國　立　成　功　大　學

醫 學 資 訊 研 究 所

碩 士 論 文

探討定序讀數的索引結構降低參考偏差在變異辨認上的可行性以及潛力

Investigating the Feasibility of Indexing Read Sequences and its Potential to Reduce Reference Bias in Variant Calling

研 究 生：周東誼

指 導 教 授：賀保羅 教授

中 華 民 國 一 百 零 九 年 七 月

**探討定序讀數的索引結構降低參考偏差在變異辨認上的可行性以及潛力**

**周東誼\* 賀保羅\*\***

**國立成功大學醫學資訊研究所**

# 中文摘要

近年來由於次世代定序技術的進步，取得人類參考基因序列的資料變得快速許多，有許多研究在分析基因體變異，基因體變異的辨認一直是一個重大的課題。在變異辨認的流程中，通過映射定序讀數到參考基因序列上，分析映射結果，我們可以得到變異發生的位置及形態。但是，參考基因序列通常不會含有個人的變異，如果定序儀定出來的讀數有個體上的變異，會導致在映射的過程中因為與參考序列有差異而不會被映射到正確的位置上，映射結果就有一定程度的誤差，我們稱之為”參考偏差”，進而影響到基因體變異辨認的準確性。本研究將探討我們提出的一個系統對於降低參考偏差是否可行及其潛力。

本系統分成三個部分，第一部分是數據前處理，以一般做法將參考序列建立索引，透過BWA將讀數映射到參考序列上產生映射結果檔案(BAM)，然後我們將讀數也建立一個索引結構；第二部分從變異辨認格式(VCF)檔案中對於每個變異，在映射結果檔案裡找到有涵蓋到該變異的位置上的讀數，我們稱之為”堆積序列”；第三部分是對於每個變異，截取參考序列上的一段序列進行修改使其符合發生變異的狀態，我們稱之為假設序列。再用假設序列對讀數索引結構進行搜尋，找到的讀數與映射結果的堆積序列進行比對，如果找到的讀數不存在於堆積序列中，說明我們可以找到因為參考偏差而沒有映射到該位置上的讀數。

在實驗中，本研究使用了GIAB 的定序資料集以及ClinVar的變異資料集，我們把變異分成單核苷酸多型變異和插入刪除變異分別進行實驗。結果顯示在插入刪除變異實驗上可以找到包含變異的讀數但在標準映射檔案(BAM)上沒有映射到該變異的位置。在大多數情況下，這些讀數存在於標準映射檔案(BAM)中，但被映射到基因組的另一個（同源）區域。因此，我們在讀數數據上建立索引的策略可以找到有用的證據（匹配讀數）來潛在地支持標準讀數映射過程中被遺漏的讀數。但是，這些變體通常在旁系同源物的區域內發現，因此必須謹慎解釋這一新證據。

關鍵字: 次世代定序、參考偏差、變異辨認、單核苷酸多型變異、插入刪除變異

\*作者 \*\*指導教授

**Investigating the Feasibility of Indexing Read Sequences and its Potential to Reduce Reference Bias in Variant Calling**

Tung-Yi Chou\* Paul Horton\*\*

Institute of Medical Informatics,

National Cheng Kung University, Tainan, Taiwan, R.O.C.

# Abstract

In recent years, due to the advancement of the next generation sequencing technology, obtaining data on human reference gene sequences has become much faster. There are many studies analyzing the human genome variation, and the variant calling has always been an important topic. In the process of variant calling, by mapping the read sequences on the reference sequences and analyzing the result, we can get the position and type of the variant. However, the reference sequences usually does not contain individual variants. If the reads sequencing by the sequencer have individual variations, it may cause that they not be mapped to the correct position due to differences with the reference sequences during the mapping process. Thus, the mapping result has a certain degree of bias we call “reference bias”, which in turn affects the accuracy of variant calling. This study will discuss the feasibility and potential of a system we propose to reduce the reference bias.

The system is divided into three parts. The first part is data pre-processing. The reference sequences is indexed in the usual way. The reads are mapped to the reference sequences by using the BWA to generate a mapping result file (standard mapping BAM). Then we also create an index structure of the reads. The second part is that using the variants from a predefined set of candidate variants stored in a Variant Calling Format (VCF) file. For each variant, we first gather the reads covering that position as determined by standard read mapping (the so-called “pileup”) use the finds the reads covering the position of the variation in the mapping result file, which we call the "pileup". The third part is for each variant, extracting a local part of the reference sequence and editing it to contain the variant; constructing a hypothetical sequence. Then we use the hypothetical sequence to query the read index structure. The found reads are compared with the pileup of the mapping result. If the found reads do not exist in the pileup, it means that we can find the reads that is not mapped to the position because of the reference bias.

In the experiment, we used the sequencing data from the GIAB dataset and the potential variants from the ClinVar dataset. We divided the variants into SNP variants and indel variants. Our results show that indel variants sometimes match the reads which match a variant but which are not mapped to that genome position in the standard mapping BAM file. In most cases those reads are present in the standard mapping BAM file but mapped to another (paralogous) region of the genome. Thus our strategy of building an index on the read data can find useful evidence (matching reads) to potentially support variants which the standard read mapping procedure misses. However those variants usually are found in regions within paralogs, so this new evidence must be interpreted with caution.

Keywords: NGS, reference bias, variant calling, SNP, indel

\* Author \*\* Advisor

# 誌謝

首先感謝

西元2020年 周東誼 謹致於

國立成功大學 醫學資訊研究所所實驗室

Contents

[中文摘要 I](#_Toc45555833)

[Abstract III](#_Toc45555834)

[誌謝 VI](#_Toc45555835)

[Contents VII](#_Toc45555836)

[List of Figures IX](#_Toc45555837)

[List of Tables X](#_Toc45555838)

[Chapter 1 Introduction 1](#_Toc45555839)

[1.1 Background 1](#_Toc45555840)

[1.1.1 File Formats 3](#_Toc45555841)

[1.1.2 Read Mapping 3](#_Toc45555842)

[1.1.3 “Pile-up” based variant calling 4](#_Toc45555843)

[1.2 Motivation 5](#_Toc45555844)

[1.3 Research Objectives 7](#_Toc45555845)

[1.5 Organization of this Thesis 7](#_Toc45555849)

[Chapter 2 Related Work 8](#_Toc45555850)

[2.1 BWA 8](#_Toc45555851)

[2.2 Reference Bias 8](#_Toc45555852)

[Chapter 3 Method 9](#_Toc45555853)

[3.1 Overview 10](#_Toc45555854)

[3.2 Read Index Structure 11](#_Toc45555855)

[3.2.1 BWA index 12](#_Toc45555856)

[3.2.2 Creating Read Index 13](#_Toc45555857)

[3.3 Accessing the BAM “Pileup” 13](#_Toc45555858)

[3.4 Query Hypothesis 14](#_Toc45555859)

[3.4.1 Hypothetical Sequence 15](#_Toc45555860)

[3.4.2 Find SEMS and Filtering Reads 16](#_Toc45555861)

[Chapter 4 Experiment 18](#_Toc45555862)

[4.1 Experimental Design 18](#_Toc45555863)

[4.2 Data Description and Selection 19](#_Toc45555864)

[4.2.1 Data Description 19](#_Toc45555865)

[4.2.2 Data Selection 20](#_Toc45555866)

[4.3 Query Hypothesis 20](#_Toc45555867)

[4.3.1 Confidence of SMEM 21](#_Toc45555868)

[4.3.1.1 Simulation of the Querying Sequence 21](#_Toc45555869)

[4.3.1.2 Distribution of the Matched Length 23](#_Toc45555870)

[Chapter 5 Result 24](#_Toc45555871)

[5.1 SNP and Indel 24](#_Toc45555872)

[5.2 Cases Analysis 26](#_Toc45555873)

[5.2.1 The Cases of Indel variant 27](#_Toc45555874)

[5.2.1.1 Case 1 27](#_Toc45555875)

[5.2.1.2 Case 2 28](#_Toc45555876)

[5.2.1.3 Case 3 29](#_Toc45555877)

[5.2.2 The Cases of SNP variant 30](#_Toc45555878)

[5.2.2.1 Case 1 31](#_Toc45555879)

[5.2.2.2 Case 2 32](#_Toc45555880)

[5.2.2.3 Case 3 32](#_Toc45555881)

[5.2.2.4 Case 4 33](#_Toc45555882)

[Chapter 6 Discussion 34](#_Toc45555883)

[6.1 Conclusions 36](#_Toc45555884)

[6.2 Future works 37](#_Toc45555885)

[References 38](#_Toc45555886)

# List of Figures

Figure 1‑1 pileup illustration 4

Figure 3‑1 briefly illustrate the read index query 10

Figure 3‑2 reference bias detection overview 11

Figure 3‑3 Fasta and Fastq format 12

Figure 3‑4 Fastq to Fasta 13

Figure 3‑5 the pileup from a standard read mapping BAM file 14

Figure 3‑6 producing the hypothetical sequence (SNP variant) 15

Figure 3‑7 producing the hypothetical sequence (indel variant) 16

Figure 3‑8 SMEMs 16

Figure 3‑9 pileup reads (left) and reads obtained by querying our read (right) 17

Figure 4‑1 proportion of SNPs and indel variants in ClinVar 20

Figure 4‑2 First-Order Markov Model 22

Figure 4‑3 adding the mutation in the simulated sequence 23

Figure 4‑4 Distribution of match length 24

Figure 5‑1 Case 1 of the indel variant 28

Figure 5‑2 Case 2 of the indel variant 29

Figure 5‑3 Case 3 of the indel variant 30

Figure 5‑4 Case 1 of the SNP variant 31

Figure 5‑5 Case 2 of the SNP variant 32

Figure 5‑6 Case 3 of the SNP variant 33

Figure 5‑7 Case 4 of the SNP variant 34

# List of Tables

Table 5‑1 Indel variants experimental data 25

Table 5‑2 SNP variants experimental data 26

Table 6‑1 execution time of function 35

# Introduction

## Background

*Variant calling* refers to the inference of the genotype of an individual sample based on DNA sequencing. The DNA sequencing is usually done using Illumina sequencers producing read sequences of 150~250bp in length. These sequences are typically compared against a reference genome sequence to infer genome variants. This inference is complicated by the fact that many genomes (i.e. the human genome) are somewhat repetitive in nature and not all length 200 DNA fragments can be uniquely mapped to the reference genome (especially when those reads include variants).

In recent years many methods have been developed to perform variant calling. Some comparisons have been published to evaluate the effectiveness of different methods of variant calling pipelines [1] , and using different alignment tools [2] ; according to these results variant calling from NGS DNA sequencing data generally works well but some challenges remain.

The human genome is made up of 23 chromosome pairs, and with a total of about three billion DNA base pairs. There are 22 autosomal chromosome and sex chromosome (X and Y chromosome). The information in DNA base pairs is stored as four kind of chemical bases: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). In 2003, the Human Genome Project was declared complete, and since then whole genome sequencing has been used in much research.

Next Generation Sequencing (NGS) is a high-throughput and fast technology for a large number of short-reads, and has lower cost compared to the traditional Sanger Sequencing. NGS technology has made various genome wide “omics” measurements (genome sequencing, RNA seq, CHiP seq, etc.) affordable enough to become routine procedures a has had a major impact on biological science.

There are many variants in the human population. In 2015, the 1000 Genomes Project found that a typical human genome has about 5 million variants that differ from the reference human genome. The variants can be classified into types, such as Single Nucleotide Polymorphism (SNP) which is a single nucleotide substitution at particular sites, Insertion or Deletion (indel) which is an insertion or deletion of a DNA sequence bases measuring the length from 1 to 10,000 base pairs, and Structural Variants (SV) which is variation in the structure of chromosome on one species that range includes over 50 base pairs. In most known genomes, indel frequency is markedly lower than SNP frequency, except some repetitive regions. indels, however, involve more base changes per variant than SNPs, so when weighted by length indel variants make up a very significant fraction of genome variation. Moreover, variant calling methods are generally more sensitive in detecting SNPs than indels, so the number of indel variants may be somewhat under-estimated [3] .

### File Formats

Here we introduce the file format we use in this thesis.

BAM/SAM : a kind of alignment data, Binary Alignment Map / Sequence Alignment Map, BAM is compressed to binary from SAM.

Fasta : A text-based format for storing nucleotide sequences.

Fastq : A text-based format for storing nucleotide sequences and sequencing error probabilities encoded as quality scores (represented as ASCII characters).

VCF : Variant call format, a test file for storing the information of gene sequence variations.

### Read Mapping

Sequence alignment is arranging pairs of DNA, protein, or RNA sequences, to highlight their similarities. *Read mapping* is a special case of sequence alignment in which DNA sequence reads are aligned to a reference genome. Due to the high number of DNA sequence reads generated by even routine DNA sequencing, read mapping software (e.g. BWA [4] , bowtie2 [5] ) adopt specialized algorithms to quickly find matches between read sequences and the reference genome. At the core of those algorithms are data structures called *indexes* which contain information about a set of sequences (called the text), such that exact matches between an arbitrary query sequence and any part of the text can be found. Remarkably, as long as enough computer memory is available to store the index, query to text matching can be done in time proportional to the length of the query, independent of the length of the text. Data-structures used to implement such indexes include hash tables and suffix arrays. Of course a query may contain matches to multiple parts (substrings) in the text. In particular, repetitive sequences in the genomes lead to multiple matches to short DNA read sequences.

### “Pile-up” based variant calling

From the above sections the reader can probably imagine a straightforward way to do variant calling. First, perform read mapping for all sequencing reads against the reference genome. Then stack those mapped reads on top of each other in a special kind of multiple alignment often called the “pile-up” with the bottom sequence being the reference genome (as shown in Figure 1-1).

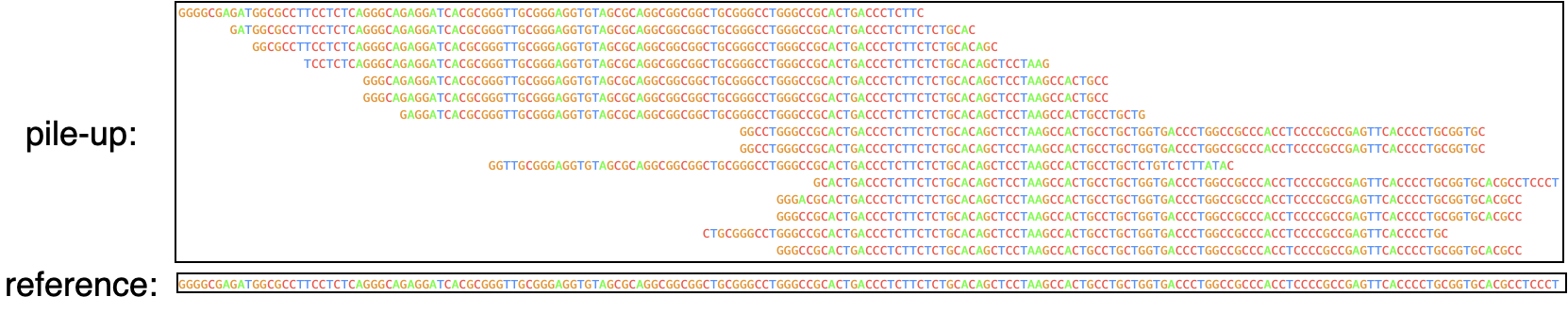


Figure 1‑1 pileup illustration

A simple way to do variant calling is to simply scan this pile-up multiple alignment looking for columns in which read sequences consistently differ from the reference genome. This straightforward procedure works well in many cases and is indeed the essence of most variant calling methods. Unfortunately, however this approach depends entirely on successfully mapping variant containing sequencing reads to their corresponding position in reference genome sequence. Since the read mapper is simply comparing read sequences to the reference sequence, reads which contain variants are less likely to be correctly mapped to their genome region of, an effect known as *reference bias* [7] . Reference bias is expected to be a source of false negatives (failing to call true variants) especially for longish indel variants.

## Motivation

Many methods have been developed for variant calling such as GATK [8] [9] [10] , Platypus [11] , FreeBayes [12] . These methods mostly work well but their heavy reliance on the “pile-up” can leads to mistakes in repetitive regions. EAGLE [13] reported a modest improvement in accuracy of evaluating candidate variants. The approach lowers the reliance on the pile-up by considering a large number of possible read-to-genome alignments and combining this evidence in a principled way according to an explicit probability model. Unfortunately, EAGLE does still rely on the pile-up to identify which reads to consider as potentially relevant to a given variant, and thus theoretically still suffers from reference bias. In principle, that problem can be solved by searching for reads matching each candidate variant, instead of simply grabbing reads mapped to the variant position in the pile-up. However one might ask: 1) is this theoretical reference bias really a problem in practice? and 2) can searching for relevant reads for each candidate variant be made feasible in terms of computing resources of time and memory space? This thesis aims to answer these questions.

Sequence Alignment Map (SAM) is a bioinformatic format that stores sequencing alignment results that contain reads, position, quality score and some mapping flags. It has been widely used in various studies. There are many tools to alignment reads, such as BWA, bowtie2, etc. In our research, we used BWA as our alignment tool and used the parameter MEM, which has better results for short sequences.

## Research Objectives

In this research, we develop a system to find the reads which support the statistical hypothesis. Our system is divided into three parts. First, we pre-process the data, build an index of read sequences. Second, we find the pileup of every variant. At the last, we query the hypothetical sequence to the read index. The objectives of our research as follows:

1. Build a read index structure.
2. Find the reads which support hypothetical cases.
3. Analyze some interesting cases of the finding result.

## Organization of this Thesis

This thesis is organized as follows: In Chapter 1, we introduce the background, motivation, and this research objectives. In Chapter 2, we review the research on the tool BWA and the reference bias. In Chapter 3, we describe the method of our system. In Chapter 4, we experiment of real data. In Chapter 5, we show the result and analyze some cases. In Chapter 6, we discuss the result, provide a conclusion and future works.

# Related Work

In this chapter, we review the related work in recent years. In section 2.1, we introduce the Burrows-Wheeler Aligner (BWA), a well-known alignment tool. In section 2.2, we briefly introduce the related research on the reference bias.

## BWA

BWA is a software to align the reads on reference sequences based on Burrows–Wheeler Transform (BWT). Using the FM-index [14] which combines the BWT and suffix array enables the backward search, making querying more efficient [15] [16] . The index function of BWA will build a BWT file, suffix array file, and some temporary files. The indexing reference sequences is a necessary step before mapping reads. There are three methods of mapping function of BWA, SW, ALN, and MEM. Li [17] claim that the BWA-MEM shows better performance in mapping 100bp sequences of several read aligners. In this paper, our read sequences data length is about 100bp. Therefore, we choose the BWA-MEM as our read mapping function.

## Reference Bias

There are some researches discuss the impact of reference bias, such as [18] , who investigated the effect of reference bias on population genomic studies of prehistoric human populations. They found that while most regions of genome suffer little bias, reference bias does have the potential affect downstream analysis. In other aspect, some researches use the genome graph [19] [20] to reduce the reference bias. These researches build a new structure of reference sequences which contain common variants. In this paper, we investigate whether building an index structure from read sequence data may be a practical way to reduce reference bias and investigate its potential to improve variant calling.

# Method

In this chapter, we introduce how to use the alignment tool mapping Next Generation Sequencing data, and get the BAM file. Then find the position of variation with VCF and reference sequencing data, replacing the variation on the reference genome data, and cutting the region as a hypothesis. Next, we create an index by the reads sequencing data to queried by the hypothesis previously we mention. In the last part, we find some reads to support our hypothesis.

## Overview

Figure 3-1 shows the core of method in this chapter. We develop a system to find out the reads that are not mapped on the BAM file, which can be used to get the read containing the specific variation. As Figure 3-2 shows, our system flow is divided into three parts. First of all, we use BWA to map the genome read sequences (Fastq) on the reference genome in the standard way. Secondly, to find the pileup, we get the variant position from the VCF file and find the same position on BAM, then we can get some reads which are covered in the position. At the last, we cut a region as our hypothetical sequence which is substituted the reference sequence with a variant. Querying the read index with our hypothetical sequence, we compare with the pileup and find the reads which are containing a variant but not exist in the pileup from BAM.

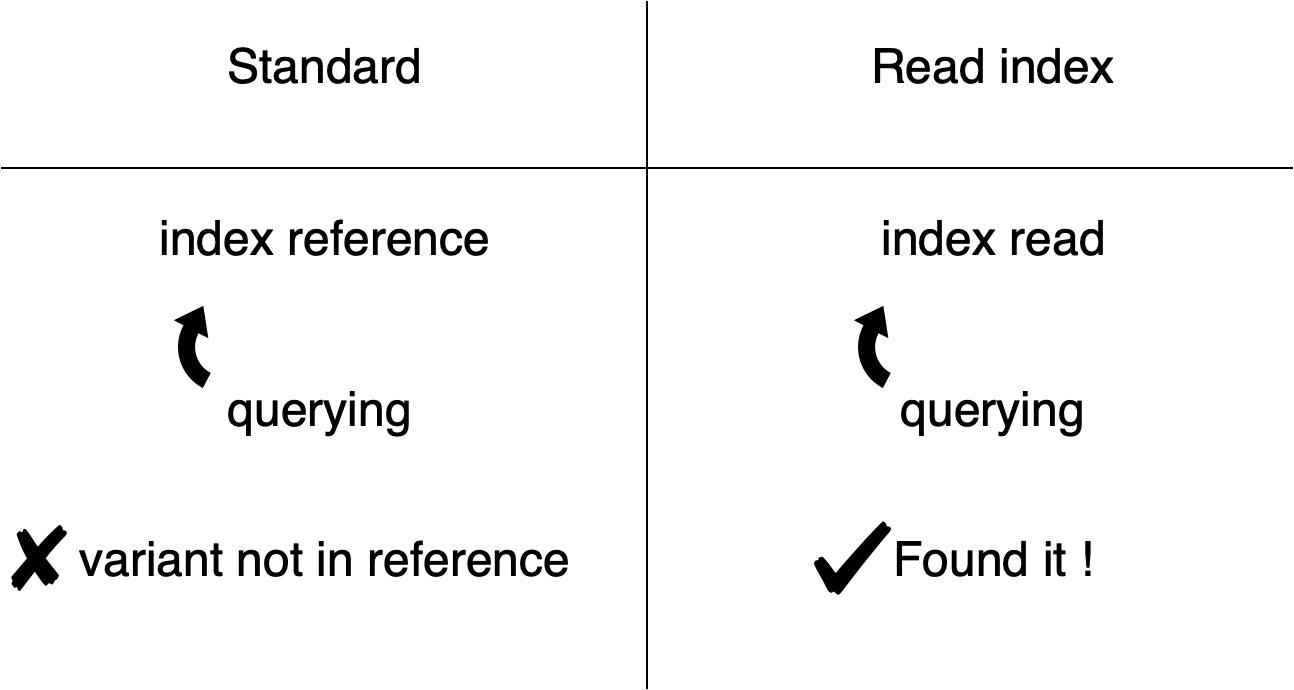


Figure 3‑1 briefly illustrate the read index query

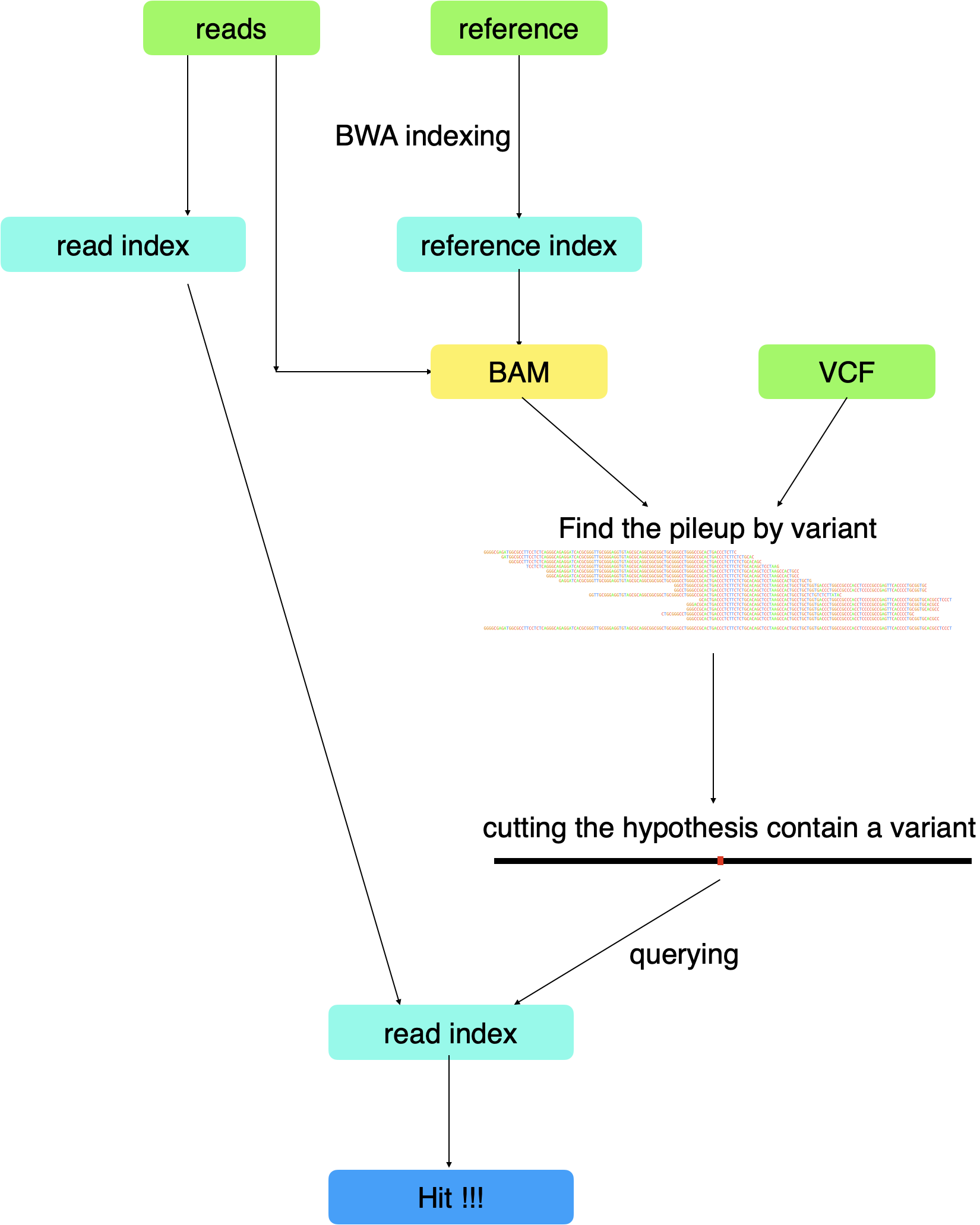


Figure 3‑2 reference bias detection overview

## Read Index Structure

In sequence alignment, the usual approach is to make an index of the reference genome sequence, but in this section, we introduce the approach of making an index structure of the genome reads sequencing data. Below we describe this in two parts: first we outline how the BWA alignment tool builds an index structure of the reference genome sequence, and compare the structure between the reference sequence format (Fasta) and read sequencing format (Fastq). The second step is preprocessing the read sequencing data and building an index.

### BWA index

In this step, we build a traditional index structure of the reference sequence by using the BWA with its index command, which is for aligning the reads on the reference by reference index position. As Figure 3-3 shows, the reference sequence format (Fasta) contains a name and sequence per chromosome. The DNA read sequencing format (Fastq) contains a name, a sequence, and its quality score which is stored as ASCII format.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Fasta | Fastq |
| format | Line 1 | >description of sequence | @sequence id |
| Line 2 | sequence | sequence |
| Line 3 |  | + |
| Line 4 |  | Quality value |
| example | | >NT\_113878.1 chromosome 1  GATAGCATCTGGCCTTAT… | @SEQ\_ID  ACCACCTGCTCC…  +  ???DB+=2C>??E;E… |

Figure 3‑3 Fasta and Fastq format

### Creating Read Index

BWA expects Fasta format sequences for indexing. In this step, we make the Fastq structure similar to the Fasta structure by restructuring, so that BWA can build an index of it. We keep the name and sequence of Fastq, and the quality score is taken out and placed in another file so that we can convert the format of Fastq to Fasta.

Figure 3-4 shows how to convert Fastq format sequence read int to Fasta format for building the read index structure.

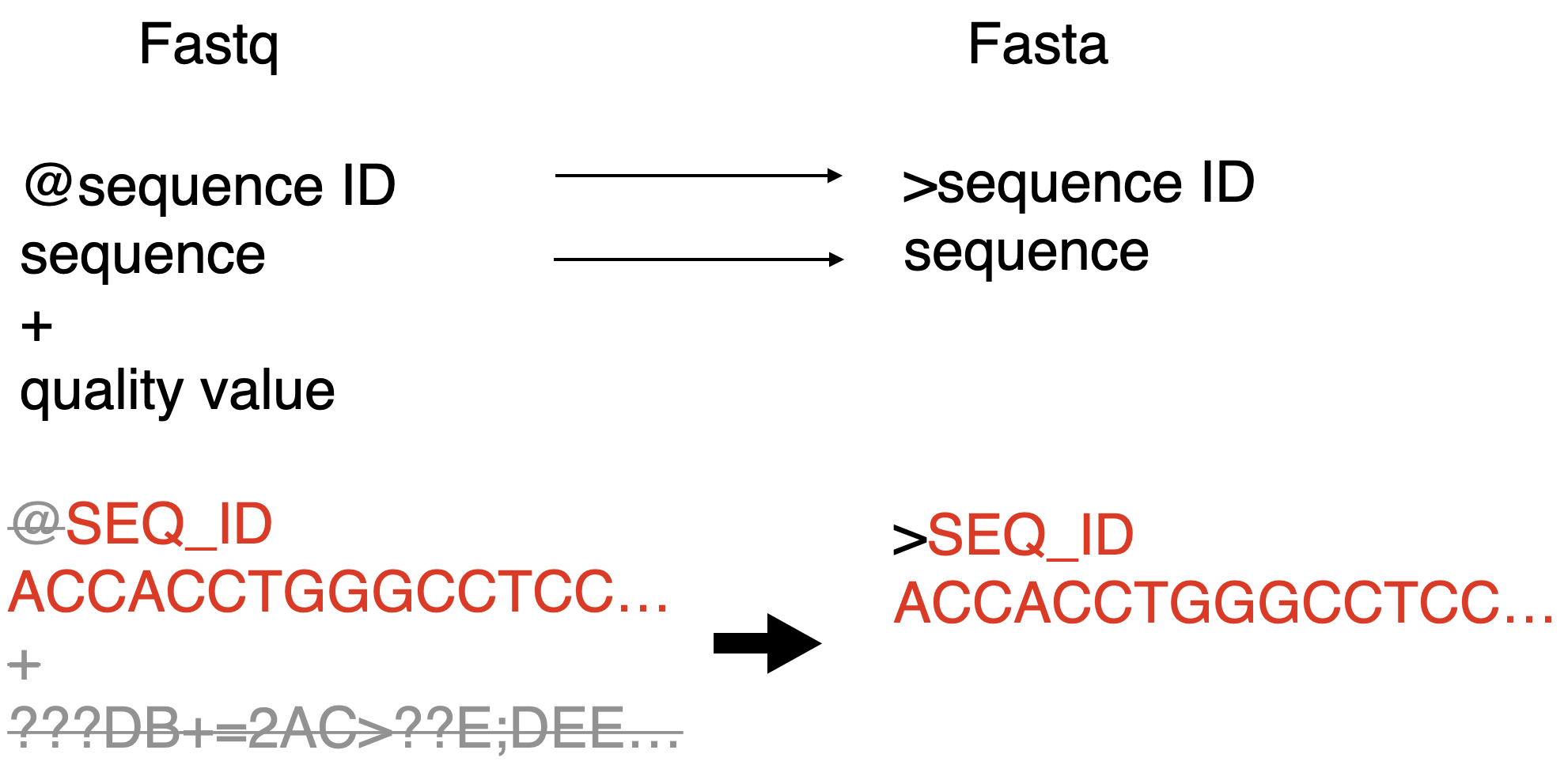


Figure 3‑4 Fastq to Fasta

## Accessing the BAM “Pileup”

In this section, we are going to introduce how we find out what reads map to a given genomic position in standard read mapping. In the BAM file, we can observe that the reads are mapped to a position of a chromosome. Many reads may overlap at the same position, which is called the pileup. Our purpose is to find the pileup which overlaps with a variant position.

Li H , Handsaker B, et al [6] develop a tool for processing read alignments, which is contain in a utility library HTSlib, a unified C library for accessing high throughput sequencing data. We use this library to efficiently processing the data, the BAM file is quickly accessible because of its binary format. As described in the following section we will compare these pileup reads with the set of reads we obtain via querying our read data index. Figure 3-5 shows the pileup reads which overlap a variant position.

一張含有 儀錶 的圖片

自動產生的描述

Figure 3‑5 the pileup from a standard read mapping BAM file

## Query Hypothesis

In this section, we introduce how we produce the hypothetical sequence and query the index that is divided into two steps. The first step is preparing the hypothetical sequence, produced by editing a region of the reference sequencing data to include the variant sequence at the specific position. The second step is querying the read index structure, as mentioned in the previous section.

### Hypothetical Sequence

In this section, we produce the sequence by editing a region of the reference sequence to include a variant of interest. We use the variants from an input VCF file as our set of hypothetical variants. We take out the information of each variant, including the genome position coordinates, chromosome, the sequence before mutation, and the sequence after the mutation occurs. For each variant, we replace with the mutation in the position of the chromosome and extract the region a fixed length before and after of the variant position (in this work we used a length of ±100 bp). Figure 3-6 and Figure 3-7 illustrate who we generated hypothetical sequences to query for the given variant.

一張含有 時鐘 的圖片

自動產生的描述

Figure 3‑6 producing the hypothetical sequence (SNP variant)

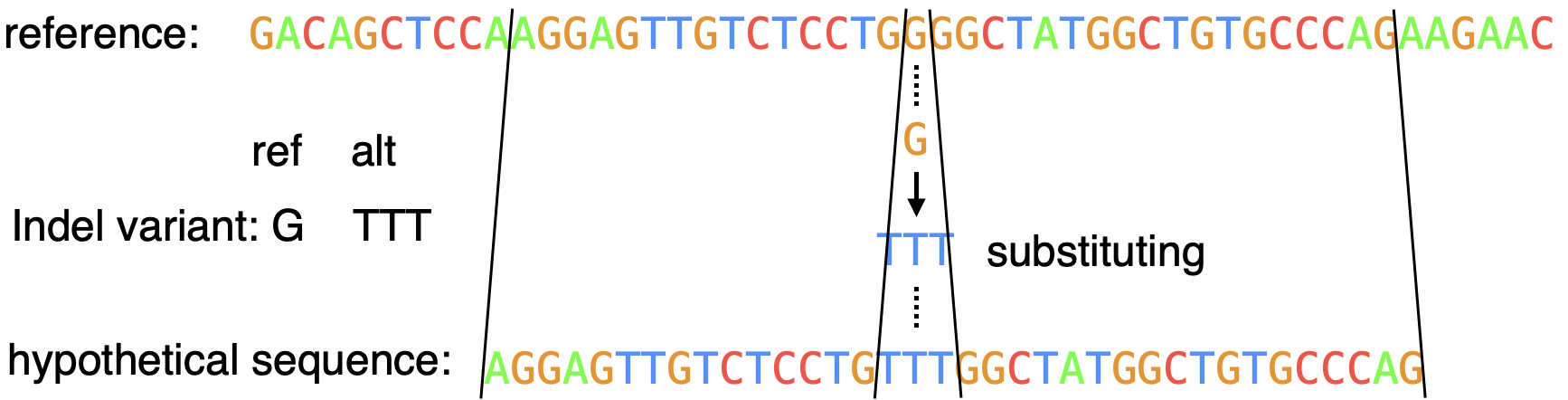


Figure 3‑7 producing the hypothetical sequence (indel variant)

### Find SEMS and Filtering Reads

In previous section, we have produced the hypothetical sequences which contain a variant. Then, we explain how do we query the read index with the hypothesis. Our purpose is finding the reads that support the hypothesis but are not present in the pileup reads overlapping the variant genome position. Fortunately, the BWA source code provides a function which can be called to obtain the position in the indexed text of super-maximal exact matches (SMEMs) [21] to a query. As Figure 3-8 shows, SMEM is a longest maximal exact match sequence.

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

Figure 3‑8 SMEMs

Here the indexed text is the read sequences concatenated together in the order they appear in the Fasta file given to BWA when building the index. To go from a position in this text to the corresponding read number we can simply divide read length as fortunately (as is often the case) all reads in the dataset we used are the same length. Then, using the method to find the SMEMs, we also get the SMEMs coordinate of suffix array to find the read which supports the hypothesis. By comparing the read and the pileup reads, there are two cases that will occur. The first case is the read already exists in the pileup reads. The second case is the read is not in the pileup reads, and therefore would be missed from consideration in a standard NGS workflow. Figure 3-9 illustrate the read compare with the pileup reads.

一張含有 煙火 的圖片

自動產生的描述

Figure 3‑9 pileup reads (left) and reads obtained by querying our read (right)

# Experiment

In this chapter, we design an experiment to find some reads which contain a variant but are not mapped to the position of the variant in the standard read mapping BAM file. To test our system, we use real sequencing data. In section 4.1, we are going to describe the experimental design. Then we describe the data and data selection in section 4.2. The information of the resource of data we describe in section 4.2.1 and the information variant and replaced correlation we mention in section 4.2.2. We show the result of query hypothesis in section 4.3. The query the length of confidence are introduced in section 4.3.1 and the result of different variant type (SNP and indel) we show in section 4.3.2. In section 4.4, we show the result of cases analysis for some interesting cases.

## Experimental Design

Our system can be divided into three parts. The first part is that we map the read genome sequence on the reference sequencing data which is the human genome resource from NCBI by using the BWA. Next, we get the variant from VCF file which is from the dataset on NCBI website, then our purpose is find the pileup reads which are covered at the same position of variant on the reference. The last part is that cutting a region of the reference sequence which is replaced with a variant as our hypothetical sequence. We filter the hypothesis by querying our read index structure, then we find the reads which contain a variant and not exist in pileup reads by comparing with the pileup from BAM file.

## Data Description and Selection

In this section, we first give the description of the data where the genome resource from a public dataset. Then, we classify the variant of the VCF file by different type which we defined.

### Data Description

There are three kinds of data what we used. The first one is the reference data (Fasta) which contains the chromosome sequence of human genome. The second one is the read sequencing data (Fastq) which is the raw data by sequencing with the sequencer. The last one is the variant calling format (VCF) which contain information of variants.

First, the reference genome sequencing data is Genome Reference Consortium Human Build 37 patch release 13 (GRCh37.p13) version from NCBI website. There are 1 to 22, X and Y chromosome of primary assembly and some extra sequences. We use the primary assembly chromosome as our reference sequence.

Second, the read sequencing data is real sequencing data from NA12878 which is a cell line of an individual female from a CEPH pedigree that is Utah residents with Northern and Western European ancestry, using an exome sequencing dataset (Garvan HG001) by sequencer HiSeq2500 from Genome-In-A-Bottle (GIAB) [22] .

Third, we use these variants of VCF file from ClinVar [23] which is a freely available resource of medically important variants. ClinVar collect the variants which have a relationship about human health.

### Data Selection

We divided the ClinVar variants into two types, SNPs (single nucleotide polymorphism) and indels (insertion and deletion). Figure 4-1 shows the proportion of SNPs and indels.



Figure 4‑1 proportion of SNPs and indel variants in ClinVar

## Query Hypothesis

In this section, we introduce results and describe the experiment to evaluate the confidence of match length. The match confidence experiment is shown in section 4.3.1. We introduce experiment results of SNP and indel variant in section 4.3.2.

### Confidence of SMEM

As we mentioned in section 3.4.2, we find the super-maximal exact matches of the query hypothesis. We want to know what length of exact match is long enough to support our hypothetical sequence. Then we design an experiment to test the match length and divided into two parts. The first part is simulating genome-like random sequences as our query, we do using a First-Order Markov Model trained on the reference genome. The second part is counting the match of the query which we simulate in the first part, and analyze the result of the distribution of the matched length.

#### Simulation of the Querying Sequence

In this section, we introduce the method of producing the simulating sequence. To get a match length which has high confidence supporting our querying sequence, we test the generated sequence and choose an appropriate length as our match length. The sequence generated by completely random function may not suitable for the biological sequence because the distribution of bases is not entirely uniform in the human genome, in particular CG sequences are strongly depleted. Therefore, we choose the First-Order Markov Model to simulate random sequences with genome-like composition of dimers. Figure 4-4 shows, the state translation of the First-Order Markov Model on DNA sequences.

一張含有 膝上型電腦 的圖片

自動產生的描述

Figure 4‑2 First-Order Markov Model

We count the DNA bases (ACGT) of the read sequence data and we calculate the probability of the next base from the previous base, e.g., the probability that the next base is Cytosine (C) from Adenine (A) is computed by,



We also add some mutation to make the simulated sequences more like real human genome sequences. We first set a sliding window which length is 40 to slide a read we generated. Then we set the mutation occur probability as 0.2%. When the mutation occurs, the base is replaced one of the other 3 bases at a random position in the window. As Figure 4-5 shows, illustrating the mutation adding in the simulated sequence.

一張含有 儀錶, 時鐘, 畫畫 的圖片

自動產生的描述

Figure 4‑3 adding the mutation in the simulated sequence

* + - 1. **Distribution of the Matched Length**

After generating sequence in previous section, we generate 100 sequence to test an appropriate match length supporting our query sequence. The First-Order Markov Model we computed from the read sequences data, and we query the 100 sequences we generated to the read index structure. The result is the match length of the simulated sequences distributing over 11 base pairs to 22 base pairs. As Figure 4-6 shows, we observe that the most numbers of match length are 13 base pairs to 16 base pairs. Therefore, we choose length 40 as match length to support our query sequence number. We have high confidence that choosing length 40 because it is more 2 times about the simulated sequence match length.



Figure ‑ Distribution of match length

# Result

In this chapter, we describe the test of our system and some cases analysis. We show the result of querying hypothetical sequences of different variant types, include SNP and indel variant in section 5.1. By observing the result, we found some matching reads which support the hypothesis but have different situations, and we are going to introduce these situations in section 5.2.

## SNP and Indel

In this section, we show the result of different variant types. There are 70,735 indel variants form the ClinVar dataset. The other one, SNP variants are random select 100,000 variants from the dataset. The amount of matching read of the querying sequence of indel variants are 6950. We divided the SMEM into two part, the forward match and the reverse match. The amount of forward match which are matching on the forward strand are 3711, and the amount of reverse match are 4026 which are matching on the complement sequence. We call the matching sequence which does not in the pileup of the position of a variant as “hit”, also divided into the forward hits and reverse hits. The forward hits are 301. The reverse hits are 179 shown in Table 5-1. In 70,735 indel variants we find 65,947 indel variants have the pileup, there is about 93.23% variants which position can be mapped the reads by using the alignment tool.

Table ‑ Indel variants experimental data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | reads | Forward SMEM | Reverse SMEM | Forward Hits | Reverse Hits |
| Numbers | 6950 | 3711 | 4026 | 301 | 179 |

The amount of matching read of the querying sequence of SNP variants is 6587. Forward SMEM is 3964, reverse SMEM are 3743, forward hits are 624, and reverse hits are 560. As Table 5-2, we show the experimental data. In 100,000 SNP variants, there are 94,399 variants can find the pileup, it is about 94.4% variants with pileup by using the alignment tool.

Table ‑ SNP variants experimental data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | reads | Forward SMEM | Reverse SMEM | Forward Hits | Reverse Hits |
| Numbers | 6587 | 3964 | 3743 | 624 | 560 |

By comparing two experimental data, we observe the probability of hits of indel variants is about 0.73%, the probability of hits of SNP variants is about 1.25%. We estimate the hits of Indel variants will more than the hits of SNP variants because the indel variants length usually more than one character, and the alignment tool BWA usually allows the one character mutation (SNP) in alignment processing. We believe that the read contains an SNP variant that mostly can be mapped in the position. That is to say, most of the reads which contain SNP variants can be found in the pileup. But the experimental data show the different results from we previously estimated, we explanate the situation in the next section 5.2.

## Cases Analysis

In this section, we illustrate the match read we found to compare with the pileup by substituting the reference sequence, analyzing some cases, and describing the feature of these cases. We divided the cases into two parts, the one is belong to the Indel variants, the other one is belong the SNP variants. We show the DNA bases by different color, Adenine (A) is color Green, Cytosine (C) is color red, Guanine (G) is color orange, and Thymine (T) is color blue. The two down arrow range in several following figures is a matching read length for every read we found.

### The Cases of Indel variant

In this section, we illustrate the match read we found to compare with the pileup by substituting the indel variant from the reference sequence.

#### Case 1

As Figure 5-1 shows, we see that the variant is an insertion with “A” inserting “ATTT”, the number of pileup is 1 that is only one read be mapped in the position. By using our system, we can find these reads different from the read of pileup. The two down arrow shows the range of the match length. All of these reads we found which length are over 40 base pair to support our condition. There are 4 reads not in alignment data, that is to say, these reads can be found by querying the read-index structure. The bottom read on Figure 5-1 is in the alignment data, but not in the position of the variant, it was mapped on the other position by the alignment tool because the other position was aligned better. The alignment tool mapped the read on the unreplaced reference sequence if the individual has the variant. The alignment tool can not align the read in the position because the reference sequence data do not contain the individual mutation. In this case, although the read is mapped on the other position, there is possible that the read is mapped on the position when the variant occurs.

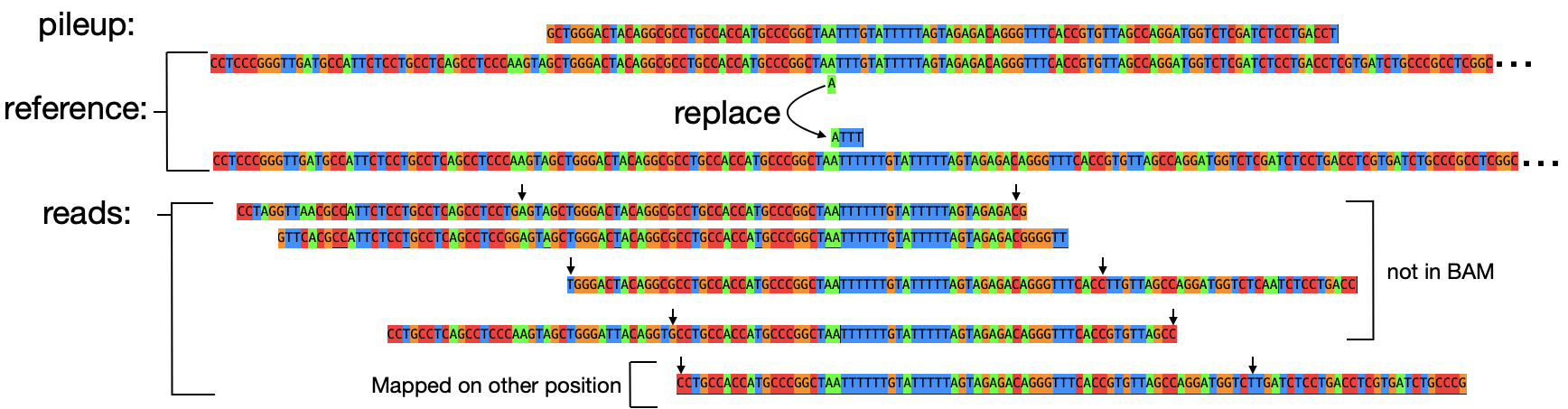


Figure ‑ Case 1 of the indel variant

#### Case 2

As Figure 5-2 shows, we see that the variant is a deletion with removing a “C” base pair. The BWA mapped one read to the position of the variant. “NC\_000011.9” is the description of chromosome 11 in GRCh37.p13 version. We find 6 reads to support the hypothesis by using our system. We observe that these reads are mapped to the chromosome tag “NW\_003871076.1” in the alignment data. We investigate the tag and find that the tag is also chromosome 11 but it is from the patch HG299\_PATCH. This is a technical problem that the patch and the main reference are different references to the alignment tool, despite they both represent the same chromosome. The patch position does not convert to the main reference position, so the alignment tool does not map these reads to the position. These reads may have an effect on the variant calling workflow. By using our system, we can find these reads which support the variant occurs but is mapped to the patch reference.

一張含有 光 的圖片

自動產生的描述

Figure 5‑2 Case 2 of the indel variant

#### Case 3

In this case, we observe that the variant is a deletion with removing “GT” base pairs. There are 19 pileup reads which are mapped to the position in chromosome 7. We find 11 reads supporting the hypothesis, investigate these reads, and divide them into three conditions. First, as Figure 5-3 shows, there are four reads also be mapped the position in chromosome 7 but the position is different from the position of this variant. The mapping position distance is about 1,553,700 base pairs from the position of variant and the length of reference chromosome 7 is 159,138,663 base pairs. Second, there are four reads be mapped to “NW\_003871064.1” which is from HG1257\_PATCH. Third, we find three reads not in the alignment data.

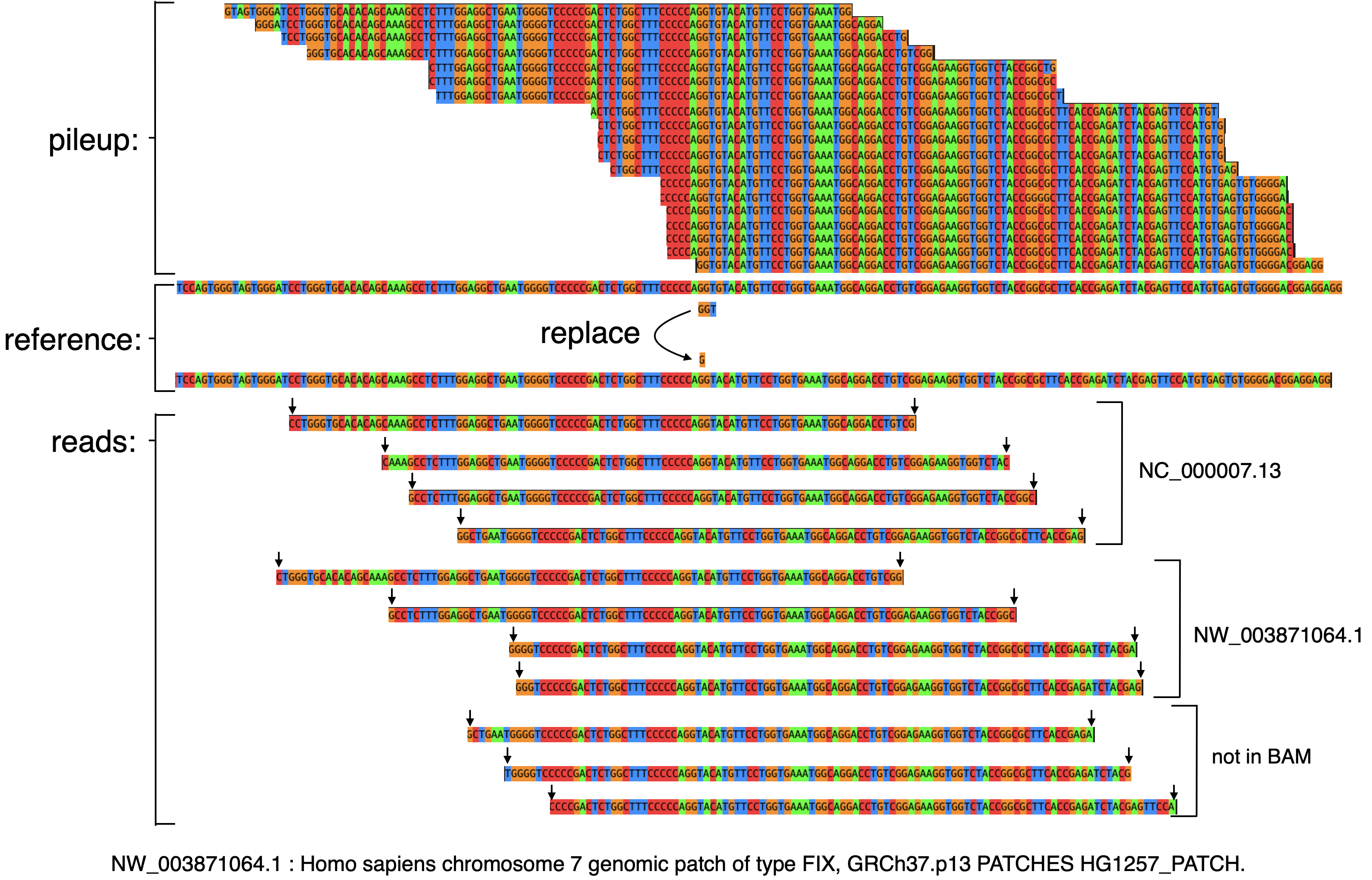


Figure 5‑3 Case 3 of the indel variant

### The Cases of SNP variant

In SNP cases, the hits of querying sequence of a SNP variant are more than the hits of querying sequence of an indel variant. After analyzing the reads, we speculate that there are two reasons, one reason is that SNP variant occurs probability is higher than indel variant occurs probability. The other reason is that there are some regions of the human body would be repeated producing, led to the reads may are mapped on the other similar region. It means that the alignment tool always chooses a most likely region to map, if a mutation occurs, the alignment tool may map on the different position.

#### Case 1

As Figure 5-4 shows, the SNP variant is “T” substituted by “G”, the pileups are 7 reads.

The variant is in chromosome X. We find 7 read differently from the pileup. The SNP variant is the only difference between the reads and the pileups. Although we found these reads that differ from the pileup and not in position of the variant, these reads still exist in the alignment data. They are mapped to the other reference chromosome tag, “NW\_003871103” which is highly similar to the position of a variant. The tag is also the chromosome X and it is from HG1497\_PATCH. By querying the read index structure we build, we can find these reads which may originally be mapped on the patch reference.

一張含有 畫畫 的圖片

自動產生的描述

Figure ‑ Case 1 of the SNP variant

#### Case 2

As Figure 5-5 shows, the SNP variant is “C” replacing to “A”, the pileup are 6 read sequences. The variant is in chromosome 6. There are 6 reads we found to support our hypothesis and these reads are mapped to the other position tag “NT\_113891.2”. The tag is also chromosome 6 but from alternate locus group ALT\_REF\_LOCI\_2.

一張含有 光, 畫畫 的圖片

自動產生的描述

Figure 5‑5 Case 2 of the SNP variant

#### Case 3

As Figure 5-6 shows, the SNP variant is “T” substituted by “C”, the number of pileup is 7. The pileup is mapped to the position in chromosome 11, and we find reads are mapped to the position tag “NW\_003871077.1”. The tag also represents chromosome 11 and it is from the patch reference HG305\_PATCH. We observe that there are 3 reads in the pileup which are supporting the variant that occurs, and they are almost the same reads we find. But the BWA mapped them to the different position. By using our system, we can find the reads which are miss by the BWA despite they are from the patch reference. In variant calling processing, we have more confidence to infer that the variant occurs in the individual.

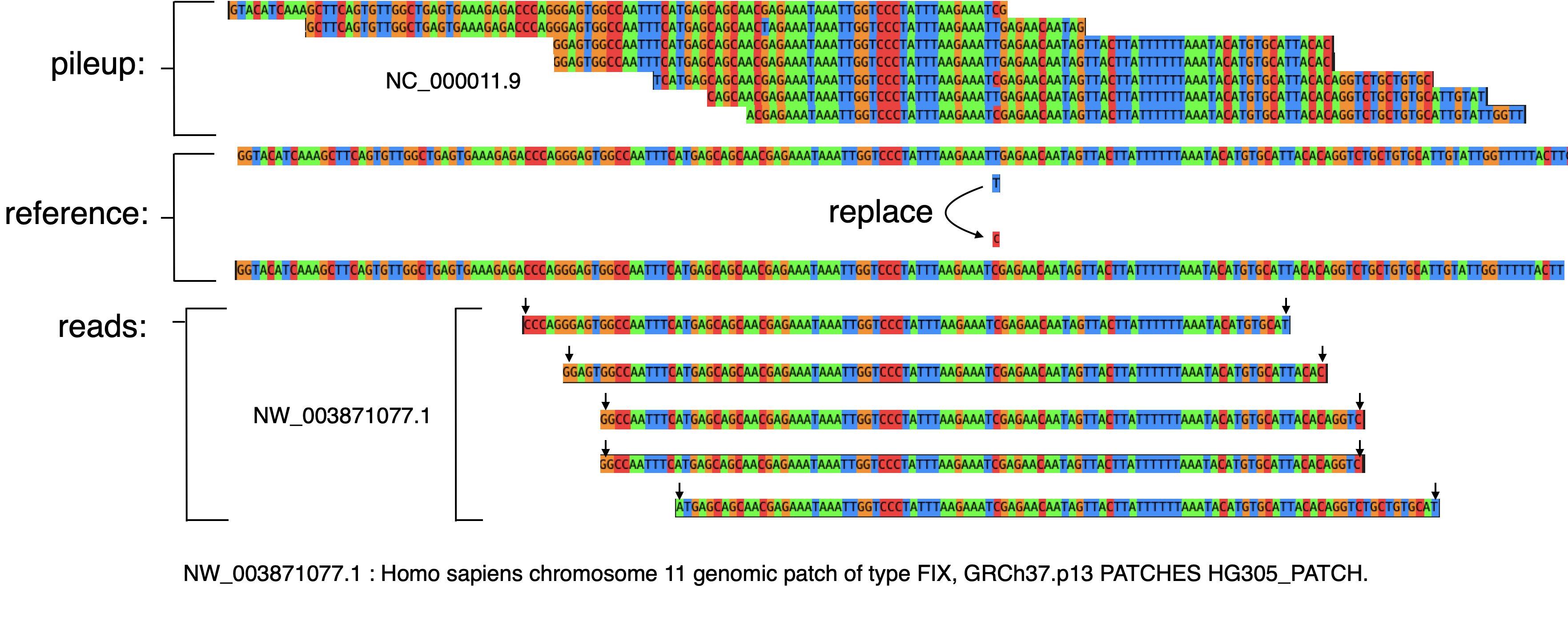


Figure 5‑6 Case 3 of the SNP variant

#### Case 4

In this case, the SNP variant is “A” replacing to “G”, there are 7 read sequences in the pileup. These reads are mapped to the position in chromosome 8. We can find 3 reads not in the pileup but supporting our hypothetical sequence. These reads we found are mapped to the position tag “NW\_003315923.1” which is also in chromosome 8 but it is from the patch reference HG104\_HG975\_PATCH. This is an interesting case, as we observe that the pileup reads are all support the replacing sequence. That is to say, in BWA mapping process, it allows the SNP variant and mapped these reads to the position. The reads by using our system we found are increasing reads support the variant.

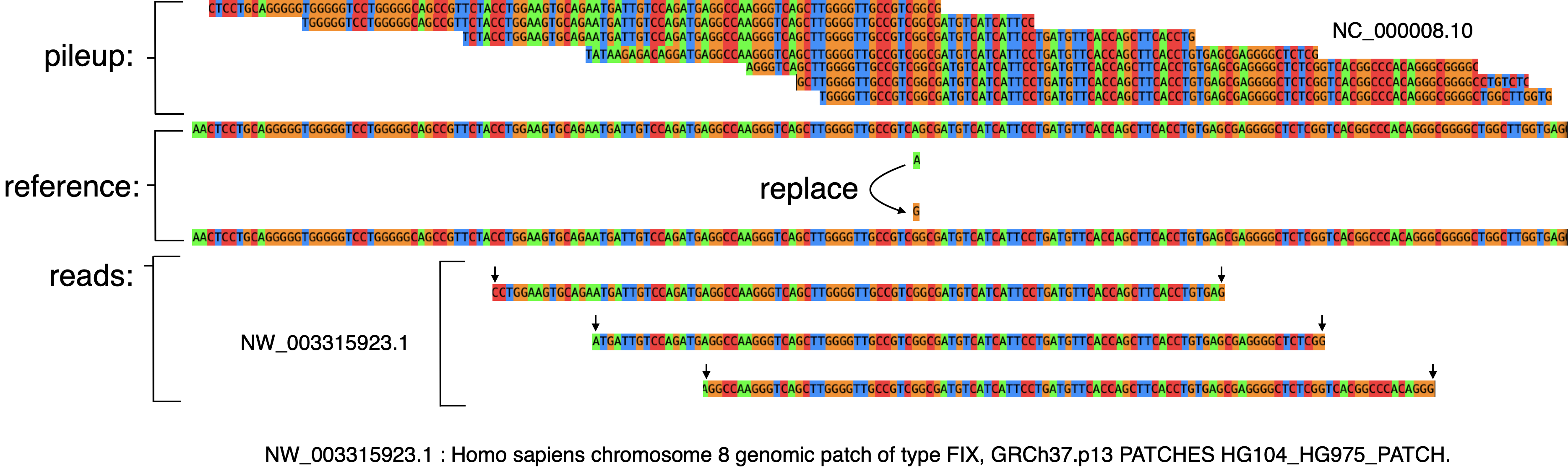


Figure 5‑7 Case 4 of the SNP variant

# Discussion

In this chapter, we are going to discuss two problems we worry about in this research beginning. The first one is the speed of querying the read index structure whether slow our whole system. There are 65,947 variants of the indel variants data and 94,399 variants of the SNP variants. As Table 6-1 shows the execution time of different program function. In indel experiment, it takes 29 minutes on finding the pileup and take 9 seconds on querying the index. In SNP experiment, it takes 45 minutes on finding the pileup and take 14 seconds on querying the index. As expected, the more variants take more execution time. We can observe that the most execution time is finding the pileup comparing with the querying index. Finding the pileup will access the BAM file frequently by every variant and still needs to access the reference file to cut a region to replace. To accelerate our system, we use the multi-thread to accelerate the function of finding pileup.

Table 6‑1 execution time of function

|  |  |  |
| --- | --- | --- |
|  | indel variants data | SNP variants data |
| Find pileup | 29 minutes | 45 minutes |
| Query index | 9 seconds | 14 seconds |

The second one is whether we can find the read sequences to support our hypothesis. The result of the experiment shows there are some reads we can found. However, the result has a little bit different from what we were looking for. In the experiment of SNP variants, many of the reads we found are mapped on the other positions. We find the reads which are not mapped on the position but still exist in the BAM file because the alignment tool mapped them in the other position. There are two possibilities in this situation, one is that the individual has the variant, and these reads which contain a variant should be mapped in the variant position but the alignment tool mapped them in the other position because the reference does not contain the variant and but another (paralogous) region of the reference genome approximately matches the read. The other possibility is the individual does not contained the variant, and so those reads are in fact mapped correctly in the other position by the alignment tool. Thus the interpretation of the results in this case require caution, but we assert that even in this case it is good that our approach can gather potentially informative evidence (variant containing reads) to use for variant calling or other purposes.

## Conclusions

We design a system to test the read index structure that can be querying to find the reads to support variant containing hypothetical genome sequences. Our system is divided into three parts. First, the data pre-processing part, we build an index of the reference sequences by standard practice, and generate the alignment result file (BAM) by using BWA. Then, we build the index of read sequences. Secondly, we find the reads mapped to the position of the putative variant in the “pileup” generated by standard read mapping. Third, to obtain a hypothetical genome sequence containing the, we extract the relevant region of the reference sequence and edit it to contain the variant. Finally, we query the read index for matches to that hypothetical sequence. The result of the querying is that the reads we found support the hypothesis.

In experiment, we using the real sequencing data and the ClinVar set of potential variants to test our system. ClinVar contains approximately 70,000 indels and 630,000 SNPs; of these we processed all of the indel variants and 100,000 randomly selected SNPs. After obtaining these results we chose several cases of each to investigate manually.

In the SNP variant case, new reads found by searching the read index typically are mapped to other positions in the genome. This implies that the position of the variant corresponds to a genome region which has at least one highly similar paralog in the genome. This case is complicated to interpret and one cannot simply assert that the variant is true. Still we would argue that it is still good to be able to find such reads as part of the relevant evidence.

The results for indel variants we more clear cut, and in most cases the new reads we find with our read index are not mappable to anywhere in the reference genome. Also our matching criteria of length 40 exact match ensures the match is not due to random chance. Thus the simplest explanation is that those indel variants are indeed present in the individual, suggesting that our strategy of query a read index can help support true indel variants which would otherwise be missed.

## Future work

The logical next step would be to integrate our approach into some variant calling procedure to benefit from the potential increased sensitivity of our approach. In particular our lab hopes to integrate this function into the EAGLE [13] as it is designed to properly weight relevant evidence when paralogous regions are involved.

# References

|  |
| --- |
| 1. Hwang, S., Kim, E., Lee, I. et al. Systematic comparison of variant calling pipelines using gold standard personal exome variants. Sci Rep 5, 17875 (2015). https://doi.org/10.1038/srep17875 |
| 1. Chen, J., Li, X., Zhong, H. et al. Systematic comparison of germline variant calling pipelines cross multiple next-generation sequencers. Sci Rep 9, 9345 (2019). https://doi.org/10.1038/s41598-019-45835-3 |
| 1. Prodduturi N, Bhagwate A, Kocher JA, Sun Z. Indel sensitive and comprehensive variant/mutation detection from RNA sequencing data for precision medicine. BMC Med Genomics. 2018;11(Suppl 3):67. Published 2018 Sep 14. doi:10.1186/s12920-018-0391-5 |
| 1. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics. 2009;25(14):1754‐1760. doi:10.1093/bioinformatics/btp324 |
| 1. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9(4):357-359. Published 2012 Mar 4. doi:10.1038/nmeth.1923 |
| 1. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078-2079. doi:10.1093/bioinformatics/btp352 |
| 1. Degner JF, Marioni JC, Pai AA, et al. Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. Bioinformatics. 2009;25(24):3207-3212. doi:10.1093/bioinformatics/btp579 |
| 1. McKenna A, Hanna M, Banks E, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 2010;20(9):1297-1303. doi:10.1101/gr.107524.110 |
| 1. DePristo, M., Banks, E., Poplin, R. et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet 43, 491–498 (2011). https://doi.org/10.1038/ng.806 |
| 1. Van der Auwera GA, Carneiro MO, Hartl C, et al. From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. Curr Protoc Bioinformatics. 2013;43(1110):11.10.1-11.10.33. doi:10.1002/0471250953.bi1110s43 |
| 1. Rimmer, A., Phan, H., Mathieson, I. et al. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. Nat Genet 46, 912–918 (2014). https://doi.org/10.1038/ng.3036 |
| 1. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. 2012. arXiv:1207.3907v2 [q-bio.GN]. |
| 1. Kuo, T., Frith, M.C., Sese, J. and Horton P. EAGLE: Explicit Alternative Genome Likelihood Evaluator. BMC Med Genomics 11, 28 (2018). https://doi.org/10.1186/s12920-018-0342-1 |
| 1. P. Ferragina and G. Manzini, "Opportunistic data structures with applications," Proceedings 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, CA, USA, 2000, pp. 390-398, doi: 10.1109/SFCS.2000.892127. |
| 1. Lippert RA. Space-efficient whole genome comparisons with Burrows-Wheeler transforms. J Comput Biol. 2005;12(4):407-415. doi:10.1089/cmb.2005.12.407 |
| 1. T. W. Lam, W. K. Sung, S. L. Tam, C. K. Wong, S. M. Yiu, Compressed indexing and local alignment of DNA, Bioinformatics, Volume 24, Issue 6, 15 March 2008, Pages 791–797, https://doi.org/10.1093/bioinformatics/btn032 |
| 1. Li, Heng. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. ArXiv. 1303. |
| 1. Günther T, Nettelblad C. The presence and impact of reference bias on population genomic studies of prehistoric human populations. PLoS Genet. 2019;15(7):e1008302. Published 2019 Jul 26. doi:10.1371/journal.pgen.1008302 |
| 1. Garrison, E., Sirén, J., Novak, A. et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. Nat Biotechnol 36, 875–879 (2018). https://doi.org/10.1038/nbt.4227 |
| 1. Paten B, Novak AM, Eizenga JM, Garrison E. Genome graphs and the evolution of genome inference. Genome Res. 2017;27(5):665-676. doi:10.1101/gr.214155.116 |
| 1. Li H. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. Bioinformatics. 2012;28(14):1838-1844. doi:10.1093/bioinformatics/bts280 |
| 1. Zook JM, Chapman B, Wang J, et al. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. Nat Biotechnol. 2014;32(3):246-251. doi:10.1038/nbt.2835 |
| 1. Landrum MJ, Lee JM, Riley GR, et al. ClinVar: public archive of relationships among sequence variation and human phenotype. Nucleic Acids Res. 2014;42(Database issue):D980-D985. doi:10.1093/nar/gkt1113 |