Geny: Genotyping Tool for Allelic Decomposition of Killer-cell Immunoglobulin-Like Receptor Genes

by

Mazyar Ghezelji
B.Sc., Urmia University, 2019

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Geny: Genotyping Tool for Allelic Decomposition of Killer-cell Immunoglobulin-Like Receptor Genes

by

Mazyar Ghezelji
B.Sc., Urmia University, 2019

Supervisory Committee

Dr. Ibrahim Numanagić, Supervisor
(Department of Computer Science)

Dr. Ulrike Stege, Departmental Member
(Department of Computer Science)

Dr. Xuekui Zhang, Outside Member
(Department of Mathematics and Statistics)

Supervisory Committee

Dr. Ibrahim Numanagić, Supervisor
(Department of Computer Science)

Dr. Ulrike Stege, Departmental Member
(Department of Computer Science)

Dr. Xuekui Zhang, Outside Member
(Department of Mathematics and Statistics)

ABSTRACT

The accurate genotyping of Killer Immunoglobulin-like Receptors (KIR) plays a pivotal role in enhancing our comprehension of immune responses, disease correlations, and the advancement of personalized medicine. This thesis delves into the intricacies of KIR genotyping methodologies and introduces "Geny," an innovative computational tool formulated for precise allele-level genotyping. Through a comprehensive evaluation, Geny consistently demonstrates superior performance compared to existing tools, notably surpassing T1K, especially within crucial gene segments. The tool's resilience in addressing both fundamental and advanced genotyping tasks highlights its robustness in the face of various challenges. The exceptional precision demonstrated by Geny in identifying critical genes positions it as a valuable resource for advancing the field of patient-centric medicine. By contributing to the evolution of KIR genotyping, this study not only establishes a new benchmark but also highlights the continuing requirement for innovative approaches. We emphasize Geny's remarkable capabilities, recognizing the ever-evolving landscape of genomics. Furthermore, we outline potential future directions, encompassing the detection of gene fusions and the enhancement of mutation identification. These insights pave the way for KIR genotyping to play a pivotal role in shaping the landscape of modern medical research.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all those who have contributed to the completion of this master's thesis.

First and foremost, I am deeply thankful to my supervisor, Dr. Ibrahim Numanagić, for their invaluable guidance and unwavering support throughout this research journey.

I extend my appreciation to my family for their love and encouragement, which have provided me with the strength to overcome challenges.

I would also like to acknowledge the faculty members of the Department of Computer Science for their profound knowledge and insights shared during my academic pursuit as well as my labmates for their invaluable assistance.

Lastly, I am thankful to the University of Victoria for the resources and facilities that have facilitated the smooth progress of this research.

To all those whose names may not be mentioned individually, your contributions have not gone unnoticed. Thank you for being a part of this accomplishment.

DEDICATION

To those who tirelessly explore the mysteries of our bodies, this work is for you. Your dedication to understanding how our immune system works and how it can be harnessed for better health is truly inspiring. Your efforts pave the way for better treatments and a healthier future.

Chapter 1

Introduction

The human immune system is a remarkable defense mechanism that protects the body from infections, viruses, and diseases. At the core of this intricate immune response are the Killer Immunoglobulin-like Receptor (KIR) genes, a family of genes located on chromosome 19 in the human genome. KIR genes encode cell surface receptors that play a pivotal role in regulating the activity of natural killer (NK) cells and certain subsets of T cells [15, 22].

These receptors play a pivotal role in immune surveillance by interacting with major histocompatibility complex class I (MHC-I) molecules found on the surface of various cells in the body [4]. The expression and interactions of KIR genes are essential for distinguishing between healthy and abnormal cells, allowing NK cells to spare normal cells while effectively targeting and eliminating infected or cancerous cells [3]. This genetic diversity contributes to the wide array of immune responses observed among individuals and influences disease susceptibility.

The KIR gene family, located on chromosome 19 within a specific 150kb region of the Leukocyte Receptor Complex (LRC), showcases intriguing genetic characteristics [34]. One of the standout features of KIR genes is their pronounced genetic diversity and polymorphism. Polymorphism in this context refers to the presence of multiple genetic variants within a gene or gene family. This results in a myriad of KIR haplotypes and genotypes within the human population [29]. Importantly, this variation is not limited to just the coding regions; it also encompasses the regulatory regions that direct their expression. Uhrberg [32] proposes that this vast genetic diversity likely stems from the evolutionary pressures posed by constantly evolving viruses. Such intricate genetic architecture means that fewer than 2% of unrelated individuals share an identical KIR genotype [21].

To offer a detailed perspective, approximately 30 unique haplotypes have been identified, broadly categorized into groups A and B, depicted in Figure 1.1 as per [6]. Group A maintains a mostly consistent set of genes, while group B introduces more variability. Nevertheless, a number of genes, referred to as framework genes, remains consistent across most KIR haplotypes. The combined interplay of both maternal and paternal haplotypes introduces an added dimension of complexity, further diversifying individual KIR genotypes.

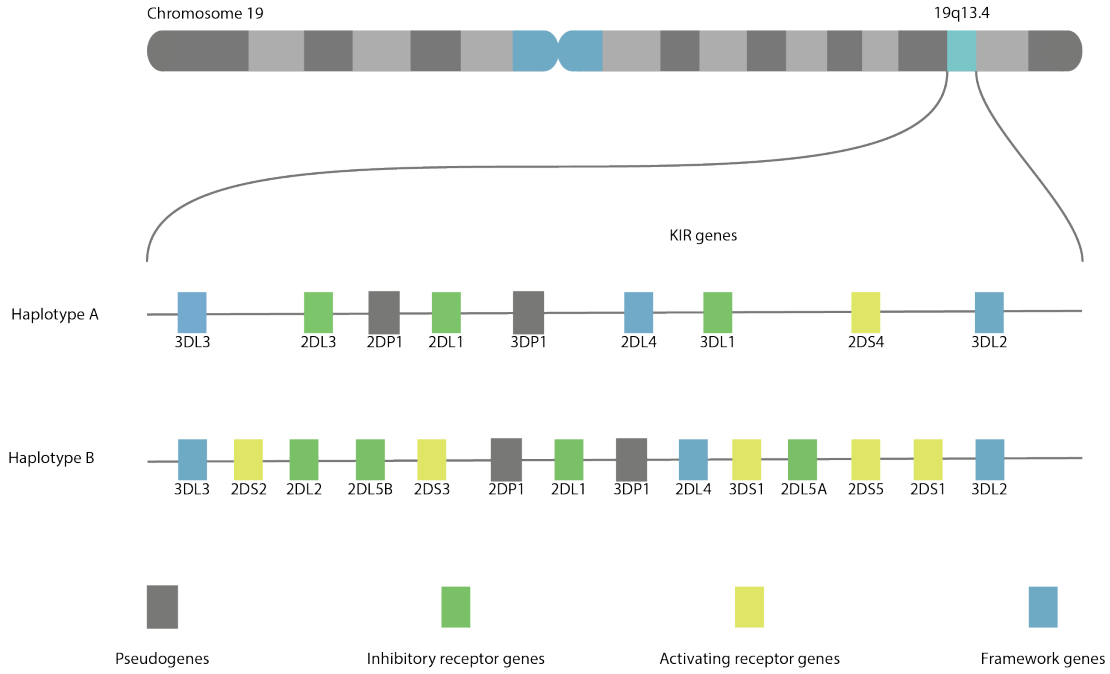

Figure 1.1: Illustration of the KIR gene positions on chromosome 19, showcasing the distinct structures of haplotype groups A and B.

KIR genes are named based on their extracellular Immunoglobulin-like (lg-like) domains (designated as 2D or 3D) and the lengths of their cytoplasmic tails, marked as L for long cytoplasmic tails, S for short cytoplasmic tails, and P for pseudogene. A general rule is that short-tailed KIRs are activating receptors and long-tailed KIRs are inhibitory receptors. Sequential numbers and asterisks are used for further differentiation and to highlight allelic variations [21]. Based on these designations, the KIR genes can be categorized as follows:

- Genes with two domains and long cytoplasmic tails include *KIR2DL1*, *KIR2DL2*, *KIR2DL3*, *KIR2DL4*, *KIR2DL5A*, and *KIR2DL5B*.

- Those with two domains and a short cytoplasmic tail are *KIR2DS1*, *KIR2DS2*, *KIR2DS3*, *KIR2DS4*, and *KIR2DS5*.
- KIR genes with three domains and long tails are *KIR3DL1*, *KIR3DL2*, and *KIR3DL3*,
- *KIR3DS1* is characterized by having three domains and a short tail.
- Additionally, the KIR gene family includes two pseudogenes: *KIR2DP1* and *KIR3DP1*.

The intricate relationship between KIR genes and human leukocyte antigen (HLA) molecules has significant implications for disease associations. Specific KIR-HLA combinations have been associated with altered susceptibilities to various infectious diseases, autoimmune disorders, and cancers. For example, KIR genes have been found to play a crucial role in HIV-1 control [23]. Certain KIR-HLA profiles can confer protection against viral infections, while others may increase susceptibility to certain diseases.

As our understanding of the role of KIR genes in immune regulation and disease associations grows, the demand for precise and efficient genotyping methods has increased. High-throughput sequencing (HTS) technologies have emerged as powerful tools for genotyping KIR genes, enabling comprehensive analysis of KIR loci and their interactions with HLA molecules [19].

Detecting mutations can be a crucial step for ascertaining the presence of specific genes, and determining their alleles and copy numbers which are pivotal in biomedical research and clinical practice. These genetic variations can influence susceptibility to diseases, affect therapeutic responses, and predict the risk of hereditary conditions. For example, certain mutations can increase cancer susceptibility [28]. Specific gene alleles can influence drug metabolism, leading to variations in drug response and potential adverse reactions [8]. Furthermore, copy number variations (CNVs) can result in developmental disorders and have been implicated in conditions such as autism and schizophrenia [25]. Therefore, accurate detection and understanding of these genetic elements play a crucial role in personalized medicine, disease prevention, and therapeutic interventions.

A multitude of researchers have advanced computational methodologies to address the intricate challenges posed by genotyping highly polymorphic genes. Notably, tools developed by Numanagić et al. (Aldy), Lee et al. (Stargazer), Twesigomwe

et al. (StellarPGx), and Twist et al. (Astrolabe) stand out as premier genotyping approaches for pharmacogenes [18, 11, 30, 31]. Within the realm of KIR genes, contributions from Song et al. (T1K), Norman et al. (PING), Roe and Kuang (KPI), and Vukcevic et al. (KIR*IMP) offer distinct strategies for the analysis of these genes and their respective variants [26, 17, 24, 33].

Aldy is a novel computational tool based on integer linear programming techniques that aims to analyze data from high-throughput sequencing (HTS) technologies to call genotypes and potentially identify genetic variations. It demonstrates higher accuracy and computational efficiency than other methods, making it suitable for large-scale studies and various types of genomic variation, including copy number variations, gene fusions, and complex mutations. While Aldy is optimized for pharmacogenes and adept at identifying specific known variations within them, it may not be tailored to handle the complexities of KIR genes, as it was designed to work with one gene with fewer complexities than KIRs. The abundance of closely related genes, coupled with a multitude of highly similar and polymorphic alleles, necessitates specialized mapping techniques and a tailored tool for precise representation.

T1K is a cutting-edge genotyping tool, grounded in the expectation maximization methodology, tailored for precise and rapid identification of KIR and HLA (Human Leukocyte Antigen) genotypes from sequencing data. While T1K offers speed and commendable accuracy, it currently lacks the capability to determine the copy number of alleles and genes. This limitation can be crucial when multiple copies of the same allele or gene are present.

PING (Pushing Immunogenetics to the Next Generation) is a groundbreaking method tailored to decode the complexities of the highly polymorphic KIR and HLA class I genes. Notably, PING identified 116 novel KIR alleles, greatly enriching the current KIR allele database. The method stands out for its efficiency and cost-effectiveness in genotyping KIR genes. Moreover, its flexibility allows adaptation for other polymorphic gene systems. However, it's worth noting that PING, being specifically designed for KIR-targeted amplified data, may occasionally overlook specific KIR genes, especially those with close similarities like *KIR3DL1* and *KIR3DS1*, during allele predictions [26].

The KPI (KIR Probe Interpretation) algorithm is a novel technique designed to precisely predict KIR genes and haplotypes using Whole Genome Sequencing (WGS) data. The method employs a synthetic SSO-like (k-mer) library and marker-based genotyping, utilizing a comprehensive multiple-sequence alignment of full-length hap-

lotypes. Through rigorous testing, the algorithm has showcased remarkable accuracy in gene and haplotype-pair predictions, significantly advancing our comprehension of KIR gene diversity and its relevance in immunogenetics research. However, it is important to note that the primary focus of this algorithm lies in gene-level analysis, rather than extensively exploring allelic variations which play a key role in tailoring medical interventions to an individual's genetic makeup.

KIR*IMP is a statistical imputation tool, designed to interpret the complexities of the KIR gene region, adeptly identifying genetic variations with an emphasis on SNP mutations on alleles. Its distinctive ability to unravel the vast diversity of specific haplotypes and discern between various KIR copy-number types is considerable. Despite its benefits, KIR*IMP primarily operates by harnessing the information of SNP genotypes and might inadvertently omit some pivotal SNPs due to data reliability concerns. Furthermore, it can only impute variants that exist in its reference panel, ensuring accuracy only when there are enough examples. Such limitations indicate potential areas for improvement and refinement in its approach and sets the stage for the emergence of novel tools that might offer heightened precision and comprehensive insights into the complex area of KIR genotyping.

While several computational tools have been devised to address the challenges of KIR genotyping, each presents specific limitations. Aldy, despite its strength in pharmacogene variation identification, is not inherently tailored for the intricacies of KIR genes. T1K's swift and precise methodology has a notable shortcoming in determining allele and gene copy numbers. PING's unique approach can sometimes bypass certain KIR genes with close similarities. The KPI algorithm, although adept at gene-level analysis, does not extensively explore vital allelic variations. And KIR*IMP, even with its specialization, might exclude essential SNPs owing to data reliability issues. Recognizing these shortcomings, we have designed and implemented a new algorithm, aiming to offer a more comprehensive solution to the multifaceted challenges of KIR genotyping.

While current tools have substantially advanced progress in genomics and KIR gene genotyping and analysis, there remain discernible areas of enhancement. To bridge these identified gaps, we introduce "Geny" (GENotYper for KIR genes). This tool, built on a robust foundation of expectation minimization and integer linear programming, offers fast and accurate allele-level genotyping and copy number calling. Furthermore, it is able to detect and leverage all mutation types found in the KIR database. In the upcoming chapter, we delve deeper into the details of the implemen-

tation of Geny.

Chapter 2

Method

2.1 Preliminaries

Let the alphabet Σ be defined as $\Sigma = \{A, C, G, T\}$, representing the nucleotide bases Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). These bases are the fundamental building blocks of human DNA sequences. Each sequence $G = G_1, G_2, \ldots, G_m$ where $G_i \in \Sigma$ represents a DNA sequence. Within the scope of this thesis, G represents a KIR gene sequence.

A read r refers to a relatively short fragment of DNA that has been determined through a sequencing method. This fragment originates from randomly shearing DNA samples into smaller pieces, which are then sequenced using technologies such as Illumina's next-generation sequencing (NGS) platforms or Oxford Nanopore's long-read sequencing [9]. Reads can be generated in two different forms; single-end and paired-end. Single-end reads involve sequencing from one end of the DNA fragment, providing a linear snapshot of the genetic sequence. They are simpler and quicker to generate but may not capture the full context or complexity of the genomic region. In contrast, paired-end reads sequence both the forward and reverse ends of a DNA fragment. This method offers greater accuracy, especially in regions of repetitive DNA or where structural variants exist, by capturing the entire span of the fragment. The distance between the paired reads, known as the insert size, can provide critical information about the structural layout of the genome. Once these reads are generated, they are aligned to a reference genome, essentially piecing them together in the correct order to reconstruct the original DNA sequence. In the context of genotyping, these aligned reads are examined at specific genomic positions or loci to identify the

specific nucleotide(s) present. By comparing these nucleotides with known reference sequences, one can determine the genotype of an individual at that position.

Each gene G can exhibit multiple variations, indicating that mutations have altered the gene. These variations typically arise from genetic mutations, which can manifest through a range of mechanisms such as:

- Deletion: This process involves the removal of one or more nucleotide bases from a DNA sequence. Deletions can have profound effects on gene function and lead to various genetic disorders [27, 16].

- Insertion: This refers to the addition of one or more nucleotide bases into a DNA sequence. Like deletions, insertions can have significant consequences on gene function [27, 16].

- SNP (Single Nucleotide Polymorphism): A change at a single position in a DNA sequence that varies among individuals. SNPs serve as biological markers and help scientists locate genes associated with specific diseases [1].

Another type of genomic alteration is fusion which arises when two previously distinct genes merge, often due to events like translocations. These variations are common in several cancer types [14].

In the context of This thesis, we use the term "valid allele" for alleles that pass the allele-filtering stage, "valid reads" for reads that pass the read-filtering stage, and "true allele" for the alleles present in the sample DNA.

As highlighted in the previous chapter, the detection of mutations is essential for identifying specific genes and discerning their alleles and copy numbers. Such determinations are fundamental in both biomedical research and clinical settings.

Our model uses Integer Linear Programming (ILP) as its solver. To understand ILP, it's important to first comprehend Integer Programming (IP). IP is a mathematical optimization technique where some or all of the decision variables are constrained to take integer values. It is widely used to model and solve combinatorial optimization problems arising in various domains, such as scheduling, production planning, and network design. On the other hand, ILP is a subset of IP where the objective function and constraints are linear. Essentially, ILP is an extension of linear programming, but with the added restriction that one or more variables must be integer-valued, making the problem NP-hard and, at times, computationally challenging to solve [35].

Our objective is to develop a novel algorithm that can precisely and efficiently detect the KIR genes present in a patient sample, identify the alleles of those genes and determine the copy number of each one, leveraging high-throughput sequencing reads as input. In the subsequent sections, we will delve into the specifics of our methodology.

2.2 Overview

Our method comprises three essential components, each contributing to the overall process of accurately determining the true alleles and their copy numbers. These components are designed to progressively refine the input data and provide an optimal foundation for the final stage, which involves the design and execution of an Integer Linear Programming (ILP) algorithm.

The first stage of our method involves mapping the reads to the reference sequence. This initial mapping step allows us to identify the potential locations in the genome where each read could be aligned. By determining these potential mapping positions, we establish a starting point for further analysis.

The subsequent stage of our method focuses on filtering the alleles and reads to enhance the quality of the input data for the final stage.

The allele filtering stage aims to narrow down the list of potential alleles by applying specific criteria, such as coverage of functional mutations and the percentage of non-covered positions. This filtering process ensures that the selected alleles align with the observed sequencing data and meet the defined thresholds.

Following the allele filtering stages, we proceed to the read filtering stage. Here, we apply additional filters to the reads, selecting those that align more accurately with the chosen alleles. This filtering step further enhances the quality and reliability of the input data for the subsequent analysis.

Finally, armed with the refined input data, we enter the last stage of our method. In this stage, we employ an ensemble method to obtain the optimal result. The first solver, which utilizes the Expectation Maximization (EM) algorithm, further refines the selection of alleles. Through iterative parameter estimation, the EM algorithm improves the alignment of the chosen alleles with the sequencing data, maximizing the likelihood of observing the given data.

The last solver is used for generating the final results. This stage involves designing and solving an Integer Linear Programming (ILP) algorithm. The ILP algorithm

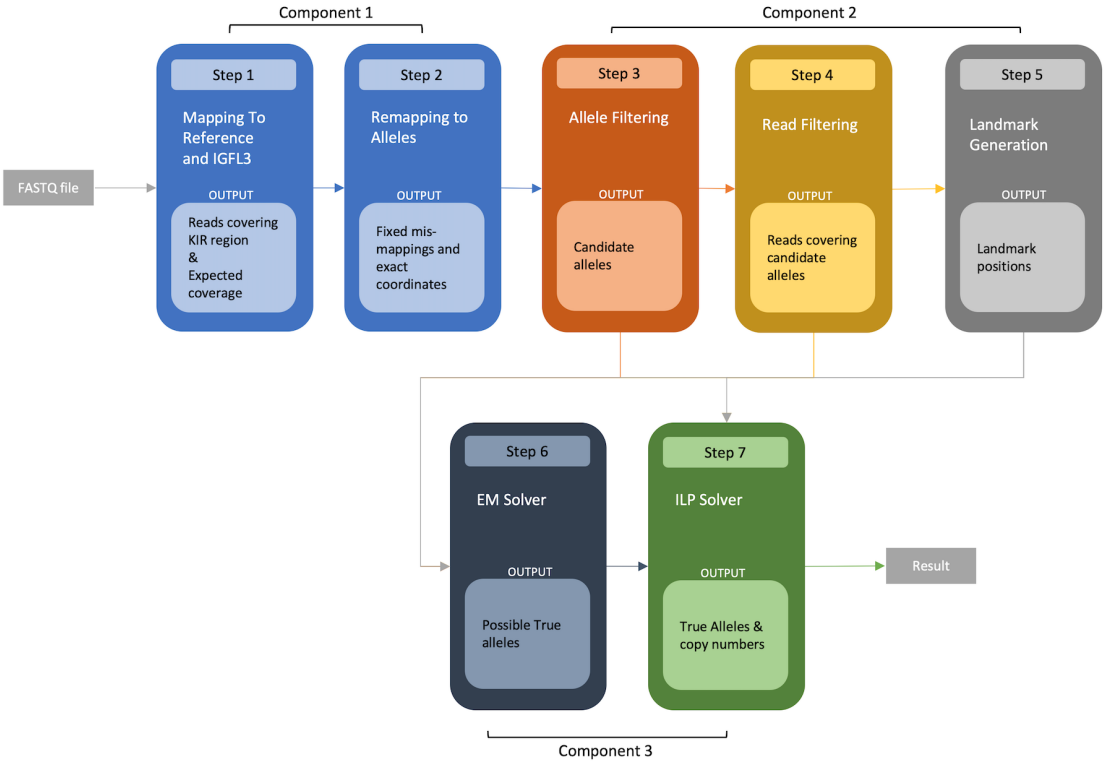

Figure 2.1: Schematic representation of the pipeline of our proposed tool.

assigns the reads to the potential true alleles, taking into account their copy numbers. By optimizing the assignment of reads, the ILP algorithm determines the true alleles and their respective copy numbers, providing valuable insights into the genetic variations present in the data. An illustration of the workflow is shown in Figure 2.1.

Next, we will delve into each stage of our method in detail, outlining the specific processes and considerations involved.

2.3 Mapping to reference alleles

In our algorithm, we have the flexibility to utilize two types of inputs: next-generation sequencing reads or mapped reads to the reference sequence. Regardless of the input type, our first step is to map the reads to the reference sequence. This process allows us to identify the corresponding locations in the genome where these reads align.

However, in the case of the KIR region of chromosome 19, we need to take an additional step to ensure accurate mapping. Specifically, for reads that can be mapped to the KIR region, we remap them to the allele sequences associated with the KIR region. By aligning these reads to the allele sequences, we enhance the precision of mapping, which is vital for subsequent analysis. Accurate mapping serves as a fundamental building block for downstream analysis and ensures reliable results.

To achieve this, we employ a reliable mapper called minimap2 [12]. This tool is well-regarded for its ability to support multi-mapping, which means it can handle cases where a read can be aligned to multiple locations in the genome. Detecting multi-mapping is crucial, particularly when it comes to identifying pseudo-genes such as *KIR2DP1* and *KIR3DP1*. By utilizing minimap2, we can effectively identify and handle these scenarios.

An additional component of this stage involves mapping the reads to a copy number-neutral gene, IGFL3, sequence. This gene is part of the insulin-like growth factor family of signalling molecules [7]. Mapping to this sequence provides us with the expected coverage of the sequencing process, which we subsequently use in the optimization phase.

After performing the all-to-all mapping of the reads to the respective sequences, we obtain a comprehensive collection of locations that each read can be mapped to. This information provides us with the basis necessary for further analysis. These bases serve as the foundation for subsequent steps, enabling us to delve deeper into the analysis of the data and extract meaningful insights.

2.4 Allele filtering

In our analysis, we encounter a substantial number of KIR genes, amounting to a total of 17, with an extensive set of alleles, totaling 1549. However, directly using all these alleles as input to an Integer Linear Programming (ILP) solver can lead to computational load and the existence of noise can potentially impact the accuracy of the results. To mitigate this issue, we employ a filtering approach to limit the number of valid alleles that we consider for further analysis. This filtering process involves several criteria outlined below:

1. Functional mutations coverage: Functional mutations are alterations in gene sequence that cause changes in the final produced proteins by the gene. In order to detect trues present alleles we need to make sure that all functional mutations are covered by the reads. To ensure comprehensive coverage of functional mutations, we prioritize the inclusion of alleles where all functional mutations are covered by at least one read. In the case of wildtype alleles, which are sequences without mutations, we adopt a strategy to facilitate their analysis. We select specific positions along their sequence and treat them as virtual mutations. This approach allows us to assess the coverage of these wildtype alleles in a similar manner to alleles with actual mutations.

   To include wildtype alleles in the selection of valid alleles, the number of virtual mutation positions not covered should be less than a predefined threshold, denoted as T_1. The less the value of T_1, the less we take sequencing errors into account.

2. Overall Coverage Percentage: In addition to functional mutations, we evaluate the coverage of all positions within an allele. The percentage of positions that are not covered by any read should be less than a specified threshold, denoted as T_2. T_2 can be determined experimentally using known sequences and controlled simulations.

3. Non-Functional Mutation Coverage: Non-Functional mutations are alterations in gene sequence that do not cause changes in the final produced proteins by the gene. This optional step can be used for ranking selected alleles, particularly when dealing with a large number of valid options. We assess the coverage of non-functional mutations within the alleles, ensuring that the percentage of uncovered non-functional mutations is below the threshold value T_3. This

additional criterion enhances the selection process, allowing us to prioritize alleles with better coverage of non-functional mutations for further analysis.

4. Selection of A_G Alleles per Gene G: some genes can have many similar alleles that all can pass the filtering step. To maintain a manageable number of alleles for the solver phase and to accommodate other genes in subsequent stages, we set a criterion to choose up to A_G alleles for each gene G. This approach streamlines the allele count for further analysis.

It is important to note that these filtering steps are performed sequentially, with the output of one step serving as the input for the next. Consequently, the result of this filtering process is a list of candidate alleles, where for each gene G, there are A_G or fewer alleles that satisfy the defined criteria.

Selecting values for thresholds T_1 and T_2 requires careful consideration since they play a key role in defining which alleles will be included in the following steps. We established these thresholds after experimentation with various datasets. Our selected values for T_1 and T_2 are 10 and 50 respectively.

By implementing these filtering criteria, we can effectively reduce the number of alleles considered for analysis, focusing on those that are most relevant and likely to provide accurate results. This approach optimizes the efficiency of the ILP solver and enhances the overall accuracy of the subsequent analyses.

2.5 Filtering reads

After identifying the valid alleles in the previous stages, the next crucial step is to filter the reads to optimize efficiency and focus only on the ones that provide valuable information for the allelic decomposition process. It is unnecessary to always use all the reads, as some may not contribute significantly to the task at hand. Therefore, we aim to select the minimum set of reads that can effectively distinguish between alleles.

To filter the reads, we first need to determine the important mutation positions that can help differentiate between alleles. These positions play a crucial role in the selection process. One approach is to consider all mutation positions within the valid alleles for each gene individually. However, this method can lead to unnecessary computational overload, as not all positions may be essential for distinguishing between alleles. Another approach is to focus solely on functional mutations that are present

in our KIR database. While more efficient as there are fewer positions, this approach may not be sufficient to distinguish alleles with identical functional mutations.

In our approach, we generate a set of important positions for each gene G. These important positions can be generated by taking the union of mutations found on the valid alleles of that gene. By using this approach, we can capture the important differences between alleles in an efficient manner. This allows us to include only the necessary mutation positions, reducing computational complexity while still ensuring the ability to distinguish between alleles effectively.

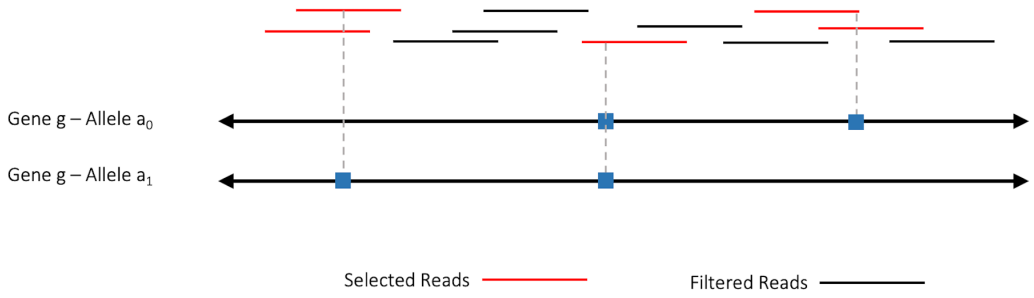

Figure 2.2: Diagram illustrating the read filtering stage. Two alleles of the same gene are shown. Reads selected for analysis are highlighted in red, with blue squares indicating the mutations in each allele. The chosen reads encompass the combined mutations of both alleles.

Once we have the list of important positions, we examine all the reads that can be mapped to those positions with a mapping cost equal to or less than a specified threshold, denoted as T_e. We add these reads to our list of valid reads as shown in Figure 2.2. In cases where the reads are paired-end, we must also consider the other paired read to ensure that both reads can be mapped to the allele under consideration.

By carefully filtering the reads based on important mutation positions in the read filtering step, we obtain a refined set of valid reads that provide the necessary information for accurate allelic decomposition. This step optimizes computational efficiency, reduces noise from non-informative reads, and ensures that the subsequent analysis focuses on the most relevant and informative data.

A significant challenge encountered during this step is the possibility of a read from a non-mutation position of one allele being erroneously mapped to a mutation position of another allele. We expect this situation due to the high similarity between different alleles, making it crucial to address this problem and prevent mismatches in

subsequent steps.

To overcome this challenge and ensure accurate mapping, we incorporate non-mutation positions of the alleles into consideration. By including some of these non-mutation positions, reads that truly originate from these positions have the opportunity to be mapped back to their correct locations as the solver tries to assign reads to mutation positions that they cover in order to find the true locations they originate from, resulting in a more realistic coverage of important positions. However, it is essential to exercise caution during this process to avoid excessive increases in the number of important positions, which could lead to computational inefficiencies.

Moreover, including non-mutation positions as input to our ILP solver may result in lower coverage for those positions. This occurs because we did not initially consider all reads covering these non-mutation positions in the first step. Thus, it is necessary to strike a balance between capturing important reads and positions while avoiding excessive computational complexity and potential coverage loss which we will discuss more in the next section.

To address these challenges effectively, we introduce the concept of landmarks as an alternative to using mutation positions. The generation of landmarks will be discussed in detail in the following section. By incorporating landmarks, we can appropriately identify specific positions within alleles that are distinctive and informative for subsequent analysis. This approach allows us to include important reads and positions in our algorithm while mitigating potential mismatches and maintaining computational efficiency.

2.6 Landmark Generation

When analyzing the obtained set of valid alleles and reads, it is common to encounter overlapping regions among the selected reads where multiple mutation positions are located in that region. However, such overlaps can introduce redundancy as the mutation positions are close to each other and assigning a read to one mutation position is equivalent to assigning it to the other, resulting in increased computational overhead and potential compromises to final accuracy. Furthermore, some valid reads can have multimappings (a situation in which a read can be mapped to different positions) on the same and other alleles. To address these challenges and also accommodate reads from important non-mutation positions, we introduce the concept of landmarks.

Landmarks are projections of mutations (SNPs) on alleles that do not possess mu-

tations but some valid reads can be mapped to them. The objective of landmarks is to provide the opportunity for the reads to be assigned to those alleles while maintaining a minimal number of those positions.

For each candidate allele ($A_C \in A_G$), we construct an overlap graph (G_C) to capture the relationships between reads that align to the mutations of A_C. Each read corresponds to a node in the graph, and an edge is created between two reads (r_1 and r_2) if there is an overlapping region between them on allele A_C. Constructing the overlap graph enables us to identify the connections among reads aligned to the mutations of each candidate allele.

Subsequently, we identify the strongly connected components (SCCs) within the overlap graph for each candidate allele. These SCCs represent groups of reads that have mutual alignments with each other. To effectively handle the challenge of multi-mapping, which is particularly prevalent in genes like *KIR2DP1* and *KIR3DP1* based on our experiments on the KIR database, we consider all mappable positions of the reads on each candidate allele when generating the SCCs.

The next step is to select a minimal number of landmarks so that every read within each SCC covers at least one landmark. This selection process involves strategically choosing the minimum number of positions that maximize the coverage of valid reads. For each candidate allele (A_C), we identify a set of landmarks (L_C) that best represent the critical regions of interest. An illustration of the process is shown in Figure 2.3.

As a final step in this stage, we revisit the identified landmarks and expand the set of valid reads by including all the reads that can cover these landmark positions. By doing so, we enhance the coverage of important positions, resulting in a more comprehensive and accurate subsequent analysis.

2.7 First Allele Caller: Expectation Maximization

After obtaining a set of valid alleles and reads, we proceed with the first solving step using the Expectation Maximization (EM) algorithm. This step helps refine the selection by identifying alleles with lower expectations in the input sample, thus reducing the solution space for the final solver and improving specificity [13].

The EM algorithm, initially proposed by Dempster et al. [5], in a setting where the input data in partially known and the parameters of the distribution function (model) that generated the data are unknown, iteratively estimates the parameters of the model to maximize the likelihood of the observed data. Here our goal is to

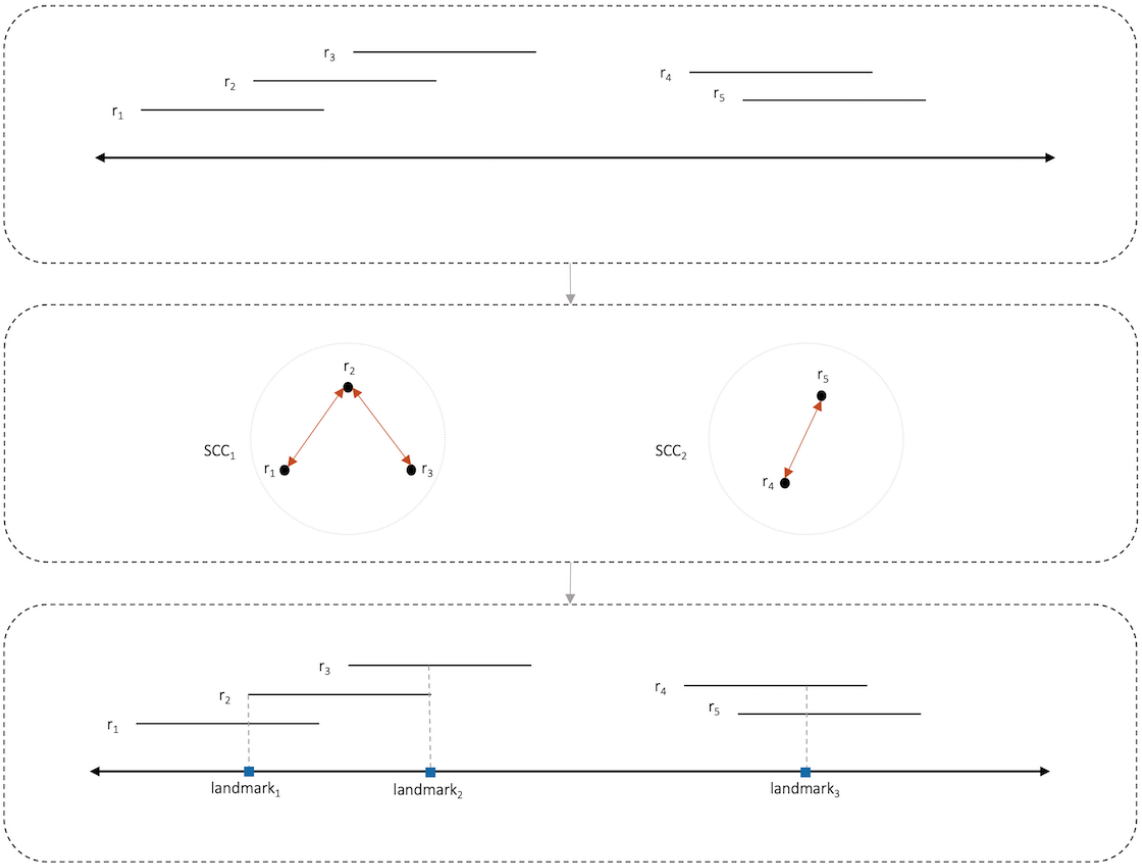

Figure 2.3: Diagram illustrating the process of landmark generation. Initially, all valid reads are collected as input. These reads are then utilized to construct a graph that represents their overlap. Then, we identify strongly connected components (SCCs) of that graph which represent groups of reads that continually cover a region by overlapping each other. In the concluding step, an iterative approach is used to determine the minimal set of landmarks needed to capture all the valid reads within each SCC. With each iteration, the number of landmarks is increased until all reads in a given SCC are captured.

perform maximum likelihood estimation on the abundance of each allele.

Let $\theta = \{\theta_1, ..., \theta_n\}$, denote the abundance (percentage) of each of the n candidate alleles, $\mathcal{R}$ be the set of reads, $\mathcal{L}(\theta)$ The log-likelihood of $\mathcal{R}$ given the parameter θ, Z_j^i is the latent variable representing the probability that read R_i is generated by allele A_j, θ_s^t be the probability that a read is generated by allele A_s at iteration t.

We define the log-likelihood $\mathcal{L}(\theta)$ as follows:

$$\mathcal{L}(\theta) = \log P(\mathcal{R} \mid \theta) = \sum_{i=1}^{m} \log P(\mathcal{R}_i \mid \theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{n} P\left(\mathcal{R}_i | mid Z_j^i\right) P\left(Z_j^i; \theta\right) \quad (2.1)$$

For simplicity, we define $c_j^i = P\left(\mathcal{R}_i | Z_j^i\right) = \frac{N_j^i}{L_j}$ where N_j^i denotes number of positions read $\mathcal{R}_i$ can be aligned to allele A_j and L_j denotes the length of allele A_j. During the E-step of the EM algorithm, we compute the expectation of the complete log-likelihood with respect to the latent variable, as follows:

$$\mathbb{E}_{z|\mathcal{R};\theta^t}[\log P(\mathcal{R}, Z \mid \theta)] = \sum_{i=1}^{m} \sum_{j=1}^{n} P\left(Z_j^i \mid \mathcal{R}_i, \theta^t\right) \log P\left(\mathcal{R}_i, Z_j^i \mid \theta\right) \quad (2.2)$$

In the M-step, we update the parameter θ_s^t, with maximum likelihood estimation on (2). by:

$$\theta_s^{t+1} = \mathbb{E}_j[P\left(Z_j^i \mid \mathcal{R}_i; \theta^t\right)] = \mathbb{E}_j\left[\frac{c_s^i \theta_s^t}{\sum_k c_k^i \theta_k^t}\right] = \frac{1}{m} \sum_{i=1}^{m} \frac{c_s^i \theta_s^t}{\sum_k c_k^i \theta_k^t} \quad (2.3)$$

Once the updated θ_s values are obtained, we select all alleles associated with activated components, where $\theta > \epsilon$, for further refinement using the Integer Linear Programming (ILP) solver. Here, we set ϵ to a very small value, specifically $\epsilon = 10^{-10}$, which is chosen to be close to zero. This selection criterion assumes that alleles with abundances below this threshold are not present in the sample. By applying this filtering step, we focus on alleles that have a substantial presence in the sample, improving the accuracy and reliability of the results.

It is important to note that selecting the appropriate threshold for allele density is crucial for obtaining accurate outcomes. While previous studies have reported thresholds in the range of 0.1-0.25 [26], the robustness of hyper-parameter selection remains questionable. The process of determining the threshold for selecting alleles

lacks theoretical proof of optimality.

Therefore, to enhance specificity and achieve more reliable results, further refinement of the identified alleles is necessary. This can be accomplished by applying additional steps or algorithms to improve the accuracy of allele selection and overcome potential limitations or uncertainties in the initial filtering process.

2.8 Second Allele Caller: Integer Linear Programming

In the ILP formulation for allele refinement, given a set of reads, alleles, positions on those alleles, and expected coverage of reads on those positions, we aim to determine the optimal assignment of reads to positions on alleles in order to minimize the difference between expected coverage and assigned coverage on each position and accurately identify the true alleles present in the sample. The ILP formulation is as follows:

Let $r_1...r_R$ represent the set of all R reads from the valid reads, and $a_1...a_A$ denote the set of all alleles from the previous solver (EM). We introduce binary variables $V_{i,j}$, where $V_{i,j}$ indicates whether read r_j is assigned to allele a_i. To allow the possibility of dropping reads, we define the variable $D_j = 1 - V_j$, where D_j equals 1 if the read is dropped.

To ensure the correct assignment of reads to alleles, we introduce binary variables S_i, where $S_i = 1$ indicates that allele a_i is selected. We enforce constraints to ensure the correct assignment of reads, such that $S_i \leq \sum_j V_{i,j}$ and $S_i \geq V_{i,j}$. These constraints ensure that if an allele is selected, at least one of its assigned reads is also selected.

To account for the copy number variation, we introduce an integer variable C_i to represent the copy number of allele a_i. This variable is bounded by $C_i \geq 1$ and $C_i \leq M$, where M is the maximum allowed copy number, typically set to 12. Additionally, a selection cost SC is associated with each selected allele, which can influence the determination of the copy number. This additional cost is determined through a manual assignment. We employ a heuristic approach by using the read drop cost multiplied by the expected coverage, which has demonstrated satisfactory results in our test cases.

The objective of the ILP formulation is to minimize the total absolute error. This

objective is expressed as:

$$\min \left[\left(\sum_i S_i \cdot \left(\left(\sum_{l \in i} \left| \left(\sum_{j \in l} V_{i,j} \right) - E \cdot C_i \right| \cdot W_i \right) + SC \right) \right) + \left(\sum_j D_j \cdot DC \right) \right] \tag{2.4}$$

In this objective function, E represents the expected coverage for a single copy of an allele, determined by comparing it to a copy number-neutral region. The selection cost SC influences the preference for selecting particular alleles, potentially aiding in the accurate determination of copy numbers. The read drop cost DC accounts for the cost associated with dropping reads.

The objective function involves summations over alleles i, mutations l within each allele, and reads j in the set of valid reads. The weight W_i represents the relative importance of allele a_i and is defined as $\frac{\max_i N_i}{N_i}$, where N_i is the number of landmarks on allele i. This weighting factor ensures that alleles with more landmarks have a higher impact on the optimization process.

By formulating the problem as an ILP and minimizing the total absolute error, we effectively optimize the assignment of reads to alleles, resulting in more accurate and reliable allele identification. The ILP refinement step plays a crucial role in enhancing the specificity of the analysis and obtaining a refined set of alleles that correctly represent the true alleles present in the sample.

Note that the specific implementation and parameter settings may vary depending on the algorithm and software used for the ILP solver. We employed Gurobi as a reliable tool for solving ILP problems in an efficient manner. [20].The ILP formulation presented here provides a general framework for allele refinement based on integer linear programming.

2.9 Implementation

The entire methodology is implemented in the Python programming language. We selected Python due to its efficiency in model implementation and its widespread popularity, which facilitates code-sharing with fellow researchers. Inputs are accepted in the FASTQ file format, and we utilize the minimap2 library for mappings [12]. While we incorporated the EM algorithm in our code, we leveraged the Gurobi solver for addressing the ILP problem [20].

Chapter 3

Results

We assessed the performance of our system using real chromosome 19 data. To simulate real high-throughput sequencing reads, we employed the ART Illumina simulator [10]. Utilizing this tool, we generated simulations with a 10x coverage, which subsequently served as the input for our model. The outcomes were then compared with the true alleles inherent in the sample, and the accuracy was subsequently computed.

Our database comprises 48 real chromosome 19 samples sourced from GeneBank [2], encompassing both haplotype classes A and B. Leveraging this database, we also have access to annotated alleles for each KIR gene present. In a few instances where annotations are absent, we undertake a manual labelling process by identifying the nearest matching sequence within our KIR database.

Our database encompasses all 15 KIR genes along with two pseudogenes, cumulatively containing 405 true alleles spread across these genes. The distribution of alleles for each gene can be referenced in Table 3.1.

To gain a comprehensive understanding of our results, we need to compare our findings with a state-of-the-art genotyper for KIR genes. As described in the introduction section, there are multiple unique tools that can be used. As we need a tool that is able to do accurate allele-level genotyping in a timely manner using high throughput sequencing reads similar to our method, we chose T1K for comparisons. Comparison with T1K, being the latest and the most accurate model for KIR genotyping, can clearly show the capabilities of our model. In this comparison, the dataset supplied to T1K was identical to that used in our system.

Furthermore, we rigorously assessed our system's efficacy in scenarios where both two and three haplotypes are present. This evaluation aims to determine the system's accuracy in discerning the correct copy numbers of the extant genes and alleles. For

Simulation	Total	2DL1	2DL2	2DL3	2DL4	2DL5A	2DL5B	2DP1	2DS1	2DS2	2DS3	2DS4	2DS5	3DL1	3DL2	3DL3	3DP1	3DS1
one copy	405	34	15	28	39	10	9	33	14	15	7	32	9	30	46	42	32	10
Two copies	355	28	13	25	37	11	4	26	11	13	5	29	9	27	41	36	29	11
Three copies	386	28	16	25	39	14	6	28	17	15	10	26	9	25	45	40	29	14

Table 3.1: Number of alleles present in all simulated samples

the two-copy scenario, input data was generated by randomly selecting two samples from our initial set of 48, and subsequently using all reads from these two samples. The methodology for generating three-copy samples is similar to that of the two-copy process by randomly selecting three samples from our initial set and all their reads. In our study, we examined 21 two-copy samples and 15 three-copy samples. The distribution of alleles for each gene, in the context of two and three copies, is detailed in Table 3.1.

Figure 3.1: Comparison of performances for each gene in different copy number settings.

Algorithm	Total	2DL1	2DL2	2DL3	2DL4	2DL5A	2DL5B	2DP1	2DS1	2DS2	2DS3	2DS4	2DS5	3DL1	3DL2	3DL3	3DP1	3DS1
Geny: One copy	**95.8**%	97.0%	100%	100%	100%	100%	88.0%	96.9%	78.5%	93.3%	100%	96.8%	100%	96.6%	95.6%	85.7%	100%	100%
T1K: One copy	93.5%	97.0%	100%	100%	100%	100%	88.0%	100%	78.5%	93.3%	100%	96.8%	100%	96.6%	80.4%	78.5%	100%	100%
Geny: Two copies	**91.8**%	96.4%	100%	100%	97.2%	81.8%	100%	100%	90.9%	84.6%	80.0%	100%	88.8%	92.5%	82.9%	83.3%	89.6%	81.8%
T1K: Two copies	83.0%	92.8%	92.3%	88.0%	89.1%	100%	75.0%	88.4%	81.8%	76.9%	100%	89.6%	100%	92.5%	78.0%	47.2%	82.7%	72.7%
Geny: Three copies	**87.3**%	92.8%	93.7%	100%	92.3%	85.7%	100%	96.4%	94.1%	93.3%	50.0%	96.0%	100%	92.0%	82.0%	70.0%	72.4%	85.7%
T1K: Three copies	67.6%	57.1%	100%	56.0%	64.1%	78.5%	100%	64.2%	70.5%	80.0%	70.0%	80.7%	88.8%	80.0%	53.3%	57.5%	65.5%	64.2%

Table 3.2: Algorithm performance by gene and simulation setting

We conducted an in-depth evaluation of our model, comparing its accuracy against the T1K algorithm. In our context, accuracy is calculated by the proportion of major alleles that were correctly identified out of the total extant alleles. To enhance our understanding of the model's performance across the genetic spectrum, we also computed the accuracy for each specific gene. This detailed approach aids in identifying both the strengths and potential areas for refinement in our model. The final result can be seen in Table 3.2 and Figure 3.1.

In an exhaustive assessment of KIR genotyping methodologies, two primary algorithms, Geny and T1K were critically evaluated across three distinct simulations to ascertain their efficiency and precision.

Upon examination of the results in the one-copy simulation, Geny excels with an aggregate accuracy of 95.8%. Although its accuracy across genes like *2DL3*, *2DL4*, and *2DL5A* touches perfection at 100%, a noticeable drop is seen in the *2DS1* category, descending to 78.5%. Meanwhile, the T1K, which exhibits a total accuracy of 93.5%, maintains par excellence in categories such as *2DL4* and *2DL5A*, mirroring the exactitude of Geny. However, its performance in the *3DL2* and *3DL3* genes, falling to 80.4% and 78.5% respectively, reveals some potential areas of improvement.

Transitioning to the two-copy simulation, Geny manifests a stable performance, boasting a cumulative accuracy of 91.8%. Even though it maintains a high accuracy for genes like *2DL3* and *2DP1*, both at 100%, a notable decline is evident in *2DL5A*, with a percentage of 81.8%. Contrarily, T1K, having an overall accuracy of 83.09%, faces palpable challenges, especially with the *2DL5B* and *3DL2* genes, recording a below-average 75% and 47.2% accuracy respectively.

Finally, in the three-copy simulation, Geny projects an accuracy rate of 87.3%. Despite exhibiting superior performance in genes such as *2DL3* with a flawless 100% score, it encounters a daunting challenge with the *2DS3* gene, where accuracy plummets to 50%. Conversely, T1K struggles significantly in this simulation, having an overall accuracy of 67.6%. The results for genes such as *2DL1* and *2DL4* are notably low, at 57.1% and 64.1% respectively.

While the overarching findings clearly demonstrate the superiority of Geny over T1K, it is imperative to delve deeper into the nuanced differences and the potential implications of these results for practical applications in genomics and personalized medicine.

The consistent accuracy of Geny across the one and two-copy simulations exemplifies its robustness as shown in Figure 3.2. Such reliability can be especially vital

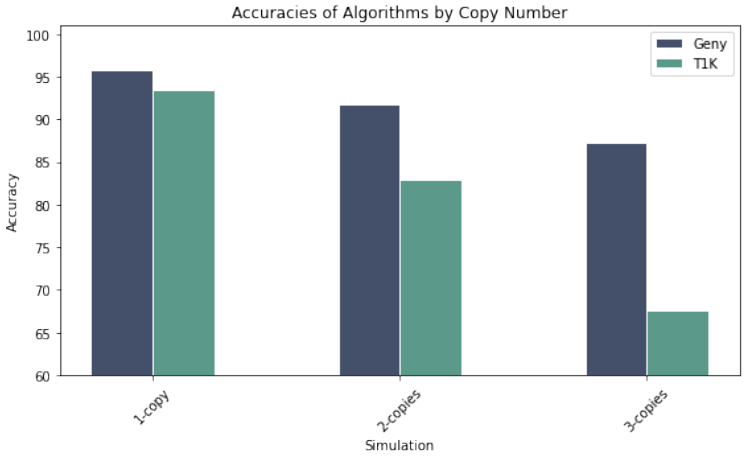

Figure 3.2: Comparison of performances in different copy number settings.

when genotyping samples with degraded or limited DNA, as accuracy in these circumstances directly correlates with the success of subsequent medical interventions or therapeutic approaches. Moreover, the impeccable 100% accuracy in genes such as *2DL3* and *2DP1* further underscores the algorithm's finesse, which is integral for genetic studies where even minor deviations can have significant downstream effects.

On the other hand, T1K's faltering performance, especially in the three-copy simulation, raises questions about its utility in more complex genetic landscapes. Genes such as 2DL1 and 2DL4, which displayed markedly reduced accuracy, are instrumental in immune response regulation. Therefore, inaccuracies in these segments might lead to misinterpretations, potentially influencing clinical decisions. Such challenges accentuate the need for continuous refinement of genotyping tools to ensure they meet the exacting standards required in contemporary biomedical research.

While Geny has shown superior performance in most cases, there are some cases in which T1K outperforms it in terms of accuracy. This situation can be noticed for *2DP1* in one-copy simulations, *2DL5A*, *2DS3*, and *2DS5* in two-copies simulations, and *2DL2* and *2DS3* in three-copies simulations. Delving deeper, we observe that in all of the mentioned scenarios, Geny predicts one of the alleles wrong among all alleles of that gene but T1K can call all of the present alleles correctly except *2DS3* in three-copies simulations where Geny calls 5/10 of the alleles and T1K calls 7/10 of them. The reason for the lower accuracy in this specific gene in three-copies scenarios for both Geny and T1K is that distinguishing between alleles *0020101* and *0010301* in more intricate settings is problematic and this specific allele gets repeated multiple times among simulations.

In regard to runtimes, Geny efficiently completes the filtering and final ILP solver stages within seconds for a single copy and approximately 10 minutes for multiple copies. Nevertheless, the most time-consuming aspect of our system is the remapping phase, which has polynomial-time complexity and spans from 10 minutes for a single copy to a maximum of one hour for three copies. Conversely, T1K requires up to 10 minutes to execute its comprehensive analysis.

In summary, through intricate simulations and scrutiny, Geny consistently outperforms T1K across multiple gene segments. Such findings lay the groundwork for enhancing the precision and reliability of KIR genotyping algorithms in future biomedical applications.

Chapter 4

Conclusion

KIR genotyping is a complex and precise process, vital for biomedical studies. Our deep dive into its techniques resulted in Geny, a new tool for KIR gene identification. In evaluations, Geny consistently outperformed another state-of-art tool, T1K, across various gene segments. Geny proved reliable in everything from basic to intricate genotyping tasks, highlighting its resilience to the inherent challenges of the process.

Geny's standout performance is not only of academic interest but also of practical importance. As we move towards tailored medical treatments, the accuracy of tools like Geny in identifying genes can shape the future of patient care. Specifically, Geny's precision in critical genes positions it as a promising tool for advancing the role of KIR genotyping in patient-focused medicine.

On the other hand, T1K had variable outcomes, reminding us of genotyping's complexities. Though it showed high accuracy in some situations, its inconsistency in specific simulations raised concerns. Key immune-related genes, such as 2DL1 and 2DL4, displayed this variability. This does not just spotlight areas for technical improvement, but it also highlights potential clinical implications and the need for careful interpretation.

As genomics advances, it is vital to have tools that progress alongside these advancements. Geny sets a benchmark, but it also underlines the constant need for innovation in this dynamic field.

In summary, our study showcases Geny's impressive abilities and underlines the continuous strive for excellence in KIR genotyping. Through our findings, we are mapping out the future of KIR genotyping and emphasizing its critical role in modern medicine and research.

Our method is not complete and can always be improved. For potential future

paths, there is considerable space for refining and enhancing the existing segments. One particularly promising path to explore is the detection of gene fusions. This area offers opportunities for significant research and advancements. Another important part that can be enhanced is remapping the reads to alleles for mutation detection purposes. This segment can be improved in implementations or replaced by only mapping reads to the reference genome in order to reduce the time complexity and achieve higher efficiency.

Bibliography

[1] International SNP Map Working Group Cold Spring Harbor Laboratories: Sachidanandam Ravi 1 Weissman David 1 Schmidt Steven C. 1 Kakol Jerzy M. 1 Stein Lincoln D. 1, National Center for Biotechnology Information: Marth Gabor 2 Sherry Steve 2, Sanger Centre: Mullikin James C. 3 Mortimore Beverley J. 3 Willey David L. 3 Hunt Sarah E. 3 Cole Charlotte G. 3 Coggill Penny C. 3 Rice Catherine M. 3 Ning Zemin 3 Rogers Jane 3 Bentley David R. drb@ sanger. ac. uk 3 m, and Washington University in St. Louis: Kwok Pui-Yan 4 Mardis Elaine R. 4 Yeh Raymond T. 4 Schultz Brian 4 Cook Lisa 4 Davenport Ruth 4 Dante Michael 4 Fulton Lucinda 4 Hillier LaDeana 4 Waterston Robert H. 4 McPherson John D. 4. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928–933, 2001.

[2] Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. Genbank. *Nucleic acids research*, 41(D1):D36–D42, 2012.

[3] Jeanette E Boudreau and Katharine C Hsu. Natural killer cell education and the response to infection and cancer therapy: stay tuned. *Trends in immunology*, 39(3):222–239, 2018.

[4] Jeffrey C Boyington and Peter D Sun. A structural perspective on mhc class i recognition by killer cell immunoglobulin-like receptors. *Molecular immunology*, 38(14):1007–1021, 2002.

[5] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[6] Jonathan Downing and Lloyd D'Orsogna. High-resolution human kir genotyping. *Immunogenetics*, 74(4):369–379, 2022.

[7] Peter Emtage, Paolo Vatta, Matthew Arterburn, Matthew W Muller, Emily Park, Bryan Boyle, Sophie Hazell, Renee Polizotto, Walter D Funk, and Y Tom Tang. Igfl: A secreted family with conserved cysteine residues and similarities to the igf superfamily. *Genomics*, 88(4):513–520, 2006.

[8] William E Evans and Mary V Relling. Pharmacogenomics: translating functional genomics into rational therapeutics. *science*, 286(5439):487–491, 1999.

[9] Taishan Hu, Nilesh Chitnis, Dimitri Monos, and Anh Dinh. Next-generation sequencing technologies: An overview. *Human Immunology*, 82(11):801–811, 2021.

[10] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[11] Seung-been Lee, Marsha M Wheeler, Karynne Patterson, Sean McGee, Rachel Dalton, Erica L Woodahl, Andrea Gaedigk, Kenneth E Thummel, and Deborah A Nickerson. Stargazer: a software tool for calling star alleles from next-generation sequencing data using cyp2d6 as a model. *Genetics in Medicine*, 21(2):361–372, 2019.

[12] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

[13] Na Li and Matthew Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–2233, 2003.

[14] Fredrik Mertens, Bertil Johansson, Thoas Fioretos, and Felix Mitelman. The emerging complexity of gene fusions in cancer. *Nature Reviews Cancer*, 15(6):371–381, 2015.

[15] Derek Middleton and Faviel Gonzelez. The extensive polymorphism of kir genes. *Immunology*, 129(1):8–19, 2010.

[16] Ryan E Mills, Christopher T Luttig, Christine E Larkins, Adam Beauchamp, Circe Tsui, W Stephen Pittard, and Scott E Devine. An initial map of insertion and deletion (indel) variation in the human genome. *Genome research*, 16(9):1182–1190, 2006.

[17] Paul J Norman, Jill A Hollenbach, Neda Nemat-Gorgani, Wesley M Marin, Steven J Norberg, Elham Ashouri, Jyothi Jayaraman, Emily E Wroblewski, John Trowsdale, Raja Rajalingam, et al. Defining kir and hla class i genotypes at highest resolution via high-throughput sequencing. *The American Journal of Human Genetics*, 99(2):375–391, 2016.

[18] Ibrahim Numanagić, Salem Malikić, Michael Ford, Xiang Qin, Lorraine Toji, Milan Radovich, Todd C Skaar, Victoria M Pratt, Bonnie Berger, Steve Scherer, et al. Allelic decomposition and exact genotyping of highly polymorphic and structurally variant genes. *Nature communications*, 9(1):828, 2018.

[19] Ibrahim Numanagić, Salem Malikić, Victoria M Pratt, Todd C Skaar, David A Flockhart, and S Cenk Sahinalp. Cypiripi: exact genotyping of cyp2d6 using high-throughput sequencing data. *Bioinformatics*, 31(12):i27–i34, 2015.

[20] Gurobi Optimization et al. Gurobi optimizer reference manual, 2020.

[21] Peter Parham. Immunogenetics of killer cell immunoglobulin-like receptors. *Molecular immunology*, 42(4):459–462, 2005.

[22] Peter Parham. Mhc class i molecules and kirs in human history, health and survival. *Nature reviews immunology*, 5(3):201–214, 2005.

[23] Kimberly Pelak, Anna C Need, Jacques Fellay, Kevin V Shianna, Sheng Feng, Thomas J Urban, Dongliang Ge, Andrea De Luca, Javier Martinez-Picado, Steven M Wolinsky, et al. Copy number variation of kir genes influences hiv-1 control. *PLoS biology*, 9(11):e1001208, 2011.

[24] David Roe and Rui Kuang. Accurate and efficient kir gene and haplotype inference from genome sequencing reads with novel k-mer signatures. *Frontiers in immunology*, 11:583013, 2020.

[25] Jonathan Sebat, B Lakshmi, Jennifer Troge, Joan Alexander, Janet Young, Par Lundin, Susanne Manér, Hillary Massa, Megan Walker, Maoyen Chi, et al. Large-

scale copy number polymorphism in the human genome. *Science*, 305(5683):525–528, 2004.

[26] Li Song, Gali Bai, X Shirley Liu, Bo Li, and Heng Li. Efficient and accurate kir and hla genotyping with massively parallel sequencing data. *Genome Research*, 2023.

[27] Paweł Stankiewicz and James R Lupski. Structural variation in the human genome and its role in disease. *Annual review of medicine*, 61:437–455, 2010.

[28] Michael R Stratton, Peter J Campbell, and P Andrew Futreal. The cancer genome. *Nature*, 458(7239):719–724, 2009.

[29] Peter H Sudmant, Jacob O Kitzman, Francesca Antonacci, Can Alkan, Maika Malig, Anya Tsalenko, Nick Sampas, Laurakay Bruhn, Jay Shendure, 1000 Genomes Project, et al. Diversity of human copy number variation and multicopy genes. *Science*, 330(6004):641–646, 2010.

[30] David Twesigomwe, Britt I Drögemöller, Galen EB Wright, Azra Siddiqui, Jorge da Rocha, Zané Lombard, and Scott Hazelhurst. Stellarpgx: a nextflow pipeline for calling star alleles in cytochrome p450 genes. *Clinical Pharmacology & Therapeutics*, 110(3):741–749, 2021.

[31] Greyson P Twist, Andrea Gaedigk, Neil A Miller, Emily G Farrow, Laurel K Willig, Darrell L Dinwiddie, Josh E Petrikin, Sarah E Soden, Suzanne Herd, Margaret Gibson, et al. Constellation: a tool for rapid, automated phenotype assignment of a highly polymorphic pharmacogene, cyp2d6, from whole-genome sequences. *NPJ genomic medicine*, 1(1):1–10, 2016.

[32] Markus Uhrberg. The kir gene family: Life in the fast lane of evolution. *European Journal of Immunology*, 35(1):10–15, 2005.

[33] Damjan Vukcevic, James A Traherne, Sigrid Næss, Eva Ellinghaus, Yoichiro Kamatani, Alexander Dilthey, Mark Lathrop, Tom H Karlsen, Andre Franke, Miriam Moffatt, et al. Imputation of kir types from snp variation data. *The American Journal of Human Genetics*, 97(4):593–607, 2015.

[34] Hagen Wende, Marco Colonna, Andreas Ziegler, and Armin Volz. Organization of the leukocyte receptor cluster (lrc) on human chromosome 19q13. 4. *Mammalian Genome*, 10:154–160, 1999.

[35] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.