.T., Matsen, F.A., Jackson, K.J., Bashir, A., Smith, M.L., Glanville, J., Breden, F., Kleinstein, S.H., Collins, A.M., and Busse, C.E. (2017). Comment on a database of human immune receptor alleles recovered from population sequencing data. J. Immunol. *198*, 3371–3373.

Watson, C.T., Steinberg, K.M., Huddleston, J., Warren, R.L., Malig, M., Schein, J., Willsey, A.J., Joy, J.B., Scott, J.K., Graves, T.A., et al. (2013). Complete haplotype sequence of the human immunoglobulin heavy-chain variable, diversity, and joining genes and characterization of allelic and copy-number variation. Am. J. Hum. Genet. *92*, 530–546.

Yu, Y., Ceredig, R., and Seoighe, C. (2017). A database of human immune receptor alleles recovered from population sequencing data. J. Immunol. *198*, 2202–2210.

iScience, Volume *23*

Supplemental Information

Genotyping and Copy Number Analysis of Immunoglobin Heavy Chain Variable Genes Using Long Reads

Michael Ford, Ehsan Haghshenas, Corey T. Watson, and S. Cenk Sahinalp

Supplemental Information

Genotyping and Copy Number Analysis of Immunoglobin Heavy Chain Variable Genes using Long Reads

Michael Ford, Ehsan Haghshenas, Corey T. Watson, S. Cenk Sahinalp

S1. Supplemental Figures

S1.1. Plots of Remapped Consensus Sequence Error Reduction, related to Section 2.3

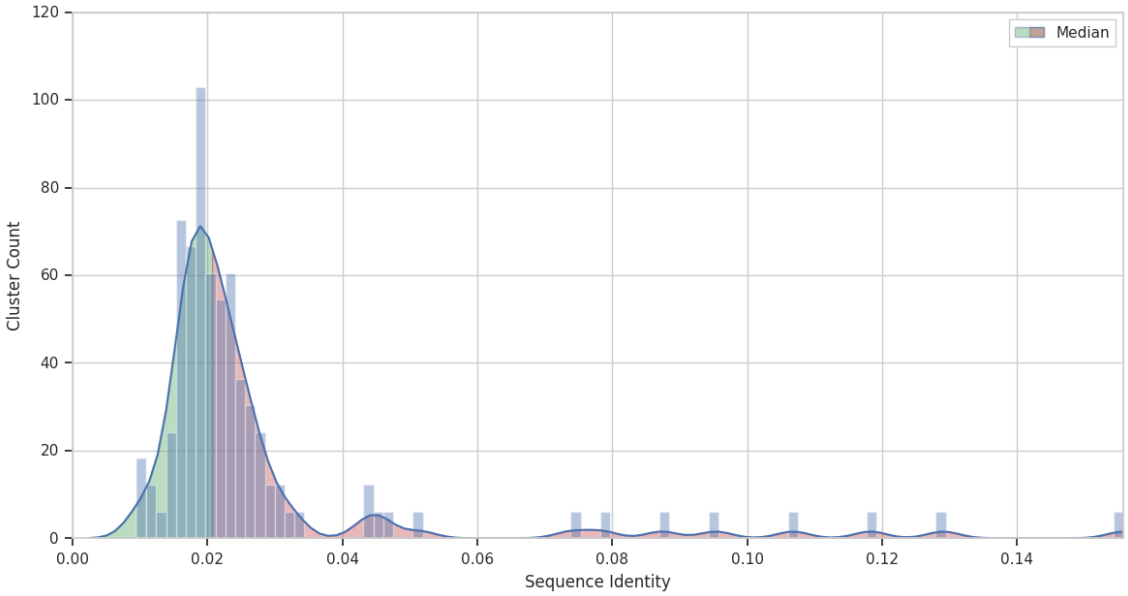


Figure S1: Related to Section 2.3. Histogram of sequence similarity between CHM1 simulated cluster consensus sequences and their best mapping location on the IGH CHM1 reference

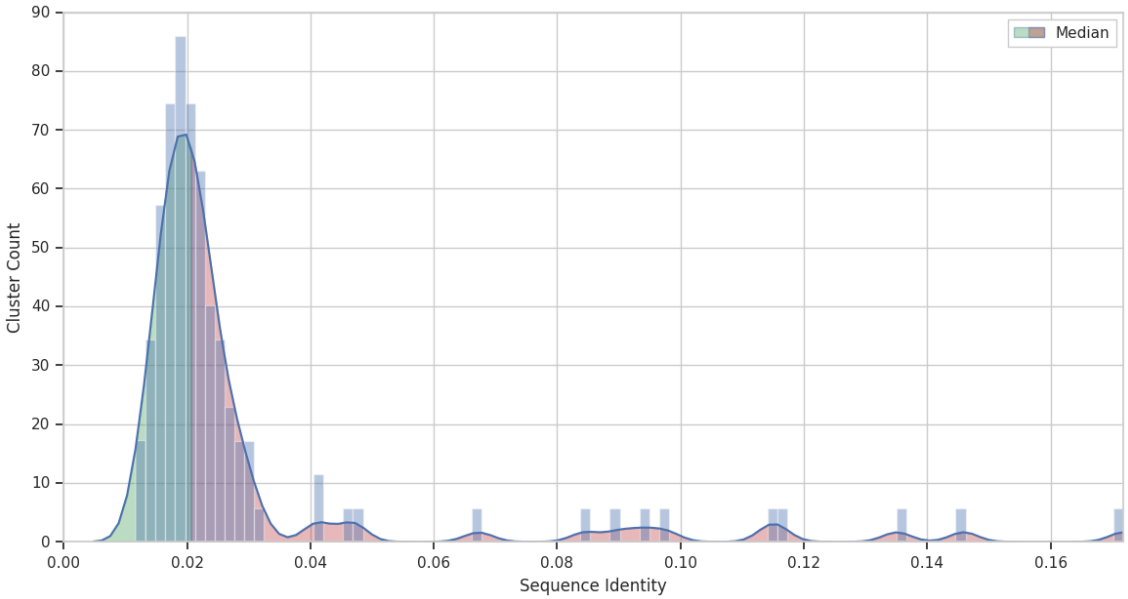


Figure S2: Related to Section 2.3. Histogram of sequence similarity between GRCh37 simulated cluster consensus sequences and their best mapping location on the IGH GRCh37 reference

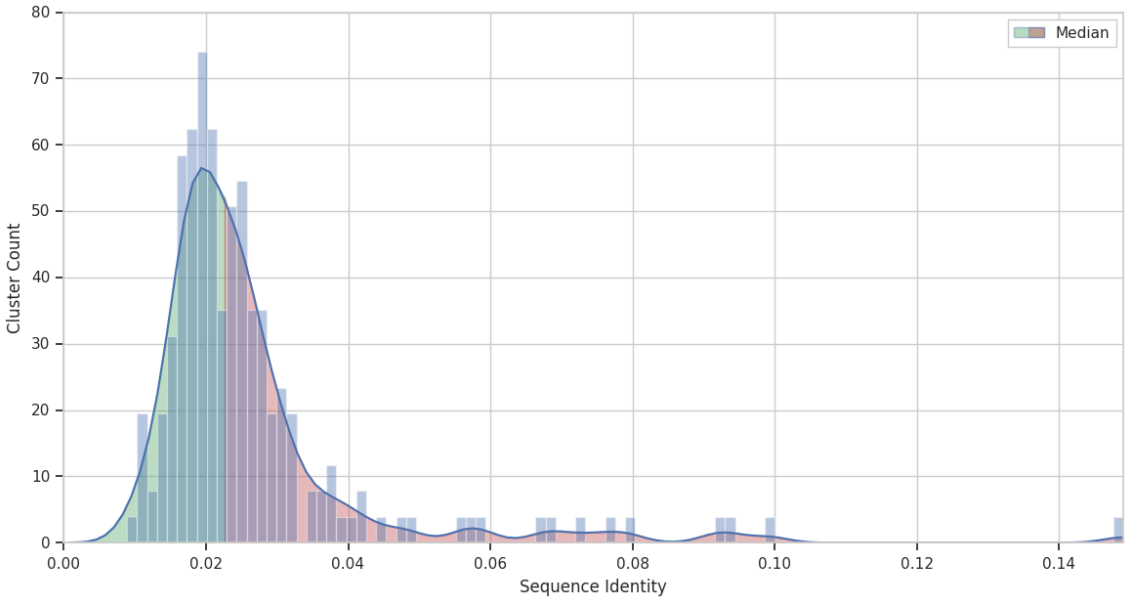


Figure S3: Related to Section 2.3. Histogram of sequence similarity between the CHM1 + GRCh37 simulated cluster consensus sequences and their best mapping location on the IGH CHM1 or IGH GRCh37 reference

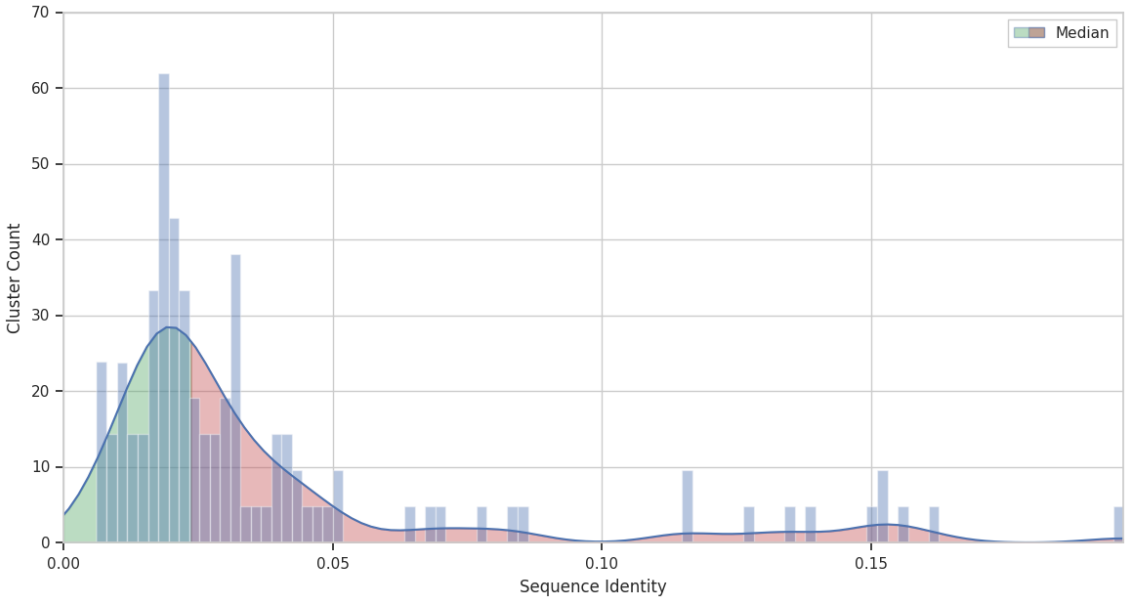


Figure S4: Related to Section 2.3. Histogram of sequence similarity between the CHM1 cluster consensus sequences and their best mapping location on the IGH CHM1 reference

S1.2. Investigation into False Positive Allele Calls Figures, related to Section 2.4

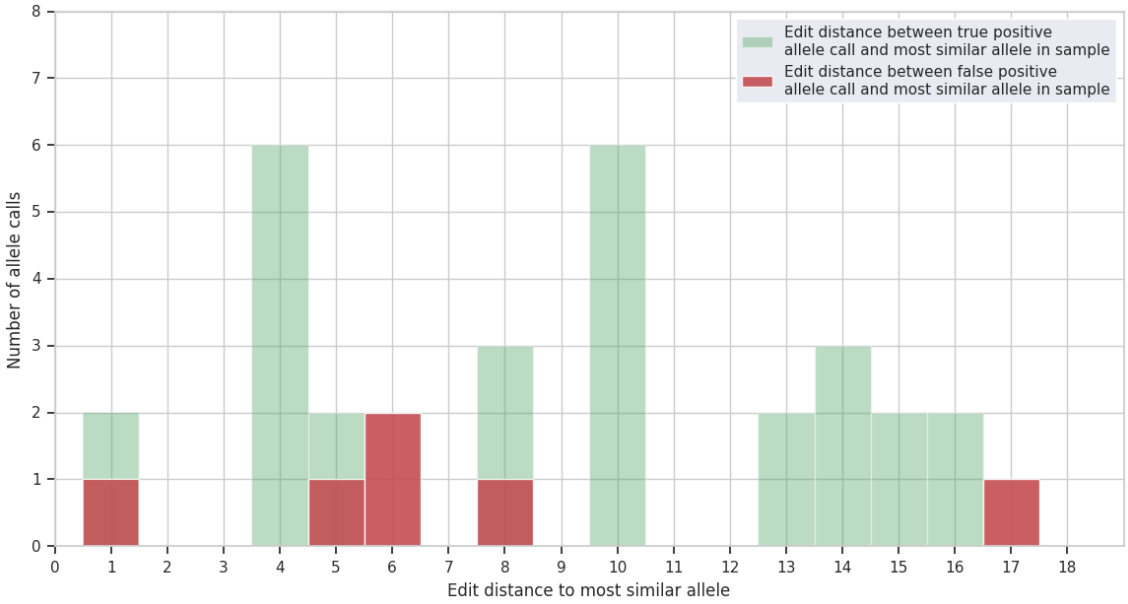


Figure S5: Related to Section 2.4. Comparing sequence similarity between TP and FP calls for the simulated CHM1 sample.

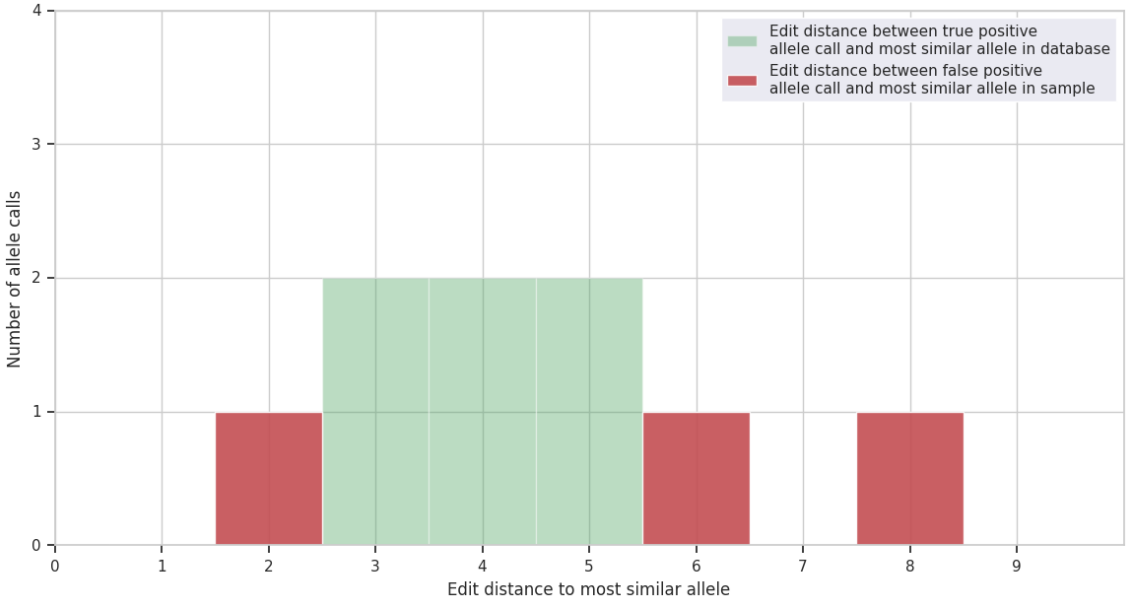


Figure S6: Related to Section 2.4. Comparing sequence similarity between TP and FP calls for the simulated GRCh37 sample.

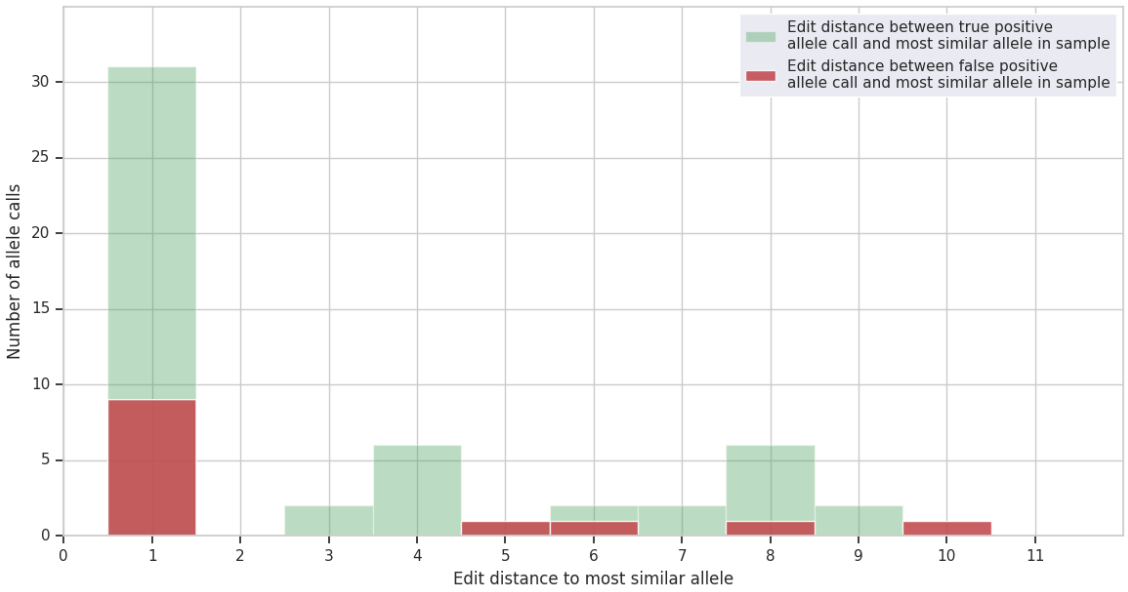


Figure S7: Related to Section 2.4. Comparing sequence similarity between TP and FP calls for the simulated CHM1+GRCh37 diploid sample.

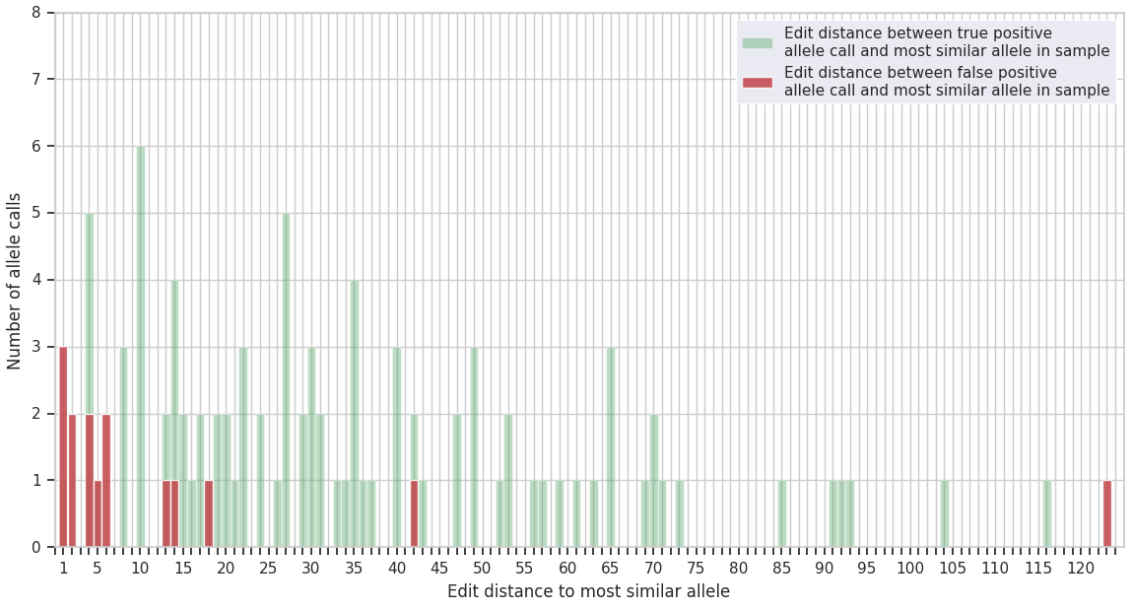


Figure S8: Related to Section 2.4. Comparing sequence similarity between TP and FP calls for the WGS CHM1 sample.

S2. Transparent Methods

S2.1. Algorithmic Foundations

Our goal in this paper is selecting a set of alleles that *best* describes a set of reads from the IGHV region. The principal challenge lies in deciding what represents the *best* selection. The complexity of the problem depends on the number and heterogeneity of allele candidates. There are two key considerations that need to feature in evaluating a potential *solution*:

1. The read sequences must be similar to their matched allele as well as to each other, as much as possible.
2. The number of reads assigned to an allele must match the expected read coverage.

Both these features are quantitative and their linear combination can be used as an error function to describe the quality of an assignment of reads to alleles in the context of what we call the *Allele Assignment Problem*, which we formally define as follows.

Definition: Allele Assignment Problem (AAP). Given a set of input reads $R = \{1, ..., n\}$, and a set of candidate alleles $A = \{a_1, ..., a_m\}$ as the input, consider, for any subset of reads $s_i \subseteq R$ and an allele a_j, a function $f(s_i, a_j)$ describing the error corresponding to the assignment of s_i to allele a_j. The Allele Assignment Problem asks to partition R into non-intersecting subsets s_i and assign each subset s_i to one allele a_j such that $\sum_i f(s_i, a_j)$ is minimized. More specifically, given S, the set of all $2^m - 1$ non-empty subsets of R, consider the set of all possible assignments between each $s_i \in S$ and each $a_j \in A$ with weight $f(s_i, a_j)$. Let $x_{i,j}$ be a binary variable which takes value 1 if s_i is assigned to a_j and is 0 otherwise. The allele assignment problem thus asks to determine the values of $x_{i,j}$ that minimize the objective

$$\sum_{s_i \in S, a_j \in A} x_{i,j} f(s_i, a_j)$$

subject to the constraint that $\bigcup_{\forall x_{i,j}=1} s_i = R$ [1]. As such, AAP modifies the well known *many-to-one assignment* problem (Pentico, 2007) in the following manner: (i) AAP does not have the constraint that *each* allele a_j needs to be assigned a non-empty subset s_i, nor does it have the constraint that each subset s_i is assigned to a distinct allele a_j, and (ii) the cost of assigning a read to an allele depends on the other reads assigned to the same allele. Note that any error function f that captures the features summarized above leads to a computationally difficult combinatorial optimization problem; as a result we first greedily establish some read to allele assignments through a number of distinct steps so as to reduce the size of the eventual allele assignment problem we solve.

S2.2. Overview of the ImmunoTyper Approach

ImmunoTyper aims to solve the Allele Assignment Problem (AAP) through which it can identify all alleles of the IGHV genes and their respective copy numbers.[2] For that it follows a number of distinct steps as described below.

1. **IGHV-containing Read Identification and Subread Extraction**
   Reads relevant to the IGH region are identified by mapping to the GRCh38 reference. Reads originating from possible novel IGH sequence are identified by mapping the unmapped reads to the IGHV allele database. IGHV sequences are identified by mapping all extracted reads to the IGHV allele database, and subsequences containing the coding region and flanking sequence, dubbed *subreads*, are extracted.
2. **Mapping-based Clustering**
   Subreads are mapped to the IGHV allele database, and then are greedily assigned to their best mapped allele under the conditions that (i) the mapping is unambiguous and (ii) the number of assigned reads for any given allele is sufficiently close to the estimated read coverage. (Read coverage is estimated using high confidence allele mappings and the provided sequence coverage.) Subreads not meeting these criteria are passed to the next step.

[1] in certain applications, with the additional constraint that $(x_{i,j} = 1) \rightarrow (x_{i',j} = 0)$

[2] Note that ImmunoTyper is currently tailored for V gene analysis even though it can easily be extended to perform D or J gene analysis or could be generalized to other multi-copy genes as well.

3. **Allele assignment for Ambiguous Subreads**
   The set of ambiguous subreads (those which could not be assigned to a single allele unambiguously) are processed in three stages:

   a. **Super-cluster Building**
      In order to reduce the solution space, we partition the allele assignment problem on ambiguous subreads into smaller, independent sub-problems. This is achieved by clustering subreads based on sequence similarity, into *super-clusters*, each corresponding to a small set of alleles that share high sequence similarity.

   b. **Super-cluster Breaking**
      For each super-cluster, the ILP formulation for AAP is solved independently as follows. First, candidate alleles are identified by mapping the super-cluster subreads to all IGHV alleles. Variants with respect to the consensus sequence generated from all subreads are determined. Finally an ILP formulation for AAP is solved using the commonly used Gurobi ILP package (Gurobi Optimization, 2018), to break each super-cluster into smaller clusters of subreads, each representing a single copy of an IGHV gene or pseudogene.

   c. **Allele Calling**
      Each subread cluster is then assigned to an allele by mapping the consensus sequence of cluster subreads against the IGHV allele database (implicitly reducing mapping errors that would be due to read error biases).

ImmunoTyper additionally includes two independent subread filtering steps which are designed to remove subreads that were mistakenly included in the analysis due to mapping errors in the subread extraction step. (See Supplemental Information S2.5.1 and S2.6.1 for details.)

Solving the Allele Assignment Problem, and ultimately IGHV genotyping in this multi-stage, optimization-based approach offers several advantages. First, by employing multiple distinct methods at different stages, we can reduce the solution space and solve the problem more efficiently. For example, the 'Mapping-based Clustering' stage prioritizes speed, but only solves allele assignments for sufficiently distinct alleles. Second, by using two different methods for allele assignment, we tailor the method to the difficulty of a given allele assignment. As a result, allele assignment for IGHV sequences that are highly similar is solved using the optimization approach in "Allele Assignment for Ambiguous Subreads", which is specifically designed to differentiate highly similar sequences by considering distinguishing variants on a nucleotide level.

S2.3. Allele Database

ImmunoTyper utilizes the complete set of human IGHV gene and pseudogene alleles as provided by the The International Immunogenetics information system (IMGT:*www.imgt.org* (Lefranc, 2008)). However calls for alleles that are shorter than 200bp, redundant or poorly defined are ignored. In addition, we have modified two pseudogene sequences to avoid ambiguity in the database. See Supplemental Information S2.8 for a complete record of alleles that are ignored or modified.

S2.4. IGHV-containing Read Identification and Subread Extraction

ImmunoTyper takes as input a BAM file representing a PacBio WGS mapping to the GRCh38 reference, as well as the depth of coverage as a parameter. In order to extract relevant reads that contain IGHV sequences, ImmunoTyper first extracts all reads with primary and supplementary mappings to the IGH region (chr14:105586437-106880844). Second, all reads that are identified as being unmapped are also extracted.

S2.4.1. Subread Extraction

Extracted reads from both steps above are then mapped to the IGHV allele database. This is performed using Minimap2 (Li, 2018) with increased sensitivity parameters ("`-cx map-pb -k10 -w3 -N5`") to account for any novel IGHV sequence that may not be represented in the database. Reads with no mapping are then discarded.

Non-overlapping mapping locations on every read are then identified as being IGHV sequences. A subread is extracted for every IGHV sequence, using its best mapped allele. The subread contains the IGHV sequence along with the adjacent 1000bp flanking sequence. A subread extraction is conditional on (1) The best mapping covering at

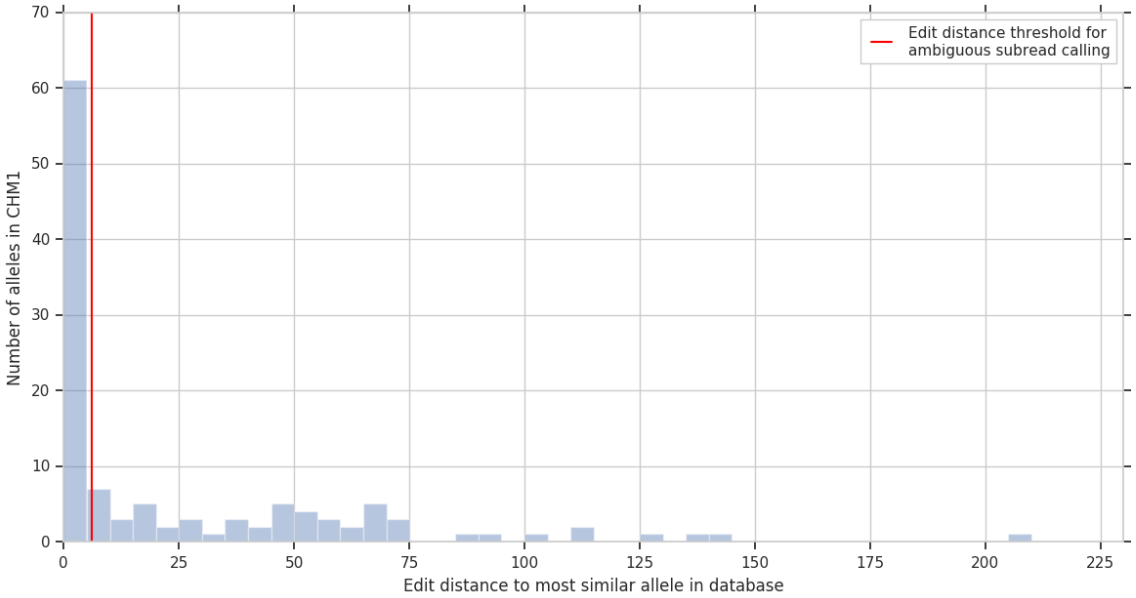


Figure S9: Histogram of edit distance between each CHM1 IGHV allele and its most similar IGHV allele (with respect to edit distance) from the complete database. Ambiguous mapping threshold is set to 6 (red line) as described in Supplemental Information S2.5.

least 90% of the target IGHV reference sequence. (2) Neither of the 1000bp flanking sequence being clipped by the read ends. After all the valid IGHV-containing subreads are extracted, they are oriented so as to all be on the same strand.

S2.5. Mapping-based Clustering

Despite the presence of highly similar and hard to differentiate V genes (Luo et al., 2016) (see Figure 1 for allele sequence similarity distribution), many IGHV (pesudo)genes have sufficiently distinct sequence composition to allow for confident and unambiguous mapping results, even for error-prone long reads (see Supplemental Figure S9). Thus ImmunoTyper initially identifies high-confidence assignment of subreads to alleles (again, provided that the mapping is unambiguous and the coverage of the allele by the assigned subreads is close to the estimated coverage), leading to the identification and accurately genotyping of >50% of the IGHV (pseudo)genes in the sample, resulting in a significant reduction of the computational problem (i.e. of handling the subreads that could not be confidently assigned to alleles).

More specifically, we identify high-confidence mappings among the subread-to-allele mappings returned by the S2.4.1 step by sorting them according to the number of errors in the alignment, combined with the number of bases in the reference sequence that are not included in the alignment, i.e.

$$error = NM + a_{start} + (a_{length} - a_{end})$$

where NM is the total number of mismatch and indel bases in the alignment, a_{start} is the start of the alignment on the reference allele, and therefore represents the number of bases in the reference allele that are not included in the alignment, and $a_{length} - a_{end}$, represents the end positing of the alignment on the reference allele subtracted from the total length of the reference allele - overall providing us the total number of reference allele bases not included in the alignment.

Subreads are then assigned to their best-mapping allele, provided that mapping is unambiguous, i.e. if the second-best mapping reference allele has at least 6 additional edit errors to the subread in comparison to the best-mapping

reference allele. Subreads with an unambiguous mapping to one of the *ignored alleles* as described in Supplemental Information S2.8 are discarded.

Since subreads are assigned to reference alleles based on mapping ambiguity (more specifically, a lack of mapping ambiguity) and not sequence similarity, this approach for subread clustering may still produce a valid cluster from subreads that originate from a *novel* (i.e. not in the Allele Database) IGHV (pseudo)gene, provided (i) the novel allele is sufficiently similar to an existing allele in the Database to produce acceptable mappings, and (ii) the existing allele is sufficiently distinct from all other alleles in the database so as to result in unambiguous mappings.[3] Even though we are not explicitly aiming to identify novel alleles, it is possible to generate the consensus sequence of each cluster of subreads at this stage and compare it with the reference allele they are assigned to so as to identify any difference in the sequence composition (see Supplemental Information S2.7.4), allowing for subsequent identification of any novel allele sequence.

Subreads that have ambiguous best-mapping loci are passed to the S2.6 step.

S2.5.1. Read coverage depth estimation

In order to confidently describe a cluster of subreads as one originating from a reference allele, it is not sufficient that the subreads have unambiguous allele assignments; the number of subreads in the cluster must also be congruent with the expected depth of coverage. In fact, depth of coverage could, in principle, be used to determine the copy number of each allele. Unfortunately it is possible that the observed depth of coverage differs from the actual sequencing depth due to natural fluctuations in sequencing coverage, or read dropout in the mapping process due to factors such as sequencing error rate and repetitive DNA in the mapping locus.

To account for any potential divergence from the actual sequencing depth, ImmunoTyper uses the results from subread mapping-based clustering to calculate a read depth statistic in order to ensure that the *expected coverage* is empirically derived from the data. To calculate the updated sequencing depth, clusters $\leq$ 50% of the user-provided *actual* sequence depth are considered unlikely to be representative of an actual allele in the sample and are not considered, as are clusters >150% of the actual sequence depth, as these are likely to originate from alleles with multiple copies. ImmunoTyper then calculates the empirical read coverage as the median coverage of the remaining clusters.

S2.5.2. Cluster Filtering

In order to ensure that S2.5 step provides only high-confidence results, clusters are finally filtered based on the newly calculated *empirical* read coverage value. Clusters with coverage $\leq$ 85% of this value are discarded, and their subreads are passed to the S2.6 step. This step primarily eliminates allele assignments whose lower concordance with the expected depth are deemed lower-confidence. This step would also filter any subreads that do not contain true IGHV sequences, but were incorrectly extracted in S2.4 step due to chance sequence error. These subreads can be discarded later in the subread filtering steps explained in Supplemental Information S2.6. After the completion of all the filtering, the remaining subread clusters and their assigned reference alleles are then called with a copy number estimated to be the integral multiple of the empirical read coverage that is closest to the size of the cluster.

S2.6. Super-cluster Building

The subreads that could not be assigned in S2.5 step require a more refined approach. ImmunoTyper utilizes a second clustering approach for these more difficult cases, considering both the coding region of the V genes, as well as the adjacent non-coding flanking regions - of length 1000bp .

Variants present in the non-coding flanking regions have the potential to aid subread differentiation; unfortunately distinguishing non-coding variants from sequencing errors is a major challenge. Reference-guided approaches are not possible here as there is no non-coding reference sequence/variant database available. This implies that variants must be identified through subread-to-subread comparison. Additionally, due to the high sequencing error rate of long read (PacBio) data, there is necessarily a large number of errors present in the subreads associated with the full 2000bp flanking sequence. The high error rate, combined with the lack of non-coding references and the limited utility of

[3] Note that if this reference allele from is also present in the dataset with subreads originating from it, its coverage will be close to an integral multiple of the overall expected coverage.

coding reference alleles may result in allele asssignments with a low signal-to-noise ratio. Finally, there may be thousands of subreads originating from dozens of alleles which need to be processed in this stage, implying that any method with subread-to-subread comparisons will have a large solution space.

In order to reduce the solution space of the implied problem and improve the signal-to-noise ratio, ImmunoTyper first performs a rough clustering based on a subread-to-subread sequence similarity graph as follows. Subreads are first aligned to each other (we have used Minimap2 (Li, 2018) (with the "`-cx ava-pb -k14 -w3`" options to increase sensitivity). A graph is then constructed by creating a node for each subread r, and creating an edge between r and each subread r' provided the two subreads align *well* with a weight equal to the normalized error metric similar to that used in S2.5 step.

$$weight(r, r') = \frac{2NM + (r_{start} + (r_{length} - r_{end})) + (r'_{start} + (r'_{length} - r'_{end}))}{r_{length} + r'_{length}}$$

Here NM is the total number of mismatched and indel bases in the alignment, r_{start} is the start of the alignment on r, and therefore the length of the prefix of r not included in the alignment and $r_{length} - r_{end}$ is the length of the suffix of r not included in the alignment. Corresponding definitions apply for r'. Finally the error is normalized by the sum of the lengths of r, r'.

In order to ensure precision (and compensate for the increased sensitivity parameters used with Minimap2) we only maintain edges with weight ≤ 0.3 - the rest are deleted. Then, any node with degree 0 and its associated subread is discarded so as to eliminate subreads not sufficiently similar to others because they do not originate from the IGHV region but nevertheless were extracted in S2.4 step due to chance sequencing or mapping errors.

The resulting *subread distance graph* can then be clustered using the Dense Subgraph Finder (DSF) tool (Safonova et al., 2015). DSF is designed to solve the 'corrupted-clique problem' as an approximation to the problem of clustering subreads originating from the same allele that have been subject to sequencing errors. It finds dense subgraphs through identification and merging of maximal cliques in the input graph. In order to ensure high precision clustering and encourage clustering that is concordant with the calculated read depth, we use the "`--min-fillin 0.95`" parameter and set the minimum cluster size at $\leq 75\%$ of the empirical read coverage using the "`--min_clust_size`" parameter. Any clusters that are smaller than the minimum are returned as single subreads and are passed to the next step.

S2.6.1. Unclustered Subread Merging

The output of DSF is a set of dense clusters of subreads, each cluster composed of subreads with similar sequence composition. As each such cluster may include subreads that originate from more than one gene copy, we will call them super-clusters.

In addition to the super-clusters, DSF also outputs some unclustered subreads which, due to sequence error, are not sufficiently similar to other subreads to be assigned to a cluster, or were grouped into clusters smaller than the minimum size as described above. In order to assign these unclustered subreads, ImmunoTyper merges them with one of the available super-clusters. This is achieved by first constructing a representative consensus sequence for each super-cluster (using SPOA v1.1.3 (Vaser et al., 2017), a SIMD-accelerated, partial-order alignment-based consensus and Multiple Sequence Alignment (MSA) tool which has been shown to be particularly effective and aligning indel-rich long reads). Unclustered subreads are then mapped to these consensus sequences (again using Minimap2 with "`-cx map-pb -k10 -w3 -N5`" options), and are added to the super-cluster with the best associated mapping. Subreads without a *good* mapping are then discarded (this second filtering step is again for eliminating subreads erroneously included in the analysis).

S2.7. LP Super-cluster Breaking

The subread super-clusters are broken into smaller clusters so that each individually represents a single allele copy - by the use of a novel ILP approach. For that we first generate a likely set of candidate alleles (described below), and then assign subreads from each super-cluster to candidate alleles using the ILP formulation (described in Supplemental Information S2.7.1 and S2.7.2).

Given a super-cluster, the set of relevant candidate alleles are determined using the subread-to-allele mappings that were performed in S2.4 step. Specifically, we first generate a candidate allele pool that includes each allele that is the best-mapping allele of at least one subread in the super-cluster. In order to reduce the candidate pool, we count the number of subreads that have each allele as its best mapping; if a subread has 2 or more equally good best mapping allele, it contributes to the count of each allele by 1. We now discard any allele if (i) its (best mapping) subread count is not one of the top 10 counts among all candidate alleles, or (ii) its subread count is $\leq$ 50% of the empirical read coverage.

S2.7.1. Identifying Allele-Defining Variants

In order to distinguish candidate alleles from one another, we generate a set of *allele-defining* variants for each candidate allele. This is achieve by first obtaining the consensus sequence of the subreads in the super-cluster and then comparing each candidate allele with the consensus sequence.[4] We then generate the MSA (again obtained by the use of the POA method) of the candidate allele sequences and the consensus sequence. This allows us to identify a set of candidate allele-defining variants. Any of these variants that are shared among all candidate alleles are then discarded since they do not provide information for discriminating alleles; the remaining variants form our allele-defining variant set for the super-cluster.

Each subread is now compared against the consensus sequence using the subread MSA described above, to identify the allele-defining variants it includes. Then, the candidate variants are filtered based on their subread support: if the number of subreads including a variant $\leq 0.9\cdot$ empirical read coverage, it is discarded - since it is likely a result of sequencing errors. Similarly if a variant has $\geq 2\cdot$ empirical read coverage, subread support it is discarded as well - since it is not going to be very helpful in distinguishing alleles supported by the super-cluster.

S2.7.2. ILP Formulation

Each subread super-cluster can now be partitioned into distinct clusters, each corresponding to a single allele, using a ILP formulation defined below. Note that in order to allow for multiple copies of each candidate allele, the candidate allele set (and the associated allele-defining variants) is duplicated by the *max-copy-number* value, a user-defined parameter with a default value of 4.

Given a super-cluster C, let a_j denote the j-th candidate allele and r_i denote the i-th subread associated with C.

Variables

$$\text{Let } D_i^j = \begin{cases} 1 & \text{if } r_i \text{ has been assigned to } a_j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Let } \delta^j = \begin{cases} 1 & \text{if } a_j \text{ is called for } C \\ 0 & \text{if } a_j \text{ is not called for } C \end{cases}$$

Constraints

$$\text{For all } r_i, \quad \sum_{a_j} D_i^j = 1 \tag{1}$$

$$\text{For all } a_j, \quad \sum_{r_i} D_i^j \geq \text{empirical read coverage} \cdot 0.9 \tag{2}$$

$$\text{For all } r_i, a_j, \delta_j \geq D_i^j \tag{3}$$

[4] ImmunoTyper uses the POA method (Lee et al., 2002) that implements the *partial order alignment* algorithm introduced there. POA is slower than SPOA but it generates a higher quality consensus sequence of the subreads and as well as their implied MSA.

$$min\_num \leq \sum_{a_j} \delta^j \leq max\_num \tag{4}$$

here $min\_num \approx size(C)/(\text{empirical read coverage} * 0.9)$ and $max\_num \approx size(C)/(\text{empirical read coverage} * 1.1)$ where $\approx$ rounds the value to the closest integer.
(The above *interval constraint* allows each super-cluster to deviate from the empirical read coverage.)
Objective
Minimize:

$$\sum_{a_j} \alpha * code\_var\_cov(a_j) + non\_code\_var\_cov(a_j)$$

where:
$code\_var\_cov(a_j) =$

$$\sum_{v_k \in V_C} \begin{cases} \left| \left(\sum_{r_i \text{ if } v_k \in r_i} D_i^j \right) - \left(\delta^j * expcov \right) \right| & \text{if } v_k \in a_j \\ \left(\sum_{r_i \text{ if } v_k \in r_i} \right) D_i^j & \text{otherwise} \end{cases}$$

$non\_code\_var\_cov(a_j) =$

$$\sum_{v_k} \left| \left(\sum_{r_i \text{ if } v_k \in r_i} D_i^j \right) - \left(\delta^j * expcov \right) \right|$$

Here, given the set of all variants V for all reads and candidate alleles, v_k denotes the k-th variant in V and $V_C \subseteq V$ denotes the set of all allele defining variants for all candidate alleles. Additionally, α is a user defined parameter with default value 1000 - optimized for simulated data; $code\_var\_cov(a_j)$ is the variant coverage error for allele-defining variants in a_j and $non\_code\_var\_cov(a_j)$ is the variant coverage error for non-coding variants for subreads assigned to a_j.

S2.7.3. Cluster merging and re-breaking

A super-cluster may fail to be partitioned so as to be assigned to distinct alleles in the following two cases: (1) there is no *qualifying* candidate allele, or (2) the ILP infeasible. In both cases we deduce that we have a poor-quality clustering, discard the super-cluster and assign each of its subreads to its *best-mapping valid cluster* as follows. We first map the subread to the consensus sequence obtained for every cluster from S2.7 step (using SPOA). Any such cluster with a newly mapped subread is then merged with all its sibling clusters to re-create the original subread super-cluster - additionally containing one or more newly mapped subreads. The super-cluster is then re-partitioned using a new instance of the ILP. This iterative process is repeated until no such erroneous cluster is obtained by the ILP formulation (the user may put an upper bound on the number of attempts, which is set to 3 by default).

S2.7.4. Allele Calling

In the final step, each subread cluster, obtained by partitioning a super-cluster, is assigned to an allele by first generating its consensus sequence (using SPOA), and then mapping the consensus sequence to the allele reference database as defined in Supplemental Section S2.3 with a copy number of one.[5]

[5] Any mapper including our own lordFAST (Haghshenas et al., 2018) or Minimap2 (Li, 2018) can be used here - however we have observed that our non-standard mapping of long reads to short reference alleles works best with Blasr (Chaisson and Tesler, 2012) on simulated data.

S2.8. Filtered IMGT Alleles

S2.9. Ignored Allele Calls

Alleles whose reference sequence shorter 200 bp are ignored. This includes the functional alleles:

Allele	Length
3-72*02	165
4-39*04	196

And the following non-functional alleles:

(III)-44*01	21
(III)-44D*01	21
3-62*02	106
3-76*02	155
1-12*02	154
7-56*01	154
(III)-22-2*01	30
(III)-22-2D*01	30
(III)-5-1*01	99
(III)-67-2*01	99
(II)-40-1*01	77
(II)-67-1*01	139
(II)-46-1*01	147
(II)-1-1*01	182

Additionally, the following alleles were completely removed from the database:

- 1-69D*01 was removed because it is identical in coding sequence to IGHV1-69*01
- 3-30-52*01 was removed because it differs from 3-30-2*01 by a 2bp truncation at the 3' end
- 2-70*04 was removed because it differs from 2-70D*04 by a 13bp 3' truncation
- 3-42D*01 was removed because it differs from 3-42*02 by a single bp truncation at the 5' end

Pseudogene IGHV(II)-43-1D*01 was ignored as it differs from IGHV(II)-43-1*01 by a single bp insertion, and ImmunoTyper differentiates alleles in LP Super-cluster Breaking using only SNPs. See below for a sequence comparison:

```
IGHV_II_-43-1*01       TCTGGATTCCCCAACAGAACCAGTGCTTCCTGCTGGAGCTGGATCCATCAGCCCCCAGGG 60
IGHV_II_-43-1D*01      TCTGGATTCCCCAACAGAACCAGTGCTTCCTGCTGGAGCTGGATCCATCAGCCCCCAGGG 60
                       ************************************************************

IGHV_II_-43-1*01       AAGGGA-TGGAGTGGGTCAGGTGCACAGGTCATGAAGGGAGCACAAATTCTAACCCACTC 119
IGHV_II_-43-1D*01      AAGGGACTGGAGTGGGTCAGGTGCACAGGTCATGAAGGGAGCACAAATTCTAACCCACTC 120
                       ****** ***************************************************

IGHV_II_-43-1*01       CTCAAGAGTCCAGTCACCACCTCCAGATCTATGTCCAAAAACAGCTCTTCGTATGGCTGA 179
IGHV_II_-43-1D*01      CTCAAGAGTCCAGTCACCACCTCCAGATCTATGTCCAAAAACAGCTCTTCGTATGGCTGA 180
                       ************************************************************

IGHV_II_-43-1*01       GTGACATTAGCAACAAGCACACAGCCATGT 209
IGHV_II_-43-1D*01      GTGACATTAGCAACAAGCACACAACCATGT 210
```

Alleles belonging to either of the chr15 or chr16 orphons are also ignored:

IGHV3-42D*01IGHV1/OR15-1*01
IGHV1/OR15-1*02
IGHV1/OR15-1*03
IGHV1/OR15-1*04
IGHV1/OR15-2*01
IGHV1/OR15-2*02
IGHV1/OR15-2*03
IGHV1/OR15-3*01
IGHV1/OR15-3*02
IGHV1/OR15-3*03
IGHV1/OR15-4*01
IGHV1/OR15-5*01
IGHV1/OR15-5*02
IGHV1/OR15-6*01
IGHV1/OR15-6*02
IGHV1/OR15-9*01
IGHV1/OR16-1*01
IGHV1/OR16-2*01
IGHV1/OR16-3*01
IGHV1/OR16-4*01
IGHV1/OR16-4*02
IGHV1/OR21-1*01
IGHV2/OR16-5*01
IGHV3/OR15-7*01
IGHV3/OR15-7*02
IGHV3/OR15-7*03
IGHV3/OR15-7*04
IGHV3/OR15-7*05
IGHV3/OR16-10*01
IGHV3/OR16-10*02
IGHV3/OR16-10*03
IGHV3/OR16-11*01
IGHV3/OR16-12*01
IGHV3/OR16-13*01
IGHV3/OR16-14*01
IGHV3/OR16-15*01
IGHV3/OR16-15*02
IGHV3/OR16-16*01
IGHV3/OR16-6*01
IGHV3/OR16-6*02
IGHV3/OR16-7*01
IGHV3/OR16-7*02
IGHV3/OR16-7*03
IGHV3/OR16-8*01
IGHV3/OR16-8*02
IGHV3/OR16-9*01
IGHV4/OR15-8*01
IGHV4/OR15-8*02
IGHV4/OR15-8*03

All pseudogenes that are classified as 'non-localized' are also ignored:

IGHV1-NL1*01
IGHV3-NL1*01
IGHV7-NL1*01
IGHV7-NL1*02
IGHV7-NL1*03
IGHV7-NL1*04
IGHV7-NL1*05

Alleles belonging to pseudogene IGHV(II)-20-1 are ignored due to a lack of reference sequences in the IMGT database, despite IGHV(II)-20-1*02 being listed in the CHM1 annotation.

S2.9.1. Allele reference sequence modifications

Pseudogene IGHV(II)-30-41*01 has been modified by removing the 3' sequence that differentiates it from the IGHV(II)-28-1 alleles. While both the IGHV(II)-28-1*03 and IGHV(II)-28-1*01 sequences from CHM1 and GRCh37 respectively contain this 3' sequence, indicating the IGHV(II)-28-1 references should be modified, we decided that modifying a single reference sequence was more parsimonious and therefore suitable. See below for alignment of relevant alleles and sample sequences.

```
IGHV_II_-28-1*03_hg38/969-2245          CATCAACAACTATGTTTCTCAGCACACTTCTGGCTTGAGACGTCCTTGCA 1019
IGHV_II_-28-1*03_reference/1-283        ---CAACAACTATGTTTCTCAGCACACTTCTGGCTTGAGACGTCCTTGCA 1016
IGHV_II_-30-41*01_reference/1-299       ---CAACAACTATGTTTCTCAGCACACTTCTGGCTTGAGACGTCCTTGCA 1016
IGHV_II_-28-1*02_reference/1-253        ---------------------------------CTTGAGACGTCCTTGCA 986
IGHV_II_-28-1*01_reference/1-253        -------------------------------GGCTTGAGAC-TCCTTGCA 987
IGHVII-28-1*01_hg37/970-2241            CATCAACAACTATGTTTCTCAGCACACTTCTGGCTTGAGAC-TCCTTGCA 1018
                                                                        ******** ********

IGHV_II_-28-1*03_hg38/969-2245          GACCCTCTCCCTCACCTGCACTGTCTCTGGATTCCCCATCATAACCAGTG 1069
IGHV_II_-28-1*03_reference/1-283        GACCCTCTCCCTCACCTGCACTGTCTCTGGATTCCCCATCATAACCAGTG 1066
IGHV_II_-30-41*01_reference/1-299       GACCCTCTCCCTCACCTGCACTGTCTCTGGATTCCCCATCATAACCAGTG 1066
IGHV_II_-28-1*02_reference/1-253        GACCCTCTCCCTCACCTGCACTGTCTCTGGATTCCCCATCATAACCAGTG 1036
IGHV_II_-28-1*01_reference/1-253        GACCCTCTCC-TCACCTGCACTGTCTCTGGATTCCCCATCATAACCAGTG 1036
IGHVII-28-1*01_hg37/970-2241            GACCCTCTCC-TCACCTGCACTGTCTCTGGATTCCCCATCATAACCAGTG 1067
                                        ********** ************************************

IGHV_II_-28-1*03_hg38/969-2245          TGTCCTGCTAGAATTGTATCTGCTTGCCCCTAGAAGATGGACAGGAGTGG 1119
IGHV_II_-28-1*03_reference/1-283        TGTCCTGCTAGAATTGTATCTGCTTGCCCCTAGAAGATGGACAGGAGTGG 1116
IGHV_II_-30-41*01_reference/1-299       TTTCCTGCTAGAATTGTATCTGCTTGCCCCTAGAAGATGGACAGGAGTGG 1116
IGHV_II_-28-1*02_reference/1-253        TTTCCTGCTAGAATTGTATCTGCTTGCCCCTAGAAGATGGACAGGAGTGG 1086
IGHV_II_-28-1*01_reference/1-253        TTTCCTGCTAGAATTGTATCTGCTTGCCCCTAGAAGATGGACAGGAGTGG 1086
IGHVII-28-1*01_hg37/970-2241            TTTCCTGCTAGAATTGTATCTGCTTGCCCCTAGAAGATGGACAGGAGTGG 1117
                                        * *******************************************

IGHV_II_-28-1*03_hg38/969-2245          ATCAGGTGCATGGGTTGTGAAGGGAGCACAAATTACAACCCACTGCTCAA 1169
IGHV_II_-28-1*03_reference/1-283        ATCAGGTGCATGGGTTGTGAAGGGAGCACAAATTACAACCCACTGCTCAA 1166
IGHV_II_-30-41*01_reference/1-299       ATCAGGTGCATGGGTTGTGAAGGGAGCACAAATTACAACCCACTGCTCAA 1166
IGHV_II_-28-1*02_reference/1-253        ATCAGGTGCATGGGTTGTGAAGGGAGCACAAATTACAACCCACTGCTCAA 1136
IGHV_II_-28-1*01_reference/1-253        ATCAGGTGCATGGGTTGTGAAGGGAGCACAAATTACAACCCACTGCTCAA 1136
IGHVII-28-1*01_hg37/970-2241            ATCAGGTGCATGGGTTGTGAAGGGAGCACAAATTACAACCCACTGCTCAA 1167
                                        ************************************************

IGHV_II_-28-1*03_hg38/969-2245          GAGTCCATATCCAGATCCAAGAAACAGTTCTTACAGCTGAGCTCTGTGCC 1219
IGHV_II_-28-1*03_reference/1-283        GAGTCCATATCCAGATCCAAGAAACAGTTCTTACAGCTGAGCTCTGTGCC 1216
```

```
IGHV_II_-30-41*01_reference/1-299          GAGTCCATATCCAGATCCAAGAAACAGTTCTTACAGCTGAGCTCTGTGCC 1216
IGHV_II_-28-1*02_reference/1-253           GAGTCCATATCCAGATCCAAGAAACAGTTCTTACAGCTGAGCTCTGTGCC 1186
IGHV_II_-28-1*01_reference/1-253           GAGTCCATATCCAGATCCAAGAAACAGTTCTTACAGCTGAGCTCTGTGCC 1186
IGHVII-28-1*01_hg37/970-2241               GAGTCCATATCCAGATCCAAGAAACAGTTCTTACAGCTGAGCTCTGTGCC 1217
                                           **************************************************

IGHV_II_-28-1*03_hg38/969-2245             CAGTGAACACACAACTACGCATTTTTAAGCAAAAGACGCAATGAAGGGCC 1269
IGHV_II_-28-1*03_reference/1-283           CAGTGAACACACAACTACGCATTTTTAAGCAAAAGA-------------- 1252
IGHV_II_-30-41*01_reference/1-299          CAGTGAACACACAACTACGCATTTTTAAGCAAAAGACGCAATGAAGGGCC 1266
IGHV_II_-28-1*02_reference/1-253           CAGTGAACACACAACTACGCATTTTTAAGCAAAAGA-------------- 1222
IGHV_II_-28-1*01_reference/1-253           CAGTGAACACACAACTACGCATTTTTAAGCAAAAGA-------------- 1222
IGHVII-28-1*01_hg37/970-2241               CAGTGAACACACAACTACGCATTTTTAAGCAAAAGACGCAATGAAGGGCC 1267
                                           **********************************

IGHV_II_-28-1*03_hg38/969-2245             TTCATTGT 1277
IGHV_II_-28-1*03_reference/1-283           --------
IGHV_II_-30-41*01_reference/1-299          TT------ 1268
IGHV_II_-28-1*02_reference/1-253           --------
IGHV_II_-28-1*01_reference/1-253           --------
IGHVII-28-1*01_hg37/970-2241               TTCATTGT 1275
```

Similarly, pseudogene IGHV(III)-25-1*02 has been modified by removing the 3' insertion relative to IGHV(III)-25-1*01. This was performed for the same reasons as above; the 3' insertion is present in the GRCh37 copy of IGHV(III)-25-1*01. Sequence alignment is provided below:

```
IGHV_III_-25-1*01_reference/1-295          --GAAGTTCACCGGGGGAGACAGAGGAAATAACGGTGCAGCCGGGGGCTA 1057
IGHVIII-25-1*01_hg37/1010-2306             GTGAAGTTCACCGGGGGAGACAGAGGAAATAACGGTGCAGCCGGGGGCTA 1059
IGHV_III_-25-1*02_reference/1-344          --GAAGTTCACCGGGGGAGACAGAGGAAATAACGGTGCAGCCGGGGGCTA 1057
                                             **********************************************

IGHV_III_-25-1*01_reference/1-295          TCTGAGTCTCTCCTCCAAAGACTCTGGATTCACCTTCACTGATTGCAGCA 1107
IGHVIII-25-1*01_hg37/1010-2306             TCTGAGTCTCTCCTCCAAAGACTCTGGATTCACCTTCACTGATTGCAGCA 1109
IGHV_III_-25-1*02_reference/1-344          TCTGAGTCTCTCCTGCAAAGACTCTGGATTCACCTTCACTGATTGCAGCA 1107
                                           ************** **********************************

IGHV_III_-25-1*01_reference/1-295          TAAGCTTGGTCCAGCAAGCTCCAGGACCAGGGTTGATGTGGGCAGCAACA 1157
IGHVIII-25-1*01_hg37/1010-2306             TAAGCTTGGTCCAGCAAGCTCCAGGACCAGGGTTGATGTGGGCAGCAACA 1159
IGHV_III_-25-1*02_reference/1-344          TAAGCTTGGTCCAGCAAGCTCCAGGACCAGGGTTGATGTGGGCAGCAACA 1157
                                           **************************************************

IGHV_III_-25-1*01_reference/1-295          GGGAGAAATTGAAGAGGAAGCTCTCAGTGGTGCCCTCCATGAATACAAAG 1207
IGHVIII-25-1*01_hg37/1010-2306             GGGAGAAATTGAAGAGGAAGCTCTCAGTGGTGCCCTCCATGAATACAAAG 1209
IGHV_III_-25-1*02_reference/1-344          GGGAGAAATTGAAGAGGAAGCTCTCAGTGGTGCCCTCCATGAATACAAAG 1207
                                           **************************************************

IGHV_III_-25-1*01_reference/1-295          AATCTTCACAGTCCCCAGGACACCCTTACGTGC----------------- 1240
IGHVIII-25-1*01_hg37/1010-2306             AATCTTCACAGTCCCCAGGACACCCTTACGTGCATGGTCTCACTGATATC 1259
IGHV_III_-25-1*02_reference/1-344          AATCTTCACAGTCCCCAGGACACCCTTACGTGCATGGTCTCACTGATATC 1257
                                           *********************************
```

```
IGHV_III_-25-1*01_reference/1-295          ------------------------------------
IGHVIII-25-1*01_hg37/1010-2306             TTTACTTCTTTTATCACTTTTGTTATGTAAATCACAAT 1297
IGHV_III_-25-1*02_reference/1-344          TTTACTTCCTTTATCACTTTTGTTATGTAAAT------ 1289
```