國立成功大學醫學資訊研究所

碩士論文

基於評估讀取索引以達成提高Illumina測序資料中基因組變異辨認的靈敏度

Evaluating a read index based approach to improve the sensitivity of Illumina sequencing based genome variant calling

研究生：陳威穎　　Student: Wei-Ying Chen

指導老師：賀保羅　　Advisor: Paul Horton

National Cheng Kung University,

Tainan, Taiwan, R.O.C.

Thesis for Master of Science Degree

July, 2022

中華民國１１１年７月

基於評估讀取索引以達成提高Illumina測序資料中基因組變異辨認的靈敏度

陳威穎* 賀保羅†

國立成功大學醫學資訊研究所

摘要

次世代定序技術(NGS)是精準醫療不可或缺的一種技術，但NGS還需搭配後續的資料分析流程才能真正運用到臨床或研究上。其中NGS數據中變異辨認幾乎是所有下游分析和解釋過程都依賴的關鍵步驟，但是因為變異辨認過程中固有的不確定性，使得準確的基因體變異偵測仍然有許多問題及挑戰。因此我們先前提出過一個變異偵測評估工具Explicit Alternative Genome Likelihood Evaluator(EAGLE)，對於每個被偵測到的潛在變異位點，通過計算邊際事後機率，來評估測序數據與推定變異所隱含的替代基因組序列的匹配程度。而根據先前研究的結果表明，其有效提高變異評估的準確性，相當適合用來進一步分析及處理各種變異偵測工具所產生的結果。

由於我們比對的人類基因組參考序列通常為線性的，僅包含單個序列，因此無法捕獲人類基因組的多樣性。但是讀序可能含有個體變異，所以在映射的過程中，因為讀序與參考序列的高度不同，造成讀序可能被映射到錯誤的位置或無法映射，這種現象我們稱之為參考偏差。錯誤的讀序映射會導致假陰性

*學生

†指導教授

或假陽性的變異辨認，從而影響到我們識別基因體變異的準確性。而為了解決這個問題，在先前的研究中，我們新增假設序列及讀取索引結構，改進基因組參考序列，使其能夠包含基因組的多樣性，而研究結果表明，其能有效的降低參考序列帶來的影響。

然而先前研究著重在於評估新方法對於降低參考偏差帶來的影響，並無對於實際應用情況進行驗證及評估。所以在本次研究中，我們將對NA12878的真實資料分別使用外顯子測序集及全基因組測序集進行效能的評估及探討，並使用常見的變異偵測工具來協助實驗的進行。

對於實驗結果而言，在我們使用精準度與召回率曲線來評估新版本EAGLE的表現時，發現在同樣的召回率中，新版本EAGLE的精準度有下降的問題發生。因此，我們擷取EAGLE給予邊際機率較高的變異，並檢查它們的映射情形，期望找出問題發生的原因。然而為了確保並非新版本的EAGLE有評估錯誤的可能性，我們也對於可能發生錯誤的情形進行後續的檢驗。

總言之，根據這次的研究結果，我們推測基準資料可能有缺失某部分變異的情況發生，並且經由後續的驗證，可以大致排除新版本EAGLE發生評估錯誤的可能性。因此，可以說明EAGLE具備發現基準資料缺失的能力。由於我們僅對於研究結果影響較大的小部分案例，進行後續的檢驗，所以此推測仍需在日後用更嚴謹的方式進行驗證與解釋。

關鍵詞: 次世代定序、變異點偵測、讀取索引

Evaluating a read index based approach to improve the sensitivity of Illumina sequencing based genome variant calling

Wei-Ying Chen* Paul Horton†

Institute of Medical Informatics, National Cheng Kung University

Abstract

Next-Generation Sequencing (NGS) is an indispensable technology for Precision Medicine, but NGS needs to be paired with a subsequent data analysis process before it can be truly applied to clinical or research applications.Variant identification in NGS data is a key step that almost all downstream analysis and interpretation processes rely on, but the inherent uncertainty in the variant identification process makes accurate genomic variant detection still very problematic and challenging. Therefore, we previously proposed a variant detection tool, Explicit Alternative Genome Likelihood Evaluator (EAGLE), to evaluate the match between sequencing data and the alternative genome sequence implied by the putative variant by calculating the marginal posterior probability for each detected potential variant sites. Based on the results of previous studies, it has been shown to be effective in improving the accuracy of variant assessment and is suitable for further analysis and processing of results generated by various variant detection tools.

Since the human genome reference sequences we compare are usually linear and contain only single sequences, the diversity of the human genome cannot be captured. However, read sequences may contain individual variation, so during

*Student
†Advisor

the mapping process, read sequences may be mapped to the wrong position or fail to be mapped because of the difference in height between read sequences and reference sequences, a phenomenon we called reference bias. Incorrect read sequence mapping can lead to false-negative or false-positive variant identification, which affects the accuracy of identifying genomic variants. To solve this problem, we added new Hypothetical Sequences and read index structures to improve the genomic reference sequences to include genomic diversity, and the results showed that they are effective in reducing the impact of reference sequences.

Due to the previous study focused on evaluating the impact of the new method on reducing reference bias, we did not verify and assess the actual application. Therefore, in this study, we will evaluate and investigate the efficacy of NA12878 using exome sequencing set and whole genome sequencing set for real data, and use common variant detection tools to assist the experiment.

To the experimental results, when we evaluated the performance of the new version of EAGLE using the precision and recall curves, we found that the precision of the new version of EAGLE had a degradation problem occurring on the same recall level. Therefore, we extracted the variants of EAGLE that gave a higher marginal probibility and examined their mapping situation in order to find out the cause of the problem. However, in order to ensure that not new versions of EAGLE have the possibility of evaluation errors, we also perform a follow-up check on the cases where errors may occur.

In conclusion, based on the results of this study, it is assumed that there may be some missing variation in the benchmark data, and the possibility of evaluation errors in the new version of EAGLE can be largely excluded by subsequent validation. Therefore, it can be stated that EAGLE has the ability to detect missing benchmark data. Since we have only conducted follow-up tests on a small number of cases where the results of the study had a significant impact, this speculation needs to be verified and explained in a more rigorous manner in the future.

Keywords: NGS, variant calling, read-index

誌謝

本篇論文能順利完成，首先我想感謝我的指導老師賀保羅教授，很榮幸能成為老師的學生，感謝老師在我攻讀碩士的這兩年間，不厭其煩地指導我，感謝你總在我緊張時給予鼓勵，在我需要時給予幫助，並且讓我了解進行研究該秉持的態度。也感謝口試委員劉宗霖教授及王士豪教授在生物資訊與論文的撰寫上給予我許多建議，令我受益良多。

在實驗室的這兩年，特別感謝的是同屆的宥霖、力元、恆霖、芊瑀，謝謝你們讓我在成大的日子並不孤單，從資料探勘到巨量生物資料分析，從第一次吃冰仔角到最後熬夜趕論文，從咆嘯深淵到召喚峽谷，從逐鹿到涮乃葉，從保齡球到滑輪場，從健身房到籃球場，從傷心難過到開懷大笑，一起經歷過的每件事，多到數不完的各種回憶，這些過程都有你們的陪伴。

接下來想感謝實驗室的學長姐們，感謝文弘分享許多研究上的經驗、想法與資源，也給予我許多幫助。感謝綾娟總是傾聽我的煩惱，給予我許多生活與課業上的建議。感謝喚凱在許多方面不吝分享他的看法與他的美食指南霸主小妞炒飯。感謝國勳的湯湯食與他帶給我的歡笑，有他的碩一生活一點都不無趣，感謝士杰總是分享很酷很有趣的知識給大家，也不吝給予我各種幫助。

也想謝謝實驗室的學弟妹昱伶、益宗、筱萱、海墨、怡靜、中柏，謝謝你們讓我的碩二生活變得熱鬧許多，一起修習的軟體設計，一起聊過的大小事，一起度過的這一年，還有你們給予我的幫助，一定會讓我在日後憶起我的碩士生活有更多的笑容與懷念，這篇論文能順利完成感謝生醫暨語言資訊實驗室大家的協助。

還想特別感謝容尹，在我對於論文的撰寫上遇到瓶頸與困難時，給予我很大的幫助與鼓勵，謝謝你無條件的付出與包容，讓我能順利完成此篇論文。最後也感謝我的家人無條件的支持我完成學業，在我灰心時給予最暖心的鼓勵，讓我能在求學之路上無後顧之憂，全心全意的精進向上，成就更好的自己。

最後，將此論文獻給我摯愛的家人，並再次感謝這一路上幫助過我的

所有人，誠摯的祝福你們身體健康，一路順風。

陳威穎 謹誌於

國立成功大學

中華民國111年7月

CONTENTS

List of Tables

List of Figures

Nomenclature

General

BAM	Binary Alignment Map
DNA	deoxyribonucleic acid
EAGLE	Explicit Alternative Genome Likelihood Evaluator, the predecessor of this research
GATK	Genome Analysis Toolkit
GIAB	Genome-In-A-Bottle
indel	Insertion and Deletion
IPG	Illumina Platinum Genomes
NGS	Next generation sequencing
PR	Precision and Recall
SAM	Sequence Alignment Map
SNP	Single Nucleotide Polymorphism
VCF	Variant Call Format
WGS	Whole Genome Sequencing

Chapter 1

Introduction

1.1 NGS based variant calling

In recent years, Next Generation Sequencing (NGS) has become an integral part of Genomics analysis. This technology enables the sequencing of the entire human genome in a relatively short period of time, allowing for more advanced bioanalysis. Despite the breakthroughs achieved with NGS results, there are still some problems in the sequencing pipeline. Since sequencers cannot generate reads as long as the entire human genome without error, sequencing results are often presented as multiple shorter reads, making data processing and analysis particularly important. The identification of variants in NGS data is a critical step that almost all downstream analysis and interpretation processes rely on. Therefore, many NGS studies rely on accurate variants detection, in order to gain a better understanding of the genomic system and to apply it to clinical practice and research, especially in the diagnosis of genetic diseases and diseases related to genetic diseases [1, 2]. With the development of NGS technology, variant calling software such as Genome Analysis Toolkit (GATK) [3], SAMtools [4], Platypus [5], etc. have been rapidly developed in the past few years.

Despite the efforts to detect variants accurately, the detection of variants in NGS data is prone to errors due to various factors such as sequencing errors,

inaccurate alignment and low read coverage. This also leads to the results of variant calling tools often have a low precision-recall.

To address this Tony Kuo et al. previously proposed a variant detection evaluation tool EAGLE: Explicit Alternative Genome Likelihood Evaluator [6]. It is applied after the variant calling phase in the NGS data analysis pipeline to provide the confidence level of each entry in the list of candidate genome variants. It was demonstrated that this post-processing method improves the precision at an acceptable recall rate compared to the initial called variants results. However that EAGLE method relies on standard read mapping (to the reference genome) to decide which reads to consider when computing likelihood, and therefore in principle suffers from reference bias.

1.2 Research Objective

Our lab at NCKU has investigated extending EAGLE to solve the problem that human reference genome sequence cannot capture the diversity of human genome, we added Hypothetical Sequence and read-index to improve the reference genome sequence to include the genomic diversity [7, 8]. Based on simulation and *ad hoc* analysis, those studies showed that the read-index may effectively reduce bias towards the reference sequence. Here we extend that work by benchmarking the variant calling performance of the read-index extended version of EAGLE real data.

1.3 Thesis Overview

The organizational structure of this paper is as follows: Chapter one presents an overview of the previous study, the motivation and the objectives of the study. Chapter two we will introduce the file formats commonly used to store data in NGS data analysis and provide a more detailed description of the background of the previous study on Data format, EAGLE, Reference Bias, Hypothetical

Sequence and read-index. Chapter three we briefly introduce the Variant Callers to be used in this experiment, such as GATK HaplotypeCaller, BCFtools and Platypus. Chapter four we introduce the datasets we intend to use, and the workflow of the experiment. Chapter five we show the results of the experiments and discuss the findings in the experimental results. Chapter six we provide conclusions and future work.

Chapter 2

Background

2.1 Data format

During NGS data analysis, there are often many data storage formats, which are briefly introduced here for the subsequent explanation.

2.1.1 Sequence Alignment Map and Binary Alignment Map

Sequence Alignment Map (SAM) is a text-based format originally developed by Heng Li et al. [4]. For storing biological sequences compared to reference sequences. It is now widely used for storing data that is generated by next generation sequencing technologies.

The SAM format consists of two sections, header and aligned section. If there is a header section, it must precede the aligned section and be preceded by the ”@” symbol to distinguish the aligned section. The aligned section has 11 required fields (TAB-delimited) and several optional fields(Figure 2.1).

The SAM file is a file stored in SAM format, while Binary Alignment Map file (BAM file) is a binary file corresponding to a SAM file, which stores the same data in a compressed binary representation. The BAM file is a compact and indexable representation of nucleotide sequence alignments, which can effectively compress the storage space and quickly query specific locations.

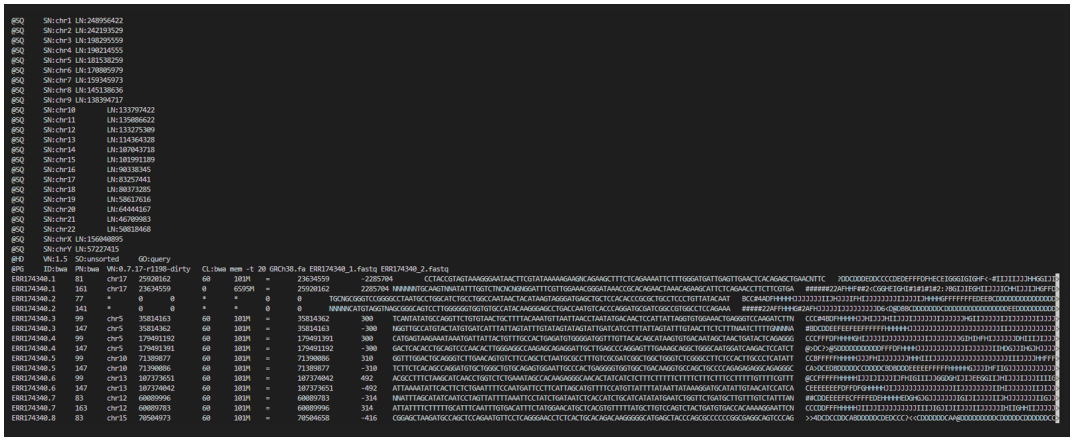

Figure 2.1: Example for SAM format.

2.1.2 Variant Call Format

Variant Call Format (VCF) specifies the format of the text file used in bioinformatics for storing gene sequence variants [9]. A VCF file is a file stored in VCF, which consists of two parts, the header and the VCF body. As shown in Figure 2.2, the header is the beginning of the file and provides the metadata describing the body of the file; the header starts with ”#” and the special keyword is indicated by ##.

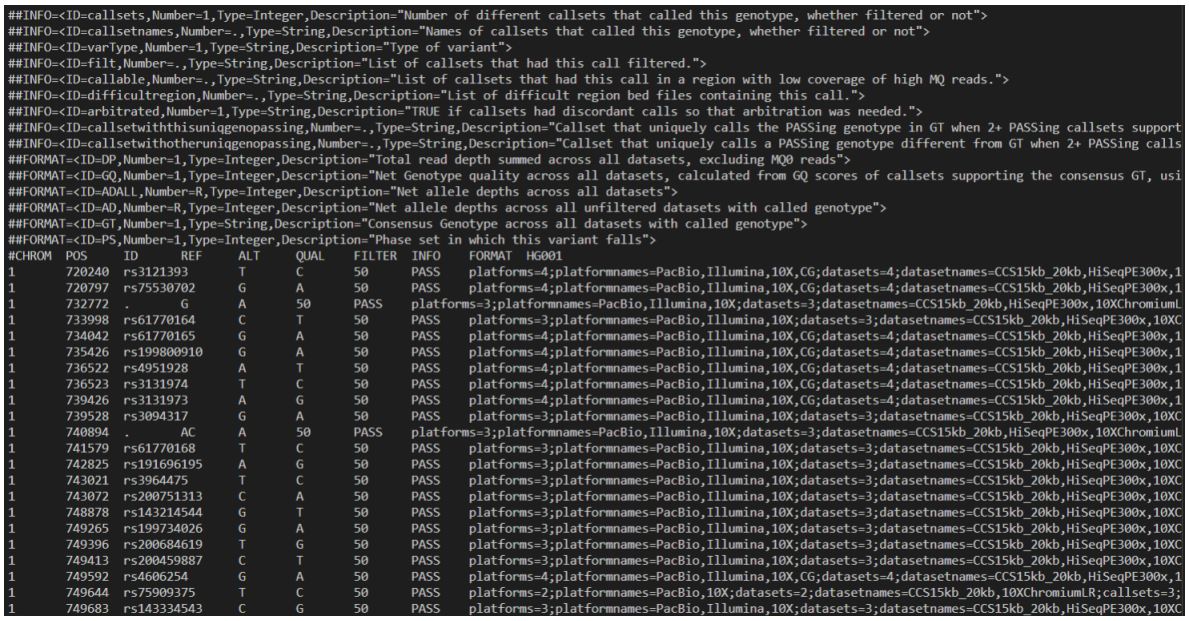

Figure 2.2: Example for VCF format.

2.1.3 FASTQ and FASTA format

The FASTA format is a text-based format for representing nucleotide sequences or amino acid (protein) sequences, where nucleotides or amino acids are represented as single-letter codes [10]. The format originated from the FASTA software package and has now become a very common standard in the field of bioinformatics. In the FASTA file(Figure 2.3), each sequence is stored with a ”>” in the first line, followed by a summary description of the sequence, and then by the sequence information.

```
>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken
MADQLTEEQIAEFKEAFSLFDKDGDGTITTKELGTVMRSLGQNPTEAELQDMINEVDADGNGTID
FPEFLTMMARKMKDTDSEEEIREAFRVFDKDGNGYISAAELRHVMTNLGEKLTDEEVDEMIREA
DIDGDGQVNYEEFVQMMTAK*
```

Figure 2.3: Example for FASTA format.

The FASTQ format is a text-based format for storing information about biological sequences and their quality scores, where both the sequence and the quality scores are represented by a single ASCII character [11]. Since it combines the sequence and the associated quality score of each base, FASTQ has become a common file format for sequencing read data.

```
@ERR174340.1 HSQ1004_144:3:1101:1071:2106/1
GAANGTTCAGCTCTGTGAGTTCAACTCAATCATCCCAAAGAATTTTCTGAGAAAGCTTCTGNCTTCTTTTTATACGAAGTTATTCCCTTTACTACGGTAGG
+
@@C#4=DDFHHHHJJGIGIIHDHIIHIJJJJJJJJJJIJIIJJIIJIJIGGHHJJJIIJII#-<FHGIGIGGGIECEHFDFFFEDEDCCCCDDEDDDCDD?
@ERR174340.2 HSQ1004_144:3:1101:1133:2111/1
TGCNGCGGGTCCGGGGCCTAATGCCTGGCATCTGCCTGGCCAATAACTACATAAGTAGGGATGAGCTGCTCCACACCCGCGCTGCCTCCCTGTTATACAAT
+
BCC#4ADFHHHHHJJJJJJJIIJHJJJIFHIJJJJJJJJIJJJJIJHHHHGFFFFFFFEDEEBCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEDEDC
@ERR174340.3 HSQ1004_144:3:1101:1036:2115/1
TCANTATATGCCAGGTTCTGTGTAACTGCTTTTACAAATGTTAATTAACCTAATATGACAACTCCATTATTAGGTGTGGAAACTGAGGGTCCAAGATCTTN
+
CCC#4BDFHHHHHJJHIJJJHIIJJJIJJJJJJJIJJJJJJHGIIJJJJJJIJIJJJJJJJIJJJJJJJJJJJJJJIJCCGGGHHHHHHFFFFACAECDDDDEC#
```

Figure 2.4: Example for FASTQ format.

In the FASTQ file(Figure 2.4), a sequence usually consists of four lines. The first line starts with ”@”, followed by the identifier of the sequence and the description information. The second line is the sequence information, consisting of A, T, C, G, N, etc. The third line starts with a ”+”, after which the sequence identifier and description information can be added again. The fourth line is the

quality score information, which corresponds to the sequence in the second line, and must be the same length as the second line.

2.2 EAGLE

This section introduces the function and concept of EAGLE, a variant calling evaluator based on an explicit probability model, which was previously studied.

As shown in Figure 2.5, the main function of EAGLE is to take the three files provided by the user as input:the reference genome in FASTA format file, the called variants in a variant call format file(VCF file) and the aligned sequencing reads in a Binary Alignment Map file(BAM file). Then return the likelihood score of each candidate variants.

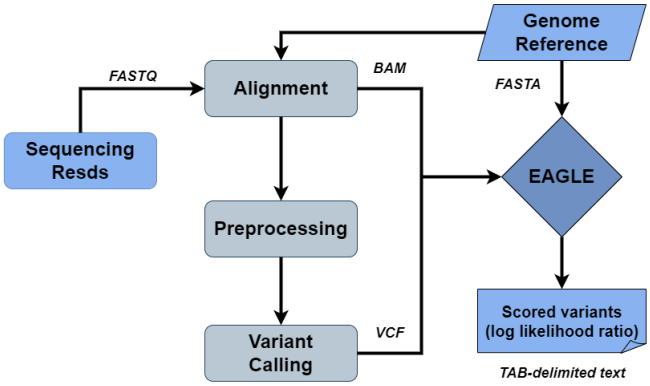

Figure 2.5: A high level overview of the EAGLE workflow. (Figure reproduced from Kuo et al. [6])

The main concept is that, during variant calling, various uncertainties such as sequencing errors, inaccurate alignments and low read coverage can cause errors, which can affect downstream analysis. Therefore, EAGLE includes candidate variants in the explicit hypothesis of individual genomes, and then calculates the odds of the sequencing reads under each hypothesis. To explicitly assess the degree of match between given individual sequencing data and the list of candidate variants, and then to deal with these possible errors and uncertainties.

In conclusion, the more uncertain the quality of the data, the less credible the analysis will be. Therefore, in principle, the likelihood score calculated by EAGLE can be used to rank the reliability of putative variants and enhance the credibility of downstream analysis.

2.3 Reference Bias

Sequencing data analysis often begins with aligning reads to a reference genome. Conventionally, read alignment is performed using a linear reference genome.Because the linear reference genome contains only a single sequence, it does not capture the genomic diversity of a population. Because of the difference between the reference genome and the individual being sequenced, alignment tools map reads to the wrong location or fail to match the occurrence(Figure 2.6). This problem can lead to false negative or false positive variant calls.

There have been many studies discussing how to mitigate the impact of reference bias, such as genome graph aligners, or using high-coverage NGS data, to improve the reference genome to include the diversity of the human genome [12, 13].

Figure 2.6: Reference bias. Only a small portion of the read containing inserts (dark blue) are mapped to the correct positions during the alignment to the reference genome [14].

2.4 Hypothetical Sequence

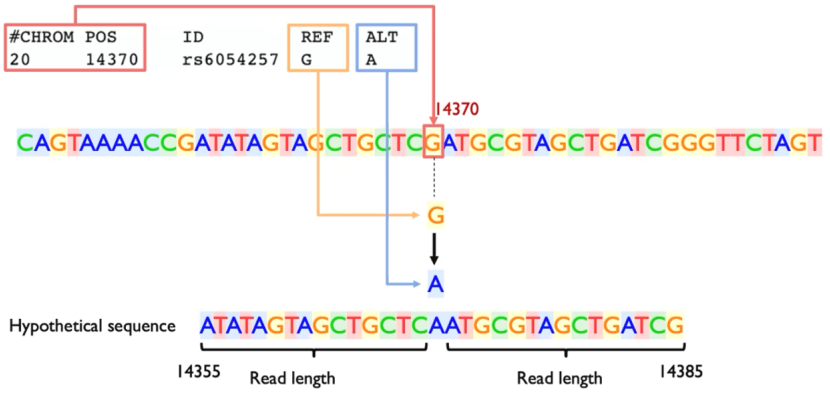

Figure 2.7: A Hypothetical Sequence constructed from single nucleotide polymorphisms. (Figure reproduced from Su. [8])

In our previous study, the reference genome was modified to construct the Hypothetical Sequence to reduce the effect of reference bias.

We construct a Hypothetical Sequence by replacing the pre-variant sequence with the after-variant sequence according to the variant position, and retrieve the part of the reference genome before and after the position with a read length.

Since we simulate the occurrence of mutations in the reference sequence by combining the variant to generate Hypothetical Sequence, the reference genome captures the genomic diversity of the entire population and find the reads that match our Hypothetical Sequence in the FASTQ file to reduce the impact of the reference bias.

2.5 read-index

If we search for reads matching our Hypothetical Sequence in FASTQ file by traditional method, we need to compare all the reads with Hypothetical Sequence. In our case, for each variant, we will create a Hypothetical Sequence to search for matching reads, which will take a lot of our time.

Therefore, in our previous study, we investigated the variation of overall time

complexity when indexing the Hypothetical Sequence and reads respectively in order to improve the search efficiency. Finally, we decided to index the reads to generate the read-index, which helps us to find the read matching Hypothetical Sequence quickly.

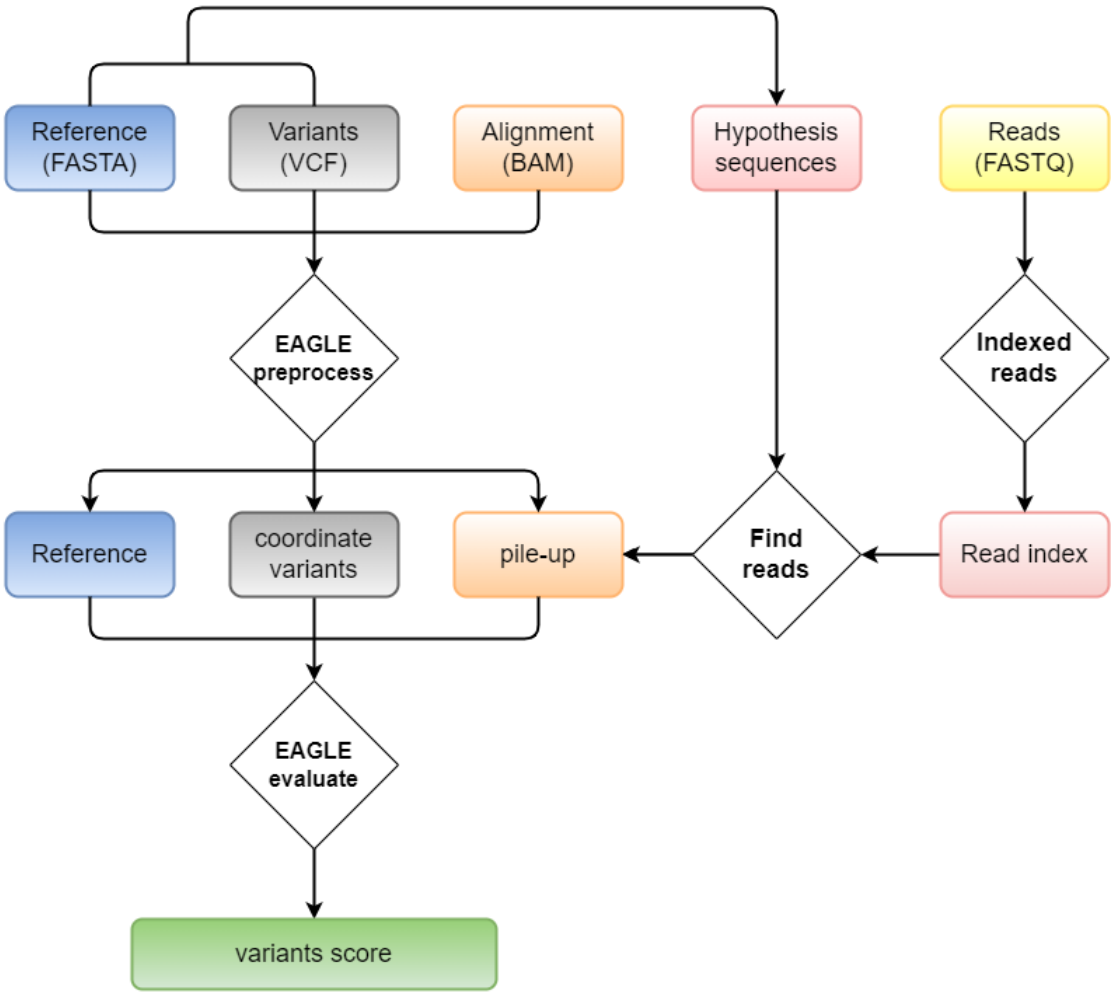

Figure 2.8: The core workflow of EAGLE after adding read-index to reduce reference bias.

Chapter 3

Related Works

This chapter will briefly introduce the variant callers to be used in this study, first introducing GATK HaplotypeCaller [15], then BCFtools [16], and finally Platypus [5]. Each variant caller will input a BAM file for variant calling, and output a VCF file as the result.

3.1 GATK HaplotypeCaller

GATK HaplotypeCaller is able to call SNPs and indels by performing local de novo assembly on haplotypes of active regions. In other words, whenever it encounters a region that is likely to be mutated, GATK HaplotypeCaller discards the existing mapping information and completely reassembles the read of the region. This allows GATK HaplotypeCaller to be more accurate when calling regions that were previously difficult to call, for example, when they contain different types of variants very close to each other. This also allows GATK HaplotypeCaller to perform better than location-based callers in calling indels.

The following is the workflow of GATK HaplotypeCaller(Figure 3.1):

1. Identify active regions based on variation evidence (mapping information) and determine which regions the program needs to operate on.
2. For each active region, the program builds a De Bruijn-like graph to reassemble active regions and determine which haplotypes are likely to be present in the

data. The Smith-Waterman algorithm is then used to reassemble each haplotype with the reference haplotype to identify the potential variation sites.

3. Calculate the likelihoods of the haplotypes based on the given read data. i.e. for each active region, the program uses the PairHMM algorithm to align each read and each haplotype to obtain the allele likelihoods for each potential variant site.

4. For each potential variant site, the program applies Bayes' rule and calculates the likelihood of each genotype for each sample using the likelihood of the allele for the given read data, and then assigns the most likely outcome to that sample.

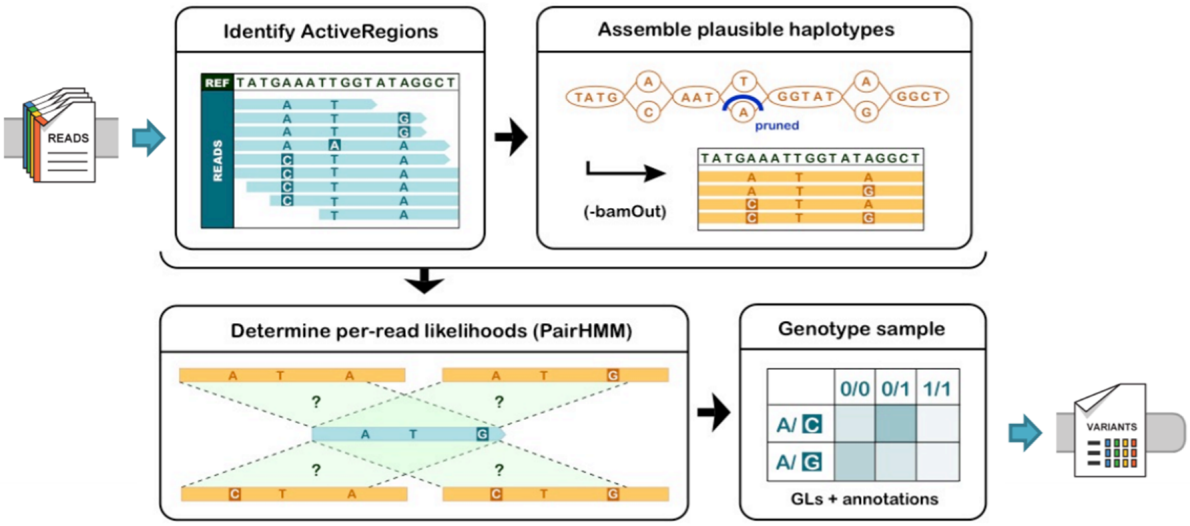

Figure 3.1: The workflow of GATK HaplotypeCaller. (Figure reproduced from Poplin, Ryan, et al. [15])

3.2 BCFtools

BCFtools generates candidate variant directly from the results of independent mapping of each read to the reference genome, and simulates sequencing errors and variant calling by using Bayesian inference [17, 18].

BCFtools performs variant calling in two steps. The first "mpileup" part generates genotype likelihoods at each genomic position with coverage. The second "call" part makes the actual calls.

In previous versions, BCFtools was usually combined with SAMtools' mpileup command for variant calling. Since both SAMtools and BCFtools are updated very quickly, if you install the wrong version of the tool, it will easily cause errors. Therefore, in order to reduce the problems caused by inappropriate versions, the BCFtools development team added the mpileup feature to BCFtools.

3.3 Platypus

As shown in Figure 3.2, Platypus generates candidate variants by using local de novo assembly followed by local rearrangement and probabilistic haplotype estimation [5].

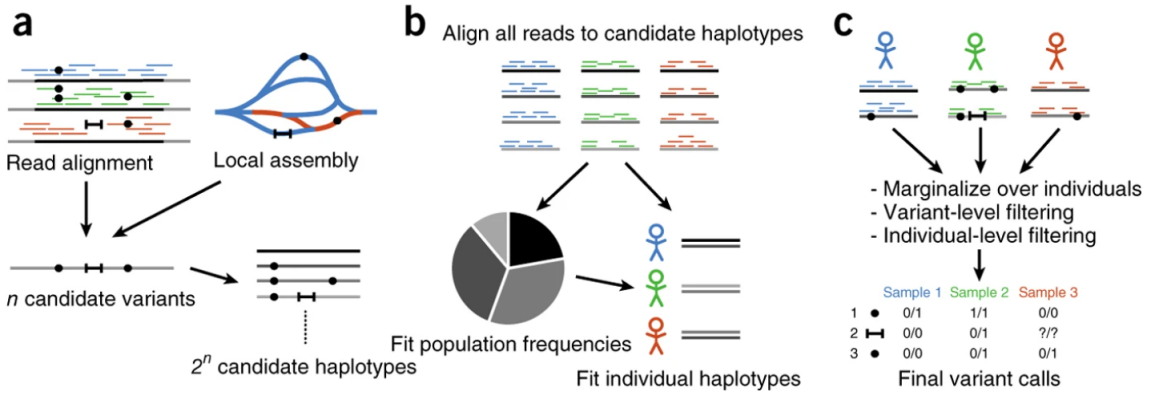

Figure 3.2: The pipeline of the Platypus algorithm contains three stages. (Figure reproduced from Poplin, Rimmer, Andy, et al. [15])

First, Platypus generates a set of candidate variants according to the algorithm. Three sources of candidate variants can be considered: variants directly supported by the read alignment, variants identified by the assembly, and variants from external sources. For n variants, this results in 2^n haplotypes (disregarding excluded combinations due to overlap).

Then, the likelihood of haplotype was calculated, that is, given the probability of the haplotype read: $p(r|h)$, where r and h denote read and haplotype, respectively. After calculating $p(r|h)$ for all combinations of reads and haplotypes, the Expectation-Maximization (EM) algorithm was used to estimate the

probability of each haplotype under the diploid genotype model.

$$L(R|\{h_i, f_i\}) = \prod_{\text{sample } s} \sum_{\text{haplotypes } i,j} f_i f_j \prod_{\text{reads } r \in R_s} (\frac{1}{2}p(r|h_i) + \frac{1}{2}p(r|h_j)) \quad (3.1)$$

Here, f_i denotes the frequency of haplotype h_i in the population; a is the number of alleles considered, R and R_s denote the set of all reads, and reads from sample s respectively, and the sum over haplotypes extends over all ordered pairs (i, j), i.e. genotypes.

Finally, we break up called haplotypes into their constituent variants and calculate variant-specific genotype calls and likelihoods by marginalizing over the other variants in the region.

$$P[v|R] = \frac{P(v)L(R|\{h_i, f_i\}_{i=1...a})}{P(v)L(R|\{h_i, f_i\}_{i=1...a}) + (1 - P(v))L(R|\{h_i, \frac{f_i}{1-F_v}\}_{i \in I_v})} \quad (3.2)$$

where $P(v)$ is the prior probability of observing variant v. The likelihood of reads given haplotypes and their frequencies is computed as

$$L(R|\{h_i, f_i\}_{i=1...a}) = \prod_{sample\ s} \sum_{haplotypes\ i,j} f_i f_j \prod_{reads\ r \in R_s} (\frac{1}{2}p(r|h_i) + \frac{1}{2}p(r|h_j)) \quad (3.3)$$

Chapter 4

Experimental Workflow

4.1 Dataset

In this study, in order to test the performance of the new version of EAGLE on real sequencing data, we use an exome sequencing dataset (Garvan HG001) from Genome-In-A-Bottle (GIAB) [19] and a 200× whole genome sequencing dataset from Illumina Platinum Genomes (IPG) to NA12878 (cell line of an individual from a CEPH pedigree) real data were tested [20].

From / Use	GIAB	IPG
Reference	GRCh37	GRCh38
Read	Exome Data	200xWGS Data
System	HiSeq2500	HiSeq2000

Table 4.1: The Dataset of Experiment.

The benchmark from IPG is a high confidence callset constructed using variant callers such as FreeBayes, Platypus, and GATK to construct the GRCh38 human gene reference sequence(Figure 4.1). The benchmark from GIAB is a high confidence callset for the GRCh37 human genome reference sequence using FreeBayes, SAMtools, and GATK.

Since the performance was evaluated using the above data in the previous study, we would like to use the same dataset to present and discuss the results in

this study.But the benchmarks were constructed from variant calls made by the same tools we are comparing against, there may be some bias in the following results.

Aligner	Variant Caller	Sequence Data	SNVs	Indels
bwa-mem	GATK3	ILMN PCR-free (2x100bp)	Yes	Yes
bwa-mem	FreeBayes	ILMN PCR-free (2x100bp)	Yes	Yes
bwa-mem	Platypus	ILMN PCR-free (2x100bp)	Yes	Yes
Isaac	Strelka	ILMN PCR-free (2x100bp)	Yes	Yes
NA	Cortex	ILMN PCR-free (2x100bp)	Yes	Yes
CGTools 2.0	CGTools 2.0	CGI (2x50bp)	Yes	No

Figure 4.1: The IPG sequence data and variant calling pipelines. (Figure reproduced from Eberle, MA et al. [20])

4.2 Data Preprocessing

In the previous section we introduced two sets of datasets to be used, in this section we will briefly describe the flow of data preprocessing.

First, we refer to the GATK 'best practices' preprocessing steps to practice our process [21], the process of constructing a BAM file is as follows:

1. Create an index of the reference genome for subsequent usage.
2. Use BWA-MEM to perform Read mapping [22].
3. Convert the obtained SAM file to BAM file to save storage space.
4. Sort the BAM file from the smallest to the largest according to the chromosome name and coordinate order.
5. Do Label read groups with Picard AddOrReplaceReadGroups on BAM file [23].
6. For BAM file use Picard MarkDuplicates identifies duplicate reads and remove it [24].

In the second step, we use the generated BAM format alignment data to make call variants using the variant caller introduced in Chapter 3.

Finally, we canonicalize the complex variants using `vt normalize` for each variant callset we obtained [25].

4.3 Plotting PR Curves

According to the EAGLE mentioned in Subsection 2.2, we will input the variant callset obtained in Subsection 4.2 in the format of a vcf file. This is used to obtain the marginal posterior probabilities for each candidate variants. Since the variant evaluator EAGLE evaluates the confidence level of the variants in the input file, we want to focus on the correct determination of the positive samples. However, even if the number of negative samples in the data set is much larger than the number of positive samples, the Precision-Recall Curves (PR Curve) is still a valid reference indicator. Therefore, we rank the marginal posterior probabilities of the obtained results and the quality scores of each variant caller separately. Finally, a PR Curve was obtained to evaluate the performance of our new version of EAGLE.

Chapter 5

Results and Discussion

5.1 Experimental Results

In this section, the dataset and data preprocessing process introduced in the previous section will be applied to plot the PR Curve to present the changes between the new version of EAGLE and the previous version. Due to the large amount of biological data, the variation data of chromosome 22 will be extracted for the presentation of the results in this study.

In addition to showing the performance of Insertion and Deletion (indel) and Single Nucleotide Polymorphism (SNP) separately (Figure 5.1(a,c) and Figure 5.1(b,d)), we have also plotted two comparative graphs to facilitate the discussion. The first one is to plot EAGLE with read-index and the three variant callers using marginal posterior probabilities and quality scores (Figure 5.1(a,b)). The second one is to plot the best performing curves of EAGLE, EAGLE with read-index and variant caller (Figure 5.1(c,d)).

For the above results, we can notice that the new version of EAGLE has a significantly higher precision than the variant caller in ranking putative variants on whole genome sequencing data(Figure 5.1). However, contrary to the expectation, compared with the previous version of EAGLE, there was a decrease in precision at the same recall level in both indel and SNP. There was no significant change in the exome sequencing data.

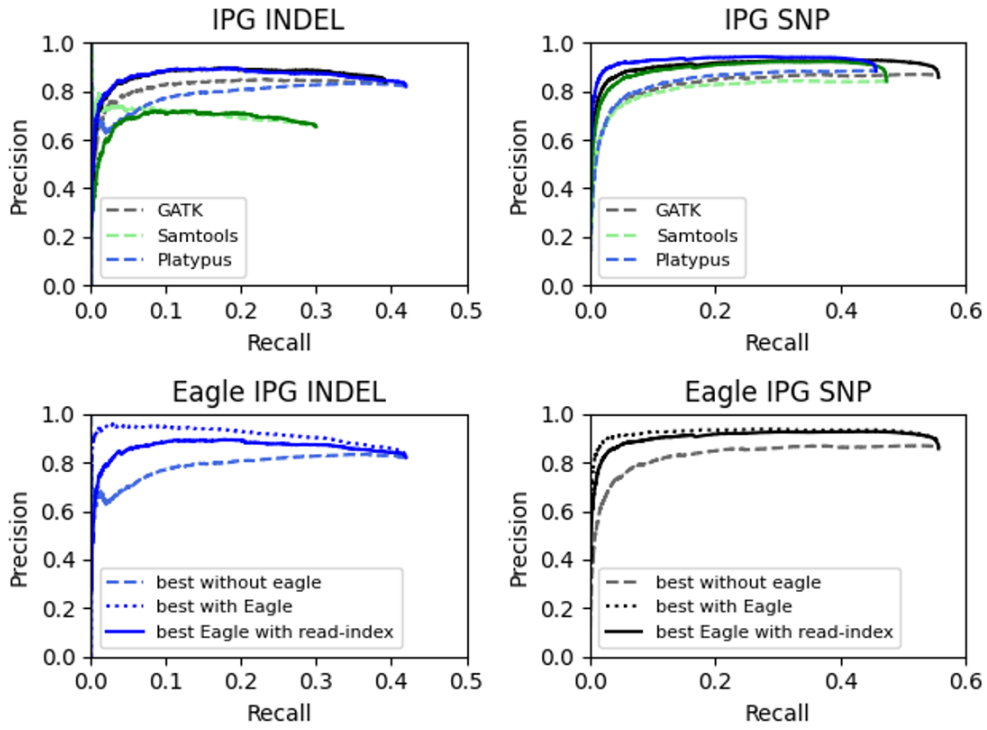

Figure 5.1: Precision and recall for NA12878 for whole genome sequencing data in Illumina Platinum Genome benchmark.

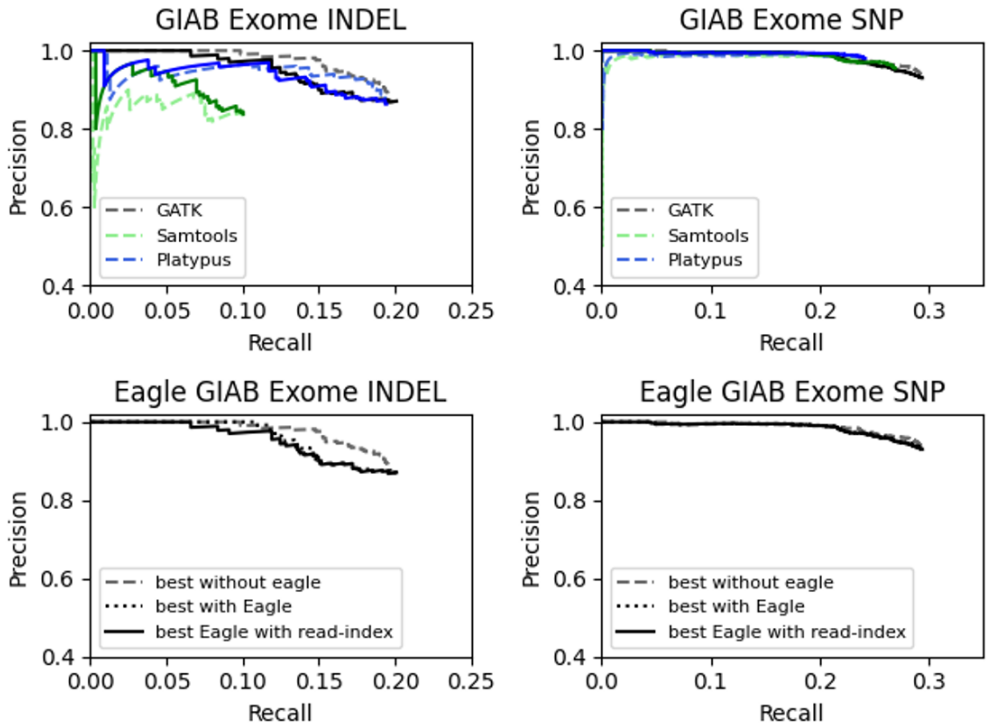

Figure 5.2: Precision and recall for the NA12878 Genome-In-A-Bottle benchmark.

Figure 5.3: Example for pile-up.

5.2 Checking the pile-up

As mentioned above, we got the opposite result than expected in the new version of EAGLE. Therefore, in order to find the reason for the problem, we examined the pile-up before and after using read-index, expecting to get an explanation for the lower-than-expected results.

We ranked EAGLE with read-index (EAGLE-WRI) with the marginal posterior probabilities calculated by EAGLE and found that EAGLE-WRI increased the variant with lower probability in EAGLE very much. Therefore, we wanted to check the top variants with higher probability in EAGLE-WRI, which did not appear in the benchmark. So we check this variant in the case that no new reads are added to the pile-up.

As shown in Table 5.1, we manually checked 30 indel and 25 SNP cases respectively. Among the 30 cases of indel, we found that 21 cases with high probability were not found in the benchmark data, but after checking the original BAM file, we found that 13 of them showed support for the possibility of variation. Among the 25 cases of SNP, we found that 17 examples with higher probability were not found in the benchmark data, but after checking the original BAM file, we found that 12 of them showed support for the possibility of variation. In the following, we will show 3 indel cases(Figure 5.4, 5.5, 5.6) and 3 SNP cases respectively(Figure 5.7, 5.8, 5.9).

As mentioned in subsection 1.1, because there are many uncertainties in variant calling, such as sequencing error, inaccurate alignment, and low read

coverage. Therefore, we believe that the degradation of performance may be related to the absence of some variant in the benchmark.

State / Type	Before checking		After manual checking	
	In Benchmark	Not in Benchmark	Benchmark or Plausible mutations	Appears wrong
indel	9	21	22	8
SNP	8	17	20	5

Table 5.1: Checking the pile-up for mutations

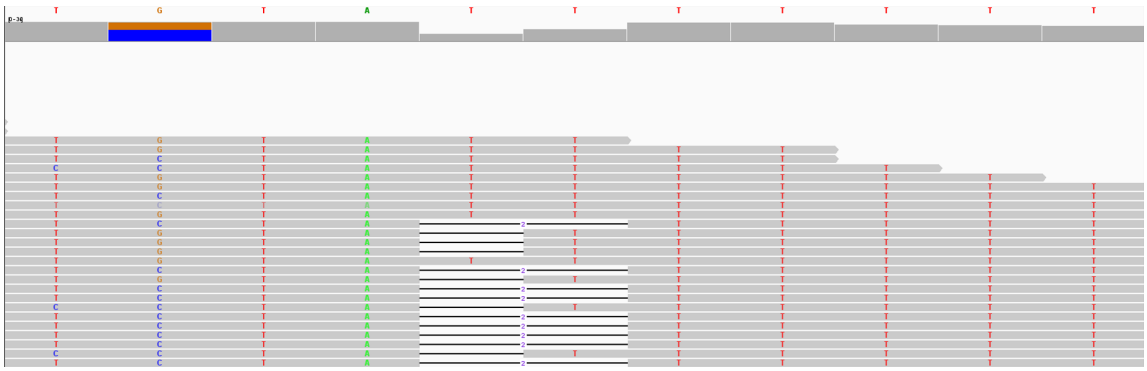

Figure 5.4: The Example1 of pile-up for indel.

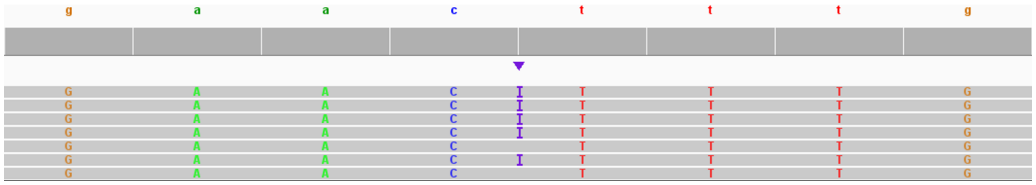

Figure 5.5: The Example2 of pile-up for indel.

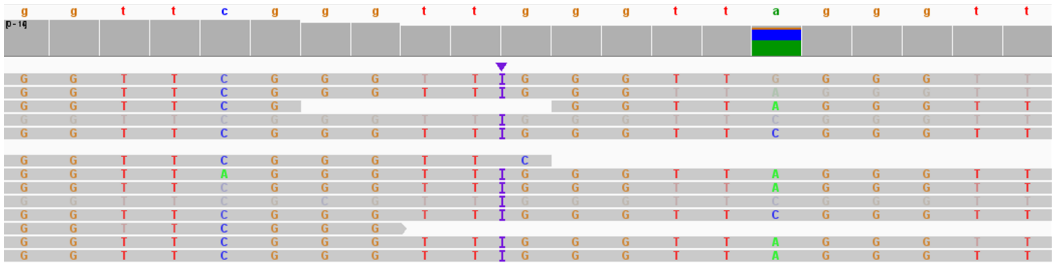

Figure 5.6: The Example3 of pile-up for indel.

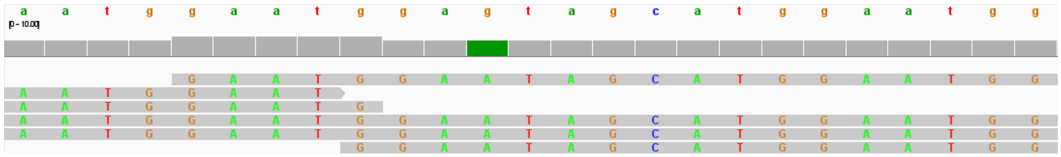

Figure 5.7: The Example1 of pile-up for SNP.

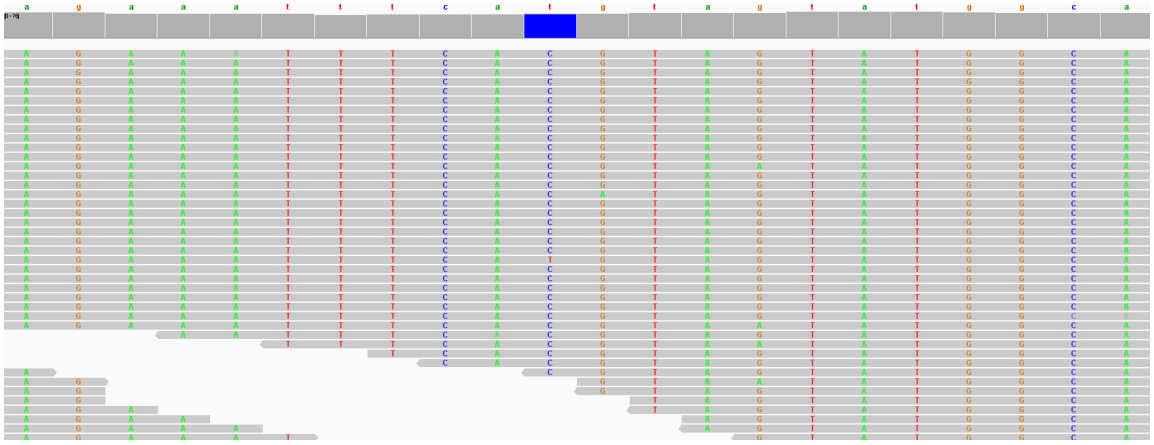

Figure 5.8: The Example2 of pile-up for SNP.

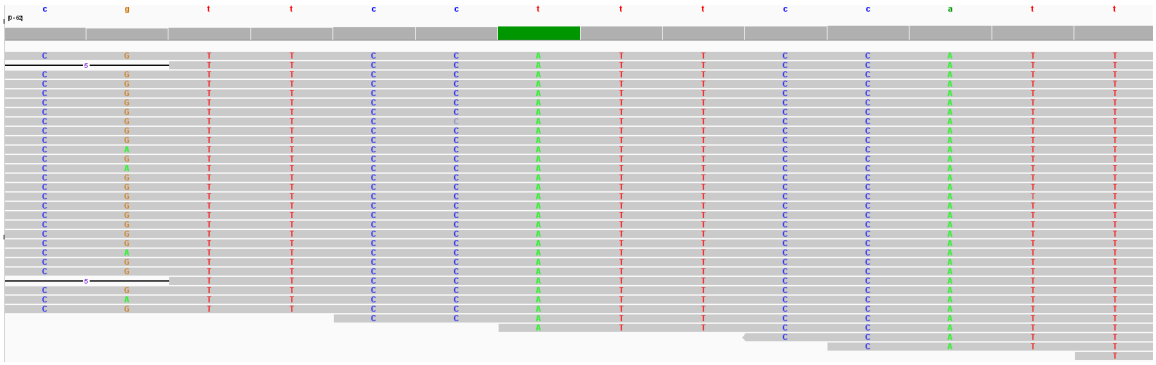

Figure 5.9: The Example3 of pile-up for SNP.

5.3 Multi-mapping detection

In the results showcase, we found that the benchmark data may have some missing variation. In the above process, it is also shown that EAGLE-WRI supports more variants not found in the benchmark than before. In other words, EAGLE-WRI adds many reads to the pile-up as described in subsection 2.4.

However, since there are many similar regions in the human genome reference sequence, it is possible that the reads added to this region are from reads that have been mapped to other positions in the reference genome. To rule out the possibility of such a problem, we will perform a check. We extracted the high probability examples from EAGLE-WRI to examine the problem.

The results showed that among the 30 examples of SNPs and indel, we found 2 and 1 variant with multi-mapping phenomenon respectively. Therefore, in the examination of this problem, we can probably exclude the possibility of

evaluation errors in EAGLE-WRI due to multi-mapping.

```
ERR174340.142744803     chr10   42316780
ERR174340.3836317       chr10   42316796
ERR174340.103029077     chr10   42316793
ERR174340.25224458      chr10   42321060
ERR174340.134480904     chr10   42321102
ERR174340.80930617      chr10   42316788
ERR174340.10509597      chr10   42316822
ERR174340.53796980      chr10   42316807
ERR174340.64359847      chr10   42321093
ERR174340.49154486      chr10   42321043
ERR174340.137769151     chr10   42316800
```

Figure 5.10: The example of SNP with multimapping phenomenon.

```
ERR174340.83456667      chr16   34626646
ERR174340.87448678      chr16   34626676
ERR174340.7045150       chr16   34626627
ERR174340.71487459      chr16   34626648
ERR174340.28047982      chr16   34626624
ERR174340.77448449      chr16   34626627
```

Figure 5.11: The example of indel with multimapping phenomenon.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In our previous study, we extended the functionality of EAGLE to reduce the effect of reference bias and to improve the accuracy of EAGLE in assessing variation. We have also demonstrated in previous simulations that the modified EAGLE is effective in eliminating the effects of reference bias, but we have not yet verified the accuracy of EAGLE on real data.

However, when we evaluated EAGLE by precision versus recall, we found a significant decrease in precision at the same recall level in the whole genome sequencing data. Therefore, in our subsequent study, we examined several cases that led to the decrease in performance and found that the decrease in performance might be related to the missing variant in the benchmark data. However, the missing variants could not be ruled out because the benchmark used more stringent conditions to call variant.

We also detect the areas where EAGLE gives a higher chance of variation. This is to avoid that EAGLE highly supports variants that are not found in the benchmark because of adding too many similar regions of reads to pile-up. The results show that only a very small number of the examples we examined had the Multi-mapping case. Therefore, we can generally rule out the possibility of evaluation errors in EAGLE.

As mentioned above, in this study, we speculate that the decrease in performance may be due to a variation in the missing part of the benchmark data. As a result of the above examination, we generally exclude the possibility of evaluation errors in the new version of EAGLE. Therefore, it can be shown that EAGLE has the ability to detect missing benchmark data. However, more significantly, we only examined a small number of cases during the study, so we cannot be completely sure of our findings. As a result, we need to examine all cases of performance degradation more rigorously in future studies to support our findings and to examine whether there are systemic defects in the new version of EAGLE.

6.2 Future Work

In this paper, we examined both the pile-up and the multi-mapping detection manually. This consumes a lot of our time and limits our ability to make a more comprehensive evaluation of the results. As future work, we should implement an automatic process to achieve both of these tests, as well as a rigorous follow-up examination of performance using GIAB exome sequencing data, which may also reveal the same problems. This will give a more complete evaluation of EAGLE with read-index and make it more useful for NGS data analysis.

Bibliography

[1] Vandana Shashi et al. “The utility of the traditional medical genetics diagnostic evaluation in the context of next-generation sequencing for undiagnosed genetic disorders.” *Genetics* 16.2 (2014), pp. 176–182.

[2] Peter D. Stenson et al. “The Human Gene Mutation Database: towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies.” *Human genetics* 136.6 (2017), pp. 665–667.

[3] Aaron McKenna et al. “The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data.” *Genome research* 20.9 (2010), pp. 1297–1303.

[4] Heng Li et al. “The sequence alignment/map format and SAMtools.” *Bioinformatics* 25.16 (2009), pp. 2078–2079.

[5] Andy Rimmer et al. “Integrating mapping-, assembly-and haplotype-based approaches for calling variants in clinical sequencing applications.” *Nature genetics* 46.8 (2014), pp. 912–918.

[6] Tony Kuo et al. “EAGLE: explicit alternative genome likelihood evaluator.” *BMC medical genomics* 11.2 (2018), pp. 1–10.

[7] Tung-Yi Chou. “Investigating the Feasibility of Indexing Read Sequences and Its Potential to Reduce Reference Bias in Variant Calling.” (2020).

[8] Wen-Hong Su. “Towards Eliminating Reference Bias in Genome Variant Calling — Integrating a BWA-MEM Read Index into the EAGLE Variant Evaluation Method.” (2021).

[9] Petr Danecek et al. “The variant call format and VCFtools.” *Bioinformatics* 27.15 (2011), pp. 2156–2158.

[10] David J. Lipman and William R. Pearson. “Rapid and sensitive protein similarity searches.” *Science* 227.4693 (1985), pp. 1435–1441.

[11] Peter JA Cock et al. “The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants.” *Nucleic acids research* 38.6 (2010), pp. 1767–1771.

[12] Torsten Günther and Carl Nettelblad. “The presence and impact of reference bias on population genomic studies of prehistoric human populations.” *PLoS genetics* 15.7 (2019).

[13] Nae-Chyun Chen et al. “Reference flow: reducing reference bias using multiple population genomes.” *Genome biology* 22.1 (2021), pp. 1–17.

[14] Jessica Lau. *Reference bias: Challenges and solutions*. `https://www.sevenbridges.com/reference-bias-challenges-and-solutions/`. 2017.

[15] Ryan Poplin et al. “Scaling accurate genetic variant discovery to tens of thousands of samples.” *BioRxiv* (2018).

[16] Petr Danecek et al. “Twelve years of SAMtools and BCFtools.” *Gigascience* 10.2 (2021).

[17] Heng Li. “A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data.” *Bioinformatics* 27.21 (2011), pp. 2987–2993.

[18] Heng Li. “Mathematical Notes on SAMtools Algorithms.” (2010).

[19] Justin M. Zook et al. “Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls.” *Nature biotechnology* 32.3 (2014), pp. 246–251.

[20] Michael A. Eberle et al. “A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree.” *Genome research* 27.1 (2021), pp. 157–164.

[21] Geraldine A. Van der Auwera et al. “From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline.” *Current protocols in bioinformatics* 43.1 (2013), pp. 11–10.

[22] Heng Li. “Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM.” (2010).

[23] *AddOrReplaceReadGroups (Picard)*. `https://gatk.broadinstitute.org/hc/en-us/articles/360037226472-AddOrReplaceReadGroups-Picard-`.

[24] *MarkDuplicates (Picard)*. `https://gatk.broadinstitute.org/hc/en-us/articles/360037052812-MarkDuplicates-Picard-`.

[25] Adrian Tan, Gonçalo R. Abecasis, and Hyun Min Kang. “Unified representation of genetic variants.” *Bioinformatics* 31.13 (2015), pp. 2202–2204.