TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

COMPUTATIONAL ANALYSIS OF HUMAN GENOMIC VARIANTS AND LNCRNAS FROM SEQUENCE DATA

Ning Wang

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 23 | TECHNICA – INFORMATICA | TURKU 2023

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

COMPUTATIONAL ANALYSIS OF HUMAN GENOMIC VARIANTS AND LNCRNAS FROM SEQUENCE DATA

Ning Wang

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 23 | TECHNICA – INFORMATICA | TURKU 2023

University of Turku

Faculty of Technology
Department of Computing
Computer Science
Doctoral programme in Technology

Supervised by

Professor, Laura L. Elo
Turku Bioscience Centre
University of Turku and
Åbo Akademi University
Finland

Adjunct Professor, Sofia Khan
Turku Bioscience Centre
University of Turku and
Åbo Akademi University
Finland

Reviewed by

Associate Professor, Anish MS Shrestha
The College of Computer Studies
De La Salle University Manila
Philippines

Adjunct Professor, Esa Pitkänen
Institute for Molecular Medicine Finland
University of Helsinki
Finland

Opponent

Professor, Mauno Vihinen
Faculty of Medicine
Lund University
Sweden

The originality of this publication has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

ISBN 978-951-29-9321-5 PRINT
ISBN 978-951-29-9322-2 PDF
ISSN 2736-9390 (Painettu/Print)
ISSN 2736-9684 (Sähköinen/Online)
Painosalama, Turku, Finland, 2023

UNIVERSITY OF TURKU
Faculty of Technology
Department of Computing
Computer Science
NING WANG: Computational analysis of human genomic variants and lncRNAs from sequence data
Doctoral Dissertation, 216 pp.
Doctoral Programme of Technology
2023

ABSTRACT

The high-throughput sequencing technologies have been developed and applied to the human genome studies for nearly 20 years. These technologies have provided numerous research applications and have significantly expanded our knowledge about the human genome. In this thesis, computational methods that utilize sequence data to study human genomic variants and transcripts were evaluated and developed.

Indel represents insertion and deletion, which are two types of common genomic variants that are widespread in the human genome. Detecting indels from human genomes is the crucial step for diagnosing indel related genomic disorders and may potentially identify novel indel makers for studying certain diseases. Compared with previous techniques, the high-throughput sequencing technologies, especially the next-generation sequencing (NGS) technology, enable to detect indels accurately and efficiently in wide ranges of genome. In the first part of the thesis, tools with indel calling abilities are evaluated with an assortment of indels and different NGS settings. The results show that the selection of tools and NGS settings impact on indel detection significantly, which provide suggestions for tool selection and future developments.

In bioinformatics analysis, an indel's position can be marked inconsistently on the reference genome, which may result in an indel having different but equivalent representations and cause troubles for downstream. This problem is related to the complex sequence context of the indels, for example, short tandem repeats (STRs), where the same short stretch of nucleotides is amplified. In the second part of the thesis, a novel computational tool VarSCAT was described, which has various functions for annotating the sequence context of variants, including ambiguous positions, STRs, and other sequence context features. Analysis of several high-confidence human variant sets with VarSCAT reveals that a large number of genomic variants, especially indels, have sequence features associated with STRs.

In the human genome, not all genes and their transcripts are translated into proteins. Long non-coding ribonucleic acid (lncRNA) is a typical example. Sequence recognition built with machine learning models have improved significantly in recent years. In the last part of the thesis, several machine learning-based lncRNA prediction tools were evaluated on their predictions for coding potentiality of transcripts. The results suggest that tools based on deep learning identify lncRNAs best.

KEYWORDS: sequence data, human genome, indel, variant, sequence context, lncRNA, computational tool, method development, bionformatics, computational biology

TURUN YLIOPISTO
Teknillinen tiedekunta
Tietojenkäsittelytieteen laitos
Tietotekniikka
NING WANG: Ihmisen genomivarianttien ja lncRNA:iden laskennallinen analyysi sekvenssiaineistosta
Väitöskirja, 216 s.
Teknologian tohtoriohjelma
2023

TIIVISTELMÄ

Korkean suorituskyvyn sekvensointiteknologioita on kehitetty ja sovellettu ihmisen genomitutkimuksiin lähes 20 vuoden ajan. Nämä teknologiat ovat mahdollistaneet ihmisen genomin laaja-alaisen tutkimisen ja lisänneet merkittävästi tietoamme siitä. Tässä väitöstyössä arvioitiin ja kehitettiin sekvenssiaineistoa hyödyntäviä laskennallisia menetelmiä ihmisen genomivarianttien sekä transkriptien tutkimiseen.

Indeli on yhteisnimitys lisäys- eli insertio-varianteille ja häviämä- eli deleetio-varianteille, joita esiintyy koko genomin alueella. Indelien tunnistaminen on ratkaisevaa geneettisten poikkeavuuksien diagnosoinnissa ja eri sairauksiin liittyvien uusien indeli-markkereiden löytämisessä. Aiempiin teknologioihin verrattuna korkean suorituskyvyn sekvensointiteknologiat, erityisesti seuraavan sukupolven sekvensointi (NGS) mahdollistavat indelien havaitsemisen tarkemmin ja tehokkaammin laajemmilta genomialueilta. Väitöstyön ensimmäisessä osassa indelien kutsumiseen tarkoitettuja laskentatyökaluja arvioitiin käyttäen laajaa valikoimaa indeleitä ja erilaisia NGS-asetuksia. Tulokset osoittivat, että työkalujen valinta ja NGS-asetukset vaikuttivat indelien tunnistukseen merkittävästi ja siten ne voivat ohjata työkalujen valinnassa ja kehitystyössä.

Bioinformatiivisessa analyysissä saman indelin sijainti voidaan merkitä eri kohtiin referenssigenomia, joka voi aiheuttaa ongelmia loppupään analyysiin, kuten indeli-kutsujen arviointiin. Tämä ongelma liittyy sekvenssikontekstiin, koska variantit voivat sijoittua lyhyille perättäisille tandem-toistojaksoille (STR), jossa sama lyhyt nukleotidijakso on monistunut. Väitöstyön toisessa osassa kehitettiin laskentatyökalu VarSCAT, jossa on eri toimintoja, mm. monitulkintaisten sijaintitietojen, vierekkäisten alueiden ja STR-alueiden tarkasteluun. Luotettaviksi arvioitujen ihmisen varianttiaineistojen analyysi VarSCAT-työkalulla paljasti, että monien geneettisten varianttien ja erityisesti indelien ominaisuudet liittyvät STR-alueisiin.

Kaikkia ihmisen geenejä ja niiden geenituotteita, kuten esimerkiksi ei-koodaavia RNA:ta (lncRNA) ei käännetä proteiiniksi. Koneoppimismenetelmissä ja sekvenssi-tunnistuksessa on tapahtunut huomattavaa parannusta viime vuosina. Väitöstyön viimeisessä osassa arvioitiin useiden koneoppimiseen perustuvien lncRNA-ennustustyökalujen ennusteita. Tulokset viittaavat siihen, että syväoppimiseen perustuvat työkalut tunnistavat lncRNA:t parhaiten.

AVAINSANAT: sekvenssiaineisto, ihmisen genomi, indeli, variantti, sekvenssi-konteksti, lncRNA, laskentatyökalu, menetelmäkehitys, bioinformatiikka, laskennallinen biologia

Table of Contents

Abbreviations

bp	base pair
BAM	binary representation of Sequence Alignment/Map format
CHM	complete hydatidiform mole
CIGAR	concise idiosyncratic gapped alignment report
DNA	deoxyribonucleic acid
FN	false negative
FP	false positive
GRC	Genome Reference Consortium
GIAB	Genome in a Bottle
HGVS	Human Genome Variation Society
Indel	insertion and deletion
LncRNA	long non-coding RNA
MNV	multi-nucleotide variant
mRNA	messenger RNA
NGS	next-generation sequencing
ORF	open reading frame
RNA	ribonucleic acid
SAM	Sequence Alignment/Map format
SNP	single nucleotide polymorphism
SNV	single nucleotide variant
STR	short tandem repeat
SV	structural variants
TN	true negative
TP	true positive
TRF	Tandem Repeats Finder
VCF	variant call format
WES	whole exome sequencing
WGS	whole genome sequencing

List of Original Publications

This dissertation is based on the following original publications, which are referred to in the text by their Roman numerals:

I Wang N, Lysenkov V, Orte K, Kairisto V, Aakko J, Khan S, Elo L.L. Tool evaluation for the detection of variably sized indels from next generation whole genome and targeted sequencing data. *PLOS Computational Biology*, 2022; 18: e1009269. https://doi.org/10.1371/journal.pcbi.1009269.

II Wang N, Khan S, Elo L.L. VarSCAT: A computational tool for sequence context annotations of genomic variants. *bioRxiv*, 2022. https://doi.org/10.1101/2022.11.11.516085.

III Ammunét T, Wang N, Khan S, Elo L.L. Deep learning tools are top performers in long non-coding RNA prediction. *Briefings in Functional Genomics*, 2022; 21:230-241. https://doi.org/10.1093/bfgp/elab045

1 Introduction

The human genome, which is all of the deoxyribonucleic acid (DNA), is a set of sequences containing genetic information. The normal human genome contains 22 autosome pairs plus one sex chromosome pair in the cell nucleus as the nuclear genome, as well as a small amount of DNA in the mitochondria, the mitochondrial genome [1]. Genomics is the study of the whole content of a person's genome, which includes several genomic features that may influence biological functions and developmental processes. In this thesis, genomic variants and long non-coding ribonucleic acids (lncRNAs) are the topics that are studied.

Genomic variants, which are the DNA sequence variations among individuals, can be inherited from a parent as germline variants, or they occur during one's lifetime as somatic variants [2,3]. Some variants have high frequencies in the human population, with small or even no effects on health, whereas some variants are less common in populations and may lead to the development of certain diseases [4,5]. Different types of genomic variants, including nucleotide substitution, nucleotide gain and loss, and some structural re-arrangements, can be recognised based on the differences between human reference genome assembly and individual human genomes, or between paired samples such as cancer cell genomes and normal cell genomes [6]. Insertion and deletion variants, which are often known as indels, are the second most common variant type following single nucleotide polymorphism (SNP) [7]. Genomic variants can affect biological processes by influencing protein functions in protein-coding genes, or they are located in non-coding regions and play a role in regulatory functions to influence transcriptional activity of protein coding genes [8,9].

Protein-coding genes account for around 1.4% of the human genome [10]. Some non-coding part of genome may be transcribed into non-coding ribonucleic acids (RNAs) from non-coding genes at some point during development [11]. As representative types of non-coding RNAs, lncRNAs are transcripts with sizes of more than 200 nucleotides but may lack abilities to be translated into proteins [12]. Currently, more than 18,000 lncRNA genes located in the human autosomes and sex chromosomes are annotated in the encyclopedia of genes and gene variants (GENCODE) database (release 43) [13]. Some of lncRNAs have shown their key

roles in the regulation of cell processes and their functions as biomarkers in clinical diagnosis [14,15]. Recognising and identifying lncRNAs are important for understanding the complexity of the human genome and its biological processes.

Compared to Sanger sequencing and DNA microarray, the next-generation sequencing (NGS) technologies can sequence human genomes at a low cost and less time [16–18]. The experimental process of DNA sequencing involves determining the order of DNA sequences, while that of RNA-sequencing (RNA-seq) involves examining the quantity and sequences of RNA [19,20]. The applications of NGS in human genomic studies have enabled the ability to detect rare genomic variants and some low-expressed lncRNAs that were difficult to detect previously [21,22]. With the widespread application of NGS, a vast amount of human genomic data has been produced. Along with the use of advanced algorithms and computational tools, our knowledge of genomic variants and lncRNAs of the human genome has been improved [23,24].

Although many novel tools and algorithms have shown improved abilities for identifying genomic variants and lncRNAs, many challenges remain. One major hurdle is identifying indels in a wide range of sizes. Indels can be short variants as a few base pairs (bps) genome changes or be large as subtypes of structural variants (SVs), which are large-scale changes in the genome with a typical size range > 50 bp. Different tools and algorithms may use different features from sequencing data; thus, the optimal calling ranges of such tools might be different. To this end, in **Publication I**, a set of widely used variant calling tools were evaluated with the aim of finding their optimal indel calling ranges.

The evaluation results of **Publication I** showed that the majority of false positive (FP) indel calls made by variant calling tools were in simple repeats, including short tandem repeats (STRs). The sequence patterns of STRs may cause indels ambiguous breakpoints, further leading to equivalent indels and causing trouble for downstream analysis. In addition, the mutation rates of genomic variants, especially indels, may depend on sequence contexts [25,26]. The general proportions of breakpoint ambiguous indels and indels in STRs at the human individual level are unclear and remain a scientific question that needs to be answered. To better understand the genome sequence context of variants, a computational tool, VarSCAT, was implemented in **Publication II**. The tool provides a variety of functions to understand breakpoint ambiguity and tandem repeats in the sequence context of variants. The analysis of high-confidence human variant sets with VarSCAT showed that the sequence contexts had a strong relationship with genomic variants, especially indels.

The evaluation conducted in **Publication I** showed that the deep learning methods were the top performers in indel calling. Recently, deep learning methods have been widely applied in lncRNA prediction; the feature extraction and prediction

abilities of deep learning methods have shown great potential for identifying lncRNAs. The strengths of deep learning should be identified by benchmarking with tools developed based on other methods and on datasets with different properties. These comparison results can provide suggestions for tool selection based on different types of data and encourage the development of high-performing algorithms. In **Publication III**, a comparison of the most recent deep learning tools with other tools that use other machine learning methods for predicting lncRNAs was conducted.

In conclusion, this work evaluated popular and advanced methods for detecting genomic indels and lncRNAs, as well as developed a novel method that helps to demonstrate the sequence context of variants.

2 Aims of the thesis

The overall aim of this thesis is to investigate computational tools for sequence data analysis. NGS variant calling tools for indel calling, methods for annotating the sequence contexts of genomic variants, and tools for lncRNA prediction are the topics of focus. With the development of bioinformatics algorithms, especially with the application of machine learning algorithms that are used in the bioinformatics field, the abilities of tools have improved significantly in recent years. To provide suggestions for tool selection, a comprehensive and unbiased evaluation of existing tools should consider various research contexts and purposes. The results of this evaluation may also reveal weaknesses in the current development of algorithms and provide suggestions for future development. To better understand the relationships of variants and their sequence context in the human genome, the sequence context of variants such as STRs and low complexity regions needs to be parsed, and tools must be developed to serve this purpose.

The specific aims of this thesis to address the above needs are as follows:

1. Investigate and evaluate existing variant calling tools for detecting variably sized indels with different types of NGS data (**Publication I**).
2. Design and implement new methods for a better understanding of the sequence contexts of genomic variants (**Publication II**).
3. Assess the performance of deep learning-based and other machine learning-based tools for lncRNA prediction (**Publication III**).

3 The human genome and sequencing technologies

3.1 Human genomic variants and lncRNAs

The genome of a human individual contains all the genetic information for developmental regulation and biological processes. The human genome comprising sequences of DNA base pairs, contains regions with various distinct functions such as genes which encode for proteins and vast intergenic regions. The genes are comprised of exons and introns. The former is transcribed into mature RNA, whereas the latter is trimmed off during transcription processes. Ribonucleic acids can be divided into several groups: messenger RNAs (mRNAs), transfer RNAs, ribosomal RNAs and other non-coding RNAs. Messenger RNAs can be divided as 5' and 3' untranslated regions and several open reading frames (ORFs), only one of which can be further translated into proteins. The other RNAs may not be translated, but they have functions in structural support or functional regulations. In this thesis, two types of human genomic variants, namely, insertions and deletions (indels), the sequence contexts of genomic variants, and lncRNAs are the topics discussed.

3.1.1 The human genome

The two copies of haploid human genome consists of approximately 3 billion bps of DNA, the shape of which is a double-stranded helix structure of nucleotide chains. The monomeric units that consist of chains of nucleic acid polymers are called nucleotides. A nucleotide, which is the basic building block of nucleic acids, is composed of three chemical sub-units: a five-carbon sugar molecule, a nitrogenous base and one phosphate group. The order of the four types of nitrogenous bases, namely, adenine (A), cytosine (C), guanine (G), and thymine (T), defines the sequences of a genome. For DNA, the forward strand and the reverse strand are specifically paired so that an A always pairs with a T, and a C always pairs with a G [27]. The majority of human genome DNA is in the cell nucleus as a nuclear genome, which is divided into 23 linear molecule pairs as chromosomes, with the longest composed of around 250,000,000 nucleotides and the shortest composed of around 50,000,000 nucleotides. For a normal nuclear genome, the chromosomes consist of 22 autosome pairs and one sex chromosome pair as XX for females or XY for males. A small amount of DNA in a

circular structure is in the mitochondria as a mitochondrial genome. In the human body, most cells are diploid, meaning that a cell contains two copies of each autosome and two sex chromosomes of XX or XY. By contrast, sex cells are haploid, meaning that a cell has only one copy of each autosome and one sex chromosome. Currently, the human genome is estimated to have approximately 60,000 genes, with protein-coding genes, non-coding genes, and pseudogenes each accounting for one-third [13,28]. Genes are sequences of nucleotides in DNA; many encode the synthesis of RNAs and proteins. The human genome contains the biological information that governs biological functions and development processes [1].

Repetitive DNA, which is a DNA pattern that occurs in multiple copies, accounts for over half of the human genome. A large number of repetitive DNAs are located in regulatory or intergenic regions, and a substantial proportion of repetitive DNAs can be transcribed and translated into RNAs and proteins [29]. Based on previous research, almost all human genes (99%) contain at least one repetitive sequence in their 5000 bp flanking regions, 69% human genes contain at least one repetitive sequence in their 5' UTR or 3' UTR [30]. Among human coding genes, 12.5% of them carry short mononucleotide repeats [31].

Repetitive DNA can be divided into two classes: interspersed repeats and tandem repeats. Short interspersed nuclear elements, which are typically 100–300 bp in length, and long interspersed nuclear elements, which are typically > 300 bp in length, are the two main representative types of interspersed repeats that comprise more than 30% of the human genome [32,33]. The most well-studied interspersed repeat in the human genome is the class of Alu repeats, which are the main type of short interspersed nuclear element and account for approximately 11% of the genome. Alu elements are originally characterised by the action of the Alu restriction endonuclease and are associated with human diseases, gene expression, cell regulation and human population genetics [34,35]. Although interspersed repeats have been historically considered genomic junk, comparative genomics studies suggest that many classes of interspersed repeats can help in the understanding of the evolutionary history of the human genome [36].

Tandem repeats are sequence motifs that lie adjacent to one another and are called microsatellites, minisatellites, or macrosatellites based on the sizes of the motifs. Microsatellites, also known as STRs, consist of repeat motifs with 1–6 bp, occupying approximately 3% of the human genome [32,37]. In the past 10 years, the definition of the size of an STR motif have been 1–5 bp or 1–10 bp, but recently, the acceptable definition has been 1–6 bp [37,38]. Some studies illustrated that to have different mutation rates from the background genome, microsatellites should have lengths of minimum 10 bp [39–41], some cancer research discovered that many of the functionally relevant microsatellites can be 7-10 bp [42–44]. Minisatellites, the sizes of which are larger than those of microsatellites, have suggested repeats with motif sizes > 6 bp, while some

studies define them to have motif sizes > 10 bp. However, the upper size limit for minisatellites is not clear; it is usually a few hundred to a thousand base pairs [37,45,46]. Some studies have also called minisatellites as variable number tandem repeats, defined as tandem repeats with motif sizes > 6 bp [47–50]. Macrosatellites are the largest tandem repeats in the human genome, with repeat motifs of several thousand base pairs in size, which cover significant portions of the genome and are enriched in CpGs [45]. Tandem repeats, especially STRs, are the fastest-evolving DNA sequences in the human genome because of their relatively higher mutation rates (10^{-6} – 10^{-3} events per locus per gamete per generation) compared with single nucleotides (10^{-9} – 10^{-8}) [51–53]. The high abundance and mutation rates in the human genome make tandem repeats useful biomarkers in many research fields, such as forensic applications and human population studies [54,55]. Because of the high diverse sequence structure of the human genome, tandem repeats are not always required to be perfect. The structure of STRs, based on the similarities and gaps between each repeat unit, can be classified as perfect repeats, imperfect repeats, interrupted repeats, and compound repeats. Perfect repeats require every repeat unit to be the same as the repeat motif, and no gap is allowed between repeat units. Imperfect repeats allow some variations among repeat units but under a certain threshold. The interrupted repeats are often included in the category of imperfect repeats, which allow not only variations but also gaps among the repeat units. Compound repeats are complex repeats that contain several repeats with different motifs located in proximity, but some studies have excluded compound repeats from their biological research [56,57].

The central dogma of molecular biology, which was first published in 1958 and restated in 1970 by Francis Crick, is an explanation of the genetic information flow of a biological system [58]. Nowadays, the central dogma is often explained as follows: DNA can be copied to DNA as DNA replication, DNA can be copied to mRNA as transcription, and proteins can be synthesised with mRNA as a template as translation. Meanwhile, RNA can also be copied from RNA as RNA replication, and DNA can be synthesised with RNA templates as reverse transcription. A single-stranded nucleotide structure molecule, RNA, is the transcript product of DNA. Ribonucleic acid contains four types of nitrogenous bases: A, C, G, uracil (U). Human genomes contain both protein-coding genes that can be transcribed into mRNA and then translated into proteins and non-coding regions that may have regulatory or structuring functions. The proportion of coding regions only takes approximately 1% of the whole human genome, and a vast part of the genome remains as a non-coding region [59].

3.1.2 Human genomic variants

Human genomic variants are the DNA sequence differences among human individuals or between groups of samples such as normal and tumour tissues. These variants can be grouped into different categories based on different aspects.

Based on the forms of DNA sequences changes, genomic variants can be classified as substitutions, deletions, insertions, duplications, inversions, translocations, or complex variants [6,60,61]. A substitution is either a single nucleotide is substituted as a single nucleotide variant (SNV), or several nucleotides are substituted as a multi-nucleotides variant (MNV). An insertion indicates at least one nucleotide is added and a deletion indicates at least one nucleotide is removed. Insertions and deletions are often discussed together as indels. A duplication indicates that at least one nucleotide is duplicated. It can be duplicated adjacently as a tandem duplication or duplicated several bases away as an interspersed duplication. An inversion indicates that the orientation of a part of a DNA sequence is inverted. A translocation indicates that a part of a DNA sequence is rearranged to other location (Figure 1). If the form of a variant is too complicated and cannot be described as one of the above basic variant types, it can be categorised as a complex variant.

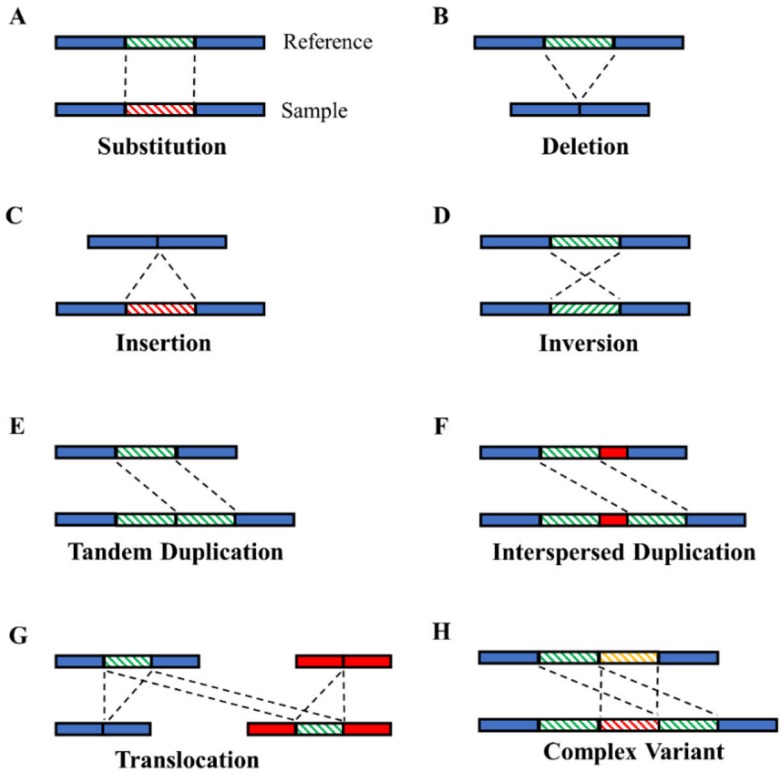

Figure 1. Different types of genomic variants. The first line of each variant type in A to H represents the reference sequence, and the last line represents the sample genome sequence. Dash lines indicate the changes between the reference and sample sequences. The slash-coloured blocks indicate the changeed part of the genome.

Besides, genomic variants can be classified based on their sizes. An indel usually indicates an insertion or deletion with size smaller than 50 bp, but it also can be a large insertion or deletion with size larger than 50 bp [7,62–64]. Genome changes, including indels, duplication, inversion, and other types of variants with sizes larger than 50 bp, also be called as SVs [65]. Copy number variations usually indicate intermediate-scale genome changes. The size ranges of copy number variations differ across publications. The common used definition is that a copy number variation should have at least 1 kilobase [66–72]. However, some studies may define them as genome changes larger than 50 bp [73–75] and consider them a sub-type of SVs.

In terms of origins of genomic variant, they can be classified as germline variants and somatic variants. Germline variant are mutations occuring in gametes, and these are transmitted to the offspring and cause every cell of the offspring to contain these mutations. Somatic variants are mutations occur in other cells of the body that are only confined to only one cell and its progeny cells.

The functional effects of variants on genome can be also used for variant classification. As for genomic variants in coding regions, a silent variant is a substitution which the change of a single nucleotide in a protein-coding region of a gene causes difference in the transcribed mRNA but does not affect the final translated protein sequence. A missense variant is a substitution that causes the final translated protein has different sequence between the original one. A nonsense variant is a substitution that results in a premature stop codon and causes a shorter, unfinished protein sequence. A frameshift variant is an indel that changes the reading frame of an mRNA. Genomic variants that located outside of coding regions can be called as non-coding variants.

In addition, in terms of the allele frequency, a SNP indicates a variant with an allele frequency > 1% in the population [76]. A common variant can be defined as a variant has an allele frequency > 5% in the population and a rare variant has an allele frequencies < 1% in the population [77].

3.1.3 Insertions and deletions in the human genome

Indel represents the genome variant types of insertion (gain nucleotides) and deletion (lose nucleotides). Studies [78,79] have showed that indels are the second most common genomic variant type in human genomes after SNPs, which account for approximately 15%–20% of all variants; among these, single-base indels represent one-third. The majority indels of a human individual are small sizes (10 bp or smaller), and the larger they are, the rarer they appear in a human genome (**Publication I**) [80,81]. Insertions and deletions are almost equally distributed in a

human genome, with the number of deletions slightly more than the number of insertions [79].

In the human genome, several spontaneous molecular mechanisms can lead to an indel. The most common molecular mechanism is DNA strand slippage, which explains three-fourths of all the indels of a human individual [82] (Figure 2). *In vitro* research has shown that all studied DNA polymerases can generate indels via strand slippage [83]. During the DNA synthesis process, DNA strand slippage may occur on either the primer or template strand as primer or template slippage, respectively, and generate a misaligned intermediate that contains one or more unpaired nucleotides. These misaligned intermediates can spontaneously realign or undergo other processes, including proofreading and mismatch repair, to fix these unpaired nucleotides and generate an accurate synthesis. However, if these unpaired nucleotides escape proofreading and mismatch repair, further synthesis will occur on these incorrect DNA sequences and result in insertions or deletions from primer or template slippage, respectively. The frequency of DNA strand slippage, also referred to as the indel error rate, varies widely among polymerases and typically occurs more frequently in repeat sequences such as homopolymeric sequences.

Previous research has shown that the indel error rate increases with a longer length of repeat sequences and decreases with a larger motif size of repeat sequences [84]. As the repeat length increases, the strand misalignment is located further away from the primer terminus where the DNA polymerase facilitates the extension of the DNA replication complex; thus, the proofreading efficiency of polymerases usually diminishes, and indel error rates increase [85]. Because the proofreading function efficiency decreases with an increase in repeat sequence length, DNA mismatch repair, which is a system consisting of several mismatch repair proteins, is an important post-replication function to correct indel errors in DNA replications. Previous research has shown that inactivated DNA mismatch repair could increase spontaneous indel error rates by 10,000-fold in repeat DNA sequences [86,87]. The loss of DNA mismatch repair in humans leads to microsatellite instability, a phenotype that is often observed and used as a biomarker for the diagnosis and prognosis of colorectal cancer [88]. Microsatellite instability refers to the high indel error rates in abundant microsatellites of human genomes, indicating a high damage level of DNA mismatch repair system and, thus, a high rate of somatic mutation in the development of cancer [89,90].

Besides DNA polymerase slippage, indels or SVs may also arise through the cellular repair of DNA structural changes such as double-stranded break, which may be caused by ionizing irradiation, metabolic by-products, or recurrent re-arrangements [91–93]. Homologous recombination and non-homologous recombination are the two general mechanisms which causes changes in the structure of DNA sequences [94]. Homologous recombination requires extensive DNA

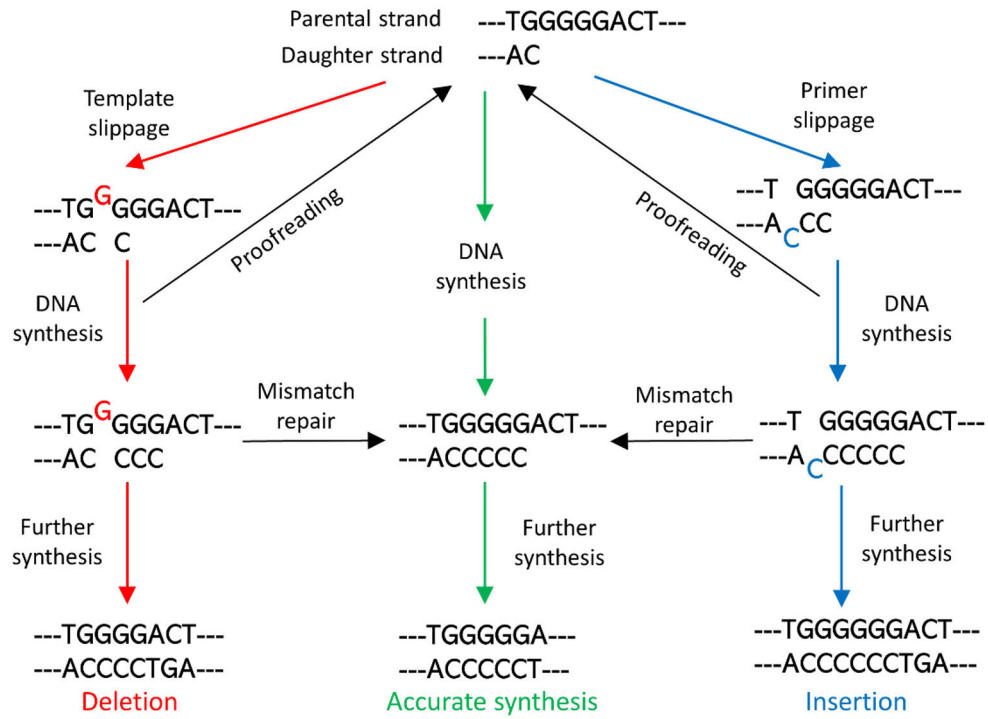

Figure 2. Indel formations by DNA strand slippage. Once DNA strand slippage occurs, a misaligned intermediate with unpaired nucleotides is generated. The misaligned intermediates may undergo proofreading or mismatch repair depending on synthesis processes and fix the misalignment. If this mislalignment escapes all the possible repair processes, it will result in a deletion or insertion for template and primer slippage, respectively. (Figure altered from [96])

sequence identity and a strand exchange protein (Rad51 in eukaryotes) to repair two-ended and broken replication forks [95]. If the damaged DNA sequence is repaired with homologous sequence in the same chromosomal position of the sister chromatid or the homologous chromosome, there will be no change in DNA structure. But if the repair process utilizes homologous sequences in different chromosomal position as non-allelic homologous recombination, duplication and deletion can be formed [97,98]. Single-strand annealing is another mechanism of break repair that acts at directly repeated sequences such as Alu and results in small deletions [99].

There are also mechanisms that repair broken DNA sequence by using very limited or no homologous sequences. Non-homologous end joining is one mechanism which is active in all cell cycle phases except mitosis and can re-join double-stranded break ends either accurately, or with small indels [100]. Microhomology-mediated end joining is another mechanism which requires 5–25 bp homologies sequence at the ends of double-stranded break and can leads to deletions

that are larger than non-homologous end joining indels but still relatively small (< 30 bp) [101].

In addition, one sub-class of retrotransposons, the non-long terminal repeat retrotransposons (includes the long interspersed nuclear element-1), is currently active in human genomes and can generate DNA insertions through RNA intermediates mechanism. A large number of insertions generated by retrotransposition events have been indicated as disease-causing variants in the human genome [102].

Many indels have no effects on the function of the human genome and are located in intergenic regions or the non-coding components of genes. However, a large proportion of indels in the human genome is located within the known genes, promoters, or exons of these gene, whose gene functions are considered to be affected significantly [79]. In coding regions, if an indel causes nucleotide changes in numbers that are not multiples of three, the indel is called as a frameshift indel and could disrupt the normal reading frame and then result in the incorrect reading of the entire gene sequence. The frameshift indel has a significant impact on the protein because it will generate a completely different amino acid sequence or create a premature stop codon that prevents the protein sequence from growing. For example, a single nucleotide duplication occurring at the *TNNC1* gene will cause hypertrophic cardiomyopathy and sudden death [103]. If the nucleotide change of an insertion or deletion is three or a multiple of three, then one or more codons will be removed or added. If the codon changes are located at important protein regions, such as the active site of an enzyme or an essential secondary structure of a protein, the indel may also have a significant influence. For example, cystic fibrosis, one of the most common human genetic diseases, is caused by a 3 bp deletion in the *CFTR* gene and leads to the removal of a single amino acid of the encoded protein. This single amino acid removal will cause an abnormal fold of protein and result in the degradation of protein, thus leading to the disease [104]. For indels located in regulatory regions, they might lead to mutated genes being expressed in the wrong tissues or overexpressed in the cell cycle, leading to uncontrolled cell division and hence to cancer [105]. For example, a 20 kb deletion located in the upstream regulatory region of the *IRGM* gene shows evidence for affecting gene expression, which modulates biological processes and causes Crohn's disease [106]. In addition, large indels may alter the copy numbers of a gene in the genome and affect gene dosage. For example, the copy number changes in an active cytochrome P450 *CYP2D* gene will cause the ultrarapid metabolism of debrisoquine; the copy number of the *AMY1* gene is positively correlated with the starch content of the diets of several populations around the world [107,108].

As described in the previous section 3.1.2, the size definitions for indels are not very clear. Because of the technical limitations, for example, the short reads from

NGS cannot cover large genome variants; some studies, such as method evaluations, novel algorithm developments, or benchmarking dataset validations, had to focus on indels with short size ranges, typically 1-50 bp [63,109,110]. Other studies may use specific methods to focus on large genome variants and treat indel as a sub-type of SVs or copy number variation [111,112]. Thus, current studies may classify indels into two artificial categories based on their sizes. **Publication I** used both simulated and real data that contained indels with a wide size range to evaluate tool performance.

3.1.4 Influence of the sequence context of genomic variants

The previous section 3.1.3 has shown that indels are enriched in STRs. This indicates that the sequence context of genomic variants has an important impact on the functions of human genomes. As an important sequence context feature of genomic variants, especially indels, STR is also an essential component of the human genome. For example, proofreading of polymerase for indel mismatches in non-repeats or STRs are as efficiently as that for base-base mismatches. However, proofreading of polymerase is less efficiently in long repeat sequences than non-repeats or STRs [86,113]. Similarly, the rate of non-allelic homologous recombination positively correlates with repeat length, GC content, and DNA sequence identity, while it is inversely correlated with the distance between different repeats [114,115]. Indels that are located in STRs and that change STR copy numbers are known to be important for biological human health functions and are the reasons for some human genetic disorders. For example, Huntington's disease is caused by an expansion in the *HTT* gene, which contains a CAG trinucleotide repeat and encodes an extended polyglutamine tract in the huntingtin protein [116,117]. A copy number increase of a CGG trinucleotide repeat in the 5' untranslated region of the *FMR1* gene can cause fragile X syndrome by silencing the gene [118].

As described in the previous section, microsatellite instability can be seen as a reflection of the impaired DNA repair system and is used as a biomarker in cancer diagnosis and treatment [119–123]. Another study showed that frame-shifting indels in STRs can be tolerated by transcriptional and translational processes due the highly similar nucleotide pattern in their sequence contexts [124]. Moreover, the sequence contexts, such as specific nucleotide features or patterns, can also influence the mutation rate of genomic variants. For example, several studies have demonstrated that (C+G)-rich trinucleotides have higher mutation rates than other types of trinucleotides [125,126]. Study [127] has shown that the mutability of a variant is jointly affected by nearby nucleotide patterns and the genomic features of the surrounding sequence contexts, such as GC content and histone modifications. The results of [128] established a mutability model and illustrated that the mutation rates

of sequences varied significantly between different trinucleotides in human genomes.

Aside from STRs, variants located in nearby sequence contexts that are close to one another may also have a joint impact. One study has illustrated that the existence of nearby indels can increase the single nucleotide mutation rate, and another study has demonstrated that common indels had a strong linkage with nearby SNPs and can be generally well tagged according to it, which makes it possible to assess indels into human haplotype reference panels and use them as markers for genome-wide association studies [82,129]. Another study has identified 1135 genetic hotspot clusters with high variant density, which were highly associated with tumour suppressor genes and oncogenes [130]. Moreover, indels rarely occur as novel DNA patterns because of the mechanism of indel mutations, such as DNA polymerase slippage. Indels usually occur as patterns of multiple repeat motifs in a sequence context [82]. Indels with novel DNA sequences can be unique mutation events at certain genome positions across human evolutionary history and different populations [4]. Some studies have demonstrated that indels with novel DNA sequences might occur at the same position in the genome. These multi-allelic indels are informative about human evolution and migration. Other studies have shown that within a short distance to the indel hotspots, some secondary indels can occur, resulting in a multi-indel locus. These loci can be used as useful markers to improve discrimination for forensic purposes. [4,131,132].

The sequence context of variants not only has biological effects on human health but may also result in technical issues, such as bioinformatics analysis. [133] demonstrated that the repeated sequence around indels causes positional and breakpoint ambiguity in identifying the accurate positions of indels. Positional ambiguity is caused by high similar nucleotide blocks, where a portion of a read can align to more than one certain region in the reference sequence (Figure 3A). This issue makes it difficult to know the exact size of an indel but might be solved by applying sequencing methods with sufficient long reads. The breakpoint ambiguity caused by the microhomologies surrounding the indel breakpoint makes it difficult to identify the exact breakpoint position of the indel (Figure 3B). The study revealed that 40% of deletions with sizes larger than 32 bp could not be identified with certain sizes or positions by pairwise alignments of sequencing with a 100 bp read length. In addition, the STRs around an indel may cause the indel to have different representations and lead to biologically equivalent but redundant indels in the database. [134] illustrated that the sequence context of an indel, such as STR, may create an equivalent indel region, where an indel can have multiple biological equivalent representations in a continuous genome region. The equivalent indel region of an indel can make sequencing read to align to a region instead of a single site. Thus, it lowers the read depth at the indel site and makes the indel filtered in

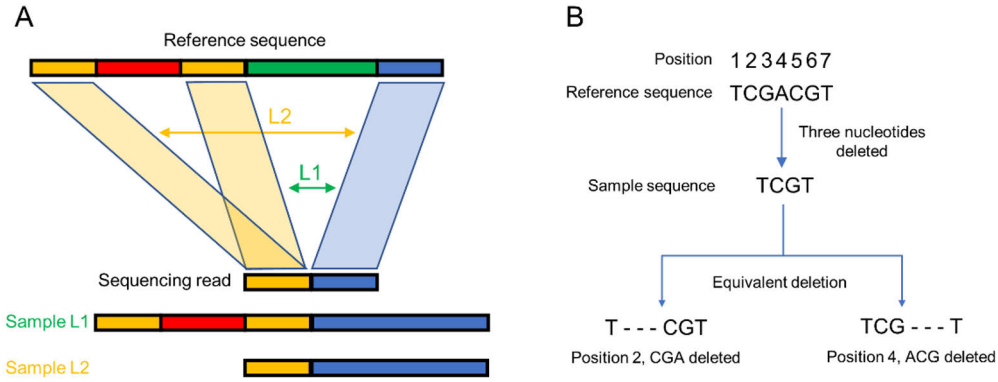

Figure 3. **(A)** Positional ambiguity. The top coloured line indicates the reference, the second small coloured line indicates the sequencing read, and the two bottom coloured lines indicate the samples. The shadow-coloured grams indicate the mapping possibility of the read. There are two orange blocks in the reference sequence, which makes the 5' of the sequencing read mapped to two locations. This positional ambiguity will result in two different sizes deletions, L1 and L2, in the sample sequence with the same read. (Altered from [133]). **(B)** Breakpoint ambiguity. Compared with the reference sequence, three nucleotides were deleted in the sample sequence. Because the nucleotide pattern 'CG' existed in two locations, this deletion can be represented as two equivalent deletions in difference positions in the reference sequence coordinates.

downstream computational analysis due to the low read depth. The authors emphasised the unambiguous annotation of an indel and suggested that databases, such as dbSNP [135], should annotate an indel with the equivalent indel region depending on its sequence context, which is a supplement to the single coordinate. [136] demonstrated that STR and breakpoint ambiguity were main factors that causing the low recall of insertions with short-read based variant calling tools. [137] investigated the breakpoint ambiguity of indels and developed a universal positioning system for annotating indels. With their annotation system, the authors found that 15% of indels in the dbSNP database and 29% of indels in the COSMIC database were redundant [138]. [139] developed a computational algorithm to recognise indels that had breakpoint ambiguity and may need to be normalised into unambiguous and concise representations. With the algorithm, they found that 14.9% of indels in the dbSNP database needed to be normalised. In addition, the variant nomenclature recommended by the Human Genome Variation Society (HGVS) has the 3' rule for variant representation, which means that variants should be represented at the most 3' possible aligned position of the reference sequence. However, in high-throughput genomic sequencing data analysis, variants are usually represented at the most 5' possible aligned position. For a breakpoint ambiguous variant, especially a breakpoint ambiguous indel, the 5' aligned position and 3' aligned might be different and further cause troubles for the conversion of variant

nomenclature [61]. In my **Publication I**, the results showed that nearly half of FP indel calls with different indel calling algorithms were located in simple repeats, indicating that the low complexity and highly diverse sequence context might be related to the FP results. Because of the complexity and biological significance of the sequence context of genomic variants, in **Publication II**, a novel computational tool named VarSCAT was developed with the aim of comprehensively annotating the sequence context of genomic variants, such as breakpoint ambiguity, flanking sequences, conversion of HGVS nomenclature, and tandem repeats in a high-throughput way.

3.1.5 LncRNAs in the human genome

Transcription is the process of passing genetic information from DNAs to RNAs. For the most part, the transcription process is carried out by RNA polymerase II, which transcribes DNAs to mRNAs, and then mRNAs can be translated into proteins [140]. The C-value paradox during the 1950s troubled scientists; that is, less complex animals, such as salamander, can have a genome that is 15 times larger than that of more complex animals, such as humans [141,142]. Later, the paradox can be explained that not all DNA can be translated into proteins, most of their DNA is non-coding and thus, the genome size cannot reflect gene number. During the 1970s, it was estimated that the human genome was unlikely to have more than 30,000 genes [143,144]. Non-coding DNA was first treated as junk DNA, which accounts for 50%–70% of the human genome as transposons, pseudogenes and simple repeats [145]. During the 1950s, the discovery of transfer RNAs that were transcribed by RNA polymerase III and ribosomal RNAs that were transcribed by RNA polymerase I provided evidence that a significant number of non-coding DNAs can be transcribed into non-coding RNAs, which had important biological functions [146,147]. In the 1980s, more evidence showed that non-coding RNAs, including small nuclear RNAs and telomerase RNAs, were involved in complex biological processes, such as protein expression and genome regulation [148,149]. With the help of sequencing technology, it has been estimated that around 70%–90% of the human genome is transcribed at some point during development [150–153]. The GENCODE project aimed to deliver a definitive annotation of functional elements in human and mouse genomes via manual curation, computational analysis, and targeted experimental approaches. Current statistics from GENCODE showed that only one-third of the total genes (19,393 out of 62,703) in the human genome are protein-coding genes (GENCODE Release 43) [13].

Different types of non-coding RNAs differ greatly from one another in biogenesis and molecular functions. Basically, they can be classified as infrastructural and regulatory non-coding RNAs. Infrastructural non-coding RNAs

include transfer RNAs, ribosomal RNAs, small nuclear RNAs, and small nucleolar RNAs. These infrastructural non-coding RNAs, which are mainly involved in protein synthesis, have regulation functions for recognising and interacting with sequence-specific RNA substrates in the translation and splicing process [154]. Others may play essential roles in chromosome maintenance; for example, a small nucleolar RNA *box H-ACA* is a component of telomerase, which has a function for extending telomeres [155].

Regulatory non-coding RNAs mainly base pair with other DNAs, RNAs, or proteins to form complexes, and they have functions in regulating biological processes. For example, the RNA-induced silencing complex uses a single strand of RNAs as a guide strand to recognise and cleave mRNAs. The process is known as RNA interference, which reduces the levels of transcripts available to be translated by ribosomes [156]. Conventionally, regulatory non-coding RNAs are classified into small non-coding or lncRNAs based on their sizes. The sizes of lncRNAs are considered to be more than 200 nucleotides, and non-coding RNAs smaller than this are termed small non-coding RNAs [157–159]. Some studies have attempted to distinguish lncRNAs based on their biological functions instead of arbitrary thresholds in size. Therefore, some lncRNAs may not necessarily exceed 200 nucleotides. For example, human transcripts hsa-mir-423 and FLJ13453 are marked as lncRNAs in the RNAcentral database, with sizes of 94 and 111 nucleotides, respectively [160]. In addition, some lncRNAs may have functions as infrastructural non-coding RNAs, and some of these RNAs, such as small nuclear RNAs, may have functions in regulating biological processes, which makes the classification of non-coding RNAs not absolute [161,162].

Current lncRNA classifications are usually based on their genomic context with respect to protein-coding genes. The classification proposed by the GENCODE database is one of the most frequently used. According to GENCODE, lncRNAs can be roughly grouped into five categories: 1) antisense RNAs, which are located on the opposite strand of a protein-coding gene and overlap with any exon; 2) long intergenic non-coding RNAs, which are located in the intergenic sequence space and do not overlap any protein-coding genes; 3) sense overlapping transcripts, which are located on the same strand of a protein-coding gene and contain its introns; 4) sense intronic transcripts, which are located in the introns of a coding gene and do not overlap with any exons; and 5) processed transcripts, which do not contain an ORF and cannot be classified into any of the above categories [163]. The ORF consists of a set of consecutive non-overlapping codons that can be translated into a protein. Some studies may also list pseudogenes and divergent transcripts as other categories. A pseudogene has homology to a protein-coding gene but has a disrupted coding sequence [164]. Divergent lncRNAs are transcribed in the opposite direction to nearby protein-coding genes from

bidirectional promoters [165]. In addition, lncRNAs can be grouped as linear RNAs and circular RNAs based on their structures. The above classification of lncRNAs based on genomic context is mainly for linear RNAs [166]. Circular RNAs are formed by the back splicing of pre-mRNAs, in which an upstream acceptor is merged with a downstream donor. Circular RNAs may overlap with introns, exons or flanking regions of protein-coding genes and have been shown to have major gene regulation roles in complex diseases such as lung cancer [167]. Moreover, lncRNAs can be grouped based on functions either in the nucleus or in the cytoplasm for regulating transcriptional or post-transcriptional events, respectively [168,169].

Many lncRNA genes are located far away from protein-coding genes or are expressed from enhancers [170,171]. Long non-coding RNAs are mainly transcribed by RNA polymerase II, and they undergo post-transcriptional processing events, including 5' -capping, splicing, polyadenylation and chemical base modification [172]. LncRNAs are similar to protein-coding mRNAs but lack translated ORFs and have poorer primary sequence conservation. Usually, they have shorter ORFs with fewer but longer exons than protein-coding mRNAs. However, some lncRNAs may have long ORFs, whereas some mRNAs may have short ORFs that code for short peptides [173,174]. Because many lncRNAs have lower expression levels than mRNAs, they have been thought for a long time to be only transcriptional noise. Transcriptome-wide studies have shown that lncRNAs have specific expression profiles among different cell types, tissues, developmental stages or disease states [163,175]. Many tissues that express lncRNAs can be found, and the brain and central nervous system have the highest diversity of lncRNA expression [176].The expression profiles of lncRNAs often show correlation with mRNA expression profiles, indicating that certain lncRNAs may be co-regulated in expression networks [177].

The molecular mechanism of lncRNAs can be grouped into four main categories: signals, decoys, guides and scaffolds [178]. As signals, lncRNAs can affect the signalling pathways of gene regulation at specific times to respond to diverse stimuli or at specific places for different developmental stages. The use of lncRNAs as mediating molecules to fulfil regulatory functions can be quickly performed and can avoid translation processes for protein expression. For example, lncRNA Xist plays an essential role in X inactivation, which is a process in mammalian female cells involving the inactivates of one paternal X chromosome to equalise the gene expression between males and females. During female development, Xist RNA is expressed from the inactive X chromosome and coats this X chromosome to repress the expression of most genes. Tsix RNA, which is an overlapping antisense lncRNA of Xist, represses Xist expression in *cis* and plays a role in the active X chromosome [179]. As a decoy, the lncRNA binds to a gene

regulator and prevents the effector molecule from binding to it, which negatively regulates the effector function in the neighbouring gene expression. For example, the human *DHFR* gene has a major promoter for initiating mRNAs and a minor upstream promoter for initiating lncRNAs. The transcribed lncRNAs can form a stable lncRNA-DNA complex with the major promoter sequences and directly interact with the general transcription factor IIB to inhibit the assembly of the preinitiation complex, thus repressing the gene expression [180]. As guides, lncRNAs interact with protein and change the gene expression either in *cis* (on nearby genes) or in *trans* (genes with distances). For example, an lncRNA RepA originating from the 5' end of Xist interacts with polycomb repressive complex 2 in *cis*, which plays an important role in the creation of a heterochromatic state of inactive X chromosome [181,182]. Another example is LincRNA-p21, which has an effect in *trans* on chromatin structure and gene expression across chromosomes [183]. As scaffolds, lncRNAs provide central platforms to support the molecular components that are assembled. For example, telomerase RNA possesses structures that contribute to the catalytic activity of the telomerase reverse transcriptase protein for extending telomeres [184].

3.2 Development of sequencing technologies

To reveal the sequence of the human genome, efforts have been made more than 50 years ago, which also reflect the developmental history of sequencing technology. Sequencing is a laboratory process that determines the exact order of nucleotides in a small region, such as a gene, or even a large region, such as a whole genome. DNA sequencing is the process of determining the order of nucleotides of DNA sequences, whereas RNA sequencing can examine the quantity and sequence order of RNA sequences.

3.2.1 The early efforts of sequencing the human genome

In the history of DNA sequencing technology developments, there have been several milestone techniques. The first one is first-generation sequencing with the representative method of chain-termination, which was developed by Frederick Sanger, and the chemical cleavage procedure, which was developed by Maxam and Gilbert [185,186]. Both methods produce DNA fragments of different sizes in four reactions, then order and visualise the DNA fragments based on their sizes by using gel electrophoresis [187]. The order of the DNA fragments indicates the order of the four nucleotides, which is the sequence order of the sample DNA (Figure 4).

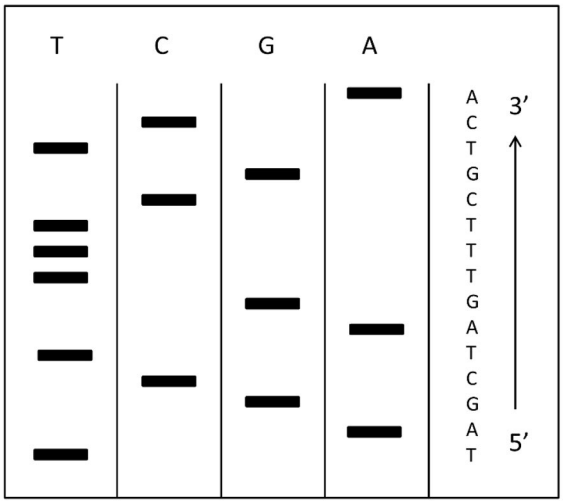

Figure 4. An example of Sanger sequencing. The DNA fragments on each panel are terminated with a certain type of nucleotide, and shorter fragments travel longer on the gel because of their lighter molecular weight. The order of DNA fragments indicates the order of their terminated nucleotides, which further indicates the DNA sequence.

With the experimental concepts and techniques of first-generation sequencing, shotgun sequencing was established for *de novo* assembly genomes. In shotgun sequencing, the sample genome is broken randomly into numerous small DNA fragments and are sequenced individually using the Sanger sequencing method [188]. The sequence of the bases of a small DNA fragment is called a read. Computational analysis is applied to assembles sequencing reads s into a continuous sequence as the genome [189].

The Human Genome Project, which was a publicly funded international scientific research project led by the US National Institutes of Health with the goal of determining the sequence of the human genome was initiated in 1990. In the Human Genome Project, the hierarchical shotgun sequencing was applied to sequence the genomes of human donors [190]. Private funded companies led by Craig Venter and Celera Genomics also joined the competition for sequencing the human genome. Celera used the whole genome shotgun sequencing method, in which the entire genome is sheared randomly into small fragments for sequencing and then reassembled using a computer programme [191]. With the efforts of both public and private funded projects, the sequence of the human genome was first released as a draft in 2001 and then as a finished euchromatic sequence in 2004 [32,191,192].

3.2.2 Next-generation sequencing

Massively parallel deep DNA sequencing, also known as NGS, became commercially available during the Human Genome Project [193]. Compared with capillary electrophoresis-based sequencing methods, multiplexing NGS is the key change. In NGS processes, a complex library of DNA templates is prepared by randomly fragmenting the sample genome. With or without polymerase chain reaction (PCR) as the amplification option, these short DNA templates are ligated with adapters and immobilised onto a 2D surface [194,195]. *In vitro* amplification generates adequate copies of DNA templates to be sequenced. Detectors such as imaging techniques together with, for example, fluorescently labelled nucleotides, determine the types of bases by detecting biochemistry signals from DNA template sequencing [196].

In vitro amplification facilitates millions of DNA template sequencing in parallel by ensuring thousands of identical copies of a DNA template as a cluster located in a pre-known area on the flow cell; the signals from millions of individual reactions of each cluster can be distinguished from background noise [197]. Since the first NGS platform became commercial in 2005, intense competition in this field has started, resulting in the rapidly development of sequencing instruments and platforms [198]. During this time, several platforms, such as 454, SOLiD, Ion Torrent, and Illumina, have been developed, matured, marketed, and applied to various scientific projects [198–202]. Since 2012, the pace of improvement has slowed down, and Illumina has become the dominant commercial platform in the sequencing market, offering various scalable sequencing options with a fair financial cost [203]. Illumina platforms apply a sequencing-by-synthesis strategy, which uses terminator molecules to prevent elongation of DNA fragments. After DNA templates are amplified on the solid surface to form clusters, a mixture that contains DNA polymerases, primers for initiating polymerase binding, and all four base-specific fluorophore-labelled elongation blocking nucleotides are added. During each cycle, this mixture is added to the solid surface to incorporate with DNA fragments, so identical DNA fragment clusters can only be elongated by one type of nucleotide for each cycle and emit base-specific fluorescence. After incorporation, unincorporated nucleotides are removed, and the solid surface is imaged by an image-capturing device to identify which nucleotide is incorporated in each cluster by recognising the fluorescence emission spectrum. After the removal of the fluorophore and blocking group on the DNA fragments, a new cycle can begin again to identify the types of bases on the next positions of the DNA fragments [200].

Based on the covered regions of the genome, NGS experiments can be classified as whole-genome sequencing (WGS), which captures whole regions of the genome; whole-exome sequencing (WES), which captures the whole regions of the exome; and targeted gene panel sequencing, which uses a gene panel to capture certain clinically relevant or other interested genomic regions. The cost of WGS per sample

is higher than that of WES and the cost of WGS per base is cheaper than that of WES [18]. Because of cost, in practice, the sequencing coverage of WGS (typically 30–50×) is usually lower than that of WES (typically 100–300×), and the sequencing coverage of WES is usually lower than that of targeted gene panel sequencing (typically over 1,000×). Moreover, WGS helps to detect larger genomic variations, less sensitive to GC content and has to a more uniform coverage than WES [204]. Targeted gene panel sequencing is commonly used for clinical diagnosis, which includes the majority of known disease-causing genes and facilitates rapid identification with simpler deployment and lower costs [205]. Although the targeted gene panel NGS can test multiple genes simultaneously and replace Sanger sequencing in laboratory testing processes, Sanger sequencing is still needed to analyse regions where NGS has difficulties obtaining sufficient sequencing coverage and good-quality data. Sanger sequencing is also used as the gold standard to confirm variants from NGS analysis before they are clinically reported. With the maturity of NGS development, some studies have shown that Sanger sequencing confirmation is not necessary in clinical practice. However, the data from NGS must meet a high quality threshold and specific regions should be carefully considered [206,207].

Many NGS sequencing library preparation kits, especially Illumina, have the option of generating paired-end reads. In paired-end sequencing, each read has a pair read, which offers more information about the genome structure for downstream bioinformatics data analysis algorithms (Figure 5). Paired-end sequencing can produce twice the number of reads in library preparation compared with single-end sequencing, and sequence alignment with read pairs enables more accurate alignments. Paired-end DNA sequencing has a good ability to detect common DNA rearrangements, such as SVs, and RNA paired-end sequencing can benefit gene fusion detection in cancer and novel splice isoforms discoveries [20,208]. Despite the benefit of using paired-end sequencing, single-end sequencing, which involves simply sequencing DNA from only one end, has its own application in studies of small RNA-seq or chromatin immunoprecipitation sequencing with a fast and economical option. Today, paired-end sequencing is the most popular approach in human genome studies.

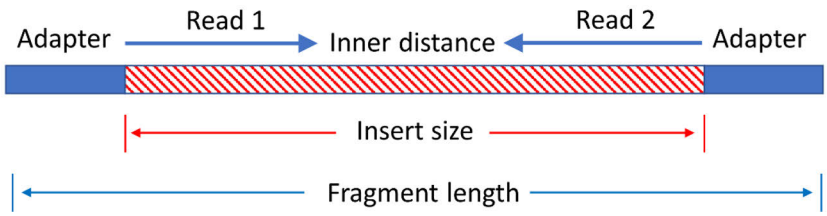

Figure 5. A demonstration of paired-end sequencing. The fragment is the sequencing molecule that is hybridised with the oligonucleotide on the surface of the flowcell. It contains the adapters with all the sequencing sections and the DNA inserted from the sample. Reads 1 and 2 are sequenced from different directions, and the distance between two reads is the inner distance.

Although different NGS platforms have their own advantages, disadvantages and technical foci, error rates are among the most important factors that help researchers choose a suitable sequencing platform for their experimental designs. The error rate comparison among different NGS platforms has been a research focus since the sequencing technique was developed. Although the NGS platform has been developed and improved through the years, many studies have been conducted to survey error rates, and generally, the error rates of Illumina platforms are around 0.1%–0.5% [209–214]. One large database analysis [214], which was published in 2021, examined 1,943 different datasets from seven Illumina sequencing platforms. The results showed that the more expensive platforms, such as Hiseq, had lower error rates at around 0.1%, and error rates were associated with sequence context where bases towards the same type. Some studies have also demonstrated that particular library preparation methods or computational analysis procedures can dramatically suppress sequencing error rates [213,215].

3.2.3 Third-generation sequencing

In the human genome, many complex elements, such as large indels and tandem repeats, play important roles and are relevant to evolution and diseases [216,217]. The sizes of these complex elements are so large that the short reads of NGS, typically up to hundreds of base pairs, cannot fully cover them. In addition, the amplification steps in NGS may introduce sequencing errors or sequence-dependent biases, thus limiting the accuracy of sequencing. To solve these problems, efforts have been made to develop other sequencing approaches than NGS, such as producing single continuous reads to cover a large genome region. Nowadays, long-read sequencing platforms, also known as third-generation sequencing, can produce long reads without amplified DNA fragments and have become increasingly important in the field. Currently, there are two widely used long-read sequencing platforms: the single-molecule, real-time sequencing platform developed by PacBio and the platform developed by Oxford Nanopore Technologies.

The PacBio sequencing platform uses a specialised flow cell containing thousands of individual wells with a zero-mode waveguide at the bottom of each well to conduct sequencing. The DNA template fragment is ligated to hairpin adapters at each end, forming a circular structure with single-stranded DNA at each end and a double-stranded DNA template in the middle. Primers, fluorophore-labelled nucleotides, and a modified DNA polymerase are added to each zero-mode waveguide. The DNA polymerase is fixed at the bottom of the zero-mode waveguide, and the DNA strand grows through it in a constant location. The fluorophore-labelled nucleotides are incorporated into each single-molecule DNA template as an elongation process, and the emitted colour signals are captured by a

laser and a camera system to visualise sequencing. There are two sequencing modes for the PacBio platform. One is the circular consensus sequencing mode, which produces highly accurate long reads by sequencing templates multiple times. Another is the continuous long read sequencing mode, which can generate the longest possible reads; over 10 kb is typical and some even approaching 100 kb [218,219].

The Oxford Nanopore Technologies sequencing platform detects the DNA composition of a single-stranded DNA fragment about 8–10 kb in length without a secondary signal, such as a fluorescence spectrum or pH change. In the electric field-driven sequencing process, the leader sequence with the adapter directs the DNA fragment to the nanometre-scale protein pore with the current passing through. As the DNA fragment is translocated through the pore, a shift in voltage through can be observed and interpreted as a particular *k*-mer sequence. The platform has thousands of possible signals for each *k*-mer. When the next base passes through the pore, a new voltage shift for the *k*-mer is identified. The hairpin structure of the DNA fragment and adapters allows the forward and reverse strands to be sequenced in order to create a consensus sequence [220,221].

The sequencing error rate for the long-read sequencing platform is relatively high (85%–90% accuracy for PacBio platforms and 96%–99% for Oxford Nanopore Technologies platforms) compared with NGS platforms when this thesis was written, which may limit downstream analysis, such as genome assembly and variant calling. However, efforts have been made to improve sequencing quality; for example, the circular consensus sequence approach, which is branded by PacBio as HiFi sequencing, can produce reads with an accuracy of > 99.9% [222,223]. These initiatives have made the application of long-read sequencing widely used in scientific research in recent years [224].

3.2.4 RNA sequencing

The early sequencing of RNA was not an easy task. The first whole nucleic acid sequence, which consists of alanine transfer RNA containing 76 nucleotides, was produced by Robert Holley and colleagues in 1965. In their sequencing method, an RNA molecule was first cut by RNase enzymes at specific sites that were already known and available, and then the fragments of RNA were separated by chromatography and electrophoresis. The nucleotides of each fragment were determined using sequential exonuclease digestion, and the sequence of the RNA molecule was deduced by overlapping each fragment [225]. However, the single-stranded structure of RNA is very unstable and easily degraded in cells, which limits the accurate sequencing of larger transcripts. With the discovery of reverse transcriptase, single-stranded RNA molecules can be converted into double-stranded

DNAs, which are complementary DNAs [226]. Facilitated by the DNA sequencing system and the PCR method, the reverse synthesised DNA made it possible to sequence RNAs by sequencing their complementary DNAs and contributing the birth of RNA-seq technology.

Compared with DNA sequencing, RNA-seq has additional steps. A typical RNA-seq experiment consists of isolating RNA from a cell or tissue population, converting it to complementary DNA, constructing the sequencing library, conducting PCR amplification and sequencing it on an NGS platform [227]. RNA-seq on NGS platforms allows for the analysis of complex samples, which makes RNA-seq cost-effective. RNA-seq libraries can pool multiple indexed samples in a single sequencing lane by introducing unique 6 bp barcodes to each RNA-seq library, which helps identify from which sample the read originated [228]. The expression level of RNA differs in cells. The optimal sequencing depth, which is the number of how many times that a given nucleotide in the genome has been sequenced, is a key factor in the experiment for the precise detection and quantification of transcripts. The suggested sequencing depth for the human genome varies from as few as five million reads per sample for quantifying highly expressed genes, to generally 20–30 million reads per sample for studying differentially expressed genes, and even up to 100 million reads for quantifying precisely low expressed genes and transcripts [227,229,230]. To detect the full sequence diversity of complex sample libraries even including low expressed transcripts, some studies have suggested that the number of reads of NGS resources should be up to 500 million [231].

Currently, the Illumina sequencing platform is the dominant RNA-seq platform in the sequencing marketing. However, NGS platforms have limitations. Short-read sequencing encounters difficulties in sequencing very long transcripts or highly diverse isoform sequences, and PCR amplification of NGS may introduce biased RNA expression levels, requiring additional specific steps to control it. These limitations boost the application of long-read technologies for RNA-seq, which enable the sequencing of whole individual RNA molecules by converting RNAs into complementary DNAs and then sequencing them or directly sequencing RNAs. The third-generation, long reads of sequencing platforms developed by PacBio and Oxford Nanopore Technologies can identify longer and more complete transcripts with isoform diversity, reduce the ambiguity in the mapping of short sequencing reads and simplify *de novo* transcriptome analysis [232–234]. Although long-read RNA sequencing techniques have some clear advantages over short-read RNA sequencing, they also have distinct limitations. Compared with the throughput of short-read sequencing, that of long-read sequencing is lower. This may limit the use of long-read sequencing in differential gene expression studies because of reduced sensitivity, but it may not be necessary for RNA isoform discovery and characterisation [235]. In addition, long RNA molecules may not always be present

as full-length transcripts because of degradation or shearing during sample preparation, which limits the usage of long-read sequencing in full-length transcriptome analysis and requires careful control to produce high-quality samples [230]. Significant efforts and developments have been made to improve long-read RNA sequencing techniques as the current-generation sequencing methods, which have great potentiality for future RNA-seq analysis [222,223].

4 Data analysis of genomic variants and lncRNAs

4.1 Format and quality control of sequencing data

After the sequencing data are collected, which consist of many sequencing reads from the sequencing platform, a typical bioinformatics analysis will be executed. As with any other data analysis, the first step in sequencing data analysis is quality control. The quality of the sequencing data is very important and can have a significant impact on various downstream analyses, such as sequence alignments, variant calling, and gene expression studies. In NGS sequencing data, sequence artefacts, such as base calling errors, small indels generated during the amplification process, reads with poor quality and primer contamination are common issues. Commercial vendors provide quality control pipelines for their sequencing platforms to filter the output, but quality issues that are affected by other factors regardless of the sequencing platform may still exist in the sequencing data. Thus, additional tools and steps are needed for quality control before any downstream analysis is conducted, such as the FastQC package developed by the Babraham Institute bioinformatics group or NGS Quality Control Toolkit [236,237].

4.1.1 Format of sequencing data

The sequencing reads produced by the sequencing platform usually come in FASTQ format, which is designed for sharing the sequence of reads with associated per-base quality scores [238]. For each sequencing read in FASTQ format, there are four lines. The first line starts with '@', which is the identifier of the read with a free format, usually containing information such as sequencing instrument name, run ID, and read length and so on. The second line is the sequence line, which contains the raw sequence letters for the order of nucleotides. Usually, the letters, 'A', 'T', 'G', 'C' or "U" are used to represent different types of nucleotides, and the letter 'N' is used in the large reference genome to represent a sequence gap or unannotated regions. The third line usually begins with a '+' character and is optionally followed by the same sequence identifier as in the first line. Usually, the third line just contains '+' to reduce the file size. The fourth line is the quality line, which contains the

corresponding quality scores per base for the sequence line. During the sequencing process, the light signals recorded by the image-capturing device of a sequencing platform are converted into corresponding nucleotide bases. The certainty or estimated probability of the error of each base call is measured using statistical models based on base information, such as signal intensities from the recorded image and sequencing cycle. The quality score of each base is introduced by the PHRED quality score, which is defined in terms of the estimated probability of error and encoded with a sub-set of ASCII printable characters [238,239]. This step, known as base calling, is usually automatically performed by the sequencing platform itself.

4.1.2 Quality control of sequencing data

The per-base quality score is the most important parameter to check for raw sequencing read quality. A plot that draws boxplots of base quality scores versus sequencing cycles is a common way to visualise base quality. For Illumina NGS platforms, the base quality usually starts out high but drops gradually as the sequencing cycle increases. The reason is that a proportion of reads in the cluster does not grow at the same rate as the others; this will slowly lead to a desynchronisation as the errors accumulate and cause the base quality scores to drop towards the end of the reads. For paired-end read sequencing, it is common for the first end of the read to have higher overall quality scores than the second end, which can be explained by the long fragment length of the sequencing library [240,241]. Despite the low base quality scores towards the end of reads, the overall low or largely varying base quality scores of the reads usually indicate the low quality of DNA samples or library preparation. As this low-quality score may indicate inaccurate sequencing, which may lead to erroneous conclusions from downstream analysis, the common way of dealing with low-quality bases at the end of the read is to trim them off.

The per-base sequence content, which describes the nucleotide distribution, is another useful quality control parameter for WGS or WES but not for amplicon-based or RNA-seq samples. In the sequencing run of good-quality data, the distribution of the four nucleotides should remain relatively stable. Poor-quality sequencing data usually have large fluctuating nucleotide distributions. The per-sequence GC content sequenced can also be used as a quality control parameter. An abnormal GC content percentage, which shows a large deviation from the theoretical distribution, can indicate the contamination of sequencing samples [240]. Furthermore, the technical sequences within the reads, which may indicate adapters and PCR primer contamination, must be to be trimmed before proceeding with any downstream analysis, such as sequence assembly or alignment. Some overrepresented sequences may indicate that the sequence library is contaminated,

or technical duplicates arise from the PCR artefacts, but it could also have biological meaning, especially when analysing small RNA libraries [236]. Users are suggested to have prior knowledge when judging the quality parameters of sequencing data.

Tools such as Trim Galore! [242] and Trimmomatic [243] provide users with functions and different options for trimming low-quality bases from sequencing reads. Furthermore, these tools could detect and trim off potential technically biased sub-sequences, such as adapter or primer sequences. However, sometimes poor-quality data cannot be improved just by trimming. In addition, the read length is usually shorter after trimming, which will lower the sequencing coverage and cause troubles for downstream analysis. In practice, the sequencing data should be examined again by quality control after trimming through quality control to confirm whether the quality of data meets expectations.

4.2 Building human genomes with sequencing data

The first human genome sequence was published as a draft in 2001 [32]. The built human genome sequence can be used as a reference and the reads generated by sequencing platforms can be aligned with it using different alignment algorithms. With alignment methods, the efficiency of building individual's genome is significantly improved, and variations in sequenced samples can be detected by comparing the differences between aligned reads and the reference sequence [244]. But before the first human genome was published, there was no reference that could be used. The *de novo* assembly method, which is a reference-free method, can assemble reads into short contigs and scaffolds and then merge them into chromosomes and build the sample genome. Studies of species that lack reference sequences can use this strategy to access the genome. Furthermore, human genome studies performed with third-generation, long-read sequencing techniques used *de novo* assembly to build human genome sequences. With long read sequencing data, which can resolve most of the long repeat regions of the human genome, the complete sequence of a human genome was published in early 2022 [245].

4.2.1 Current assemblies of the human reference genome

The human reference genome assembly is a critical resource for biological and clinical research; for example, genomic variants can be represented by the corresponding positions, and transcripts can be mapped to the genome assembly with clinical annotations. The first draft sequence of the euchromatic portion of the human genome was published in 2001 by the International Human Genome Sequencing Consortium [32,191]. Since then, with the collaboration of international research, the draft has been converted into the first finished human genome assembly with

high accuracy and nearly complete coverage, which was published in 2004 [192]. The first finished human genome assembly, also known as Build 35, still contains 341 gaps, regions represented by uncommon alleles, and sequence errors, such as valid deletion alleles or incorrect multi-copy genes [246,247]. Furthermore, this build was designed as simple linear genome sequences, which are insufficient for representing human genome regions with high complexity and structural diversities [248,249].

With the aim of providing a high-quality human reference assembly for biological and clinical research communities, the Genome Reference Consortium (GRC), which is an international consortium with expertise in genome mapping, sequencing, and informatics, was formed to address this issue. In June 2009, the GRC published a major release of the human reference assembly GRCh37. GRCh37 was generating using a hierarchical-based assembly method. The GRC assembly was constructed from sequenced bacterial artificial chromosomes that were ordered and oriented along the human genome. Unlike previous releases, GRCh37 contains three regions with nine alternate locus sequences, which are alternate representations of loci found in a largely haploid assembly to represent highly variable genome regions. The study of GRC demonstrated that the inclusion of alternate representations for genomic loci can improve alignment quality and variation calling with NGS data. In addition, GRC also introduced the concept for "minor assembly" which updates the genome as patches. A genome patch updates the assembly information, which corrects errors in the assembly or adds additional alternate loci without disrupting the chromosome coordinate system [250]. GRCh37 has 13 patch releases until June of 2013, which is GRCh37.p13. In addition, GRCh37 stands for Genome Reference Consortium human build 37 but is commonly nicknamed hg19, which stands for human genome build 37 in the context of the UCSC Genome Browser.

Despite GRCh37 having many advantages and serving as a gold-standard human reference assembly, there are still some assembly issues that GRC places special emphasis on, including tiling path errors and sequence gaps associated with complex genome structures, base pair-level sequence errors, paralogous sequences with population variation and genomic features such as centromeres and telomeres. With the development of both bioinformatics and experimental resources and techniques, a major release of human reference assembly GRCh38 was published in December 2013, which was more complete and provided better gene and variant representations than GRCh37 did. The current patch release is GRCh38.p14, which contains 178 regions with 261 alternate loci. Compared with GRCh37, GRCh38 contains more genes and protein-coding mRNAs, less partially represented coding sequences and transcripts split over assembly gaps. Moreover, GRCh38 replaces the 3 Mbp centromeric gaps on all GRCh37 releases, which shows improvement in sequencing read mapping and variant calling [251]. Evaluation studies showed that the

improvement of GRCh38 resulted in a more reliable genomic analysis, such as read alignment against the reference, variant calling in clinical-related regions and population variation with alternative loci [252,253].

Although the human reference assembly from the GRC has been widely used in scientific research, its limitations cannot be ignored. The human reference assembly from the GRC has an underrepresentation of repetitive sequences that are left unfinished or incorrectly assembled [254]. Moreover, some regions of GRCh38, such as centromeric alpha satellite arrays, are incorrect, and the entire GRCh38 also shows a genome-wide deletion bias, which may indicate incomplete assembly [81,255]. Therefore, to continually improve the quality of the human reference assembly, efforts have been made with third-generation, long-read sequencing and whole genome assembly methods. Given the low individual read accuracy of long-read sequencing platforms during that time, the genome of a complete hydatidiform mole (CHM), which is the homozygous genome arising from the loss of the maternal complement and the duplication of the paternal complement post fertilization, was used to sequence the human haploid genome because it lacks heterozygous variants. The advantage of these homozygous genomes, such as CHM1 and CHM13, in resolving complex regions shows great potential for future updates of the human reference assembly. Improvement have been made for closing or extending the remaining interstitial gaps, correcting misassembles and better representing repetitive elements and segmental duplications in the current human reference genome assembly [81,256,257]. An evaluation study showed that these new haploid assemblies may be more reliable in variant calling than GRCh38 is, but they contain fewer gene representations and have lesser human diploid complexity [252].

In addition, at the time of writing this thesis, the completed sequence of a human genome was published in March 2022 [245]. Third-generation, long-read sequencing methods, namely, PacBio HiFi and Oxford Nanopore ultra-long-read sequencing, were used to sequence the uniformly homozygous CHM13hTERT cell line. The results, which were called T2T-CHM13 (T2T stands for telomere-to-telomere), generated by whole-genome, graph-based assembly methods, showed the hidden 8% of the human genome, including centromeric regions and the entire short arms of five chromosomes. Compared with GRCh38, T2T-CHM13 reduced the number of contigs from 949 to 24 and increased the number of genes by 5.7%. Although T2T-CHM13 lacks the Y chromosome, it shows great potential for future human genome studies.

4.2.2 *De novo* assembly of a genome sequence

Before the first human genome assembly was published, there was no reference sequence with which sequencing reads could align. In addition, reconstructing a

genome from sequencing reads is an integral step in the genome project of any species [258,259]. Therefore, *de novo* assembly, which does not require a reference genome, is performed to assemble a genome with sequencing reads.

There are several methods for *de novo* assembly [260]. For example, tools such as Celera Assembler [261], Arachne [262], PCAP [263], Canu [264] and AMOS [265] use overlap graph-based methods to construct genomes from sequencing reads. These methods first find overlaps among all reads and then use these reads to form layouts on a graph before finally constructing the consensus sequences. Other tools, such as SOPAdenovo2 [266], ABySS [267], ALLPATHS [268], EPGA2 [269], Flye [270] and WENGAN [271], use the *de Bruijn* graph to construct genomes [272]. In the *de Bruijn* graph, each node is represented as a *k*-mer, which is *k* consecutive bases in one read. A directed edge between two nodes will be formed if there is an overlap with *k*–1 bases between the two reads. As for assembly methods, a set of short overlapping sequences denoted by *k*-mer will be formed to replace each read. The value of *k* should be set accordingly because it is an important value for constructing a genome. A larger value of *k* can remove some short repetitive regions and the number of nodes, but it can increase the number of gap regions. A small value of *k* reduces some gap regions but increases short repetitive regions and the number of nodes. Contigs are formed by merging *k*-mers that appear adjacently in reads [269,270]. The benefit of less storage than pairwise overlaps and the graph-based representation of the repeat structure of the genome make the *de Bruijn* graph widely used in sequence assembly tools [260]. String graph-based methods, which share some similarities with overlap graph-based and *de Bruijn* graph-based methods, are also widely used in *de novo* assembly. Compared with overlap graph-based and *de Bruijn* graph-based methods, string graph-based methods remove duplicated reads and formulate assembly with the full-length of a sequence read instead of *k*-mers. Tools such as SGA [273] and FALCON [274] were developed based on this.

Because of the reference-free analysis procedure of *de novo* assembly, it provides opportunities to study complexities of the human genome that lack annotations from the current build of the human genome reference. Research has been conducted using third-generation, long-read sequencing techniques with *de novo* assembly methods to study the complexities of human genomes, such as segmental duplications [275], interstitial gaps [81], variant diversity [276] and the completed human genome [245].

4.2.3 Read alignment against a reference genome sequence

Next-generation sequencing can produce a massive number of reads from the sequenced samples, representing a powerful technology for studying the human

genome. Aligning sequencing reads to a reference genome, which is downstream to quality control, is the first crucial step in NGS data analysis or pipelines, including variant calling, isoform quantitation and differential gene expression [277–279]. The alignment process can determine the likely point of origin of each sequencing read with respect to the reference genome [280].

Although these algorithms are complex, depending on the strategies employed behind them, alignment algorithms or aligners can be largely grouped into two categories: hash table-based algorithms and Burrows–Wheeler Transform-based algorithms [281–283]. For the hash table-based algorithm, it essentially follows a seed-and-extend paradigm, in which BLAST is the representative method [284]. First, it keeps the position of each *k*-mer sequence of the read in a hash table and scan the genome sequences of *k*-mer exact matches by looking up the hash table, which is the step of seed detection. The algorithm then extends and joins the seeds with a dynamic programming algorithm, such as Smith-Waterman alignment [285]. Tools such as SOAP [286], SeqMap [287], MAQ [288], RAMP [289], PerM [290] and ZOOM [291] are further developed, and the spaced seed method is applied, allowing internal mismatches to improve the sensitivity of alignment. SHRiMP [292] and RazerS [293] use the q-gram filter, which allows gaps within the seed to implement multiple alignments. Because some reads in repetitive regions may have too many seeds that should be checked, hash table-based algorithms may give relatively poor results [282].

To enable rapid read searching and address alignment to repetitive regions of the reference genome, some better data structures, such as suffix tree, enhanced suffix array [294], and Ferragina-Manzini index [295], are applied. To reduce the memory occupation of these data structures, Burrows–Wheeler Transform [296], which is a text compression algorithm, is applied as a solution. Tools such as MUMmer [297] and OASIS [298] are based on a suffix tree, Vmatch [294] and Segemehl [299] are based on an enhanced suffix array, and Bowtie [300], BWA [301] and SOAP2 [302] are based on the Ferragina-Manzini index.

In the history of NGS development, the growing read length is one of the most significant features. Initially, the sequencing reads from Illumina platforms were only 25 bp. When the technique developed reads with more than 100 bp, gapped alignment methods benefited genomic studies, such as variant calling, especially indel calling. With gapped alignment, a read containing an indel could be mapped to the correct position instead of being mapped as a mismatch. In addition, paired-end sequencing techniques provided paired-end information for alignment tools to estimate the sizes of some repetitive regions and SVs. An evaluation study showed that alignment tools using the base quality scores of reads to calculate the error probability of each base had better alignment accuracy [134,281]. In addition, there are other methods in the category of re-alignment or local re-assembly that perform

reads alignment only for a locus of the genome. The general alignment tools map a read to a location independently without considering the correction between other reads; however, reads from the same locus are highly correlated. These methods use corrected reads in the same locus to refine the alignment and attempt to have a better representation of the sequenced genome, especially for the regions containing indels.

4.2.4 Data format of the read alignment

During the development of NGS data analysis, the data format for the short-read alignment produced by NGS against a reference genome complied with a standard. In a conference call on October 21, 2008, the data processing sub-group of the 1000 Genomes Project, which was an international collaboration project to produce an extensive catalogue of human genetic variation, decided to unify a variety of short-read alignment formats into a Sequence Alignment/Map (SAM) format [303]. The SAM format is a text-based format which the file contains the information of sequencing reads towards the reference sequence; it has a binary representation, the BAM format, which was designed to improve computational performance. The SAM and BAM formats are still being updated, and the most recent specification is version 1.6, which was published on August 22, 2022 [304].

The SAM format consists of one header section and one alignment section, and all lines are delimited by TAB. The header section starts with the character '@', which contains information on the sequencing experiments. The alignment section has 11 mandatory fields and a variable number of optional fields, including read group information and the sequencing platform. Among these fields, the most important one is the CIGAR string, especially for variant calling. The CIGAR has nine operations to describe the pairwise alignment compared with the reference, such as match, deletion, insertion and clipping [303,304].

In the NGS data analysis, the sequencing reads in FASTQ format can first be aligned against the reference sequence using alignment tools. The alignment process generates sequencing data into a SAM file. Then, the SAM file can be converted into a BAM file and further sorted and indexed, with the purposes of fast random retrieval of alignments and efficient usage of computational resources. SAMtools is a commonly used library and software package for parsing and manipulating alignment data in SAM/BAM format [303]. To implement the stable and robust application programming interfaces of the functions of SAMtools, the HTSlib, which is a dedicated programming library in C/C++ language, was published for processing common data formats used in high-throughput sequencing, such as FASTQ, SAM and BAM. The HTSlib also has the ability to bind with other bioinformatics tools encoded with various programming languages, such as Python and R, facilitating the analysis of sequencing data and boosting the development of sequence analysis tools [305].

Nowadays, SAM/BAM and the corresponding analysis tools/libraries are used not only for NGS data but also in third-generation, long-read sequencing data analysis as the primary format of alignment; they facilitate the analysis of noisy read alignments of millions of bases in length [306]. Read alignment formats, such as Compressed Reference-oriented Alignment Maps, has also been developed and is accepted by the research field.

4.3 Indel calling with NGS data

After the read alignment processes, the read alignment file, such as a sorted and indexed BAM file, can then be analysed using variant calling tools to detect and genotype genomic variants, such as SNPs, indels and SVs. Usually, the raw output variants from variant calling tools will be filtered based on information, such as read depth, strand biases and quality scores, to reduce the number of FP variant calls. To make sense of variant calls, tools for automated variant annotation have been developed. These tools can integrate genome information available in other resources, such as genome annotation tracks of the UCSC Genome Browser [307], to annotate variants about genes that they are located at or allele frequencies in certain populations. Some precomputed scores, such as the Sorting Intolerant From Tolerant score [308], can also be used for variants to predict their likely functional consequences and amino acid exchanges. With all these pieces of information, the results from a variant calling process can make sense for downstream analysis to study the biological and clinical significance of variants [309]. Notably, the processes and underlying algorithms of germline and somatic indel calling are significantly different. This thesis mainly focuses on germline variant calling.

4.3.1 Indel calling algorithms with NGS data

The widely used NGS technology, with its reduced sequencing costs, helps produce vast amounts of human genome sequencing data, which also boosts the development of algorithms for variant calling. Compared with the DNA microarray, NGS can help identify larger sizes of genomic variations, such as indels and SVs. Because of the short length of sequencing reads produced from NGS platforms, the detection of small (typically < 50 bp) and large (typically ≥ 50 bp) indels relies on different calling algorithms. In general, indel calling algorithms can be classified into seven major groups: gapped alignment-based methods, local re-assembly-based methods, *de novo* assembly-based methods, read depth-based methods, paired-end reads-based methods, split read-based methods and machine learning-based methods [310–314] (Table 1).

Table 1. List of several variant calling tools that can be used for germline indel calling. The indel calling tools may be developed based on gapped alignment (GA), local re-assembly (LR), *de novo* assembly (DA), read depth (RD), paired-end reads (PR), split reads (SR) and/or a machine learning model (ML). A tool may use only one algorithm or integrated algorithms as its indel calling strategy. Variant calling tools may have calling abilities not only limited to insertion (INS) and deletion (DEL) but also to SNV, duplication (DUP), inversion (INV), translocation (TRA) and complex indels (COM).

Tool	Algorithm	Input	Variant type	Latest update*
BreakDancer [315]	PR	BAM	INS, DEL, INV, TRA	2013-03
Clair3 [316]	GA, ML	BAM	SNV, INS, DEL	2022-08
ClipCrop [317]	SR	SAM	INS, DEL, DUP, INV	2011-12
CNVpytor [318]	RD	BAM	DEL, DUP	2022-04
Cortex [319]	DA	FASTQ	COM	2012-08
DeepVariant [320]	GA, ML	BAM	SNV, INS, DEL	2022-10
DELLY [321]	PR, SR	BAM	INS, DEL, DUP, INV	2022-09
Dindel [322]	GA	BAM	INS, DEL	2015-03
FermiKit [323]	DA	FASTQ	SNV, INS, DEL	2015-07
Freebayes [324]	GA	BAM	SNV, INS, DEL, COM	2022-01
GASVPro [325]	PR, RD	BAM	DEL, INV	2013-10
GATK [326]	GA, LR, RD	BAM	SNV, INS, DEL	2022-10
Gridss [327]	LR, PE, SR	BAM	INS, DEL, DUP, INV	2022-02
HYDRA [328]	PR	BAM	INS, DEL, DUP, INV	2010-08
IndelMINER [329]	PR, SR	BAM	INS, DEL	2015-07
INDELseek [330]	GA	BAM	COM	2017-02
LUMPY [331]	PR, SR, RD	BAM	DEL, DUP, INV, TRA	2020-09
Manta [332]	LR, PR, SR	BAM	INS, DEL, DUP, INV	2019-06
MetaSV [333]	LR,PR,SR,RD	BAM	INS, DEL, DUP, INV	2017-01
Octopus [334]	GA, LR	BAM	SNV, INS, DEL	2021-05
PEMer [335]	PR	BAM	INS, DEL, INV, COM	2019-02
PennCNV [336]	RD	BAM	DEL, DUP	2019-01
Platypus [337]	GA, LR	BAM	SNV, INS, DEL	2015-04
Pindel [338]	SR	BAM	INS, DEL, DUP, INV	2017-05
Scalpel [339]	GA, LR	BAM	INS, DEL	2018-01
ScanIndel [340]	GA, DA, SR	FASTQ, BAM	INS, DEL	2017-10
SoftSV [341]	PR, SR	BAM	DEL, DUP, INV	2015-09
Strelka2 [110]	GA, LR, ML	BAM	SNV, INS, DEL	2018-11
SvABA [342]	LR, PR, AR	BAM	INS, DEL, DUP, INV	2019-03
SVDetect [343]	PR	SAM, BAM	INS, DEL, DUP, INV, TRA	2013-01
Ulysses [344]	PR	BAM	INS, DEL, DUP, INV, TRA	2014-09
VarDict [345]	GA,LR,PR,SR	BAM	SNV, INS, DEL, DUP, INV	2020-09
VarScan [346]	GA	pileup	SNV, INS, DEL	2019-07
Wham [347]	LR, PR, SR,	BAM	INS, DEL, DUP, INV, TRA	2016-02

* The last updates of tools were recorded before 28/10/2022.

Gapped alignment-based methods

Gapped alignment-based methods are generally optimised to detect small indels. During the initial read alignment steps with gapped aligners, the mapping information of reads is generated. This information, which contains the mapping status of each read, is used by indel calling algorithms to make indel calls. Probabilistic models, such as heuristic models or Bayesian models, are applied to filter sequence alignment errors from true indels [322,346]. The differences between the probabilistic models used by tools can cause discrepant indel calling results [348]. Gapped alignment-based methods usually require indels to be covered within a read and be identified during the initial sequencing read alignment steps, which limits the calling of large sizes of indels, especially for novel insertions [349]. A read that covers a large indel may only have a few bases supporting the breakpoint; thus, this read either fails to map against the reference sequence, or only a part of the read is mapped well, but the rest of it is trimmed or soft-clipped by the aligner [310]. Representative gapped alignment-based variant calling tools are Dindel [322], FreeBayes [324], GATK UnifiedGenotyper [350], and VarScan [346].

Local re-assembly methods

As described in Section 4.2.3, local re-assembly-based methods refine the alignment of reads in the same locus to provide a better representation of the sequenced genome. In variant calling, these methods first identify active regions that show evidence of having indels. For these regions, variant calling tools discard the existing mapping information and re-assemble reads to generate possible haplotypes using *de Bruijn* graphs. The reads in these regions are then re-aligned to the possible haplotypes, and the likelihood of the haplotypes is calculated. Posterior probabilities of having indels are then calculated, and indels are called when the posterior probability exceeds a certain threshold. These algorithms help call indels in regions that are traditionally difficult to call, such as different types of variants that are close to one another. GATK HaplotypeCaller [326], Platypus [337], Scalpel [351], and are representative variant calling tools for these methods.

De novo assembly-based methods

Unlike local re-assembly methods, which only re-assemble reads in active regions, *de novo* assembly-based methods perform whole genome assembly for variant calling. These methods first assemble reads into contigs and then compare the contigs to the reference sequence to call indels. Large indels, especially large novel insertions, can be detected efficiently with these methods. However, the high computational cost of *de novo* assembly and assembly errors are the main drawbacks

of these methods. FermiKit [323] is a typical variant calling tools based on these methods.

Read depth-based methods

Read depth-based methods are mainly used for detecting large genomic variants, such as copy number variations. These methods use the density of reads in variant regions of the reference sequence to estimate copy number changes. Although the copy numbers of variants can be accurately predicted, the breakpoint resolution from these methods is not as accurate as that from other methods, and PCR-induced biases may cause problems in variant detection [311]. A representative read depth-based variant calling tool is CNVpytor [318].

Paired-end read-based methods

Paired-end read-based methods can detect large indels from discordantly mapped paired-end reads. The significant deviations of the expected and actual distances between paired-end reads may indicate indels. For example, paired-end reads that are mapped further apart may indicate an insertion, while those that are mapped closer may indicate a deletion. However, the resolutions of the indels depend on the mean and standard deviation of the library's insert size. The exact indel sequences may not be known, and the detection of small indels is not sensitive because of difficulties in distinguishing indel-caused distance deviations between paired-end reads and normal background deviations. Representative paired-end read-based variant calling tools are BreakDancer [352], HYDRA [328] and PEMer [335].

Split read-based methods

Split read-based methods are capable of detecting medium-size indels, which are difficult to detect using gapped alignment-based methods. These methods use discordant paired-end reads in which one end maps perfectly to the reference sequence, but the other end split by an indel cannot be mapped or can only be soft-clipped mapped. The mapped end is used as an anchor point to determine the direction of the other end where an indel is assumed to be present. These unmapped or soft-clipped ends can be clustered together and searched on the reference sequence for the best alignments with the split read. With these partial alignments, an exact breakpoint can be determined, and an indel can be reconstructed. The difficulty of mapping split reads to the reference sequence requires higher sequencing coverage to obtain sufficient supporting reads for an indel [311]. The lack of probabilistic

models makes split read-based methods may have high FP rates, and post-filtration of indel calls may be needed [310]. Representative split read-based variant calling tools are ClipCrop [317] and Pindel [338].

Machine learning-based methods

Machine learning-based methods construct empirical variant filtration models from training data to reduce the FP results in indel calling. The training data usually come from gold standard variant datasets, such as Platinum Genomes and the Genome in a Bottle Consortium (GIAB) [63]. Strelka2 applies pre-trained random forest models on Platinum Genomes sample NA12878, taking such as genotype information, mapping quality, strand bias, read depth and other features as input to produce the probability of an erroneous variant call [110]. In addition, deep learning methods convert the mapping condition of reads within a genome region into an image and then call indels with a deep learning model. For example, DeepVariant is a small variant calling tool built using convolutional neural networks, trained with data from NGS or PacBio sequencing data. DeepVariant converts the sequencing data at each putative variant locus into an image-like tensor which containing six channels as inputs, including read base, base quality, mapping quality, strand of alignment, read supports variant, and base differs from reference [320]. For large indels and SV detection, machine learning-based methods have also been developed. DeepSV trained the model with datasets in the 1000 Genomes Project to call large deletions by converting read alignment features including read depth, split read and discordant pairs into images [353]. Cue converts read alignments into image-like data with channels containing information about read depth, read pairs, and other and builds a convolutional neural network by training with simulated data to call large deletions, duplications and inversions [354].

The underlying algorithms of different methods have both strengths and weaknesses; many tools integrate several algorithms to make precise and sensitive indel calls. For example, DELLY [321] and Manta [332] use both information from paired-end reads and split reads to define the exact positions and sizes of SVs. ScanIndel is a hybrid tool that can call a wide size range of indels via gapped alignment, split reads and *de novo* assembly [340]. Moreover, most methods (except *de novo* assembly-based ones) take BAM files as input, which means that the gapped alignments of reads are the initial information for tools to detect indels. For tools that use paired-end sequencing data as input, the information provided by discordant paired-end reads with split reads can always be used to define the interval of indel regions [355]. A combination of different methods can help a wide range of indels.

4.3.2 Format of variants in genomics study

Standardised data formats can significantly improve the interoperability of tools for various data analysis purposes. The output format of most variant calling tools is the variant call format (VCF), which is a standardised format for storing the most prevalent types of sequence variations, including SNPs, indels and SVs, together with rich annotations. The VCF file is a textual encoding format file that has the ability to encompass millions of variants with genotype information and annotations from thousands of samples; it can also adopt complementary indexing, which allows fast data access [356]. The VCF and its binary encodings are still being updated, and the most recent version is VCFv4.4, which was published on January 27, 2023 [357].

A VCF file consists of a header section and a data section. The header section consists of meta-information lines and a header line. The meta-information lines contain information about the file format, date, descriptions and formats of variants' information, filters and individual genotypes, as well as alignment information about assembly, contigs, samples and pedigree. Meta-information lines are recommended to include all the entries that are used in the body of the VCF file with the 'key=value' pairs format. The header line has eight fixed mandatory columns, namely, 'CHROM', 'POS', 'ID', 'REF', 'ALT', 'QUAL', 'FILTER' and 'INFO', which stand for chromosome, position, identifier, reference base(s), alternative base(s), quality, filter status and additional information of a variant, respectively. If a VCF file contains genotype data, additional columns will be appended. The additional columns about genotype information are 1) a 'FORMAT' column, which describes the order and formats of genotype data, followed by 2) arbitrary, unduplicated numbers of sample IDs. For the data section, all data lines are tab delimited and filled with the corresponding information according to each column and format. If a missing value is present, it will be specified with a dot in all cases. The 'ID' column of a VCF file from the variant calling tool is usually empty and marked with '.', but some variant calling tools may provide identifiers for each called variant. If variant records are annotated with databases, such as dbSNP, the 'ID' column may contain variant identifiers associated with the database [135]. If genotype information is present, a 'FORMAT' field is given to specify the data types and order, followed by one data block per sample with corresponding values to the types specified in the 'FORMAT' field. The first key must always be the genotype 'GT', if it is present. Genotypes are encoded as allele values separated by either '/' or '|', which indicates unphased and phased genotypes, respectively. An allele value of '0' in genotype values indicates the reference allele, an allele value of '1' indicates the first allele listed in the 'ALT' column, an allele value of '2' indicates the second allele list in the 'ALT' column and so on. For diploid calls, such as the human genome, the genotype value could be '1/0', '0|1', '1/2', '2|3' and so on. Haploid calls, such as those on the Y chromosome of humans, are indicated by having only

one allele value. If a variant call determines the genotype for a sample, a dot must be specified for each missing allele in the 'GT' field.

Because different variant calling algorithms use their own technical concepts and variables, the information provided by different tools on the 'FILTER', 'INFO' and 'FORMAT' columns might be different, which makes the VCF format not identical between tools. Given the complexity of the human genome, an indel may also have several different representations depending on the sequence contexts, such as STRs. Moreover, some variant calling tools may merge adjacent variants, including indels, as a single variant, depending on the procedures of tools. Thus, it is difficult to reach a conclusion as to whether indels are the same by simply comparing their positions. These issues can cause the comparison of different variant calling results to be problematic, especially for benchmarking variant calls.

Benchmarking tools or workflows, such as SMaSH [358], vgraph [359], RTG Tools [360], vcflib [361], hap.py benchmarking toolkit [109] and a benchmarking workflow created by Stanford University [362], were developed to address this issue. These benchmarking tools or workflows can recognise positions, alleles, and the genotypes of variants from query sets and compare them to truth sets in a unified manner. Various statistics, such as precisions, recalls, F1 scores, number of called variants, receiver operating characteristic curves and other useful results, can be drawn using these tools or workflows. Improvements have been made on unified variant representations, accurate computational performance metrics, implementations of comparison frameworks and integrations with high-confidence human variant sets. With these benchmarking tools, variant calling tools and pipelines can be evaluated easily and fairly for different research purposes.

Besides the need for a standard format in computational data analysis, a consistent and unambiguous description of sequence variants is critical in clinical diagnostics for sharing the variants detected. In 2000, the HGVS proposed a sequence variant nomenclature system, which has been widely adopted by clinical laboratories and continuously extended to accommodate the needs of genomic research [61]. Human Genome Variation Society nomenclature can describe a variant in the DNA, RNA and protein levels according to an accepted reference sequence, mutated positions and nucleotides or amino acid changes. For DNA-level descriptions, there are eight basic variant types: (1) substitution, (2) deletion, (3) duplication, (4) insertion, (5) inversion, (6) deletion-insertion, (7) repeated sequences and (8) complex. The format of HGVS nomenclature at the DNA level starts with the prefix 'g' and is followed by a position and corresponding variant type.

There are several differences between VCF and the HGVS nomenclature format. One difference is that VCF is a file format that can hold a vast number of variants in a single file, whereas HGVS nomenclature is a string format applied to a single

variant. In addition, VCF shifts variants to the 5' aligned (also known as left-aligned) position with respect to the genome, whereas HGVS shifts variants to the 3' aligned t (also known as right-aligned) position with respect to a reference sequence, a gene, a transcript, or a protein. This difference may cause the same variant to have completely different locations and alleles in these two formats. Another difference is that HGVS nomenclature requires that a variant must have a variant type, but VCF calls for small variants do not state variant types. Although VCF calls for SVs can state variant types, usually, calls for SVs do not contain exact reference and alternative alleles. In addition, in HGVS nomenclature, nucleotide-gain variations may be classified as insertions, duplications or TRs, and nucleotide-lost variations may be classified as deletions or TRs. In VCF, especially for small variants, variants with a number of nucleotide changes cannot be classified as different variant types and switched into HGVS nomenclature straightforwardly. For VCF calls for SVs, the variant types determined by the variant calling tool may disagree with the standards from HGVS nomenclature. Furthermore, VCF calls from most variant calling tools prefer to represent variants individually without considering the presence of nearby variants at a close distance. However, HGVS nomenclature provides options to present a range of variants occurring at a close distance, which cannot be described as one of the basic variant types. All these differences between the two formats require additional effort to calculate the conversions when dealing with human clinical data [363]. The widespread use of NGS in clinical fields requires the transformation of genomic variants from VCF in computational data analysis to HGVS nomenclature in clinical research. Methods have been developed to meet these needs. Mutalyzer is an HGVS variant nomenclature checker that has functions for constructing, validating, and transforming sequence variant descriptions according to HGVS guidelines. The web interface design of Mutalyser makes it easy to use for either describing an individual variant or processing variant descriptions in batch data [364,365]. The Ensembl Variant Effect Predictor is a module of the Ensembl genome browser that provides functions for mapping sequence variant descriptions in the VCF to HGVS format. The web interface and command line tool, together with the up-to-date resources of human genomics and other various variant-related functions in the Ensembl genome browser, make the Ensembl Variant Effect Predictor a versatile tool for studying human genomics [366]. SnpEff is a command line-based tool that has functions to take variants in VCF files as input and output variants with HGVS nomenclature. SnpEff not only annotates genomic variants with various information but can also predict their functional effects [367]. ANNOVAR is a command line-based tool that uses update-to-date information to functionally annotate genetic variants. ANNOVAR has the ability to integrate genome annotation resources from popular databases, such as the Ensembl Genome Browser, and to provide functions that will enable users to annotate variants with their own interests

such as HGVS annotations [368]. VariantValidator is a web-based tool that can validate HGVS sequence variation descriptions and automate the conversing descriptions of variants in VCF into HGVS nomenclature. It also has functions for mapping variants between transcripts and genome sequences [369]. The hgvs Python package can validate the HGVS nomenclature of genomic variants in the context of the reference sequence [370]. The difficulties in format conversion and validation of the variant description between VCF and HGVS nomenclature are not merely simple format changes; they involve distinguishing ambiguous sequence contexts around variants, which make conversion results of tools incorrectly [363]. In addition, tools based on the web interface require additional manual steps for data transfer, which limits the development of the data analysis pipeline. The computational tool described in **Publication III** provides a format conversion function of the variant description between VCF and HGVS nomenclature at the DNA level.

4.3.3 Evaluation of tools for indel calling with human genomes

Despite the development of indel calling algorithms and tools and impressive improvements in indel calling, a fair, comprehensive, and in-depth evaluation of these tools is lacking. Tools with different underlying algorithms may not perform well in all aspects. Because of the various topics and objective targets of real-world human genomics research, it is important to determine the strengths and weaknesses of tools on indel calling with different underlying algorithms, which can help researchers select the suitable tools to fit their research purposes. Meanwhile, revealing the current limitations of indel calling in computational fields can help suggest current needs and boost future indel calling developments.

During the past 10 years, many evaluation studies have been conducted on germline indel calling with human NGS data. Because of the lack of a gold standard truth set and benchmarking tools, early evaluation studies used simulated variants or a handful of clinically validated real variants. Concordance results among tools were often applied when a truth variant set was lacking. For example, Neuman et al. evaluated four variant calling tools with simulated human WGS data and real non-human WGS data. Small indels were inserted into human chromosome 16 of GRCh39/hg19 with certain frequencies. The effects of indel frequency, read length, indel size and sequencing coverage were evaluated [371]. O'Rawe et al. assessed variant calling, including small indel calling of three tools, with WGS and WES data. The concordance of tools was used for evaluation without knowing the actual truth indels, and cross-platform validation was conducted to evaluate unique-to-pipeline indels [348]. Liu et al. performed an evaluation of variant calling, including small indel calling of four tools with real WES data and simulated WGS data. Exome-

based array, Sanger sequencing and external scripts were applied for the validation of variants. Comparisons with validated indels and the concordance among tools were assessed [372]. Fang et al. evaluated two variant calling tools with different coverages of real WGS and WES data. Simulated sequencing data with 1–100 bp indels were also used. Cross-platform validation was applied to evaluate real sequencing data. The indel calling performance metrics of tools, concordance among tools and data types with different coverages, and the influences of sequence contexts were discussed in their results [373]. Ghoneim et al. evaluated three tools on indel calling, with 639× coverages targeting the gene panel, 74× WES and 24× WGS NGS data of 48 human samples. The concordance of indel calls from the tools with different data types was used to evaluate tool performance. An identical indel call was defined as an indel called by at least two tools within a position deviation ± 10 bp. The indel calling ranges of each tool with different data types were briefly assessed by comparing the maximum, mean and median of the called indel sizes [374]. Kim et al. evaluated four variant calling tools with WES data, and the results of the detection of 840 small indels were validated using Sanger sequencing. The distribution of called indel sizes, the performance metrics of the indel calling results and the concordance between the selected tools were assessed [375]. Sandmann et al. evaluated eight variant calling tools with the high-coverage, targeted gene panel sequencing data of 165 human samples and two simulated samples. The targeted gene panel contained 19 genes known to be recurrently mutated in patients with myelodysplastic syndrome and small-size mutations with low allele frequencies. The number of called mutations, comparison with the truth set, the influence of different sequencing coverages and background noises and running time were used for the evaluation. Without having matched sample data, platform cross-validation, expert-based review and annotation information, such as allele frequencies, read depths and presence in databases, were used to categorise variants as polymorphisms, true mutations and artefacts [376].

Later, with the development of several high-confidence human variant sets, these variant sets can now serve as truth sets for the evaluation of indel calling. For example, Hasan et al. evaluated seven tools using 78 human low-coverage WGS datasets of chromosome 11 from the 1000 Genomes Project. The indel truth set of the sequencing data was fetched by selecting the shared samples from another independent indel study. In their evaluation, most of the indels in the truth set and the tools' call sets are ≤10 bp. The running time, the number of indels called, the comparison of different indel sizes with the truth set and the similarity among the tools were evaluated. They treated an indel call as true positive (TP) if the position deviation between the truth indel and the called indel by the tool is ± 5 bp [377]. Laurie et al. conducted variant calling evaluations, including small indels with three variant calling tools and two aligners. Data on WGS and WES of the human

individual NA12878 were used in their studies, and the corresponding truth set was obtained from GIAB. The performance metrics of variant calling, the concordance among tools, and the comparison between sequencing data types and computational costs were examined [378]. Li et al. assessed five variant calling tools on small indel calling with two simulated WGS datasets, one real WGS dataset from GIAB and family-based, real WES sequencing datasets. The running time and the number and performance metrics of the called indels were evaluated with known truth indels. They simply defined TPs, false negatives (FNs) and FP indels based on the presence of indels in the tools' call sets and the truth sets. The concordance rates among tools and family-based Mendelian error rates were used for evaluation with the family-based datasets without the known truth indels [379]. Hwang et al. evaluated variant calling abilities, including small indels of seven short-read aligners and 10 variant calling tools with WGS data. Two WGS datasets with different sequencing coverages and read lengths were used, and the corresponding truth sets were obtained from GIAB. The concordance between variant calling pipelines and the influences of sequence contexts were evaluated [380]. Chen et al. examined seven variant calling tools with the same WGS datasets and TP criteria as Hasan et al. [377]. The improvement made in this evaluation study was that they assessed insertion and deletion calling separately. They also applied a pooled sample-based method for more accurate evaluations by comparing the multiple samples pooled indel call set of tools with the pooled truth set. The concordance and combination of variant calling methods were likewise evaluated. They studied the somatic indel calling of four tools by using three types of cancer sequencing data. Annotation of indels from external databases and resources was used to assess the performance metrics of indel calling [381].

To evaluate indel calling in a unified manner and avoid potential ambiguous indel representation, benchmarking tools were developed. Facilitated by high-confidence human variant sets, tool evaluation for indel calling can be performed in standardised ways. For example, Cornish et al. studied combinations of five variant calling tools and four short-read aligners with WES datasets from the GIAB Consortium. The consortium generated the variant truth sets by integrating multiple variant calling pipelines and containing both SNPs and small indels in the confidence regions of the human genome [382]. A careful design pipeline was applied to generate the variant call sets from each tool, which contained additional steps, such as realignment of reads, recalibration of read quality scores, and filtration of variant calls. The indel comparison between the truth set and the tools' call sets was made using an external benchmarking tool, vcflib. The performance metrics of the tools' indel calling with raw and filtered results were assessed, together with the concordance of the tools [383]. Hwang et al. evaluated 13 variant calling pipelines, which were combinations of three read aligners and four variant calling tools.

Twelve WGS or WES datasets from different sequencing platforms with coverage of 50–300× were used for evaluation. The indel truth set was selected by GIAB. The comparison results of the tools with the truth set and the concordance of the selected variant calling tools were used for evaluation. To deal with different indel representations between tools, additional steps were applied to regularise indels, and vcflib was used to compare the indel truth set and the tools' call sets [384]. Supernat et al. evaluated three variant calling tools with three different sequencing coverages of human WGS data. The truth set was the human individual NA12878 from GIAB, which contains SNPs and small indels. Comparisons between the truth sets and the variant call sets of selected variant calling tools were performed using an external variant benchmarking tool, RTG Tools, on whole-genome-wide and coding regions [385]. Chen et al. evaluated three variant calling tools with nine WES and WGS datasets from five different sequencing platforms. The truth set of small variants was obtained from the GIAB Consortium, and the tools' call sets were compared with it using the external variant benchmarking tool hap.py. The performance, concordance and operating efficiency of 27 combinations of sequencing platforms and variant calling tools were evaluated [386]. Zhao et al. assessed three variant calling tools using real WGS data from GIAB, a synthetic diploid and simulated WGS data with common technical sequencing parameters. The tools' performance on small indels was assessed by comparing the tools' call sets with truth sets via hap.py. The performance metrics of variant calling, running time and concordance of tools were evaluated using different genome contexts [387]. Barbitoff et al. examined the pipelines of variant calling with four short-read aligners and nine variant calling tools. Fourteen WGS and WES datasets from GIAB and an additional three WGS and three WES datasets from the African ancestry of the 1000 Genomes Project with various sequencing coverages were selected for evaluation. The performance of the tools on small indel calling was studied using hap.py. The factors that may influence the accuracy of variant calling and concordance among tools and datasets were analysed and discussed [388].

With the development of variant calling algorithms, tools with SV calling abilities have been developed, and evaluation studies for these tools have been conducted. For example, Kosugi et al. evaluated 69 SV calling tools with various read lengths, sequencing coverages and insert sizes of simulated and real WGS datasets. The simulated dataset contained 8,310 different types of SVs, ranging from 50 bp to 1 Mb, and the real dataset contained around 5,000 SVs merged from different resources for the corresponding human samples. The performance metrics of tools on the different properties of read data and SVs, measurements of running time and memory consumption, and the identification of pair algorithms were evaluated. They defined a certain type of SV as TP based on the overlapped region between the called SV and the true SV with certain technical thresholds [389].

Cameron et al. assessed 10 SV calling tools using three real WGS sequencing datasets and multiple simulated WGS datasets with various technical sequencing parameters. The simulated datasets used in their study covered a wide size range of different types of SVs with various sequence contexts, and the truth sets were obtained from other independent indel studies. The SVs from the calling tools were converted into breakpoint coordinates with event sizes and compared with the true SVs. The TP results were SVs that passed certain thresholds of overlapping with the true SVs. Structural variant detection metrics, the impact of the sequence context, SV sizes and quality scores, concordance and running times were evaluated [111]. Pei et al. evaluated 11 variant calling tools with next-generation and third-generation sequencing data on both germline and somatic variant calling. Real WGS data from different platforms with various coverages and synthetic tumour sample NGS data were selected for evaluation. The performance metrics of small indel calling of tools on different datasets were assessed via RTG Tools. Sequence context, including GC content and segmental duplication, as well as the computational costs of tools, was evaluated [390].

The basic conclusion of these evaluation studies was the ranking of the selected variant calling tools with different technical sequencing parameters and research targets. Additional insights were also obtained from previous evaluation studies. For the development of indel calling algorithms and tools, the following lessons can be learned. First, gapped alignment-based tools cannot efficiently call indels larger than the read length; split read-based and paired-end read-based tools are preferred for large indel calling. Second, tools with local re-assembly algorithms have good abilities for calling indels until medium sizes. Third, machine learning-based and deep learning-based tools have the best small indel calling abilities so far. Fourth, using a combination of different calling tools may result in better indel calling results than using a single tool. However, a combination may not always lead to good results; a careful experimental design is preferred. Fifth, the selection of specific tools suitable for different types and size ranges of indels is preferred to obtain the desired results. Different algorithms are suitable for different types of indels. The detection of tandem repeat mutations especially requires specific tools. The tools for general indel calling may not be suitable for calling tandem repeat mutations. Sixth, better usage instructions for tools are needed. Because of the lack of detailed instructions, the majority of evaluation studies have tested variant calling tools using default parameters. Optimising the parameters of tools may improve indel calling results, but it also consumes time and requires expert knowledge. Best practices or instructions for using tools based on common data types and research purposes are desired from tool developers.

For selecting proper sequencing and data analysis procedures, previous research has demonstrated that 1) increasing sequencing coverage can improve the performance of indel calling, but beyond a certain range (30×), there may only be marginal improvements. Insertion calling may require higher sequencing coverage than deletion calling. 2) Pre-processes, such as sequencing error checks, and post-processes, such as the filtration of low-quality variant calls, can help improve the performance of indel calling. 3) Based on research purposes, the careful selection of sequencing platforms, variant calling tools and analysis parameters is required for reliable indel calling. The influence of variant calling tools is greater than that of sequencing read aligners.

Regarding truth indel sets for tool evaluations, the following lessons can be learned. First, better indel truth sets are needed. Although improvements in small indel truth sets and their benefits for tool evaluation have been achieved in the last five years, truth sets containing various sizes of indels, especially large indels, are still needed for better development. Second, to conduct a good evaluation, simulated data and read data should both be considered. Simulation can create desired data on which indels can be generated based on specific evaluation purposes, and the labels of indels are clearly known. Most tools performed well with simulated data, making them suitable for testing the theoretical limitations of indel calling algorithms. However, simulations cannot fully replicate the identical sequence complexities of the human genome. Thus, real sequencing data must be used in evaluations, and the performance of tools should be assessed based on that. Awareness of the potential incompletion of the indel truth set should be made. Third, sequence contexts, such as tandem repaet and GC content, may affect indel calling. High-confidence variant sets are routinely used in evaluation studies. However, these variant sets exclude genome regions where the sequencing coverage is low, and genotyping variants is difficult. Indels are enriched in these regions, and based on previous evaluations, the sequence context may affect indel calling. Thus, the sequence contexts of variants cannot be ignored.

In **Publication I**, an evaluation of eight variant calling tools covering multiple algorithms was conducted with NGS data. Previous studies were not investigated is the suitable indel size range for different variant calling tools for indel detection. In **Publication I,** the size range of the evaluated indel was remarkably larger than that in previous evaluation studies. A semi-simulated dataset was created, and two real sequencing datasets were applied to evaluate tool performance on different size ranges of insertions and deletions separately. Sequence contexts, such as STR and computational costs, were also measured.

4.4 Sequence context annotations of genomic variants

In Section 3.1.4, the influence of the sequence context of genomic variants was discussed. To determine the sequence context of genomic variants, annotation methods should be used. Some sequence context features, such as nearby variants or breakpoint ambiguities, can be understood by visualising the local sequences of variants, and some other features, such as STRs, can be annotated using information from third-party resources. Manual visualisation is not an easy task when selecting variants based on sequence context. Using graphical software, such as the Integrative Genomics Viewer [391], to view hundreds of variants with sequencing contexts is time consuming, and additional manual decisions are needed to determine sequence features. Several tools have been developed, the functions of which can be used to visualise the sequence contexts of genomic variants. Informative sequence context features can be shown in text without heavy graphical interfaces. Sequence context annotations can also be made by integrating external resources from public databases or analysis results from other tools. An analysis pipeline may be needed to annotate variants with this information. Comprehensive sequence annotations require merging diverse results from multiple tools within a good bioinformatics pipeline, but not all tools can be easily integrated together and building such a pipeline also require additional efforts.

4.4.1 Methods for viewing the sequence contexts of variants

To visualise the sequence context of genomic variants, the direct method uses a graphical viewer to output the sequences on screen. The Integrative Genomics Viewer is a high-performance, easy-to-use, interactive tool for the visualisation of genomic data. It can combine different common format genomic data, such as FASTA, BAM and VCF, to make an integrated visualisation. The sequence of the reference genome, the reads that are aligned and piled up against the reference, and the variants indicated by alignments can all be screened together for visualisation [391]. The SAMtools text alignment viewer can display alignments in a curses-based interactive viewer. The alignments of reads that are against the reference sequence can be visualised via a command line interface instead of a graphical interface [303].

Aside from these tools, public databases, such as the UCSC Genome Browser and the Ensembl Genome Browser, also have options that allow users to display custom data, such as variants in VCF format, and view them as tracks, together with other useful genomic information provided by the databases in graphical browsers [307,392]. Aside from graphical viewers, the sequence context can be visualised in text format. SeqTailor is a web server that can extract FASTA-format DNA or

protein sequences by considering the genomic variants in VCF files with user-defined genomic regions. The tool can also annotate the nearest splice sites to the given genomic variants if the splice sites reside within the extracted DNA sequences. The functions of SeqTailor can be used to retrieve information about the sequence contexts of variant sites [393]. UPS-indel is a universal positioning system used to mark the potential breakpoint ambiguities of indels. Both the web service and the source codes for the command line interface are available. This tool takes VCF files with the reference sequence as input and then outputs variants with an additional so-called UPS coordinate column in VCF format. The UPS coordinate is the equivalent genome region for an indel, which the tool uses to determine redundant indels and produce filtered VCF files after removing redundant indels [137]. The Variant Tools is software that provides a whole set of functions for the manipulation, annotation, selection, simulation, and analysis of variants in the context of NGS analysis. Several functions of Variant Tools can be used to extract flanking bases of reference and alternative alleles of genomic variants with a given range and output them on screen or in a text file. These functions are very useful in outputting the sequence context of variants and selecting variants based on contexts, such as variants in CpG islands [394].

Various pieces of the sequence context information of variants can be assessed through these methods. Issues such as ambiguous breakpoints of indels, equivalent indels, mutated sequence and flanking bases of variants can all be assessed. To obtain a comprehensive understanding of genomic variants, several tools are needed to acquire different pieces of information about the sequence context.

4.4.2 Methods for annotating variants in tandem repeats

As described in Section 3.1.4, the STRs around variants are the main feature of the sequence context of genomic variants. To annotate variants in STRs, information from public databases, tools which can analyse the STRs of genome sequences and tools that can directly annotate STRs around variants can be used.

Information on the STR can be fetched from public databases. For example, the ‘Simple Repeats’ and ‘RepeatMasker’ tracks from the UCSC Genome Browser contain different class repeats marked with coordinates of the human genome. The ‘Simple Repeats’ track was generated by the Tandem Repeats Finder (TRF), which is a programme for analysing DNA sequences with TRs. The TRF identifies TRs by percent mismatch and the frequency of indels between adjacent potential repeat units and uses statistically based recognition criteria. The ‘Simple Repeats’ track of the UCSC Genome Browser displays STRs and their associated information, including genome coordinates, sequence motifs, copy numbers, percentage of mismatch and indels, nucleotide frequencies and corresponding scores [395]. The ‘RepeatMasker’

track was generated using the RepeatMasker programme, which screens DNA sequences for interspersed repeats and low complexity and outputs a detailed annotation of the repeats present in the input DNA sequences. The results from RepeatMasker contain match scores, genome coordinates of repeats and classes of repeats [396]. Depending on the research purposes, STR information can be fetched by directly downloading the STR tracks from databases or using the corresponding tools to re-analyse the query sequences. The STR information from databases is usually generated using widely recognised parameters. However, these parameters may limit research to certain circumstances and may not be suitable for research with specific purposes. To fit special experimental needs, re-analysing the query sequence using tools with more flexible parameter settings is desired.

Aside from tools that are involved in public databases, other tools can annotate STRs with genome sequences. For example, SciRoKo is a user-friendly software tool for the identification of perfect STRs in genomic sequences developed based on seed extension [397]. Krait is a tool with a user-friendly graphic interface for the genome-wide investigation of STRs [398]. PERF is a tool for identifying perfect STRs in DNA sequences; it uses an exhaustive algorithm to search matched sub-strings of repeat sequences [399]. TRAL is a Python library that integrates multiple tandem repeat annotation tools and applies circular profile hidden Markov models to detect repeats [400]. These STR detection tools usually build repeat patterns with different modelling approaches and then find these patterns in the query sequence. After potential repeats are detected, different statistical criteria are applied to select candidate repeats and filter redundant repeats. The STRs of the query sequence, together with repeat statistics and classes of repeats, are formatted and reported [401].

Information on STR from these tracks or tools can be fetched and converted into BED format files, which are tab-delimited text format files used to store genomic regions as coordinates and associated annotations [402]. These BED files can then be used by annotation tools, such as ANNOVAR, to annotate genomic variants. Given a list of variants in VCF format of the custom format, ANNOVAR can perform the annotation of variants based on genes, regions, and specific filters with various resources from public databases. In addition, ANNOVAR has functions for users to annotate variants with their own annotation resources, which provides opportunities to use information that is not available in ANNOVAR integrated resources [368]. To make annotations of the sequence context of genomic variants, building a pipeline to integrate multiple tools is necessary.

Despite annotation with external resources, genome variants can also be annotated with STRs directly from variant calling tools. For example, the variant calling tool GATK can annotate variants with STR composition and counts per

allele. The variant calling output of GATK in VCF format can be refined using the GATK "TandemRepeat" function, which can add additional STR information as flags into the original VCF output [326].

4.5 LncRNA prediction methods

As described in section 3.1.5, lncRNAs have been considered transcriptional noise for a long time and are now revealed to have essential functions in numerous biological processes. To further understand the molecular mechanisms and biological functions of non-coding RNAs, accurately distinguishing them from protein-coding mRNAs is essential. The most common definition of lncRNAs is that they are non-coding RNA transcripts that are larger than 200 nucleotides [157–159]. Because of their similar sizes and features to non-coding mRNAs, predicting the coding potentiality of RNA transcripts and distinguishing lncRNA from non-coding mRNAs efficiently have remained challenging tasks. The most general approach to predicting the protein-coding potentiality of novel transcripts is analysing the features of ORFs. Searching sequence homologies between novel transcripts and known transcripts with sequence alignment-based methods is another strategy for predicting protein-coding potentiality. However, the nucleotide sequences of lncRNAs are poorly conserved, and sequence homologies may not be found between lncRNAs [403]. In addition, *in vivo* experiments for identifying lncRNAs, such as ribosome profiling, are useful and reliable, but such experiments are usually time consuming and expensive [404].

To better classify protein-coding and non-coding RNAs, various computational tools have been developed, and they use the advantage of NGS to efficiently identify lncRNAs. Many of these tools have been developed using machine learning or deep learning methods with various genomic features of RNAs. In addition, several lncRNA datasets have been built in the last 10 years, which also provide extended genomic information for computational method development. With all these efforts, significant improvements have been achieved in lncRNA prediction.

4.5.1 Features and models in computational lncRNA prediction

To distinguish lncRNAs from protein-coding mRNAs, many features have been used by existing lncRNA prediction methods. These features can be derived from the RNA sequences in gene transfer format or general feature format. *K*-mers features, transcript features and structure features are commonly used.

K-mer features

K-mers are specific sub-sequences of k consecutive nucleotides with different values of k. *K*-mer, together with its frequency profile, compositions, transitions and distributions, is a commonly used feature in lncRNA prediction [405–407]. The GC content is the percentage of nitrogenous bases G or C in a DNA or RNA molecule. A low level of GC content might indicate the non-coding potential of a transcript [408]. The GC content and their corresponding variances calculated for three reading frames and different codon positions have been used to describe non-coding potentiality [409].

Codons are sets of nucleotide triplets that translate a genetic code into a sequence of amino acids. Traditionally, codons are represented in an RNA codon table, but they can also be represented in a DNA codon table. A stop codon (UAA, UAG or UGA) can cause the termination of the translation process; thus, stop codon-based features can be used to recognise lncRNAs. The count or frequency of stop codons in a transcript and the corresponding three reading frames can be used as features [405].

The Fickett TESTCODE score is another popular feature calculated by combining the nucleotide position frequencies and base compositions of a transcript [410]. In lncRNA prediction, this score might be computed differently in different tools [406,409,411,412]. Hexamer-based features are also used in lncRNA prediction. For example, the hexamer score measures biased hexamer usage between coding and non-coding sequences, and it has been shown to have the most discriminative potential for identifying lncRNAs [413].

Transcript sequence features

An ORF is a reading frame located between a start codon (AUG) and a stop codon (UAA, UAG or UGA), and it has the potential to be translated into proteins. Many ORF-related features are widely used in computational lncRNA prediction because of the high correlation between ORFs and protein coding abilities. Moreover, because many lncRNAs have shorter ORFs than protein-coding mRNAs, the size of the longest or the first ORF is an important indicator in lncRNA prediction. However, some lncRNAs with short ORFs and longer but fewer exons can also be translated and produce short peptides, which have certain key biological functions in human development [414,415]. Instead, the size of the ORF is usually used indirectly. Open reading frame coverage is the ratio of the size of the longest ORF to the size of the whole transcript. Open reading frame integrity is defined as a Boolean value feature, which means that the ORF starts with a start codon and ends with a stop codon [406]. The entropy density profile, which describes the composition and *k*-mer of the sequence, and the ORF

frame score, which is the size variance of the ORF among three reading frames, are also used in the tool development of computational lncRNA prediction [405,416,417].

Similarly, coding sequence-related features are useful in identifying lncRNAs [174]. A coding sequence is the nucleotide sequence of a gene that can be translated into a protein. The length and percentage of coding sequence towards the transcript length can be used as features to distinguish lncRNAs [418]. In contrast to the coding sequence, untranslated regions cannot be translated and located on both sides of a transcript, which are referred to as the 5' and 3' untranslated regions. The length, coverage and ratio of the 5' and 3' untranslated regions to the transcript length are features used by lncRNA prediction tools [417]. In addition, the exon count and the average exon length of a transcript are exon-based features that can be used to distinguish lncRNAs [418].

Structure features

Structure-related features regarding the formation and stability of lncRNAs and their hypothetical peptides are also used in the prediction of non-coding potential. The peptides encoded by coding mRNAs and theoretically encoded by lncRNAs are supposed to differ in chemical properties. The isoelectric point, which indicates whether a peptide molecule carries any electrical charge and indicates the molecular weight of the peptide, is a widely used and important feature [409]. In addition, the grand average of hydropathicity, the stability of the predicted peptide, the minimum free energy and the number of paired and unpaired bases of RNA secondary structures are also considered features for lncRNA prediction [406,419–421].

4.5.2 Models in computational lncRNA prediction

To use these features to distinguish lncRNAs and protein-coding mRNAs, different machine learning models have been applied (Table 2). Machine learning models are trained to classify the categories of the input samples as classification tasks or to predict the output continuous values with input values as regression tasks. In lncRNA prediction, tools may directly predict whether a transcript is non-coding or coding by labelling it or giving a score to indicate its coding probability.

Table 2. List of several lncRNA prediction tools. LncRNA prediction tools can be built with deep learning models, including convolutional neural networks (CNNs), deep neural networks (DNNs), deep belief networks (DBNs) and recurrent neural networks (RNNs). Other machine learning models, including logistic regression (LR), random forest (RF) and support vector machine (SVM), can also be used. The input format of lncRNA prediction tools can be transcripts in BED, FASTA or gene transfer format (GTF).

Tool	Model	Features	Input format	Last updates*
COME [422]	RF	*k*-mer, sequence	GTF	2016-05
CNCI [408]	SVM	sequence	GTF	2015-05
CPAT [411]	LR	Fickett, *k*-mer, ORF	BED, FASTA	2021-05
CPC2 [409]	SVM	Fickett, ORF, structure	FASTA	2020-01
CPPred [406]	SVM	ORF, *k*-mer, structure	FASTA	2019-02
FEElnc [423]	RF	ORF; *k*-mer	GTF	2022-07
IRSOM [424]	DNN	ORF, *k*-mer, sequence	FASTA	2022-04
iSeeRNA [425]	SVM	ORF, *k*-mer, sequence	BED, GTF	2014-04
lncADeep [417]	DBN	Fickett, ORF, *k*-mer	FASTA	2017-11
LncFinder [420]	SVM	ORF, *k*-mer, structure	FASTA	2021-12
Lncident [426]	SVM	ORF, *k*-mer	FASTA	2016-10
LncRNAnet [419]	CNN	ORF	FASTA	2018-08
lncRScan-SVM [418]	SVM	ORF, *k*-mer, sequence	GTF, FASTA	2015-08
lncScore [427]	LR	ORF, sequence	BED, FASTA	2016-09
Longdist [428]	SVM	ORF	FASTA	2017-09
mRNN [429]	RNN	*k*-mer	FASTA	2018-04
PLEK [430]	SVM	*k*-mer	FASTA	2016-01
RNAsamba [431]	RNN	ORF, *k*-mer, sequence	FASTA	2021-04

* The last updates of tools were recorded before 28/10/2022.

Logistic regression

The logistic regression model predicts the probability of an event using log odds [432]. The log odds of the event can be calculated with a linear combination of one or more independent features. The goal of logistic regression model is learning the coefficients and intercept, which maximises the probability of predicting correct labels.

Support vector machine

The support vector machine model predicts the probability of an event by distinctly classifying data points with the hyperplane in an N-dimensional space, which corresponds to N features of the data [433]. The goal of the support vector machine model is to find the best hyperplane that has the maximum margin. To find the best hyperplane, the support vector machine maps features from the original space to a higher dimensional space via a certain kernel function, which makes the samples linearly separable. The choice of kernel function is the key to support vector machine.

Random forest

Random forest is a type of ensemble learning model that builds and combines a multitude of decision trees to perform classification or regression tasks [434,435]. The idea of a random forest is that, given a training set, it selects random samples in a training set with replacement and then trains multiple decision trees for these samples. The classification task returns the predicted label, which is the class selected by most trees. The regression task returns the predicted value, which is the mean or average prediction of the individual trees.

Deep learning models

Deep learning models have been used in lncRNA studies, including deep neural networks, deep belief networks, convolutional neural networks, and recurrent neural networks [436]. Typically, a neural network consists of several connected layers: the input layer, the hidden layer(s) and the output layer [437]. A layer consists of several neurons. The input layer takes input data and presents them to the hidden layers through weighted sums or kernels. To propagate values from one layer to the next, non-linear activation functions, such as sigmoid, ReLU or tanh, are used. Finally, the output layer generates the final output as categorical values for classification or as numerical values for regression. The error back propagation algorithm, which was developed based on gradient descent, is the main training algorithm for the deep learning model [438]. To set up reasonable gradient descent learning rates for the changing weights of hidden layers, several popular optimisers, such as root mean squared propagation and the adaptive gradient algorithm, can be used for self-changing learning rates based on the training processes. The calculation for a neuron, which is the basic element of the deep learning model, is shown in Figure 6.

$$Z = \sum_{i=1}^{m} w_i x_i, \quad i = 1,2, \ldots m \tag{1}$$

$$y = f\ (Z + b) \tag{2}$$

where each x represents an input value, each w represents a weight, Z represents the linear sum of all the weighted input values, b represents the bias value of the neuron, f represents the activation function, and y represents the output value of the neuron, which is also one of the input values for the next layer.

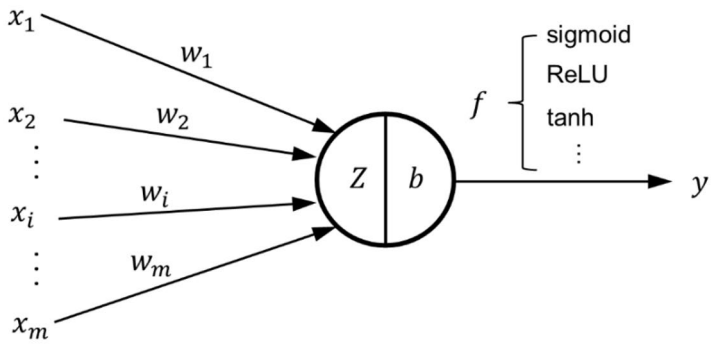

Figure 6. Example of a neuron in a deep learning model.

A simple deep neural network has multiple hidden layers that are fully connected with one another and that learn the representations of input data [439] (Figure 7). A deep belief networks consists of multiple layers, and each layer consists of a restricted Boltzmann machine, which is used to represent more abstract features. In a deep belief networks, training processes are performed in each layer at a time, and the output values of each layer are the input to the following layer [440]. A convolutional neural network is a hierarchical model with multiple layers that are trained with one-, two-, or three-dimensional convolutional kernels. Different layers, such as convolutional layers, pooling layers, and a fully connected layer, are usually involved. The convolutional layers of a convolutional neural network are used to extract the spatial features from the input data, and the pooling layer is used to reduce dimensions and reserve the most significant features of the data after convolution steps [439]. A recurrent neural network consists of an input layer, multiple recurrent hidden layers that have one or more feedback loops and an output layer. The recurrent connections of the model make data connected over a period of time and activated from time steps [441].

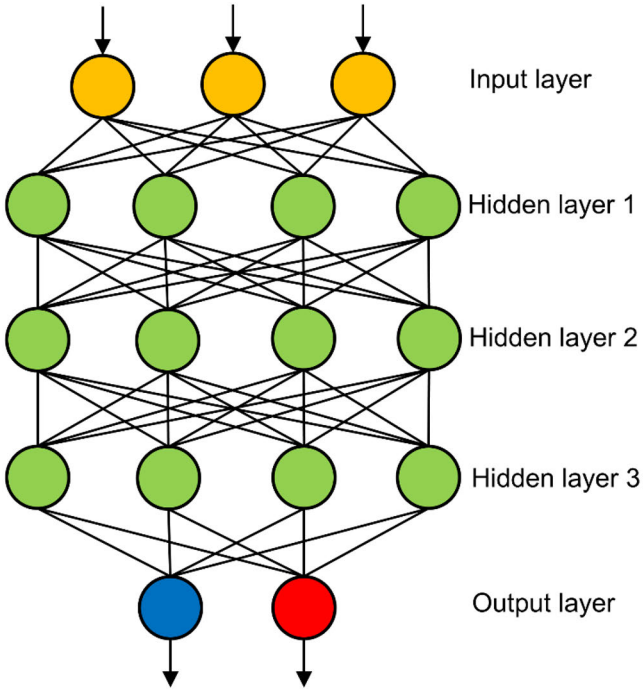

Figure 7. Example of a fully connected deep neural network that contains three hidden layers. A neuron in one hidden layer is fully connected with all the neurons in adjacent layers but is not connected with neurons in the same layer. The output layer may return numeric values or categorical classifications.

4.5.3 Current evaluations for lncRNA prediction tools

During the last 10 years, many efforts have been made to evaluate the performance of lncRNA prediction tools. Han et al. evaluated eight lncRNA prediction tools using human and mouse transcripts from several public resources and databases. The overall performance and application scope of each tool were assessed [442]. Zhang et al. comprehensively reviewed computational methods for non-coding RNA prediction and grouped methods into four main categories based on their algorithms: homology-, *de novo*-, transcriptional sequencing- and RNA family-based methods [443]. Antonov et al. assessed the performance of nine lncRNA prediction tools using human and mouse transcripts from GENCODE. Tools were trained, if necessary, and tested with balanced lncRNAs and protein-coding mRNA datasets, which further grouped the mRNAs into the long and short coding sequence sets. This study concluded that most of the lncRNA prediction tools had similar and good performance, and distinguishing mRNAs with short coding sequences from lncRNAs remained a challenge [444]. Negri et al. evaluated eight lncRNA prediction tools that were trained on plants or human transcripts. The transcript data of plants

and humans were fetched from public datasets or databases, such as FANTOM or Ensembl, and the tools were tested with plant or human data based on their categories. Overall performance, such as sensitivity, specificity, accuracy, and F1-score, was assessed. This study concluded that tools developed and trained with plant data and tools that used human data showed differences, which may indicate variations between lncRNAs in plants and humans [445]. Duan et al. evaluated the performance of 14 lncRNA prediction tools with different datasets, including high- and low-quality data from 33 species. Computational efficiency and the robustness of each selected tool were assessed. In their study, no tool was superior to others under all test conditions, but joint prediction could behave better than the use of a single tool [446]. Amin et al. comprehensively reviewed the computational prediction methods for different non-coding RNA types, including lncRNAs, circular RNAs and small non-coding RNAs. In their study, three deep learning-based lncRNA prediction tools were evaluated with human and mouse transcripts from the GENCODE database [447]. Xu et al. systematically reviewed current bioinformatics approaches for lncRNA prediction. A comprehensive review was conducted for computational lncRNA prediction methods, lncRNA databases and features for predicting lncRNAs. In addition, the authors briefly evaluated nine lncRNA prediction methods and built a Python package by integrating them, with the aim of providing a simple way for lncRNA prediction [24]. Zheng et al. assessed 17 tools on several public datasets and databases. The general performance and the performance on different size ranges of the transcripts of the selected lncRNA prediction tools were assessed. The significance of the transcript features used for coding potentiality prediction was also evaluated. The conclusion of this study was that deep learning-based tools performed better than the other algorithms did. The tools did not perform well with too short or too long transcripts. Certain transcript features, including ORF size and coverage, 3-mer, 6-mer and hexamer scores, and Fickett scores, contributed significantly to lncRNA prediction [413].

As deep learning methods become more popular in the bioinformatics field, the performance of deep learning-based tools should be assessed. In addition, because of real-world transcriptome data analysis, the proportion of lncRNAs and protein-coding mRNAs may vary. In **Publication III** of this thesis, 15 lncRNA prediction tools were evaluated with human transcripts from the GENCODE and Lncipedia databases. Different sizes of transcript sets were created with balanced and imbalanced numbers of lncRNAs and protein-coding mRNAs.

5 Materials and methods

5.1 Dataset

The semi-simulated WGS dataset

To test the theoretical limitations of each candidate variant calling tool for indel calling, a semi-simulated WGS dataset was created in the study of **Publication I**. The semi-simulated WGS dataset represents that indels are adopted from a real human, while the corresponding sequencing data were generated artificially. As described in section 3.1.3, indels are not totally randomly distributed in the human genome. To challenge variant calling tools with realistic indels from the human genome, indels from the HuRef genome, also known as J. Craig Venter's genome, were selected. The HuRef genome is the genome sequence of an individual human, which was sequenced using Sanger-based whole-genome shotgun paired-end sequencing and assembled using Celera Assembler. The variants of the HuRef genome were identified through a comparison within the maternal and paternal HuRef chromosomes and a comparison between the HuRef genome and GRCh36/hg18. The variants were then filtered by quality value and read location. Polymerase chain reaction-based experimental and computational verifications of the selected variants were conducted to assess the accuracy of variant calling. In total, more than 4.1 million variants, including 851,575 indels with a size range of 1–82,711 bp, were identified [80]. To simplify and reduce computational cost, only indels with a size range of 1– 5,000 bp from chromosomes 1 and 2 were selected in the study of **Publication I**. Chromosomes 1 and 2 of hg19 were reconstructed as the simulated genome by inserting the selected HuRef indels into their corresponding positions after liftover from hg18 to hg19. Two haplotypes were constructed by randomly selecting HuRef indels from different size ranges and inserting them into one of the haplotypes as heterozygous variants or into both haplotypes as homozygous variants. In total, 43,066 insertions and 45,223 deletions were included in the evaluation study. The simulated paired-end sequencing data were created using the NGS simulation tool ART [448] with three different coverages of 5×, 30× and 60× with a read length of 100 bp. An additional 30× coverage sequencing data with a read length of 250 bp was also created. The

reads of the two haplotypes were produced equally to represent the natural diploid genome as the human genome.

The Genome in a Bottle Consortium

The GIAB Consortium is a public–private–academic consortium hosted by the National Institute of Standards and Technology of the US. The GIAB authorities characterises human genomes with the aim of benchmarking, analytical validation, and technology development. The GIAB has currently characterised seven reference samples using sequencing data from multiple technologies and an integration computational pipeline to form high-confidence SNPs and small indel genotype calls [382]. To represent real-world sequencing data, WES data from Oslo University Hospital [449] and the corresponding variant set of human individual NA24385 from GIAB v3.3.2 were chosen in **Publication I** to evaluate variant calling tools for small indel calling. In total, 5,436 small indels located in both exome and high-confidence variant calling regions were involved. Since the development of GIAB v4.2.1, a new benchmarking dataset was produced, and it included challenging genomic regions. Genome in a Bottle Consortium v4.2.1 used both NGS and third-generation, long read sequencing technologies to sequence human samples. A carefully designed and integrated bioinformatics pipeline that included conventional and deep learning-based approaches was used to call high-confidence variants. Genome in a Bottle Consortium v4.2.1 covered 92% of the autosomal GRCh38 assembly and contained variants in clinically relevant genes not covered previously [450]. The high-confidence small variant from a son/father/mother trio of Ashkenazi Jewish and another of Han Chinese ancestry (HG002-HG007) with an average of 3,978,097 variants per individual were selected in **Publication II** to assess the sequence context of variants using VarSCAT.

The CHM1 cell line WGS dataset

The CHM1 cell line is a human haploid hydatidiform mole that lacks allelic variation. Hydatidiform moles are the result of a type of abnormal pregnancy in which an abnormal egg is impregnated by an ordinary sperm. The abnormal egg has no nuclear DNA, and the sperm doubles its own DNA, which results in two identical copies of each chromosome in every cell dividing from the mole. The CHM1 cell line is generated from one of these hydatidiform moles and has become an industry standard. Mark Chaisson et al. used single-molecule, real-time sequencing technology at a 54× WGS coverage with the CHM1 cell line to identify SVs and gaps in the CHM1 genome [81]. Using custom SV calling algorithms, a total of 26,079 large indels ≥ 50 bp within the euchromatic portion of the CHM1 genome were identified against GRCh37. Meanwhile, they also generated 41× Illumina WGS

data of the CHM1 cell line for their comparison analysis. The Illumina WGS data and 18,467 indels between 50 and 10,000 bp of the CHM1 cell line from Mark Chaisson et al. were used in the study in **Publication I**.

Platinum Genomes

The Platinum Genomes contains a set of high-confidence small variant calls for human individuals NA12877 and NA12878, generated using the WGS of 17 individuals in a three-generation pedigree and a range of publicly available bioinformatics tools. The high-confidence variant sets were produced using haplotype transmission information based on the inheritance constraints in the pedigree and the concordance of variant calls across different bioinformatics pipelines. The Platinum Genomes contains 4.7 million SNVs and 0.7 million small indels, which are consistent with the inheritance of the parents' and 11 children's pedigrees. In total, the Platinum Genomes was reported to cover around 97% of the total genes and the reference sequence of GRCh37/hg19 [63]. In 2017, the Platinum Genomes released an updated version of high-confidence small variant calls based on the human assembly GRCh38/ hg38. This high-confidence small variant set of NA12877 and NA12878 was used in **Publication II** to assess the sequence context of variants using VarSCAT.

The 1000 Genomes Project

The 1000 Genomes Project is a comprehensive catalogue of common human genetic variations. The goal of this project was to find common genetic variants with frequencies of at least 1% in the populations studied. The project was initiated in 2007, and it was planned to have four stages: a pilot phase and three phases of the main project. The phase 3 analysis was completed in 2015, and it contained 2,504 individuals from 26 populations (during the time of thesis writing, the phase 4 analysis was due for publication [451,452]). In **Publication II**, a variant set from 2,548 human individuals spanning 26 populations from a phase 3 extension were selected to assess the impact of TRs on genomic variants with the computational tool VarSCAT. The sample integrated variant set was produced *de novo* on GRCh38 by WGS and a multi-caller integrated bioinformatics pipeline. Each individual in this variant set contains on average 4,144,924 biallelic SNVs and small indels [453,454].

The ClinVar database

The ClinVar database is a public archive held by the National Center for Biotechnology Information, with the aim of reporting the relationships and supporting evidence of human variation and observed health status. ClinVar requires submissions

of variants found in patient samples with clinical significance and supporting data. The variants in ClinVar are mapped to reference sequences and reported according to the HGVS standard [455]. In total, 117,409 indels from ClinVar (date: 2022/01/09) which located on autosomes, sex chromosomes and the mitochondrial chromosome were selected in **Publication II** to assess the breakpoint ambiguity of indels. The summary of datasets used in **Publication I** and **II** is shown below (Table 3).

Table 3. Datasets used in Publication I and Publication II.

Data set	NGS data	Types of variants	Study of publication	Additional information
HuRef semi-simulated dataset	WGS	Indels (1-5000 bp)	I	Indels in chromosome 1 and 2
GIAB NA24385 (v.3.3.2)	WES	Small indels (1-50 bp)	I	High-confidence variants
CHM1	WGS	Large indels (50-10,000 bp)	I	Called by long read sequencing
GIAB HG002-HG007 (v.4.2.1)	Not used	Individual SNV small indels	II	High-confidence variants
Platinum Genomes	Not used	Individual SNV small indels	II	High-confidence variants
1000 Genomes Project	Not used	Individual SNV small indels	II	biallelic variants
ClinVar (date:2022/01/09)	Not used	Indels with clinical significance	II	Indels in database

The GENCODE Project: Encyclopedia of genes and gene variants

The database of the GENCODE project contains the definitive annotation of functional elements in the human and mouse genomes. By using manual curation, computational analysis and targeted experimental approaches, the annotation of all evidence-based gene features, including protein-coding genes, non-coding genes, pseudogenes and alternative splice variants in the human and mouse genomes, has been conducted, enhanced, and extended with high accuracy. The current GENCODE release (release 43) records 62,703 total genes, of which 19,928 are lncRNA genes and 19,393 are protein-coding genes. As for transcripts, 252,913 total transcripts are recorded in the current GENCODE, of which 58,023 are lncRNA loci transcripts and 84,411 are protein-coding mRNAs. The transcripts in GENCODE have three different confidence levels indicating the levels at which the transcripts have been validated. Level 1 is the validated level of transcripts verified experimentally using real-time PCR and sequencing through the GENCODE experimental pipeline. Level 2 is the Havana annotation level, which means that the transcripts are manually annotated. Level 3 is an automatically annotated level,

which means that the annotations of transcripts from different sources may not be consistent [13]. The transcripts from GENCODE are formatted in tab-separated standard gene transfer format or general feature format. The protein-coding mRNAs above level 2 from GENCODE were used in **Publication III**.

The LNCipedia database

The LNCipedia database is a comprehensive compendium of human lncRNAs. The version 5.2 contains 127,802 transcripts and 56,946 genes that come from integrated sources, including Ensembl, GENCODE, Refseq and other lncRNA-related studies. The aim of LNCipedia is to merge transcripts from different data sources into a highly consistent database. Transcripts from different sources are added to the database using custom import scripts and then filtered based on mapping abilities, the sizes of the transcripts and the overlapping of exons between coding sequences. Different lncRNA transcripts are then clustered based on their locations relative to genes and are named accordingly. The transcripts from LNCipedia can be downloaded in various formats, such as BED format, gene transfer format, general feature format and FASTA format, or accessed by application programming interface in JSON format [456]. The lncRNAs from the LNCipedia databases were used in **Publication III**.

5.2 Methods

5.2.1 Methods for evaluating variant calling tools for indel calling

5.2.1.1 Variant calling tools and sequencing data selections

Eight variant calling tools, namely, DeepVariant [320], DELLY[321], FermiKit [323], GATK HaplotypeCaller (GATK HC) [326], Pindel [338], Platypus [337], Strelka2 [110], and VarScan [346], were selected to represent a variety of indel calling algorithms. Tools were basically evaluated with default parameters and settings with the assumption that users do not have advanced computational knowledge. One parameter of Pindel (minimum support reads to call an indel) was tuned because the default parameter generated too many FP indel calls[338,457].

The sequencing data, including four semi-simulated WGS datasets, the GIAB WES dataset and the CHM1 WGS dataset, were processed using quality control, necessary trimming, alignment, sorting, and indexing. All the sequencing data were processed against the human genome assembly hg19, and the processed alignment

files were used as input for the selected variant calling tools. FermiKit, which required paired-end sequencing FASTQ files as the input, directly took the sequencing data to call variants. Because DELLY is special design for large indel calling, the small indel calling evaluation with GIAB WES dataset was not applied for DELLY; Based on indel calling evaluation results from the semi-simulated dataset, only DELLY, FermiKit, GATK HC, Platypus, and Pindel has abilities to call large indel calling evaluation, because of that, they were selected and evaluated with the CHM1 WGS dataset. All the selected variant calling tools were applied with the semi-simulated WGS dataset to evaluate their theoretical indel calling limits. The 'Simple Repeats' track from the UCSC Genome Browser was used to annotate the FP indel calls of all the selected tools with semi-simulated WGS data.

5.2.1.2 Evaluation criteria

Although all the variant calling results from the selected tools were in VCF format, the detailed formats of indel calls were still not consistent among all the tools. Tools such as FermiKit, GATK HC, Platypus, Strelka2, and VarScan also output SNVs, which are not the topic of this evaluation study. The truth set of the semi-simulated WGS dataset and the CHM1 WGS dataset was not in VCF format, so evaluation tools were difficult to apply. To compare the results with the truth set, the indel calls of each tool from the semi-simulated WGS data and the CHM1 WGS data were extracted with tool-specific custom scripts. Positions, sizes, variant types and genotypes were the information to collect.

The semi-simulated WGS dataset contained a wide size range of indels. An assumption was made that the position and size deviations between the tool-detected indels and the true indels may be more critical for small indels than for large ones. For example, under the widely used position deviation of ± 5 bp, a true deletion of 1 bp may be matched with a tool-detected deletion of 1 bp but located 5 bp away. This deviation is too large and may cause a match of two different indels. Besides, the size deviation was considered to tolerate size differences between the tool-detected indels and the true indels. Therefore, instead of allowing a fixed position for indels, size-related positions and size deviations were used. In this evaluation study, a TP indel was defined if 1) the position deviation of the tool-detected indel was between ±10% of the true indel size with an upper limit of 50 bp, 2) the size deviation of the tool-detected indel was < 25% the true indel size, and 3) the genotypes between a tool-detected indel and the corresponding true indel should be consistent. These evaluation criteria allowed us to assess indel calling in a flexible manner, which led to small indel calling results being evaluated more strictly than large indel calling results.

As for the CHM1 WGS dataset, a TP indel was defined as a tool-detected indel that should have at least 20% overlap with a true indel. This criterion was typically

used in other studies which used the same CHM1 WGS dataset as the truth set for evaluation purposes [133,347]. Because the truth set of GIAB WES data was in VCF format, hap.py was used as the evaluation tool to assess the performance of each tool.

5.2.2 VarSCAT: Variant Sequence Context Annotation Tool

To assess the sequence contexts of genomic variants, the computational tool VarSCAT was developed (https://github.com/elolab/VarSCAT). VarSCAT has three modules: the variant normalisation module, the ambiguous variant annotation module and the tandem repeat annotation module. The input of VarSCAT requires a VCF file and the corresponding reference sequence in FASTA format; optionally, a BED format file can also be inputted, which provides the regions of interest. With a single command line to state the annotation options and parameters, the annotation results of the variants are written into a text file. The workflow of VarSCAT is shown in Figure 8.

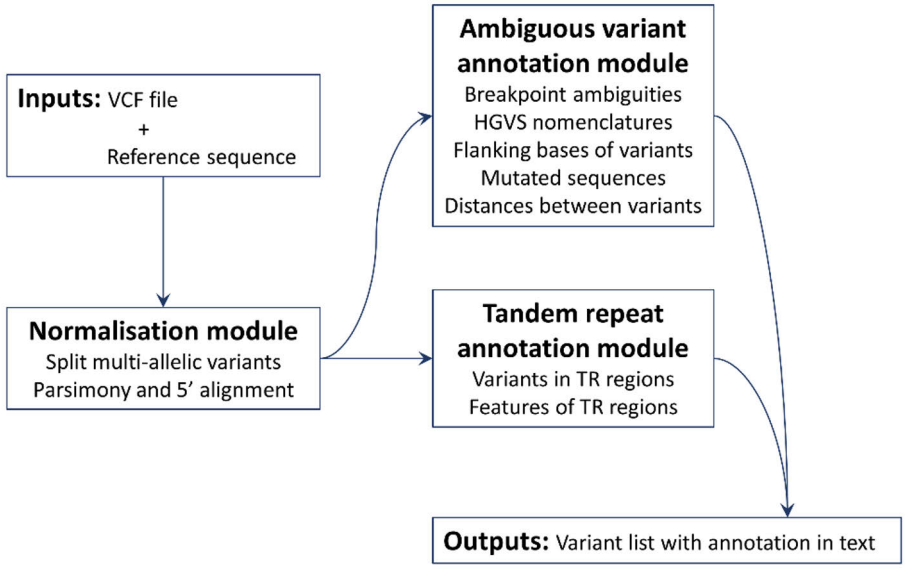

Figure 8. The workflow of VarSCAT. The ambiguous variant annotation module and the tandem repeat annotation module can be run together or individually.

The variant normalisation module

The aim of the variant normalisation module is to convert potential ambiguous variant as parsimonious and 5' aligned. A variant is parsimonious if and only if the variant is represented in as few nucleotides as possible without reducing the length of any allele to 0. A variant is 5' aligned if and only if the variant is no longer possible

to shift its position to 5' direction while keeping the length of all its alleles constant [139]. These potential ambiguous variants may have non-unique positions, references, and alternate sequences, which can cause inaccurate annotations for downstream analysis. Variants processed by the variant normalisation module will be represented as few nucleotides as possible, and 5' aligned. Any multi-allelic variants will be split into bi-allelic variants in this module.

The ambiguous variant annotation module

The ambiguous variant annotation module has functions to annotate variants, especially indels with breakpoint ambiguity-related issues. Information about variants, including 5' and 3' aligned positions, 3' edge positions (the rightmost mutated base on the coordinates of the reference sequence), the flanking bases of variants, HGVS nomenclature and distances to adjacent variants, can be annotated by this module. If a genome region is given, VarSCAT has a function to output the original reference sequence, the mutated sequence containing the variant and the complementary sequence of the mutated sequence in FASTA format.

The tandem repeat annotation module

The tandem repeat annotation module *de novo* searches the sequence context around variants based on the definition of TRs and determines the TRs with user-defined or default parameters. Tandem repeats around variants with different proportion of mismatches or gaps can be annotated, such as perfect and interrupted TRs. The algorithm of the tandem repeat annotation module first *de novo* searches all possible TR regions, which are at least 1 bp overlapped with variant sites (Figure 9). A variant site is the affected region of the variant on the coordinates of the reference sequence. For an SNV, an MNV, a deletion or a complex indel, the variant site is the changed bases of the reference sequence. For insertion, the variant site is the two adjacent bases of the inserted sequence.

Figure 9. A demonstration of how VarSCAT determines a TR. The sequence is chr19:246910-246000 the human assembly GRCh37/hg19. The VarSCAT determined TR is a motif "CCCTAA" repeated for 12 time. The red letter indicates a variant site. The rectangle blocks indicate each repeat unit of 12 repeats of "CCCTAA". Mismatches and gaps between repeat units are allowed in this TR.

The algorithm basically first chooses a candidate motif of a potential TR, which is a certain size of nucleotide sequence at least 1 bp overlapped with the variant site. Then, the algorithm global aligns this candidate motif with a potential repeat unit. The potential repeat unit is a nucleotide sequence with same length as the candidate motif, which located 5' or 3' direction within a user-defined or default distance to where the candidate motif is located. During this alignment, number of mismatches and the distance to the candidate motif are recorded. Within the certain distance, the alignment score of all the potential repeat units in terms of the candidate motif will be calculated based on a user-defined match score (MS), mismatch score (MIS) and gap score (GS) or default values. The potential repeat unit with the best alignment score will be selected, and its position was used to continue searching the next potential repeat unit towards its 5' or 3' direction. Meanwhile, the copy number of the candidate motif of the potential TR is plus by one. The searching is terminated when the alignment score of all the potential repeat units cannot pass the user-defined or default threshold. When the searching of potential repeat units terminated, and the candidate motif has both copy number and accumulated alignment score exceed user-defined or default thresholds, the whole searching region will be treated as a candidate TR. After all the candidate TRs are detected, a post-quality control process and a redundant removal process are applied based on a repeat score to make the tandem repeat annotation precise and clean. The alignment score reflects how well the repeat region against the corresponding motif. The repeat score considers the similar aspect as the alignment score but take the length of repeat region and the copy number of the motif into account. For a same repeat region with different motifs, the shorter motif is preferred, which can be used to filter out redundant motif representations (for example, for a repeat region "AAAAAAAAAA", motif "A" has higher repeat score than "AA", thus, motif "AA" will be filtered out and "A will be remained").

For a variant located in a TR, the output of this module contains information about the repeat motif, copy number, size of motif, start and end positions of the TR, the alignment score, the repeat score, GC content and match, mismatch, and gap percentages.

$$Alignment\ Score = MS \times match\ bases + MIS \times mismatch\ bases + GS\ \times gap\ bases \quad (3)$$

$$Repeat\ Score = \frac{Alignment\ Score}{length\ of\ the\ TR\ region} \times copy\ number \quad (4)$$

Benchmarking and biological analysis of VarSCAT

VarSCAT was benchmarked with GATK TandemRepeat [277], Krait [398], Tandem Repeat Finder (TRF) [395], and RepeatMarker [396] on the variant STR annotation of GIAB HG005 chromosome 1. The tandem repeat annotations of TRF and

RepeatMasker were downloaded from the UCSC Genome Browser, which were the tracks of 'Simple Repeats' and 'RepeatMasker', respectively. Krait (v1.3.3) was an STR sequence annotation tool. GATK TandemRepeat (v4.1.9.0) can directly annotate variants in STRs from a VCF file.

Indels from the ClinVar database, variants from the GIAB HG002-HG007, the Platinum Genomes NA12877 and NA12878, and the 1000 Genomes Projects were used to assess the proportions of breakpoint ambiguous indels and indels in STRs. Because different studies used various definitions of STRs [57,458], the benchmarking of VarSCAT was limited to perfect STRs with motif sizes of 1–6 bp, which was commonly used in TR-related studies [42,459,460]. One semi-random human indel set was also created. The positions of the semi-random indels were randomly selected, and the inserted sequences of the insertions were randomly generated using the DNA nucleotide alphabet. The indel size distribution and the total number of indels set were identical to the Platinum Genomes NA12878.

5.2.3 Methods for evaluating lncRNA prediction tools

Eight lncRNA prediction tools, namely, CPAT [411], CPC2 [409], IRSOM [424], LncADeep [417], LncFinder [420], longdist [428], mRNN [429] and RNAsamba [431], were selected to represent a variety of algorithms for coding potential prediction. The selected tools were run with default parameters or recommended parameters by the authors of the tools. None of the prediction models of the selected tools were re-trained in this evaluation study.

The protein-coding mRNAs from GENCODE (release 21) and lncRNAs from LNCipedia (version 5.2) were selected. The protein-coding mRNAs exist in both GRCh37/hg19 and GRCh38/hg38, and the overlapping lncRNAs in both GENCODE and LNCipedia were used in this evaluation study.

The test set was created with different sizes and proportions of lncRNAs by randomly selecting from LNCipedia and the protein-coding mRNAs from GENCODE. Four different sizes of datasets, namely, small (S), medium (M), large (L) and extra-large (XL), were created with a ratio of 4:6 for lncRNAs and protein-coding mRNAs, and another two datasets were created with more lncRNAs than protein-coding mRNAs (Lnc bias) and more protein-coding mRNAs than lncRNAs (PC bias), with ratios of 8:2 and 2:8, respectively.

5.2.4 Statistical metrics

Statistical metrics were applied to the evaluations of the variant calling tools and lncRNA prediction tools. For the assessment of the variant calling tools on indel calling, precision, recall, false discovery rate and F1 score were used as the statistical

metrics. For the biological analysis with VarSCAT, a Venn diagram and bar plots for the proportion analysis were used. Because of the imbalanced sample types in lncRNA prediction tool evaluation, sensitivity, specificity, precision, and balanced accuracy were used. All these statistical metrics were calculated from the numbers of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) labels. Of note, in the evaluation of indel calling, TN is not assessed because reference calls are not recorded in VCF format.

$$Sensitivity, true\ positive\ rate, recall\ = \frac{TP}{TP+FN} \tag{5}$$

$$Specificity, true\ negative\ rate = \frac{TN}{TN+FP} \tag{6}$$

$$Positive\ predictive\ value, precision = \frac{TP}{TP+FP} \tag{7}$$

$$Negative\ predictive\ value = \frac{TN}{TN+FN} \tag{8}$$

$$FDR = \frac{FP}{TP+FP} \tag{9}$$

$$F1\ Score = 2\ \times \frac{Precision \times Recall}{Precision+Recall} \tag{10}$$

$$Balanced\ accuracy = \frac{Sensitivity+Specificity}{2} \tag{11}$$

6 Results

6.1 Evaluation of indel calling tools

For tool evaluation on the semi-simulated WGS dataset, the performance of tools was assessed using different sizes of insertions and deletions (Figures 10 and 11). The results showed that indel calling on small indels is better than that on large indels, deletion calling is better than insertion calling, and the precision in tool performance varies less than the recalls between different sequencing settings. The trend is clear that with an increase in indel size, tool performance decreases. Although some tools were designed to call only small or large indels, the trend can also be observed with indels ≤ 50 bp. Large insertion calling does not perform well with all the selected tools.

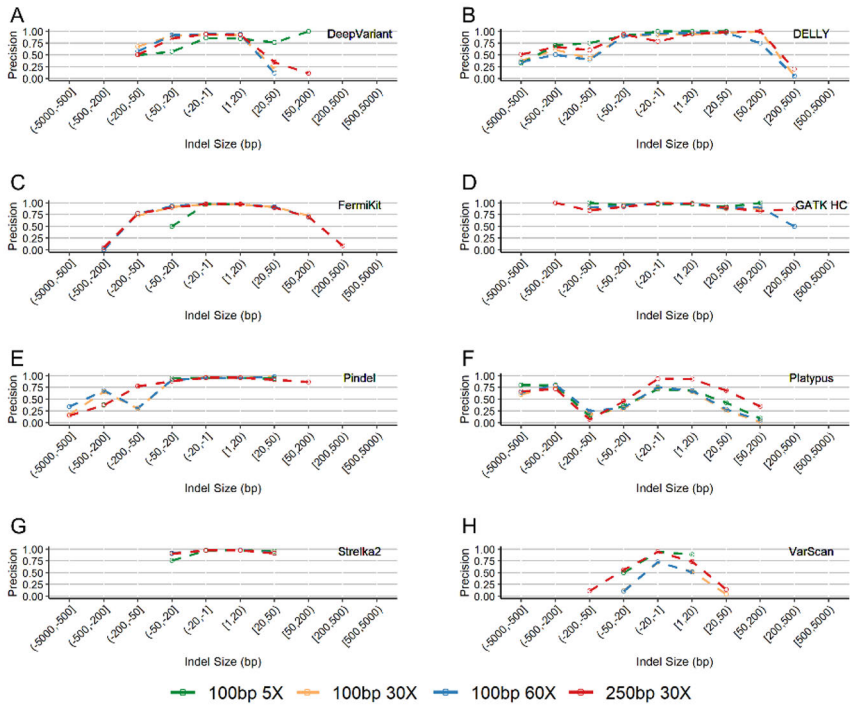

Figure 10. Precision curves of variant calling tools (A–H) on indel calling with four semi-simulated WGS datasets. The performance of tools is evaluated with insertions and deletions separately, in which intervals with negative values are deletions and positive values are insertions.

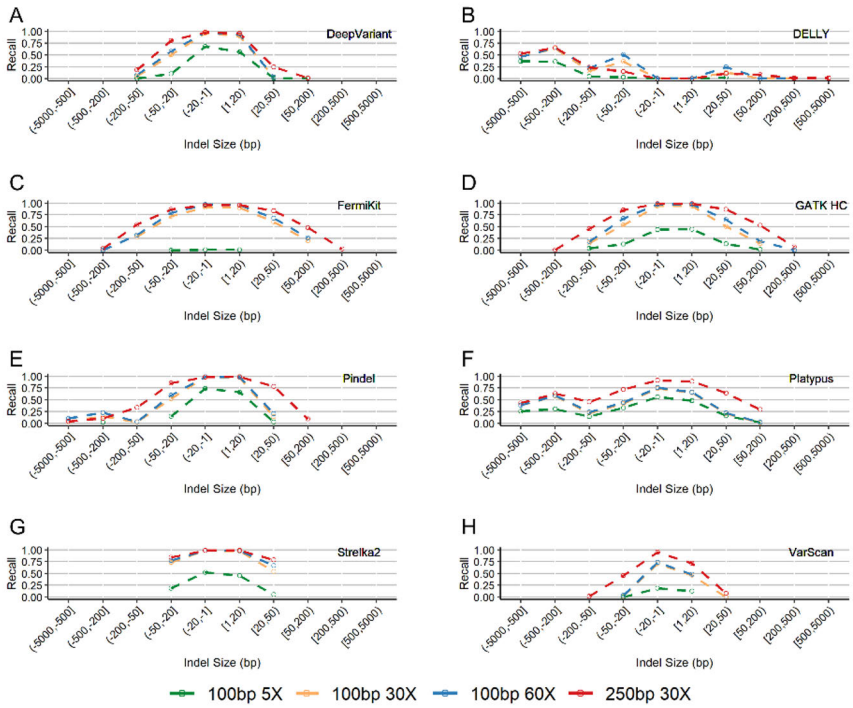

Figure 11. Recall curves of variant calling tools (A–H) on indel calling with four semi-simulated WGS datasets. The performance of tools is evaluated with insertions and deletions separately, in which intervals with negative values are deletions and positive values are insertions.

In terms of sequencing settings, higher sequencing coverage and longer read lengths can improve the performance of indel calling. However, this improvement became less obvious when the sequencing coverage was above 30×, which was consistent with previous suggestions and the recommendation of Illumina on WGS. Sequencing data with coverage of 5× may be not suitable for indel calling.

The tools without local re-assembly algorithms or good probabilistic models in this study were DELLY, Pindel, and VarScan, which did not perform well with indel genotyping, especially heterozygous indels (Table 4). The *de novo* assembly-based tool FermiKit was very good with large insertion calling, but it may not even work with low-coverage sequencing data.

Evaluation with real sequencing data showed that the machine learning-based tools DeepVariant and Strelka2 performed best with small indel calling (Table 5). Although machine learning-based tools showed good performance, DeepVariant and Strelka2 were still designed to be limited with small indel calling. The large indel calling evaluation with real sequencing data showed that all the selected tools had poor performance (Table 5). Part of the reason might be that the CHM1 dataset was

reported to have some inconsistency between single-molecule, real-time sequencing and NGS sequencing [133,347].

Table 4. Homozygous (HOM) and heterozygous (HET) indel calling precision of tools on different semi-simulated WGS data. Results with recall < 0.01 are not considered.

Data	100bp, 5X		100bp, 30X		100bp, 60X		250bp, 30X	
Type / Tool	HOM	HET	HOM	HET	HOM	HET	HOM	HET
DeepVariant	0.955	0.637	0.984	0.909	0.989	0.930	0.984	0.897
DELLY	0.829	0.176	0.966	0.423	0.971	0.463	0.953	0.450
FermiKit	0	0	0.966	0.932	0.992	0.936	0.978	0.959
GATK HC	0.972	0.977	0.994	0.967	0.993	0.968	0.990	0.966
Pindel	0.930	0.364	0.985	0.476	0.976	0.478	0.962	0.891
Platypsu	0.964	0.915	0.990	0.885	0.991	0.858	0.990	0.871
Strelka2	0.977	0.912	0.993	0.937	0.993	0.931	0.991	0.921
VarScan	0.979	0.125	0.992	0.384	0.991	0.400	0.992	0.604

Table 5. Evaluation results of variant calling tools on indel calling with real sequencing data. Metrics with a missing value '--' mean that the tool is not evaluated with the data.

Data	WES NA24385			WGS CHM1 cell line	
Metrics / Tool	Precision	Recall	F1 score	FDR	Sensitivity
DeepVariant	0.963	0.920	0.941	--	--
DELLY	--	--	--	0.738	0.061
FermiKit	0.909	0.530	0.718	0.232	0.032
GATK HC	0.896	0.910	0.903	0.281	0.048
Pindel	0.890	0.679	0.771	0.783	0.097
Platypsu	0.977	0.726	0.833	0.860	0.002
Strelka2	0.918	0.917	0.917	--	--
VarScan	0.839	0.710	0.770	--	--

Further investigation showed that more than half of the FP indel calls were located in the simple repeats of the human genome regardless of variant calling tools (Figure 12 A). The proportions of all indel calls in simple repeats are lower than that of FP indels calls in simple repeats (Figure 12), except for DELLY, which mainly called large indels. The results showed that FP indel calls were enriched in simple repeats, which may indicate that simple repeat is the main reason for FP calls.

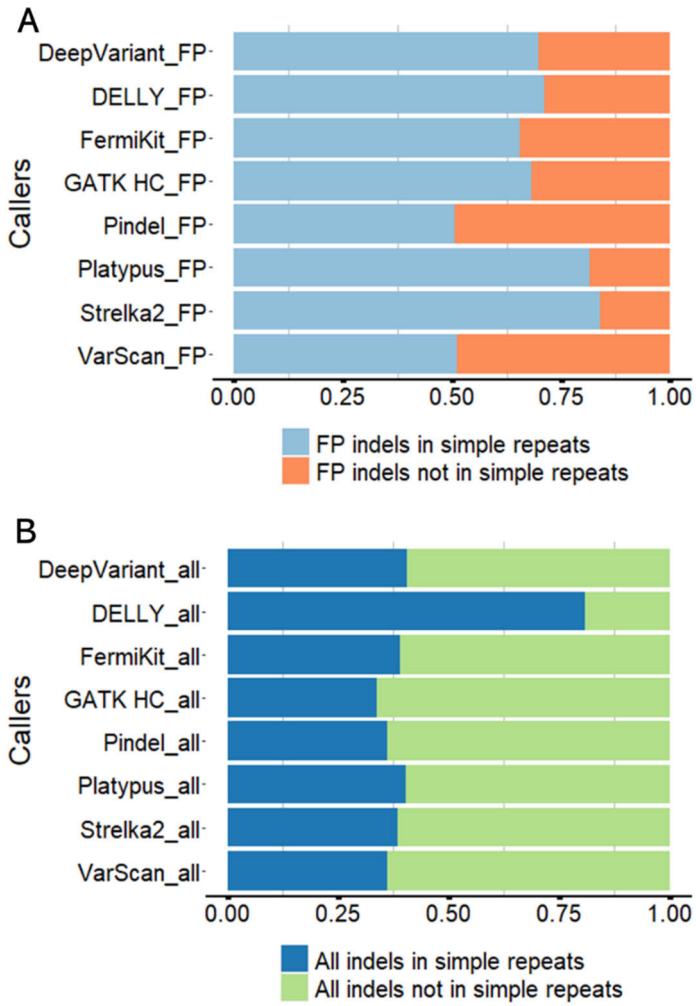

Figure 12. Proportion of (A) tools' FP indel calls and (B) tools' all indel calls in simple repeats and not in simple repeats.

In general, machine learning methods performed best in small indel calling. Machine learning methods integrated with local re-assembly algorithms were better than tools that used only local re-assembly algorithms without machine learning methods. Local re-assembly-based tools were good for detecting the correct genotypes of indels. However, because of the limited read length, these methods cannot detect large indels. *De novo* assembly-based tools were good with large insertion calling, but the genotyping of indels was not good. Split read-based and paired-end read-based methods were good for large indel calling, whereas gapped alignment-based tools were good for small indel calling.

6.2 The sequence contexts analysis with VarSCAT

6.2.1 The benchmarking of VarSCAT for STR annotations

The most critical question in human genome STR studies is the definition of STR. Different definitions, such as in terms of STR interruption and composition, may lead to significantly different results. In this study, the biological analysis was limited to perfect STRs with motif sizes of 1–6 bp and a minimum size of 10 bp. The minimum copy number was 10 for mononucleotide STRs, 5 for dinucleotide STRs and 4 for tri- to hexanucleotide STRs, as described in section 5.2.2.

Benchmarking results showed that the STR annotations for genomic variants had remarkable discordance between the different methods (Figure 13). Well-recognised STR resources, such as TRF and RepeatMasker, had strict STR criteria, which were limited only to large STRs. Short tandem repeat sequence annotation, such as Krait, may consider an STR in a wide region. Sub-STRs that are part of a large STR may not be recorded. Furthermore, although a comprehensive STR resource was acquired, the tool for variant annotation also plays an important role. In this study, ANNOVAR was selected as the annotation tool. This method only considers variants at their primary positions, which are the positions reported in VCF files. Variants, especially indels, are mutated regions instead of single positions. Thus, this method may miss some STR annotations for variants if only parts of the variants overlap with STRs.

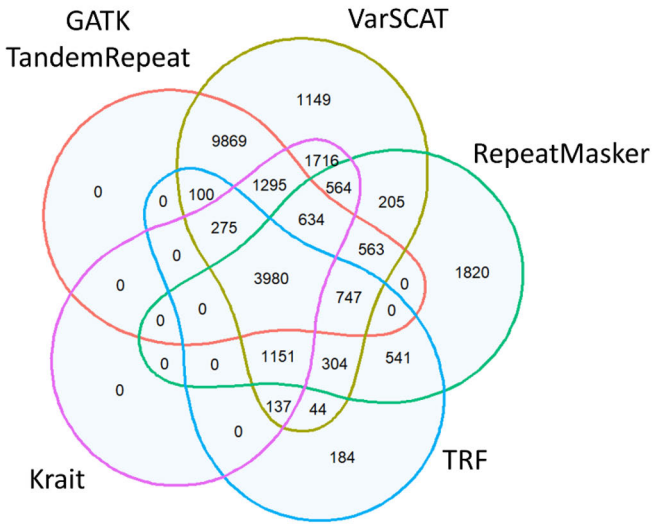

Figure 13. Benchmarking results of different STR annotation methods for variants on GIAB HG005 chromosome 1. The results are shown as a Venn diagram to demonstrate the shared STR annotations for variants among the different methods.

6.2.2 Sequence context of the variant in the genome scale

Using the comprehensive sequence context annotation tool VarSCAT, several high-confidence human individual germline variant sets, including 2,548 samples from the 1000 Genome Project, two human individuals from the Platinum Genomes and six human individuals from the GIAB, as well as the ClinVar database, which contains clinically related variants, were analysed. The population analysis with the 1000 Genomes Project showed that for each individual, around 7% of the total germline small variants and 35% of the total germline small indels were located in STRs (Figure 14). African populations had a lower proportion of variants in STRs, but the total number of variants was higher than that in other populations.

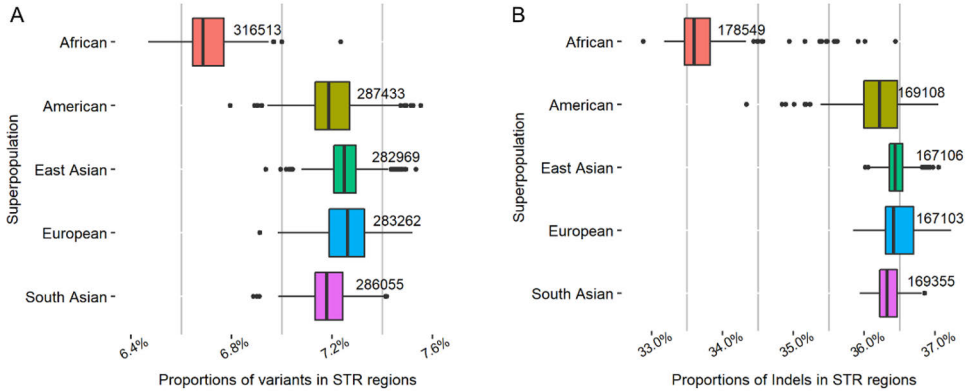

Figure 14. (A) Proportions of germline small variants in STRs among five human superpopulations. The number near each box is the average number of germline small variants in the STRs of the superpopulation. **(B)** Proportions of germline small indels in STRs among five human superpopulations. The number near each box is the average number of germline small indels in the STRs of the superpopulation.

For breakpoint ambiguity analysis, the results showed that around 90% of the germline indels of a human individual had breakpoint ambiguities, with insertions and deletions being nearly equally distributed (Figure 15A). Indels in ClinVar had a low proportion of breakpoint ambiguity. The reason might be that indels in ClinVar databases are clinically related, and the sizes of indels were longer than those of the majority of human germline indels, which increased the sequence complexity of the indel sequence pattern. The results of the simulated human indel set showed that random inserted indels had significantly lower proportions of ambiguous indels than real human indel sets, which indicated that the formation of human germline indels was heavily dependent on the sequence contexts. With eight high-confidence human germline variant sets from the Platinum Genomes and the GIAB, the results showed that around 85% of the germline variants located in STRs were indels (Figure 15B).

As far as my investigation in the literature review shows, this was the first time that the proportions of germline variants in STRs and ambiguous indels were measured at the human individual level.

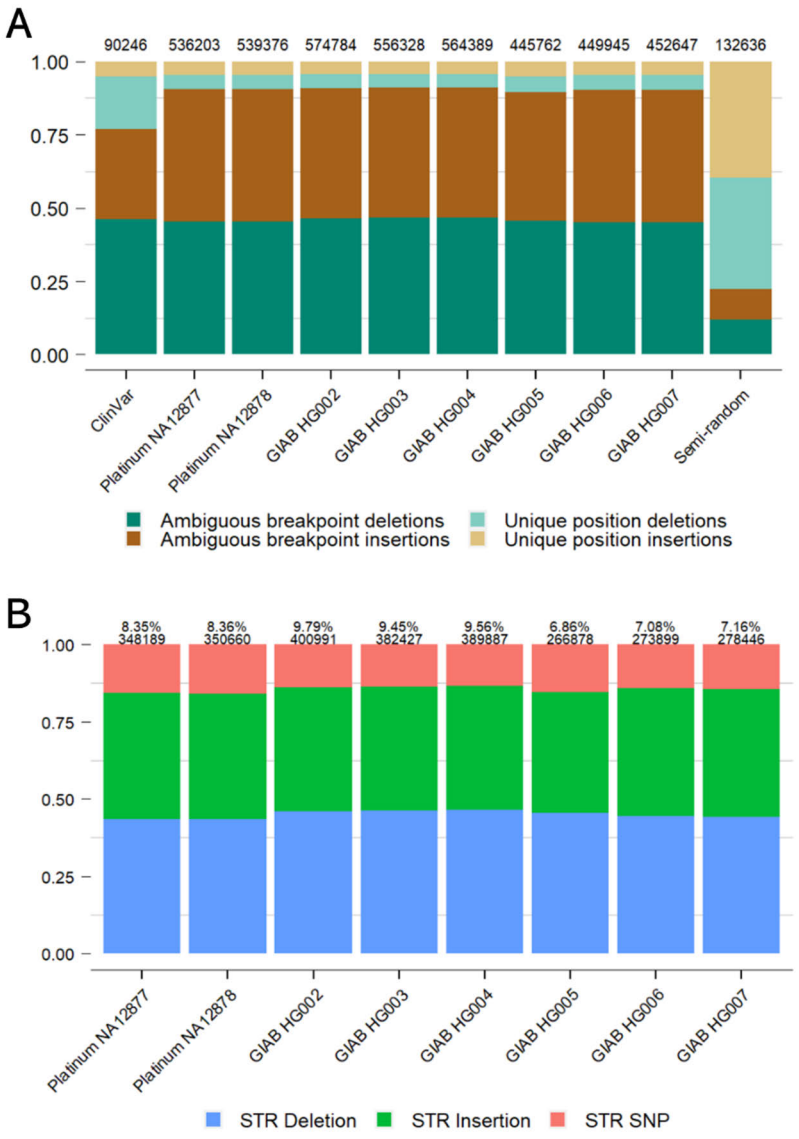

Figure 15. (A) Proportions of ambiguous breakpoint indels among eight high-confidence human germline variant sets, the ClinVar database and one semi-random indel set. The proportions of insertions and deletions are shown separately. The number on the top of each bar is the total number of breakpoint ambiguous indels for the indel set. **(B)** Proportions of variants in the STRs among eight high-confidence human germline variant sets. The number and percentage at the top of each bar are the total number and proportion of variants in the STRs of the variant set.

6.3 Evaluation of lncRNA prediction tools

The evaluation with different sizes of mixed transcript sets showed that deep learning-based tools, which are LncADeep, mRNN, and RNAsamba in this study, are the three lncRNA prediction tools that had the best TP rates, negative predictive values, and accuracy with all the datasets (Table 6). Although other evaluated metrics of deep learning-based tools are not always ranked at the top three, the values of these metrics are all comparable with those of other tools.

For the two biased datasets, it is unsurprising that compared with those of the non-biased datasets, the positive predictive values of all the tools are worse with the lncRNA biased dataset but better with the protein-coding mRNA biased set, and the negative predictive values of all the tools are worse with the protein-coding mRNA biased set but better with the lncRNA biased dataset. The reason might be that the tools were trained with protein-coding mRNA biased data, and the protein-coding mRNAs from the training data of the tools may include in the test set in this study.

In general, with all the test datasets, the TN rates of tools are higher than the TP rates, and the positive predictive values of tools are higher than the negative predictive values. These results indicate that tools may identify actual protein-coding mRNAs as non-coding transcripts, but they rarely identify actual lncRNAs as coding transcripts. In other words, the transcripts predicted by tools with coding labels were more reliable than the transcripts with non-coding labels. Compared with other methods, deep learning-based tools had remarkably more reliable predictions with transcripts with non-coding labels. Among these tested tools, LncADeep is the best tool, and longdist seems to be the worst one. The prediction results of longdist have many transcripts wrongly labelled as non-coding but are actually protein-coding mRNAs. Excluding deep learning-based tools, CPAT has the best performance.

Table 6. Evaluation results of lncRNA prediction tools with different sizes of transcript test sets. The evaluation metrics are true positive rate (TPR), true negative rate (TNR), positive predictive value (PPV), negative predictive value (NPV), and balanced accuracy (ACC). The transcript test sets include four different sizes: small (S), medium (M), large (L) and extra-large (XL). The two other transcript test sets are the biased lncRNA (Lnc bias) and biased protein-coding mRNA (PC bias) sets. All metrics consider protein-coding mRNA positives and lncRNAs negatives. Deep learning-based tools are shown in bold.

Data	**S, total transcripts: 46,563**					**M, total transcripts: 92,922**				
Metrics / Tool	**TPR**	**TNR**	**PPV**	**NPV**	**ACC**	**TPR**	**TNR**	**PPV**	**NPV**	**ACC**
CPAT	0.723	0.986	0.987	0.711	0.854	0.724	0.986	0.987	0.712	0.855
CPC2	0.487	0.996	0.996	0.574	0.742	0.484	0.998	0.997	0.572	0.741
IRSOM	0.558	0.932	0.932	0.596	0.750	0.557	0.944	0.933	0.595	0.749
LncADeep	**0.867**	**0.930**	**0.947**	**0.830**	**0.899**	**0.864**	**0.930**	**0.947**	**0.826**	**0.897**
LncFinder	0.667	0.985	0.984	0.671	0.825	0.667	0.985	0.985	0.672	0.826
longdist	0.214	0.975	0.925	0.464	0.594	0.211	0.975	0.924	0.462	0.593
mRNN	**0.849**	**0.921**	**0.940**	**0.809**	**0.885**	**0.848**	**0.923**	**0.941**	**0.807**	**0.885**
RNAsamba	**0.827**	**0.925**	**0.941**	**0.787**	**0.876**	**0.824**	**0.930**	**0.944**	**0.786**	**0.877**
Data	**L, total transcripts: 139,379**					**XL, total transcripts: 185,190**				
Metrics / Tool	**TPR**	**TNR**	**PPV**	**NPV**	**ACC**	**TPR**	**TNR**	**PPV**	**NPV**	**ACC**
CPAT	0.725	0.986	0.987	0.713	0.856	0.726	0.986	0.986	0.713	0.856
CPC2	0.486	0.998	0.997	0.573	0.742	0.488	0.998	0.997	0.574	0.743
IRSOM	0.557	0.942	0.933	0.595	0.750	0.558	0.944	0.933	0.596	0.750
LncADeep	**0.865**	**0.931**	**0.948**	**0.826**	**0.898**	**0.864**	**0.930**	**0.947**	**0.826**	**0.897**
LncFinder	0.669	0.985	0.985	0.673	0.827	0.670	0.985	0.985	0.674	0.828
longdist	0.211	0.974	0.922	0.462	0.593	0.211	0.974	0.922	0.462	0.593
mRNN	**0.850**	**0.923**	**0.941**	**0.810**	**0.886**	**0.850**	**0.922**	**0.940**	**0.809**	**0.886**
RNAsamba	**0.827**	**0.929**	**0.944**	**0.788**	**0.878**	**0.828**	**0.928**	**0.943**	**0.789**	**0.878**
Data	**Lnc bias, total transcripts: 122,265**					**PC bias, total transcripts: 128,502**				
Metrics / Tool	**TPR**	**TNR**	**PPV**	**NPV**	**ACC**	**TPR**	**TNR**	**PPV**	**NPV**	**ACC**
CPAT	0.723	0.985	0.935	0.924	0.854	0.726	0.986	0.997	0.385	0.856
CPC2	0.487	0.998	0.985	0.870	0.742	0.488	0.997	0.999	0.253	0.743
IRSOM	0.558	0.943	0.738	0.880	0.750	0.558	0.941	0.982	0.271	0.750
LncADeep	**0.866**	**0.930**	**0.782**	**0.960**	**0.898**	**0.865**	**0.932**	**0.986**	**0.546**	**0.898**
LncFinder	0.667	0.985	0.929	0.910	0.825	0.670	0.985	0.996	0.342	0.827
longdist	0.214	0.974	0.706	0.811	0.594	0.211	0.975	0.980	0.178	0.593
mRNN	**0.849**	**0.922**	**0.760**	**0.955**	**0.886**	**0.850**	**0.921**	**0.984**	**0.517**	**0.886**
RNAsamba	**0.827**	**0.929**	**0.771**	**0.949**	**0.878**	**0.828**	**0.925**	**0.984**	**0.484**	**0.876**

7 Discussion

7.1 Evaluation of variant calling tools on indel calling

The development of high-throughput sequencing technologies has greatly facilitated the studies of the human genome. A large number of algorithms and tools have been developed to analyse sequencing data. However, many tools were developed with similar algorithms and have functions with the same purpose, which may cause trouble for users to decide which tool can best fit their research. Users may select a tool based on the number of citation, journal impact, or author reputation, but these metrics may not be reliable predictors of software accuracy [461]. Tools published early are more likely to appear in high impact journals and have high number of citation due to their novelty, but without maintenance and further development, they may be outperformed by subsequent tools. To assess tools' performance, one good way is benchmarking tools and evaluate their performance with different aspects.

Indels are widely exist in the human genome and some of them have shown strong associations with human health (section 3.1.3). To assess which algorithms and tools are suitable to call particular sizes of indels, in **Publication I,** eight variant calling tools representing a variety of indel calling algorithms were evaluated comprehensively on different size ranges of indels with both simulated and real NGS data. Previous indel calling tools or algorithms evaluations were mostly focus on either small indel calling [374,376,377], or large genome changes such as SVs [111,389], but not indels with different size ranges. Insertions and deletions have their specific difficulties in variant calling process, but only limited previous evaluation research considered them as two types of variants [381]. My research in **Publication I** evaluated variant calling tools with insertions and deletions separately and benchmarked their indel calling performance with different size ranges of indels. My work made up for the deficiencies of previous studies and provide a more comprehensive indel calling evaluation of variant calling tools.

To facilitate a fair, comprehensive, and in-depth evaluation of tools for indel calling, several essential aspects need to be considered and designed carefully. First, because the concept of indel includes insertion and deletion, which cover a wide range of genome changes, an evaluation of tools for indel calling should consider the

different characteristic features of both insertions and deletions with a wide size range. Second, because the short read length in NGS cannot fully resolve all sizes of indels, the algorithms of calling small (typically < 50 bp) and large (typically ≥ 50 bp) indels use remarkably different information fetched from sequencing data, which may lead to variations in the best indel calling ranges. A good evaluation of tools for indel calling should include representative tools with various algorithms that cover the current computational indel calling field. Third, the different technical parameters of sequencing, such as sequencing coverage and read length, should also be considered to challenge tools. Fourth, an indel may have several different representations based on the complexity of sequence context. A good evaluation should clearly define reasonable criteria for labelling indels. Fifth, the selection of benchmarking datasets is important to conduct a fair evaluation. A good benchmarking dataset should 1) contain a wide range of indels that are precisely marked and can represent the nature of the human genome, 2) contain data with different technical parameters of sequencing, 3) avoid potential evaluation biases (the candidate tools of evaluation study are not solely or heavily involved in the generation process of the benchmarking dataset), and 4) choose the dataset in which the sequencing data and the corresponding indel truth set are generated by the same study and are both publicly available.

The results of **Publication I** suggested that no single tool or algorithm is suitable for all circumstances of indel calling. In general, the results agreed with those of previous indel calling evaluation studies that higher sequencing coverage and a longer read length are preferred. The performance of tools on large indels with CHM1 WGS data was worse than the performance with semi-simulated sequencing data. The reason might be that the semi-simulated data, including both large indels and corresponding sequencing reads, was too ideal for tools. The inconsistent evaluation results from real and simulate dataset can indicate that why tool evaluation cannot solely rely on simulated data. Regarding the purpose of assessing the indel calling performance of tools with different indel size ranges, the results showed that for calling large indels, specific algorithms are preferred, which agreed with previous studies [111,389]. Because of the short read length in NGS, a large indel may not be fully resolved by sequencing reads. The high-quality mapped sequencing reads have limited mapping information for a large indel, thus, specific algorithms are needed.

With the current existed indel calling tools, one method to improve the accuracy of indel calling is creating ensembled variant calling tool. An indel that detected by multiple calling tools may not be an FP. Furthermore, by integrating indel calling tools with different underlying algorithms, an ensembled variant calling tool may have potential abilities to call a wide size range of indels. To make this idea even better, an ensembled variant calling tool may even have abilities to call various types

of variants, including SNV, MNV, indels, inversions, translocations, tandem repeats, CNVs, and so on. This among of work may lead the idea of "ensemble" into genomic variant calling platforms or pipelines, and these platforms or pipelines may have huge potential market in research organizations or hospitals [462].

New sequencing methods are always developed, but this does not mean that old methods will be phased out. Sanger sequencing, as first-generation sequencing, still plays an important role in the validation of clinical diagnosis [463]. Next-generation sequencing has been developed for almost two decades, but its ability to capture variants with very low allele frequencies makes it the crucial first step in clinical diagnosis at the molecular level. Thus, research on NGS and the corresponding computational methods is still needed to improve their performance.

One question that has not been answered well is the performance of variant calling tools with somatic SVs. Efforts have been made to evaluate tools, but a high-quality, well-recognised somatic SV benchmarking tumour-normal sequencing dataset remains missing. Without a standard benchmarking dataset, the potential data or experimental design biases in evaluation studies are difficult to assess, thus making the evaluation results less persuasive. Once this kind of benchmarking dataset is available, the evaluation of tools for somatic SVs can help with clinical diagnosis.

Furthermore, machine learning shows great performance in small indel calling, which makes it a good potential to call large genome changes like SVs. Calling SVs, including large indels with NGS, requires a careful and clever designed algorithm; simply applying machine learning methods only for variant filtration may not guarantee good results. These machine learning-based SV calling tools, such as DeepSV [353] and Cue [354], should be evaluated comprehensively using recently published high-quality SV benchmarking datasets, such as that in [464], to help understand how machine learning methods should be applied in large genomic variant discovery. Furthermore, in this study, indels were grouped into small and large indels based on their sizes. However, large indels are only two sub-types of SVs. Other types of SVs, including inversions, translocations, or other large complex genomic variants, were not evaluated in **Publication I**. These types of SVs may require specific algorithms to detect, as normal indel calling may not have the ability to call them.

In addition, insertions in TRs may be classified as insertions, duplications, or repeat expansions. For a small insertion in a TR, sequencing reads may fully resolve it, so the complexities of repeat expansion may not cause trouble for detection. For a large repeat expansion, sequencing reads cannot fully resolve it, and the exact size of this variant cannot be determined using normal indel calling algorithms. Although some tools, such as Pindel and DELLY, have designed functions to call duplications, they are still limited by two copy repeat expansions. As the use of specific algorithms

is encouraged to call repeat expansions, repeat expansion is not a variant type in this evaluation work [111]. However, because of the complex structures of repeat expansions, distinguishing them from insertions in the HuRef genome is not an easy task. Thus, in the truth set of the semi-simulated dataset, repeat expansions may accidently be included. In the future, a comprehensive evaluation of specific repeat expansion calling tools should be conducted.

7.2 Sequence contexts of genomic variants

In this study, a computational tool, VarSCAT, was developed to assess the sequence contexts of genomic variants, focusing on issues related with breakpoint ambiguity and STRs. Breakpoint ambiguity occurs with indels, which causes an indel to have multiple representations as different positions and alleles [133,134]. In addition, HGVS nomenclature requires an indel to be represented at the 3' aligned position, while typical variant calling results show an indel at the 5' aligned position [61]. These positions and allele differences can cause confusion, such as redundant indels, when dealing with high-throughput data or variants from databases [137,139]. Furthermore, instead of representing an indel at a single position, showing the equivalent affected region can help to better understand how the indel can be represented differently [134]. With these purposes, I developed an ambiguous variant annotation module for VarSCAT which can output the information about breakpoint ambiguity, together with, flanking bases, HGVS nomenclature, and distance to adjacent variants.

The breakpoint ambiguity of an indel occurs because of a similar sequence pattern around the indel breakpoint. These similar sequence patterns are related to TRs. Tandem repeat lacks a general agreed, clear definition. The current definitions or thresholds for TRs always depend on artificial values. For example, the UCSC Genome Browser Simple Repeats track, which is made by the computational programme TRF, sets a minimum of 25 bp to be defined as a TR [307,395]. Under this threshold, a mononucleotide STR should have at least 25 copies, while for some large motif size repeats, the minimum copy number can be 1.8, even less than a duplication. Although the common definition of tandem repeat is not strict, the definition of STR is clear, which is a sequence motif with sizes of 1–6 bp repeated multiple times [38]. Based on this definition, the STR annotation from the Simple Repeat track of UCSC Genome Browser seems too strict. Some computational meta-analyses or molecular experiments concluded that an STR should be around at least 10 bp to show different mutation rate than the background genome [39–41]. Besides, in some cancer related studies, indels in mononucleotide STRs of length smaller than 10 bp are considered to indicate the status of microsatellite instability [42–44]. Thus, if one uses the Simple Repeats track of the UCSC Genome Browser to annotate

variants in STRs, the proportion of variants in STRs may be underestimated; nonetheless, this track also contains redundant STRs. However, algorithms of many specific STR calling tools, including pathogenic repeat expansion calling methods, limit their calling regions within large size STRs. They call repeat expansions in pre-defined STRs, which usually generated by a trusted resource, such as the UCSC Genome Browser or TRF [459,465]. By consider this research problem, the tandem repeat annotation module of VarSCAT was developed.

The biological analysis of human germline variant sets using VarSCAT demonstrated the proportion of breakpoint ambiguous indels and variants in STRs at the human individual germline variant level. Although high-confidence human variants in STRs such as centromeres have not yet been fully assessed [450], these results still provide a current estimate of the number of variants in STRs. As high-confidence human germline variant sets were used in this study, another issue that cannot be ignored is that these high-confidence variant sets may filter out variants in STRs to retain the "high confidence" of their variant sets [63,450,466]. Variants in STRs are difficult to call and genotype; some commercial NGS-based variant discovery applications, such as Illumina BaseSpace, may filter them out or suggest specific STR applications. Thus, the proportions of variants in STRs estimated by VarSCAT may be underestimated. The current version of VarSCAT is still an annotation tool that relies heavily on input variants. For future development, VarSCAT may be upgraded to take sequencing files as input and call variants in STRs with the designed criteria or try to distinguish FP variant calls from tool's variant calling result.

The results from **Publication II** shows that the occurrences of indels were not very random, small germline indels were strongly correlated with the sequence context. The majority of small germline indels in human genome had breakpoint ambiguity and many of them located in STRs (Figure 15). These results also suggested that tool evaluation cannot be conducted solely with simulated data, which variants or sequencing data are created artificially. Theoretically, simulated data has better labelled indels than real data because all indels are created artificially. Even though the truth indel set of real data has been developed carefully, incorrect labels of different causes cannot be totally avoided, which may result in inaccurate evaluation results. However, the complexity of the real human genome influences sequencing analysis from sample preparation to sequencing processes and then to computational analysis. Indeed, using simulated data can reduce the cost and complexity of experiments, but not all biological complexities can be easily simulated, such as sequencing errors in homopolymers. If indels are randomly inserted into the reference sequence, indel calling will be an easy task for tools. Besides, the technical differences between real datasets and simulated datasets might become unnecessary challenges, and these may confuse machine learning-based

methods. The strengths and weaknesses of the tools may not be reflected well or may even be reflected incorrectly with simulated data. Therefore, high-quality, standard real data with benchmarking purposes are always needed.

7.3 Evaluation of lncRNA prediction tools

In this part of the study, which is also described in **Publication III**, eight lncRNA prediction tools with different underlying algorithms were evaluated. Compared with previous research [413,442–447], this research highlighted deep learning-based tools and tools' performance with lncRNA or protein biased dataset.

The results of **Publication III** illustrated that deep learning-based tools were the top performers in lncRNA prediction. The ability of deep learning-based algorithms to recognise unknown sequence features, which might be uninterpretable for other algorithms, significantly contributed to their performance. In this study, transcript sets of different sizes were used. The performance of tools was stable, with different sizes of the transcript sets containing a balanced number of lncRNAs and protein-coding mRNAs, but they fluctuated with imbalanced lncRNAs and protein-coding mRNA sets. The reason might be that the tools were trained with data containing a balanced number of lncRNAs and protein-coding mRNAs. In the real-world transcript set, the number of lncRNAs and protein-coding mRNAs may not be equal. Thus, further model development should take the biased nature of the transcript set into account. In addition, high-quality, standard, real-world transcript sets with labels for lncRNA and protein-coding mRNA are needed.

The development of the lncRNA prediction tool may be divided into two steps: feature selection and model selection. The performance of the tool can be improved by choosing more informative features and reducing redundant or less important features. Robust feature selection algorithms for lncRNA prediction may become the direction for future method development. This and previous studies have answered the question of which model is better [445–447], also demonstrated the importance of features [413], efforts can still made for interpreting the roles of these features in lncRNA functions. Deep learning-based tools can fetch unknown features and result in good performance, but the lack of interpretation remains a problem. With a more interpretable model, knowledge regarding which features distinguish lncRNAs from protein-coding mRNAs can be obtained, and this may benefit pathological research.

LncRNA is a type of non-coding RNA that also contains several sub-types [161]. In the future, tools can be developed to predict transcripts by taking all sizes and structures of transcripts into account, such as small non-coding RNAs and circular RNAs. In addition, recognising the sub-types of lncRNAs can also be a direction for future development. Structural information and the interactions between lncRNAs and proteins, or even DNAs [444], can be used to better recognise lncRNA sub-types

and potential functions. For some lncRNAs that can be translated into small peptides [174,467], algorithms are needed for better distinguishing these bi-functional transcripts. These efforts not only require computational practice but also knowledge from molecular experiments. Efforts are needed to study the functions of transcripts with definite coding abilities and non-coding functions.

High-quality training datasets are required for non-model organism in lncRNA prediction. If these training datasets are not available, researchers may have to apply tools trained by other well-studied species to predict lncRNAs in their species of interest [445]. Aside from differences in lncRNAs in different species, cell types, the quality of sequencing data, and the proportions of lncRNAs should also be considered. Although some lncRNA prediction tools do have options to re-train their models, more work is needed to establish a generic criteria for building standard lncRNA sets.

8 Conclusion

In this thesis work, a computational analysis of human genomic features, which were the indels, the sequence contexts of genomic variants, and lncRNAs, was conducted. All of these features play important roles in the regulation and development of human biological processes. In the past 20 years, with the development of sequencing technology, vast amounts of human genomic data have been produced, and the understanding of the human genome has been significantly improved.

To further analyse these sequencing data and acquire novel knowledge from them, algorithms and tools were developed to meet research needs. For human DNA sequencing with the purpose of identifying genomic variants, variant calling tools with a variety of underlying algorithms were applied to detect different types of variants. With these known variants and annotations of the human genome sequence, the correlations between variants and certain phenotypes or diseases can be studied. In addition, the sequence contexts of genomic variants were analysed to determine the roles they play in biological processes. For human RNA sequencing, with the maturity of upstream sequencing techniques and downstream computational models, novel lncRNAs that were previously not fully known because of their low expression levels in human cells can be identified.

In the bioinformatics field, novel algorithms and tools have been published frequently, but their performance may not be benchmarked well. This issue can cause confusion in tool selection, especially for users who work with certain biological or clinical purposes and lack computational knowledge. Although the computational analysis of human genomic data has been developed for more than 15 years and routinely applied to clinical diagnosis, some questions remain unanswered, limiting the clinical uses of human genomic data. Thanks to efforts to generate open-source data, several high-confidence, standard datasets have been produced in the last five years with the purposes of benchmarking and evaluating computational methods. Together with growing data in public databases, computational tools can be compared and evaluated more accurately and in detail than in the past.

In this thesis, the evaluation of variant calling tools for indel calling on different indel size ranges and data types was conducted. No tool can perfectly fit all circumstances, but the choice of tools significantly impacts indel calling results.

Certain algorithms are suitable for certain size ranges and types of indels. Machine learning tools showed great abilities for calling small indels. Tools designed with specific indel calling algorithms are needed for calling large indels. In general, sequencing data with higher coverage and longer read lengths are preferred to call indels. The results of this study were presented in **Publication I**.

The results of **Publication I** showed that more than half of FP indel calls were in simple repeats. The similar sequence context around an indel can be explained as the reason for FP calls. The sequence contexts of genomic variants can not only cause technical trouble in data analysis but may also have biological significance. To comprehensively study the sequence contexts of genomic variants, a computational tool, VarSCAT, was developed, which was described in **Publication II**. By applying VarSCAT to a variety of human high-confidence variant sets, the proportion of breakpoint ambiguous indels and the proportion of variants in STRs were described at the human individual level. The results demonstrated that the majority of human germline small indels had breakpoint ambiguities, and they were the largest types of human germline small variants in STRs. The results also illustrated that current variant annotation methods or strategies may underestimate the proportion of variants in STRs.

In **Publication III**, deep learning-based lncRNA prediction tools were compared with tools based on other machine learning models. The results demonstrated that deep learning-based tools were the top performers in lncRNA prediction. The performance of tools does not vary with the size of the transcript dataset but with the proportions of lncRNAs and protein-coding mRNAs.

In conclusion, this thesis conducted computational analysis with human genomic DNA and RNA data, focusing especially on indels, the sequence contexts of genomic variants and lncRNA prediction. The strengths and weaknesses of current computational methods have been identified, and future method developments have been discussed. A novel tool has been developed to fill the current research gap, which might have been underestimated previously.

Bioinformatics is a rapidly evolving field, and technologies are updated every day. However, old technologies will not be eliminated; they will continue to mature into reliable downstream analysis methods and to serve industries, such as healthcare, particularly clinical diagnosis. The emergence of new technologies will advance our understanding of the human genome and provide novel solutions for human health-related issues. As a tool for studying and analysing human-related data, bioinformatics will be used more widely in the future than it is today.

Acknowledgement

First of all, I would like to thank my supervisor Prof. Laura Elo and Dr. Sofia Khan for their greatest support, patience and help during my doctoral study in Medical Bioinformatics Centre (MBC). Sometimes I lost myself in research, but you never leave me alone, and always listen to me, believe me, and help me out. Without the plenty supports from you, I will never make this thesis. No matter how many words I write here, it cannot express my gratitude.

Scientific research is hard to be carried alone. Collaboration with other researcher is the key to perform good science. Great thanks to all the co-authors, whose work is important part to make this thesis. Vladislav Lysenkov, who is my previous colleague in MBC and we worked in the genomics team leading by Sofia. Dr. Katri Orte and Dr. Veli Kairisto, who are the clinicians from University hospital, thank you very much for the explanation of patient data, clinical settings, and the "warm" email from Veli. Dr. Juhani Aakko, who is my previous colleague but also my supervisor of master thesis. Dr. Tea Ammunét, who is the leading author in the **Publication III**, and took care the research during the time when my body was not very well. Not to mentioned how much effort Prof. Laura Elo and Dr. Sofia Khan had been putting into my research. Also, my big thanks for my doctoral advisory committee, Dr. Tapio Pahikkala, Dr. Matti Tolvanen, and Dr. Mikko Venäläinen, thank you for all the advice during my study here. And thanks for Dr. Anish MS Shrestha and Dr. Esa Pitkänen for being my thesis pre-examiners, and thanks for Prof. Mauno Vihinen for being my defense opponent. It is such an honour to study and work with all of you.

My great thanks to the Faculty of Technology and the Doctoral Programme in Technology (DPT). Even though the names of faculty, department, and programme changed many times during my study, the people are always helpful. No matter when I need help, they can always provide detailed structures and help me out. The MATTI days they had been arranged (MATTI is the old name of DPT), all the helps including my study plan, graduation procedures and credits registration, without these supports my life won't be easy. I wish the DPT programme and the Faculty of Technology the best future. Also, big thanks for Turku University Foundation, University of Turku joint research grant fund, University of Turku Graduate School, and my

supervisor Prof. Laura Elo, who provide me financial supports for finishing this thesis.

Besides scientific supports, I would like to thank all the friends and colleague I met here. Hua Jin, Devon, Pearson, Shenghua Xie, Nacho, Simo, Sohrab, Tozé, Jose, Flavia, Matheus, Ye Hong, Esko, Anu, Sami, Tommi, Asta, Iivari, Kalle, Olof, Johannes, Arfa, Satu, Riku, Tapio, Veronika, Jesse, Paulina, Markus, Niklas, Mats, Nigatu, Alexander, Mehrad, Julia, Mats, António, Thomas, Deep, Maria, Haifeng, Dhany, Aidan, Xu, Markku, Jinghui Yang, Kang Chen, Weihua Zhang, Nick, Bishwa, Alan, and Tam. Thank you very much for all the knowledge you shared about research, arrangement of teamwork, all the technical supports, the advice and encouragement of PhD life, and the happy time we spend together.

When I first came to University of Turku as a master student, my degree was "Food development". Because this was a "tech" degree, I had to choose 20 credits of minor study from a different field. At first, I chose "Molecular Biology" as minor study. But the first exam of "Molecular Biology", the teacher made some questions hard to understand, and two third students failed in the exam, of course, I also failed. This was the moment I felt I might need a backup plan for minor study, then I tried "Bioinformatics". The first "Bioinformatics" lecture was talking about how to assemble a DNA sequence from fragments. This was the moment I found bioinformatics was so interesting and obsessed, and I decided to switch my major degree. Thanks a lot for the Finnish education system, it gives opportunities for students to find out and study what they are really interested. I cannot image that I can do the same thing if I didn't come to Finland. After that, I officially became a bioinformatics master student and continued my study, then I joined the bioinformatics unit (Previous name of MBC). Very grateful for Prof. Laura Elo, who gave me the chance to do my master thesis in her lab and let me continue my study as a PhD student.

In the end, the greatest thanks to my parents, who gives me all the love, encouragement, and support, who always listen to my stories, give me good advice, and sharing about daily life. We always stay on the same side, and you are the models for me in all aspects of life. For Chinese people, we rarely say "I love you" to parents in real life, because this somehow sounds fake in our culture. But in the deep of our heart, we always know we love each other, care about each other, support each other in all the situations, and this is the real and true love.

May 16, 2023
Ning Wang

List of references

1. TA. B. The Human Genome. 2 Edition. Genomes 2nd edition. 2 Edition. Wiley-Liss; 2002.
2. Cooper DN, Krawczak M, Antonarakis SE. The Nature and Mechanisms of Human Gene Mutation. The Online Metabolic & Molecular Bases of Inherited Disease. 2012. Available: http://www.ommbid.com/OMMBID/the_online_metabolic_and_molecular_bases_of_inherited_disease/b/abstract/part3/ch13
3. Rubin CM. The Genetic Basis of Human Cancer. Ann Intern Med. 1998;129: 759. doi:10.7326/0003-4819-129-9-199811010-00045
4. Auton A, Abecasis GR, Altshuler DM, Durbin RM, Bentley DR, Chakravarti A, et al. A global reference for human genetic variation. Nature. 2015. pp. 68–74. doi:10.1038/nature15393
5. Martincorena I, Campbell PJ. Somatic mutation in cancer and normal cells. Science. 2015. pp. 1483–1489. doi:10.1126/science.aab4082
6. Cardoso JGR, Andersen MR, Herrgård MJ, Sonnenschein N. Analysis of genetic variation and potential applications in genome-scale metabolic modeling. Frontiers in Bioengineering and Biotechnology. 2015. doi:10.3389/fbioe.2015.00013
7. Mullaney JM, Mills RE, Stephen Pittard W, Devine SE. Small insertions and deletions (INDELs) in human genomes. Hum Mol Genet. 2010;19. doi:10.1093/hmg/ddq400
8. Lek M, Karczewski KJ, Minikel E V., Samocha KE, Banks E, Fennell T, et al. Analysis of protein-coding genetic variation in 60,706 humans. Nature. 2016;536: 285–291. doi:10.1038/nature19057
9. Altshuler DM, Durbin RM, Abecasis GR, Bentley DR, Chakravarti A, Clark AG, et al. An integrated map of genetic variation from 1,092 human genomes. Nature. 2012;491: 56–65. doi:10.1038/nature11632
10. Sana J, Faltejskova P, Svoboda M, Slaby O. Novel classes of non-coding RNAs and cancer. Journal of Translational Medicine. 2012. doi:10.1186/1479-5876-10-103
11. Tehrani SS, Karimian A, Parsian H, Majidinia M, Yousefi B. Multiple Functions of Long Non-Coding RNAs in Oxidative Stress, DNA Damage Response and Cancer Progression. J Cell Biochem. 2018;119: 223–236. doi:10.1002/jcb.26217
12. Statello L, Guo CJ, Chen LL, Huarte M. Gene regulation by long non-coding RNAs and its biological functions. Nature Reviews Molecular Cell Biology. 2021. pp. 96–118. doi:10.1038/s41580-020-00315-9
13. Frankish A, Diekhans M, Jungreis I, Lagarde J, Loveland JE, Mudge JM, et al. GENCODE 2021. Nucleic Acids Res. 2021;49: D916–D923. doi:10.1093/nar/gkaa1087
14. Uszczynska-Ratajczak B, Lagarde J, Frankish A, Guigó R, Johnson R. Towards a complete map of the human long non-coding RNA transcriptome. Nature Reviews Genetics. 2018. pp. 535–548. doi:10.1038/s41576-018-0017-y
15. Bolha L, Ravnik-Glavač M, Glavač D. Long Noncoding RNAs as Biomarkers in Cancer. Disease Markers. 2017. doi:10.1155/2017/7243968
16. Sullivan W, Evans DG, Newman WG, Ramsden SC, Scheffer H, Payne K. Developing national guidance on genetic testing for breast cancer predisposition: The role of economic evidence? Genet Test Mol Biomarkers. 2012;16: 580–591. doi:10.1089/gtmb.2011.0236

17. Frank M, Prenzler A, Eils R, von der Schulenburg JMG. Genome sequencing: A systematic review of health economic evidence. Health Economics Review. 2013. pp. 1–8. doi:10.1186/2191-1991-3-29
18. Schwarze K, Buchanan J, Taylor JC, Wordsworth S. Are whole-exome and whole-genome sequencing approaches cost-effective? A systematic review of the literature. Genetics in Medicine. 2018. pp. 1122–1130. doi:10.1038/gim.2017.247
19. França LTC, Carrilho E, Kist TBL. A review of DNA sequencing techniques. Quarterly Reviews of Biophysics. 2002. pp. 169–200. doi:10.1017/S0033583502003797
20. Wang Z, Gerstein M, Snyder M. RNA-Seq: A revolutionary tool for transcriptomics. Nature Reviews Genetics. 2009. pp. 57–63. doi:10.1038/nrg2484
21. Chowdhary A, Satagopam V, Schneider R. Long Non-coding RNAs: Mechanisms, Experimental, and Computational Approaches in Identification, Characterization, and Their Biomarker Potential in Cancer. Frontiers in Genetics. 2021. doi:10.3389/fgene.2021.649619
22. Rhoades R, Jackson F, Teng S. Discovery of rare variants implicated in schizophrenia using next-generation sequencing. J Transl Genet Genomics. 2019. doi:10.20517/jtgg.2018.26
23. Xu C. A review of somatic single nucleotide variant calling algorithms for next-generation sequencing data. Computational and Structural Biotechnology Journal. 2018. pp. 15–24. doi:10.1016/j.csbj.2018.01.003
24. Xu X, Liu S, Yang Z, Zhao X, Deng Y, Zhang G, et al. A systematic review of computational methods for predicting long noncoding RNAs. Briefings in Functional Genomics. 2021. pp. 162–173. doi:10.1093/bfgp/elab016
25. Tanay A, Siggia ED. Sequence context affects the rate of short insertions and deletions in flies and primates. Genome Biol. 2008;9. doi:10.1186/gb-2008-9-2-r37
26. Aggarwala V, Voight BF. An expanded sequence context model broadly explains variability in polymorphism levels across the human genome. Nat Genet. 2016;48: 349–355. doi:10.1038/ng.3511
27. Watson JD, Crick FHC. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. Nature. 1953;171: 737–738. doi:10.1038/171737a0
28. Pruitt KD, Brown GR, Hiatt SM, Thibaud-Nissen F, Astashyn A, Ermolaeva O, et al. RefSeq: An update on mammalian reference sequences. Nucleic Acids Res. 2014;42. doi:10.1093/nar/gkt1114
29. Hannan AJ. Tandem repeat polymorphisms: Mediators of genetic plasticity, modulators of biological diversity and dynamic sources of disease susceptibility. Advances in Experimental Medicine and Biology. 2012. pp. 1–9. doi:10.1007/978-1-4614-5434-2_1
30. Liang KC, Tseng JT, Tsai SJ, Sun HS. Characterization and distribution of repetitive elements in association with genes in the human genome. Comput Biol Chem. 2015;57: 29–38. doi:10.1016/j.compbiolchem.2015.02.007
31. Olivero M, Ruggiero T, Coltella N, Maffe' A, Calogero R, Medico E, et al. Amplification of repeat-containing transcribed sequences (ARTS): a transcriptome fingerprinting strategy to detect functionally relevant microsatellite mutations in cancer. Nucleic Acids Res. 2003;31. doi:10.1093/nar/gng033
32. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, et al. Initial sequencing and analysis of the human genome. Nature. 2001;409: 860–921. doi:10.1038/35057062
33. Treangen TJ, Salzberg SL. Repetitive DNA and next-generation sequencing: Computational challenges and solutions. Nature Reviews Genetics. 2012. pp. 36–46. doi:10.1038/nrg3117
34. Schmid CW, Deininger PL. Sequence organization of the human genome. Cell. 1975;6: 345–358. doi:10.1016/0092-8674(75)90184-1
35. Batzer MA, Deininger PL. Alu repeats and human genomic diversity. Nature Reviews Genetics. 2002. pp. 370–379. doi:10.1038/nrg798
36. Kojima KK. Human transposable elements in Repbase: Genomic footprints from fish to humans. Mobile DNA. 2018. doi:10.1186/s13100-017-0107-y

37. Hannan AJ. Tandem repeats mediating genetic plasticity in health and disease. Nature Reviews Genetics. 2018. pp. 286–298. doi:10.1038/nrg.2017.115
38. Hannan AJ. Tandem repeat polymorphisms: modulators of disease susceptibility and candidates for "missing heritability." Trends Genet. 2010;26: 59–65. doi:10.1016/j.tig.2009.11.008
39. Lai Y, Sun F. The Relationship between Microsatellite Slippage Mutation Rate and the Number of Repeat Units. Mol Biol Evol. 2003;20: 2123–2131. doi:10.1093/molbev/msg228
40. Merkel A, Gemmell NJ. Detecting microsatellites in genome data: Variance in definitions and bioinformatic approaches cause systematic bias. Evol Bioinforma. 2008;2008: 1–6.
41. Kelkar YD, Strubczewski N, Hile SE, Chiaromonte F, Eckert KA, Makova KD. What is a microsatellite: A computational and experimental definition based upon repeat mutational behavior at A/T and GT/AC repeats. Genome Biol Evol. 2010;2: 620–635. doi:10.1093/gbe/evq046
42. Niu B, Ye K, Zhang Q, Lu C, Xie M, McLellan MD, et al. MSIsensor: Microsatellite instability detection using paired tumor-normal sequence data. Bioinformatics. 2014;30: 1015–1016. doi:10.1093/bioinformatics/btt755
43. Salipante SJ, Scroggins SM, Hampel HL, Turner EH, Pritchard CC. Microsatellite instability detection by next generation sequencing. Clin Chem. 2014;60: 1192–1199. doi:10.1373/clinchem.2014.223677
44. Kondelin J, Gylfe AE, Lundgren S, Tanskanen T, Hamberg J, Aavikko M, et al. Comprehensive evaluation of protein coding mononucleotide microsatellites in microsatellite-unstable colorectal cancer. Cancer Res. 2017;77: 4078–4088. doi:10.1158/0008-5472.CAN-17-0682
45. Dumbovic G, Forcales S V., Perucho M. Emerging roles of macrosatellite repeats in genome organization and disease development. Epigenetics. 2017. pp. 515–526. doi:10.1080/15592294.2017.1318235
46. Liehr T. Repetitive elements in humans. International Journal of Molecular Sciences. 2021. pp. 1–10. doi:10.3390/ijms22042072
47. Örd T, Puurand T, Örd D, Annilo T, Möls M, Remm M, et al. A human-specific VNTR in the TRIB3 promoter causes gene expression variation between individuals. PLoS Genet. 2020;16. doi:10.1371/JOURNAL.PGEN.1008981
48. Eslami Rasekh M, Hernández Y, Drinan SD, Fuxman Bass JI, Benson G. Genome-wide characterization of human minisatellite VNTRs: Population-specific alleles and gene expression differences. Nucleic Acids Res. 2021;49: 4308–4324. doi:10.1093/nar/gkab224
49. Course MM, Sulovari A, Gudsnuk K, Eichler EE, Valdmanis PN. Characterizing nucleotide variation and expansion dynamics in human-specific variable number tandem repeats. Genome Res. 2021;31: 1313–1324. doi:10.1101/gr.275560.121
50. Koch L. Profiling human-specific VNTR expansions. Nat Rev Genet. 2021;22: 625. doi:10.1038/s41576-021-00404-1
51. Weber JL, Wong C. Mutation of human short tandem repeats. Hum Mol Genet. 1993;2: 1123–1128. doi:10.1093/hmg/2.8.1123
52. Brinkmann B, Klintschar M, Neuhuber F, Hühne J, Rolf B. Mutation rate in human microsatellites: Influence of the structure and length of the tandem repeat. Am J Hum Genet. 1998;62: 1408–1415. doi:10.1086/301869
53. Nachman MW, Crowell SL. Estimate of the mutation rate per nucleotide in humans. Genetics. 2000;156: 297–304. doi:10.1093/genetics/156.1.297
54. Jan C, Fumagalli L. Polymorphic DNA microsatellite markers for forensic individual identification and parentage analyses of seven threatened species of parrots (family Psittacidae). PeerJ. 2016;2016. doi:10.7717/peerj.2416
55. Duitama J, Zablotskaya A, Gemayel R, Jansen A, Belet S, Vermeesch JR, et al. Large-scale analysis of tandem repeat variability in the human genome. Nucleic Acids Res. 2014;42: 5728–5741. doi:10.1093/nar/gku212
56. Fan H, Chu JY. A Brief Review of Short Tandem Repeat Mutation. Genomics, Proteomics and Bioinformatics. 2007. pp. 7–14. doi:10.1016/S1672-0229(07)60009-6

57. Merkel A, Gemmell N. Detecting short tandem repeats from genome data: Opening the software black box. Brief Bioinform. 2008;9: 355–366. doi:10.1093/bib/bbn028
58. Crick F. Central dogma of molecular biology. Nature. 1970;227: 561–563. doi:10.1038/227561a0
59. Dunham I, Kundaje A, Aldred SF, Collins PJ, Davis CA, Doyle F, et al. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012;489: 57–74. doi:10.1038/nature11247
60. Onishi-Seebacher M, Korbel JO. Challenges in studying genomic structural variant formation mechanisms: The short-read dilemma and beyond. BioEssays. 2011;33: 840–850. doi:10.1002/bies.201100075
61. den Dunnen JT, Dalgleish R, Maglott DR, Hart RK, Greenblatt MS, Mcgowan-Jordan J, et al. HGVS Recommendations for the Description of Sequence Variants: 2016 Update. Hum Mutat. 2016;37: 564–569. doi:10.1002/humu.22981
62. Gonzaga-Jauregui C, Lupski JR, Gibbs RA. Human genome sequencing in health and disease. Annual Review of Medicine. 2012. pp. 35–61. doi:10.1146/annurev-med-051010-162644
63. Eberle MA, Fritzilas E, Krusche P, Källberg M, Moore BL, Bekritsky MA, et al. A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. Genome Res. 2017;27: 157–164. doi:10.1101/gr.210500.116
64. Yang R, Van Etten JL, Dehm SM. Indel detection from DNA and RNA sequencing data with transIndel. BMC Genomics. 2018;19. doi:10.1186/s12864-018-4671-4
65. Guan P, Sung WK. Structural variation detection using next-generation sequencing data: A comparative technical review. Methods. 2016. doi:10.1016/j.ymeth.2016.01.020
66. Feuk L, Carson AR, Scherer SW. Structural variation in the human genome. Nature Reviews Genetics. 2006. doi:10.1038/nrg1767
67. Pollex RL, Hegele RA. Copy number variation in the human genome and its implications for cardiovascular disease. Circulation. 2007. pp. 3130–3138. doi:10.1161/CIRCULATIONAHA.106.677591
68. Sebat J, Lakshmi B, Troge J, Alexander J, Young J, Lundin P, et al. Large-scale copy number polymorphism in the human genome. Large-scale copy number Polymorph Hum genome. 2004;305: 525–528. doi:papers2://publication/doi/10.1126/science.1098918
69. Redon R, Ishikawa S, Fitch KR, Feuk L, Perry GH, Andrews TD, et al. Global variation in copy number in the human genome. Nature. 2006;444: 444–454. doi:10.1038/nature05329
70. Whitford W, Lehnert K, Snell RG, Jacobsen JC. Evaluation of the performance of copy number variant prediction tools for the detection of deletions from whole genome sequencing data. Journal of Biomedical Informatics. 2019. doi:10.1016/j.jbi.2019.103174
71. Shao X, Lv N, Liao J, Long J, Xue R, Ai N, et al. Copy number variation is highly correlated with differential gene expression: A pan-cancer study. BMC Med Genet. 2019;20. doi:10.1186/s12881-019-0909-5
72. Lo Faro V, ten Brink JB, Snieder H, Jansonius NM, Bergen AA. Genome-wide CNV investigation suggests a role for cadherin, Wnt, and p53 pathways in primary open-angle glaucoma. BMC Genomics. 2021;22. doi:10.1186/s12864-021-07846-1
73. Girirajan S, Campbell CD, Eichler EE. Human copy number variation and complex genetic disease. Annu Rev Genet. 2011;45: 203–226. doi:10.1146/annurev-genet-102209-163544
74. Zarrei M, MacDonald JR, Merico D, Scherer SW. A copy number variation map of the human genome. Nature Reviews Genetics. 2015. pp. 172–183. doi:10.1038/nrg3871
75. Nowakowska B. Clinical interpretation of copy number variants in the human genome. Journal of Applied Genetics. 2017. pp. 449–457. doi:10.1007/s13353-017-0407-4
76. Zhang F, Gu W, Hurles ME, Lupski JR. Copy number variation in human health, disease, and evolution. Annual Review of Genomics and Human Genetics. 2009. pp. 451–481. doi:10.1146/annurev.genom.9.081307.164217

77. Bomba L, Walter K, Soranzo N. The impact of rare and low-frequency genetic variants in common disease. Genome Biology. 2017. doi:10.1186/s13059-017-1212-4
78. Dawson E, Chen Y, Hunt S, Smink LJ, Hunt A, Rice K, et al. A SNP resource for human chromosome 22: Extracting dense clusters of SNPs from the genomic sequence. Genome Res. 2001;11: 170–178. doi:10.1101/gr.156901
79. Mills RE, Luttig CT, Larkins CE, Beauchamp A, Tsui C, Pittard WS, et al. An initial map of insertion and deletion (INDEL) variation in the human genome. Genome Res. 2006;16: 1182–1190. doi:10.1101/gr.4565806
80. Levy S, Sutton G, Ng PC, Feuk L, Halpern AL, Walenz BP, et al. The diploid genome sequence of an individual human. PLoS Biol. 2007;5: 2113–2144. doi:10.1371/journal.pbio.0050254
81. Chaisson MJP, Huddleston J, Dennis MY, Sudmant PH, Malig M, Hormozdiari F, et al. Resolving the complexity of the human genome using single-molecule sequencing. Nature. 2015;517: 608–611. doi:10.1038/nature13907
82. Montgomery SB, Goode DL, Kvikstad E, Albers CA, Zhang ZD, Mu XJ, et al. The origin, evolution, and functional impact of short insertion-deletion variants identified in 179 human genomes. Genome Res. 2013;23: 749–761. doi:10.1101/gr.148718.112
83. Kunkel TA. DNA Replication Fidelity. Journal of Biological Chemistry. 2004. pp. 16895–16898. doi:10.1074/jbc.R400006200
84. Schlotterer C. Evolutionary dynamics of microsatellite DNA. Chromosoma. 2000. pp. 365–371. doi:10.1007/s004120000089
85. Kroutil LC, Register K, Bebenek K, Kunkel TA. Exonucleolytic proofreading during replication of repetitive DNA. Biochemistry. 1996;35: 1046–1053. doi:10.1021/bi952178h
86. Tran HT, Keen JD, Kricker M, Resnick MA, Gordenin DA. Hypermutability of homonucleotide runs in mismatch repair and DNA polymerase proofreading yeast mutants. Mol Cell Biol. 1997;17: 2859–2865. doi:10.1128/mcb.17.5.2859
87. Greene CN, Jinks-Robertson S. Frameshift intermediates in homopolymer runs are removed efficiently by yeast mismatch repair proteins. Mol Cell Biol. 1997;17: 2844–2850. doi:10.1128/mcb.17.5.2844
88. Nojadeh JN, Sharif SB, Sakhinia E. Microsatellite instability in colorectal cancer. EXCLI Journal. 2018. pp. 159–168. doi:10.17179/excli2017-948
89. Loeb LA, Loeb KR, Anderson JP. Multiple mutations and cancer. Proceedings of the National Academy of Sciences of the United States of America. 2003. pp. 776–781. doi:10.1073/pnas.0334858100
90. Kang S, Na Y, Joung SY, Lee S Il, Oh SC, Min BW. The significance of microsatellite instability in colorectal cancer after controlling for clinicopathological factors. Med (United States). 2018;97. doi:10.1097/MD.0000000000010019
91. Adewoye AB, Lindsay SJ, Dubrova YE, Hurles ME. The genome-wide effects of ionizing radiation on mutation induction in the mammalian germline. Nat Commun. 2015;6. doi:10.1038/ncomms7684
92. Shibata A, Moiani D, Arvai AS, Perry J, Harding SM, Genois MM, et al. DNA Double-Strand Break Repair Pathway Choice Is Directed by Distinct MRE11 Nuclease Activities. Mol Cell. 2014;53: 7–18. doi:10.1016/j.molcel.2013.11.003
93. Mefford HC, Eichler EE. Duplication hotspots, rare genomic disorders, and common disease. Current Opinion in Genetics and Development. 2009. pp. 196–204. doi:10.1016/j.gde.2009.04.003
94. Hastings PJ, Lupski JR, Rosenberg SM, Ira G. Mechanisms of change in gene copy number. Nature Reviews Genetics. 2009. pp. 551–564. doi:10.1038/nrg2593
95. Liskay RM, Letsou A, Stachelek JL. Homology requirement for efficient gene conversion between duplicated chromosomal sequences in mammalian cells. Genetics. 1987;115: 161–167. doi:10.1093/genetics/115.1.161
96. Garcia-Diaz M, Kunkel TA. Mechanism of a genetic glissando*: structural biology of indel mutations. Trends in Biochemical Sciences. 2006. pp. 206–214. doi:10.1016/j.tibs.2006.02.004

97. Krejci L, Altmannova V, Spirek M, Zhao X. Homologous recombination and its regulation. Nucleic Acids Research. 2012. pp. 5795–5818. doi:10.1093/nar/gks270
98. Li X, Heyer WD. Homologous recombination in DNA repair and DNA damage tolerance. Cell Research. 2008. pp. 99–113. doi:10.1038/cr.2008.1
99. Lin FL, Sperle K, Sternberg N. Model for homologous recombination during transfer of DNA into mouse L cells: role for DNA ends in the recombination process. Mol Cell Biol. 1984;4: 1020–1034. doi:10.1128/mcb.4.6.1020-1034.1984
100. Lieber MR. The mechanism of human nonhomologous DNA End joining. Journal of Biological Chemistry. 2008. pp. 1–5. doi:10.1074/jbc.R700039200
101. McVey M, Lee SE. MMEJ repair of double-strand breaks (director's cut): deleted sequences and alternative endings. Trends in Genetics. 2008. pp. 529–538. doi:10.1016/j.tig.2008.08.007
102. Hancks DC, Kazazian HH. Active human retrotransposons: Variation and disease. Current Opinion in Genetics and Development. 2012. pp. 191–203. doi:10.1016/j.gde.2012.02.006
103. Chung WK, Kitner C, Maron BJ. Novel frameshift mutation in Troponin C (TNNC1) associated with hypertrophic cardiomyopathy and sudden death. Cardiol Young. 2011;21: 345–348. doi:10.1017/S1047951110001927
104. Collins FS, Drumm ML, Cole JL, Lockwood WK, Vande Woude GF, Iannuzzi MC. Construction of a general human chromosome jumping library, with application to cystic fibrosis. Science. 1987;235: 1046–1049. doi:10.1126/science.2950591
105. Neaves WB. Genomes, 2nd ed. T.A. Brown. Oxford, United Kingdom: Wiley-Liss, 2002, 600 pp., $97.50, cloth. ISBN 0-471-25046-5. Clin Chem. 2002;48: 2300–2300. doi:10.1093/clinchem/48.12.2300
106. McCarroll SA, Huett A, Kuballa P, Chilewski SD, Landry A, Goyette P, et al. Deletion polymorphism upstream of IRGM associated with altered IRGM expression and Crohn's disease. Nat Genet. 2008;40: 1107–1112. doi:10.1038/ng.215
107. Johansson I, Lundqvist E, Bertilsson L, Dahl ML, Sjoqvist F, Ingelman- Sundberg M. Inherited amplification of an active gene in the cytochrome P450 CYP2D locus as a cause of ultrarapid metabolism of debrisoquine. Proc Natl Acad Sci U S A. 1993;90: 11825–11829. doi:10.1073/pnas.90.24.11825
108. Perry GH, Dominy NJ, Claw KG, Lee AS, Fiegler H, Redon R, et al. Diet and the evolution of human amylase gene copy number variation. Nat Genet. 2007;39: 1256–1260. doi:10.1038/ng2123
109. Krusche P, Trigg L, Boutros PC, Mason CE, De La Vega FM, Moore BL, et al. Best practices for benchmarking germline small-variant calls in human genomes. Nat Biotechnol. 2019;37: 555–560. doi:10.1038/s41587-019-0054-x
110. Kim S, Scheffler K, Halpern AL, Bekritsky MA, Noh E, Källberg M, et al. Strelka2: fast and accurate calling of germline and somatic variants. Nat Methods. 2018;15: 591–594. doi:10.1038/s41592-018-0051-x
111. Cameron DL, Di Stefano L, Papenfuss AT. Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software. Nat Commun. 2019;10. doi:10.1038/s41467-019-11146-4
112. Moreno-Cabrera JM, del Valle J, Castellanos E, Feliubadaló L, Pineda M, Brunet J, et al. Evaluation of CNV detection tools for NGS panel data in genetic diagnostics. Eur J Hum Genet. 2020;28: 1645–1655. doi:10.1038/s41431-020-0675-z
113. Lam WC, Van Der Schans EJC, Sowers LC, Millar DP. Interaction of DNA polymerase I (Klenow fragment) with DNA substrates containing extrahelical bases: Implications for proofreading of frameshift errors during DNA synthesis. Biochemistry. 1999;38: 2661–2668. doi:10.1021/bi9820762
114. Liu P, Lacaria M, Zhang F, Withers M, Hastings PJ, Lupski JR. Frequency of nonallelic homologous recombination is correlated with length of homology: Evidence that ectopic synapsis precedes ectopic crossing-over. Am J Hum Genet. 2011;89: 580–588. doi:10.1016/j.ajhg.2011.09.009

115. Dittwald P, Gambin T, Szafranski P, Li J, Amato S, Divon MY, et al. NAHR-mediated copy-number variants in a clinical population: Mechanistic insights into both genomic disorders and Mendelizing traits. Genome Res. 2013;23: 1395–1409. doi:10.1101/gr.152454.112
116. MacDonald ME, Ambrose CM, Duyao MP, Myers RH, Lin C, Srinidhi L, et al. A novel gene containing a trinucleotide repeat that is expanded and unstable on Huntington's disease chromosomes. Cell. 1993;72: 971–983. doi:10.1016/0092-8674(93)90585-E
117. Walker FO. Huntington's disease. Lancet. 2007. pp. 218–228. doi:10.1016/S0140-6736(07)60111-1
118. Santoro MR, Bray SM, Warren ST. Molecular mechanisms of fragile X syndrome: A twenty-year perspective. Annual Review of Pathology: Mechanisms of Disease. 2012. pp. 219–245. doi:10.1146/annurev-pathol-011811-132457
119. Sæterdal I, Bjørheim J, Lislerud K, Gjertsen MK, Bukholm IK, Olsen OC, et al. Frameshift-mutation-derived peptides as tumor-specific antigens in inherited and spontaneous colorectal cancer. Proc Natl Acad Sci U S A. 2001;98: 13255–13260. doi:10.1073/pnas.231326898
120. Choi YY, Noh SH, Cheong JH. Molecular dimensions of gastric cancer: Translational and clinical perspectives. Journal of Pathology and Translational Medicine. 2016. pp. 1–9. doi:10.4132/jptm.2015.09.10
121. Hempelmann JA, Lockwood CM, Konnick EQ, Schweizer MT, Antonarakis ES, Lotan TL, et al. Microsatellite instability in prostate cancer by PCR or next-generation sequencing. J Immunother Cancer. 2018;6. doi:10.1186/s40425-018-0341-y
122. Fujiyoshi K, Yamamoto G, Takahashi A, Arai Y, Yamada M, Kakuta M, et al. High concordance rate of KRAS/BRAF mutations and MSI-H between primary colorectal cancer and corresponding metastases. Oncol Rep. 2017;37: 785–792. doi:10.3892/or.2016.5323
123. Abida W, Cheng ML, Armenia J, Middha S, Autio KA, Vargas HA, et al. Analysis of the Prevalence of Microsatellite Instability in Prostate Cancer and Response to Immune Checkpoint Blockade. JAMA Oncol. 2019;5: 471–478. doi:10.1001/jamaoncol.2018.5801
124. Rockah-Shmuel L, Tóth-Petróczy Á, Sela A, Wurtzel O, Sorek R, Tawfik DS. Correlated Occurrence and Bypass of Frame-Shifting Insertion-Deletions (InDels) to Give Functional Proteins. PLoS Genet. 2013;9. doi:10.1371/journal.pgen.1003882
125. Hwang DG, Green P. Bayesian Markov chain Monte Carlo sequence analysis reveals varying neutral substitution patterns in mammalian evolution. Proc Natl Acad Sci U S A. 2004;101: 13994–14001. doi:10.1073/pnas.0404142101
126. Hodgkinson A, Eyre-Walker A. Variation in the mutation rate across mammalian genomes. Nature Reviews Genetics. 2011. pp. 756–766. doi:10.1038/nrg3098
127. Carlson J, Locke AE, Flickinger M, Zawistowski M, Levy S, Myers RM, et al. Extremely rare variants reveal patterns of germline mutation rate heterogeneity in humans. Nat Commun. 2018;9. doi:10.1038/s41467-018-05936-5
128. Oman M, Alam A, Ness RW. How Sequence Context-Dependent Mutability Drives Mutation Rate Variation in the Genome. Genome Biol Evol. 2022;14. doi:10.1093/gbe/evac032
129. Jovelin R, Cutter AD. Fine-scale signatures of molecular evolution reconcile models of indel-associated mutation. Genome Biol Evol. 2013;5: 978–986. doi:10.1093/gbe/evt051
130. Long X, Xue H. Genetic-variant hotspots and hotspot clusters in the human genome facilitating adaptation while increasing instability. Hum Genomics. 2021;15. doi:10.1186/s40246-021-00318-3
131. Huang J, Luo H, Wei W, Hou Y. A novel method for the analysis of 20 multi-Indel polymorphisms and its forensic application. Electrophoresis. 2014;35: 487–493. doi:10.1002/elps.201300346
132. Yao Y, Sun K, Yang Q, Zhou Z, Shao C, Qian X, et al. Assessing Autosomal InDel Loci With Multiple Insertions or Deletions of Random DNA Sequences in Human Genome. Front Genet. 2022;12. doi:10.3389/fgene.2021.809815
133. Shrestha AMS, Frith MC, Asai K, Richard H. Jointly aligning a group of DNA reads improves accuracy of identifying large deletions. Nucleic Acids Res. 2018;46. doi:10.1093/nar/gkx1175

134. Krawitz P, Rödelsperger C, Jäger M, Jostins L, Bauer S, Robinson PN. Microindel detection in short-read sequence data. Bioinformatics. 2010;26: 722–729. doi:10.1093/bioinformatics/btq027
135. Sherry ST. dbSNP: the NCBI database of genetic variation. Nucleic Acids Res. 2001;29: 308–311. doi:10.1093/nar/29.1.308
136. Delage WJ, Thevenon J, Lemaitre C. Towards a better understanding of the low recall of insertion variants with short-read based variant callers. BMC Genomics. 2020;21. doi:10.1186/s12864-020-07125-5
137. Hasan MS, Wu X, Watson LT, Zhang L. UPS-indel: a Universal Positioning System for Indels. Sci Rep. 2017;7. doi:10.1038/s41598-017-14400-1
138. Tate JG, Bamford S, Jubb HC, Sondka Z, Beare DM, Bindal N, et al. COSMIC: The Catalogue Of Somatic Mutations In Cancer. Nucleic Acids Res. 2019;47: D941–D947. doi:10.1093/nar/gky1015
139. Tan A, Abecasis GR, Kang HM. Unified representation of genetic variants. Bioinformatics. 2015. doi:10.1093/bioinformatics/btv112
140. Hurwitz J. The discovery of RNA polymerase. Journal of Biological Chemistry. 2005. pp. 42477–42485. doi:10.1074/jbc.X500006200
141. Thomas CA. The genetic organization of chromosomes. Annual review of genetics. 1971. pp. 237–256. doi:10.1146/annurev.ge.05.120171.001321
142. Gall JG. Chromosome structure and the C-value paradox. Journal of Cell Biology. 1981. doi:10.1083/jcb.91.3.3s
143. Ohno S. So much "junk" DNA in our genome. Brookhaven Symp Biol. 1972;23: 366–70. Available: http://www.ncbi.nlm.nih.gov/pubmed/5065367
144. Comings DE. The structure and function of chromatin. Advances in human genetics. 1972. pp. 237–431. doi:10.1007/978-1-4757-4429-3_5
145. de Koning APJ, Gu W, Castoe TA, Batzer MA, Pollock DD. Repetitive elements may comprise over Two-Thirds of the human genome. PLoS Genet. 2011;7. doi:10.1371/journal.pgen.1002384
146. HOAGLAND MB, STEPHENSON ML, SCOTT JF, HECHT LI, ZAMECNIK PC. A soluble ribonucleic acid intermediate in protein synthesis. J Biol Chem. 1958;231: 241–257. doi:10.1016/s0021-9258(19)77302-5
147. PALADE GE. A small particulate component of the cytoplasm. J Biophys Biochem Cytol. 1955;1: 59–68. doi:10.1083/jcb.1.1.59
148. Reddy R, Busch H. Small Nuclear RNAs and RNA Processing. Prog Nucleic Acid Res Mol Biol. 1983;30: 127–162. doi:10.1016/S0079-6603(08)60685-6
149. Greider CW, Blackburn EH. A telomeric sequence in the RNA of Tetrahymena telomerase required for telomere repeat synthesis. Nature. 1989;337: 331–337. doi:10.1038/337331a0
150. Ota T, Suzuki Y, Nishikawa T, Otsuki T, Sugiyama T, Irie R, et al. Complete sequencing and characterization of 21,243 full-length human cDNAs. Nat Genet. 2004;36: 40–45. doi:10.1038/ng1285
151. Birney E, Stamatoyannopoulos JA, Dutta A, Guigó R, Gingeras TR, Margulies EH, et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature. 2007;447: 799–816. doi:10.1038/nature05874
152. Mercer TR, Gerhardt DJ, Dinger ME, Crawford J, Trapnell C, Jeddeloh JA, et al. Targeted RNA sequencing reveals the deep complexity of the human transcriptome. Nat Biotechnol. 2012;30: 99–104. doi:10.1038/nbt.2024
153. Djebali S, Davis CA, Merkel A, Dobin A, Lassmann T, Mortazavi A, et al. Landscape of transcription in human cells. Nature. 2012;489: 101–108. doi:10.1038/nature11233
154. Mattick JS, Makunin I V. Non-coding RNA. Human molecular genetics. 2006. doi:10.1093/hmg/ddl046
155. Meier UT. The many facets of H/ACA ribonucleoproteins. Chromosoma. 2005. pp. 1–14. doi:10.1007/s00412-005-0333-9
156. Bernstein E, Caudy AA, Hammond SM, Hannon GJ. Role for a bidentate ribonuclease in the initiation step of RNA interference. Nature. 2001;409: 363–366. doi:10.1038/35053110

157. Kapranov P, Cheng J, Dike S, Nix DA, Duttagupta R, Willingham AT, et al. RNA maps reveal new RNA classes and a possible function for pervasive transcription. Science. 2007;316: 1484–1488. doi:10.1126/science.1138341
158. Mercer TR, Dinger ME, Mattick JS. Long non-coding RNAs: Insights into functions. Nature Reviews Genetics. 2009. pp. 155–159. doi:10.1038/nrg2521
159. Wilusz JE, Sunwoo H, Spector DL. Long noncoding RNAs: Functional surprises from the RNA world. Genes and Development. 2009. pp. 1494–1504. doi:10.1101/gad.1800909
160. Amaral PP, Clark MB, Gascoigne DK, Dinger ME, Mattick JS. LncRNAdb: A reference database for long noncoding RNAs. Nucleic Acids Res. 2011;39. doi:10.1093/nar/gkq1138
161. Ma L, Bajic VB, Zhang Z. On the classification of long non-coding RNAs. RNA Biology. 2013. pp. 924–933. doi:10.4161/rna.24604
162. Dupuis-Sandoval F, Poirier M, Scott MS. The emerging landscape of small nucleolar RNAs in cell biology. Wiley Interdisciplinary Reviews: RNA. 2015. pp. 381–397. doi:10.1002/wrna.1284
163. Derrien T, Johnson R, Bussotti G, Tanzer A, Djebali S, Tilgner H, et al. The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. Genome Res. 2012;22: 1775–1789. doi:10.1101/gr.132159.111
164. Kung JTY, Colognori D, Lee JT. Long noncoding RNAs: Past, present, and future. Genetics. 2013. pp. 651–669. doi:10.1534/genetics.112.146704
165. Luo S, Lu JY, Liu L, Yin Y, Chen C, Han X, et al. Divergent lncRNAs regulate gene expression and lineage differentiation in pluripotent cells. Cell Stem Cell. 2016;18: 637–652. doi:10.1016/j.stem.2016.01.024
166. Amin N, McGrath A, Chen YPP. Evaluation of deep learning in non-coding RNA classification. Nature Machine Intelligence. 2019. pp. 246–256. doi:10.1038/s42256-019-0051-2
167. Ma Y, Zhang X, Wang YZ, Tian H, Xu S. Research progress of circular RNAs in lung cancer. Cancer Biology and Therapy. 2019. pp. 123–129. doi:10.1080/15384047.2018.1523848
168. Kaikkonen MU, Adelman K. Emerging Roles of Non-Coding RNA Transcription. Trends in Biochemical Sciences. 2018. pp. 654–667. doi:10.1016/j.tibs.2018.06.002
169. Noh JH, Kim KM, McClusky WG, Abdelmohsen K, Gorospe M. Cytoplasmic functions of long noncoding RNAs. Wiley Interdisciplinary Reviews: RNA. 2018. doi:10.1002/wrna.1471
170. Jia H, Osak M, Bogu GK, Stanton LW, Johnson R, Lipovich L. Genome-wide computational identification and manual annotation of human long noncoding RNA genes. RNA. 2010;16: 1478–1487. doi:10.1261/rna.1951310
171. Lam MTY, Li W, Rosenfeld MG, Glass CK. Enhancer RNAs and regulated transcriptional programs. Trends in Biochemical Sciences. 2014. pp. 170–182. doi:10.1016/j.tibs.2014.02.007
172. Quinn JJ, Chang HY. Unique features of long non-coding RNA biogenesis and function. Nature Reviews Genetics. 2016. pp. 47–62. doi:10.1038/nrg.2015.10
173. Ventola GMM, Noviello TMR, D'Aniello S, Spagnuolo A, Ceccarelli M, Cerulo L. Identification of long non-coding transcripts with feature selection: a comparative study. BMC Bioinformatics. 2017;18: 187. doi:10.1186/s12859-017-1594-z
174. Dinger ME, Pang KC, Mercer TR, Mattick JS. Differentiating protein-coding and noncoding RNA: Challenges and ambiguities. PLoS Computational Biology. 2008. doi:10.1371/journal.pcbi.1000176
175. Cabili M, Trapnell C, Goff L, Koziol M, Tazon-Vega B, Regev A, et al. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. Genes Dev. 2011;25: 1915–1927. doi:10.1101/gad.17446611
176. Ravasi T, Suzuki H, Pang KC, Katayama S, Furuno M, Okunishi R, et al. Experimental validation of the regulated expression of large numbers of non-coding RNAs from the mouse genome. Genome Res. 2006;16: 11–19. doi:10.1101/gr.4200206
177. Guttman M, Amit I, Garber M, French C, Lin MF, Feldser D, et al. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. Nature. 2009;458: 223–227. doi:10.1038/nature07672

178. Wang KC, Chang HY. Molecular Mechanisms of Long Noncoding RNAs. Molecular Cell. 2011. pp. 904–914. doi:10.1016/j.molcel.2011.08.018
179. Pontier DB, Gribnau J. Xist regulation and function eXplored. Hum Genet. 2011;130: 223–236. doi:10.1007/s00439-011-1008-7
180. Martianov I, Ramadass A, Serra Barros A, Chow N, Akoulitchev A. Repression of the human dihydrofolate reductase gene by a non-coding interfering transcript. Nature. 2007;445: 666–670. doi:10.1038/nature05519
181. Wutz A, Rasmussen TP, Jaenisch R. Chromosomal silencing and localization are mediated by different domains of Xist RNA. Nat Genet. 2002;30: 167–174. doi:10.1038/ng820
182. Sun BK, Deaton AM, Lee JT. A transient heterochromatic state in Xist preempts X inactivation choice without RNA stabilization. Mol Cell. 2006;21: 617–628. doi:10.1016/j.molcel.2006.01.028
183. Huarte M, Guttman M, Feldser D, Garber M, Koziol MJ, Kenzelmann-Broz D, et al. A large intergenic noncoding RNA induced by p53 mediates global gene repression in the p53 response. Cell. 2010;142: 409–419. doi:10.1016/j.cell.2010.06.040
184. Collins K. Physiological assembly and activity of human telomerase complexes. Mech Ageing Dev. 2008;129: 91–98. doi:10.1016/j.mad.2007.10.008
185. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci. 1977;74: 5463–5467. doi:10.1073/pnas.74.12.5463
186. Maxam AM, Gilbert W. A new method for sequencing DNA. Proc Natl Acad Sci U S A. 1977;74: 560–564. doi:10.1073/pnas.74.2.560
187. Maniatis T, Jeffrey A, van deSande H. Chain Length Determination of Small Doubleand Single-Stranded DNA Molecules by Polyacrylamide Gel Electrophoresis. Biochemistry. 1975;14: 3787–3794. doi:10.1021/bi00688a010
188. Anderson S. Shotgun DNA sequencing using cloned DNase I-generated fragments. Nucleic Acids Res. 1981;9: 3015–3027. doi:10.1093/nar/9.13.3015
189. Staden R. A strategy of DNA sequencing employing computer programs. Nucleic Acids Res. 1979;6: 2601–2610. doi:10.1093/nar/6.7.2601
190. Zhang JZ, Fang Y, Hou JY, Ren HJ, Jiang R, Roos P, et al. Use of Non-Cross-Linked Polyacrylamide for Four-Color DNA Sequencing by Capillary Electrophoresis Separation of Fragments up to 640 Bases in Length in Two Hours. Anal Chem. 1995;67: 4589–4593. doi:10.1021/ac00120a026
191. Craig Venter J, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, et al. The sequence of the human genome. Science. 2001;291: 1304–1351. doi:10.1126/science.1058040
192. Abdellah Z, Ahmadi A, Ahmed S, Aimable M, Ainscough R, Almeida J, et al. Finishing the euchromatic sequence of the human genome. Nature. 2004;431: 931–945. doi:10.1038/nature03001
193. Ronaghi M, Karamohamed S, Pettersson B, Uhlén M, Nyrén P. Real-time DNA sequencing using detection of pyrophosphate release. Anal Biochem. 1996;242: 84–89. doi:10.1006/abio.1996.0432
194. Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, et al. Accurate whole human genome sequencing using reversible terminator chemistry. Nature. 2008. doi:10.1038/nature07517
195. Kozarewa I, Ning Z, Quail MA, Sanders MJ, Berriman M, Turner DJ. Amplification-free Illumina sequencing-library preparation facilitates improved mapping and assembly of (G+C)-biased genomes. Nat Methods. 2009;6: 291–295. doi:10.1038/nmeth.1311
196. Shendure J, Balasubramanian S, Church GM, Gilbert W, Rogers J, Schloss JA, et al. DNA sequencing at 40: Past, present and future. Nature. 2017;550. doi:10.1038/nature24286
197. Wheeler DA, Srinivasan M, Egholm M, Shen Y, Chen L, McGuire A, et al. The complete genome of an individual by massively parallel DNA sequencing. Nature. 2008;452: 872–876. doi:10.1038/nature06884
198. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, et al. Genome sequencing in microfabricated high-density picolitre reactors. Nature. 2005;437: 376–380. doi:10.1038/nature03959

199. Drmanac R, Sparks AB, Callow MJ, Halpern AL, Burns NL, Kermani BG, et al. Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. Science. 2010;327: 78–81. doi:10.1126/science.1181498
200. Goodwin S, McPherson JD, McCombie WR. Coming of age: Ten years of next-generation sequencing technologies. Nature Reviews Genetics. 2016. pp. 333–351. doi:10.1038/nrg.2016.49
201. Rothberg JM, Leamon JH. The development and impact of 454 sequencing. Nature Biotechnology. 2008. pp. 1117–1124. doi:10.1038/nbt1485
202. Valouev A, Ichikawa J, Tonthat T, Stuart J, Ranade S, Peckham H, et al. A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning. Genome Res. 2008;18: 1051–1063. doi:10.1101/gr.076463.108
203. Illumina. An introduction to Next-Generation Sequencing Technology. 2017.
204. Meienberg J, Bruggmann R, Oexle K, Matyas G. Clinical sequencing: is WGS the better WES? Hum Genet. 2016;135: 359–362. doi:10.1007/s00439-015-1631-9
205. Brunelli L, Jenkins SM, Gudgeon JM, Bleyl SB, Miller CE, Tvrdik T, et al. Targeted gene panel sequencing for the rapid diagnosis of acutely ill infants. Mol Genet Genomic Med. 2019;7. doi:10.1002/mgg3.796
206. Strom SP, Lee H, Das K, Vilain E, Nelson SF, Grody WW, et al. Assessing the necessity of confirmatory testing for exome-sequencing results in a clinical molecular diagnostic laboratory. Genet Med. 2014;16: 510–515. doi:10.1038/gim.2013.183
207. Baudhuin LM, Lagerstedt SA, Klee EW, Fadra N, Oglesbee D, Ferber MJ. Confirming variants in next-generation sequencing panel testing by sanger sequencing. J Mol Diagnostics. 2015;17: 456–461. doi:10.1016/j.jmoldx.2015.03.004
208. Nakazato T, Ohta T, Bono H. Experimental Design-Based Functional Mining and Characterization of High-Throughput Sequencing Data in the Sequence Read Archive. PLoS One. 2013;8. doi:10.1371/journal.pone.0077910
209. Meacham F, Boffelli D, Dhahbi J, Martin DIK, Singer M, Pachter L. Identification and correction of systematic error in high-throughput sequence data. BMC Bioinformatics. 2011;12. doi:10.1186/1471-2105-12-451
210. Loman NJ, Misra R V., Dallman TJ, Constantinidou C, Gharbia SE, Wain J, et al. Performance comparison of benchtop high-throughput sequencing platforms. Nat Biotechnol. 2012. doi:10.1038/nbt.2198
211. Schirmer M, D'Amore R, Ijaz UZ, Hall N, Quince C. Illumina error profiles: Resolving fine-scale variation in metagenomic sequencing data. BMC Bioinformatics. 2016;17. doi:10.1186/s12859-016-0976-y
212. Pfeiffer F, Gröber C, Blank M, Händler K, Beyer M, Schultze JL, et al. Systematic evaluation of error rates and causes in short samples in next-generation sequencing. Sci Rep. 2018;8. doi:10.1038/s41598-018-29325-6
213. Ma X, Shao Y, Tian L, Flasch DA, Mulder HL, Edmonson MN, et al. Analysis of error profiles in deep next-generation sequencing data. Genome Biol. 2019;20. doi:10.1186/s13059-019-1659-6
214. Stoler N, Nekrutenko A. Sequencing error profiles of Illumina sequencing instruments. NAR Genomics Bioinforma. 2021;3. doi:10.1093/nargab/lqab019
215. Lou DI, McBee RM, Sawyer SL, Hussmann JA, Press WH, Acevedo A, et al. High-Throughput dna sequencing errors are reduced by orders of magnitude using circle sequencing. Proc Natl Acad Sci U S A. 2013;110: 19872–19877. doi:10.1073/pnas.1319590110
216. Mirkin SM. Expandable DNA repeats and human disease. Nature. 2007. pp. 932–940. doi:10.1038/nature05977
217. Stankiewicz P, Lupski JR. Structural variation in the human genome and its role in disease. Annual Review of Medicine. 2010. pp. 437–455. doi:10.1146/annurev-med-100708-204735

218. Levene HJ, Korlach J, Turner SW, Foquet M, Craighead HG, Webb WW. Zero-mode waveguides for single-molecule analysis at high concentrations. Science. 2003;299: 682–686. doi:10.1126/science.1079700
219. Eid J, Fehr A, Gray J, Luong K, Lyle J, Otto G, et al. Real-time DNA sequencing from single polymerase molecules. Science. 2009;323: 133–138. doi:10.1126/science.1162986
220. Branton D, Deamer DW, Marziali A, Bayley H, Benner SA, Butler T, et al. The potential and challenges of nanopore sequencing. Nature Biotechnology. 2008. pp. 1146–1153. doi:10.1038/nbt.1495
221. Clarke J, Wu HC, Jayasinghe L, Patel A, Reid S, Bayley H. Continuous base identification for single-molecule nanopore DNA sequencing. Nat Nanotechnol. 2009;4: 265–270. doi:10.1038/nnano.2009.12
222. Amarasinghe SL, Su S, Dong X, Zappia L, Ritchie ME, Gouil Q. Opportunities and challenges in long-read sequencing data analysis. Genome Biology. 2020. doi:10.1186/s13059-020-1935-5
223. Tedersoo L, Albertsen M, Anslan S, Callahan B. Perspectives and Benefits of High-Throughput Long-Read Sequencing in Microbial Ecology. Appl Environ Microbiol. 2021;87: 1–19. doi:10.1128/AEM.00626-21
224. Logsdon GA, Vollger MR, Eichler EE. Long-read human genome sequencing and its applications. Nature Reviews Genetics. 2020. pp. 597–614. doi:10.1038/s41576-020-0236-x
225. Holley RW, Apgar J, Everett GA, Madison JT, Marquisee M, Merrill SH, et al. Structure of a ribonucleic acid. Science. 1965;147: 1462–1465. doi:10.1126/science.147.3664.1462
226. Temin HM, Mizutani S. Viral RNA-dependent DNA polymerase: RNA-dependent DNA polymerase in virions of Rous sarcoma virus. Nature. 1970;226: 1211–1213. doi:10.1038/2261211a0
227. Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, et al. A survey of best practices for RNA-seq data analysis. Genome Biology. 2016. doi:10.1186/s13059-016-0881-8
228. Blencowe BJ, Ahmad S, Lee LJ. Current-generation high-throughput sequencing: Deepening insights into mammalian transcriptomes. Genes and Development. 2009. pp. 1379–1386. doi:10.1101/gad.1788009
229. Sims D, Sudbery I, Ilott NE, Heger A, Ponting CP. Sequencing depth and coverage: Key considerations in genomic analyses. Nature Reviews Genetics. 2014. pp. 121–132. doi:10.1038/nrg3642
230. Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. Nature Reviews Genetics. 2019. pp. 631–656. doi:10.1038/s41576-019-0150-2
231. Fu GK, Xu W, Wilhelmy J, Mindrinos MN, Davis RW, Xiao W, et al. Molecular indexing enables quantitative targeted RNA sequencing and reveals poor efficiencies in standard library preparations. Proc Natl Acad Sci U S A. 2014;111: 1891–1896. doi:10.1073/pnas.1323732111
232. Sharon D, Tilgner H, Grubert F, Snyder M. A single-molecule long-read survey of the human transcriptome. Nat Biotechnol. 2013;31: 1009–1014. doi:10.1038/nbt.2705
233. Oikonomopoulos S, Wang YC, Djambazian H, Badescu D, Ragoussis J. Benchmarking of the Oxford Nanopore MinION sequencing for quantitative and qualitative assessment of cDNA populations. Sci Rep. 2016;6. doi:10.1038/srep31602
234. Garalde DR, Snell EA, Jachimowicz D, Sipos B, Lloyd JH, Bruce M, et al. Highly parallel direct RN A sequencing on an array of nanopores. Nat Methods. 2018;15: 201–206. doi:10.1038/nmeth.4577
235. Tilgner H, Grubert F, Sharon D, Snyder MP. Defining a personal, allele-specific, and single-molecule long-read transcriptome. Proc Natl Acad Sci U S A. 2014;111: 9869–9874. doi:10.1073/pnas.1400447111
236. Andrews S, others. FastQC: a quality control tool for high throughput sequence data. 2010. Https://Www.Bioinformatics.Babraham.Ac.Uk/Projects/Fastqc/. Babraham Bioinformatics; 2010. p. http://www.bioinformatics.babraham.ac.uk/projects/. Available: http://www.bioinformatics.babraham.ac.uk/projects/fastqc

237. Patel RK, Jain M. NGS QC toolkit: A toolkit for quality control of next generation sequencing data. PLoS One. 2012;7. doi:10.1371/journal.pone.0030619
238. Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res. 2009;38: 1767–1771. doi:10.1093/nar/gkp1137
239. Ewing B, Green P. Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res. 1998;8: 186–194. doi:10.1101/gr.8.3.186
240. Guo Y, Ye F, Sheng Q, Clark T, Samuels DC. Three-stage quality control strategies for DNA re-sequencing data. Brief Bioinform. 2013;15: 879–889. doi:10.1093/bib/bbt069
241. Tan G, Opitz L, Schlapbach R, Rehrauer H. Long fragments achieve lower base quality in Illumina paired-end sequencing. Sci Rep. 2019;9. doi:10.1038/s41598-019-39076-7
242. Felix Krueger. Trim Galore! Babraham Bioinformatics; 2012. Available: https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
243. Bolger AM, Lohse M, Usadel B. Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics. 2014;30: 2114–2120. doi:10.1093/bioinformatics/btu170
244. Stratton M. Genome resequencing and genetic variation. Nature Biotechnology. 2008. pp. 65–66. doi:10.1038/nbt0108-65
245. Nurk S, Koren S, Rhie A, Rautiainen M, Bzikadze A V, Mikheenko A, et al. The complete sequence of a human genome. Science. 2022;376: 44–53. doi:10.1126/science.abj6987
246. Kidd JM, Sampas N, Antonacci F, Graves T, Fulton R, Hayden HS, et al. Characterization of missing human genome sequences and copy-number polymorphic insertions. Nat Methods. 2010;7: 365–371. doi:10.1038/nmeth.1451
247. Sudmant PH, Kitzman JO, Antonacci F, Alkan C, Malig M, Tsalenko A, et al. Diversity of human copy number variation and multicopy genes. Science. 2010;330: 641–646. doi:10.1126/science.1197005
248. Kidd JM, Cooper GM, Donahue WF, Hayden HS, Sampas N, Graves T, et al. Mapping and sequencing of structural variation from eight human genomes. Nature. 2008;453: 56–64. doi:10.1038/nature06862
249. Church DM, Schneider VA, Steinberg KM, Schatz MC, Quinlan AR, Chin CS, et al. Extending reference assembly models. Genome Biol. 2015;16. doi:10.1186/s13059-015-0587-3
250. Church DM, Schneider VA, Graves T, Auger K, Cunningham F, Bouk N, et al. Modernizing reference genome assemblies. PLoS Biol. 2011;9. doi:10.1371/journal.pbio.1001091
251. Li H, Wren J. Toward better understanding of artifacts in variant calling from high-coverage samples. Bioinformatics. 2014. pp. 2843–2851. doi:10.1093/bioinformatics/btu356
252. Schneider VA, Graves-Lindsay T, Howe K, Bouk N, Chen HC, Kitts PA, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. Genome Res. 2017;27: 849–864. doi:10.1101/gr.213611.116
253. Guo Y, Dai Y, Yu H, Zhao S, Samuels DC, Shyr Y. Improvements and impacts of GRCh38 human reference on high throughput sequencing data analysis. Genomics. 2017;109: 83–90. doi:10.1016/j.ygeno.2017.01.005
254. Eichler EE, Clark RA, She X. An assessment of the sequence gaps: Unfinished business in a finished human genome. Nature Reviews Genetics. 2004. pp. 345–354. doi:10.1038/nrg1322
255. Miga KH, Newton Y, Jain M, Altemose N, Willard HF, Kent EJ. Centromere reference models for human chromosomes X and y satellite arrays. Genome Res. 2014;24: 697–707. doi:10.1101/gr.159624.113
256. Steinberg KM, Schneider VA, Graves-Lindsay TA, Fulton RS, Agarwala R, Huddleston J, et al. Single haplotype assembly of the human genome from a hydatidiform mole. Genome Res. 2014;24: 2066–2076. doi:10.1101/gr.180893.114
257. Berlin K, Koren S, Chin CS, Drake JP, Landolin JM, Phillippy AM. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat Biotechnol. 2015;33: 623–630. doi:10.1038/nbt.3238

258. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. Genomics. 2010. pp. 315–327. doi:10.1016/j.ygeno.2010.03.001
259. Nagarajan N, Pop M. Sequence assembly demystified. Nature Reviews Genetics. 2013. pp. 157–167. doi:10.1038/nrg3367
260. Liao X, Li M, Zou Y, Wu FX, Yi-Pan, Wang J. Current challenges and solutions of de novo assembly. Quantitative Biology. 2019. pp. 90–109. doi:10.1007/s40484-019-0166-9
261. Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP, Flanigan MJ, et al. A whole-genome assembly of Drosophila. Science. 2000. pp. 2196–2204. doi:10.1126/science.287.5461.2196
262. Jaffe DB, Butler J, Gnerre S, Mauceli E, Lindblad-Toh K, Mesirov JP, et al. Whole-genome sequence assembly for mammalian genomes: Arachne 2. Genome Res. 2003;13: 91–96. doi:10.1101/gr.828403
263. Huang X, Wang J, Aluru S, Yang SP, Hillier LD. PCAP: A whole-genome assembly program. Genome Res. 2003;13: 2164–2170. doi:10.1101/gr.1390403
264. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: Scalable and accurate long-read assembly via adaptive κ-mer weighting and repeat separation. Genome Res. 2017;27: 722–736. doi:10.1101/gr.215087.116
265. Treangen TJ, Sommer DD, Angly FE, Koren S, Pop M. Next generation sequence assembly with AMOS. Curr Protoc Bioinforma. 2011; 1–18. doi:10.1002/0471250953.bi1108s33
266. Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, et al. SOAPdenovo2: An empirically improved memory-efficient short-read de novo assembler. GigaScience. 2012. doi:10.1186/2047-217X-1-18
267. Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJM, Birol I. ABySS: A parallel assembler for short read sequence data. Genome Res. 2009;19: 1117–1123. doi:10.1101/gr.089532.108
268. Butler J, MacCallum I, Kleber M, Shlyakhter IA, Belmonte MK, Lander ES, et al. ALLPATHS: De novo assembly of whole-genome shotgun microreads. Genome Res. 2008;18: 810–820. doi:10.1101/gr.7337908
269. Luo J, Wang J, Li W, Zhang Z, Wu FX, Li M, et al. EPGA2: Memory-efficient de novo assembler. Bioinformatics. 2015;31: 3988–3990. doi:10.1093/bioinformatics/btv487
270. Lin Y, Yuan J, Kolmogorov M, Shen MW, Chaisson M, Pevzner PA. Assembly of long error-prone reads using de Bruijn graphs. Proc Natl Acad Sci U S A. 2016;113: E8396–E8405. doi:10.1073/pnas.1604560113
271. Di Genova A, Buena-Atienza E, Ossowski S, Sagot MF. Efficient hybrid de novo assembly of human genomes with WENGAN. Nat Biotechnol. 2021;39: 422–430. doi:10.1038/s41587-020-00747-w
272. Idury RM, Waterman MS. A New Algorithm for DNA Sequence Assembly. J Comput Biol. 1995;2: 291–306. doi:10.1089/cmb.1995.2.291
273. Simpson JT, Durbin R. Efficient de novo assembly of large genomes using compressed data structures. Genome Res. 2012;22: 549–556. doi:10.1101/gr.126953.111
274. Chin CS, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, et al. Phased diploid genome assembly with single-molecule real-time sequencing. Nat Methods. 2016;13: 1050–1054. doi:10.1038/nmeth.4035
275. Nurk S, Walenz BP, Rhie A, Vollger MR, Logsdon GA, Grothe R, et al. HiCanu: Accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. Genome Res. 2020;30: 1291–1305. doi:10.1101/GR.263566.120
276. Porubsky D, Ebert P, Audano PA, Vollger MR, Harvey WT, Marijon P, et al. Fully phased human genome assembly without parental data using single-cell strand sequencing and long reads. Nat Biotechnol. 2020. doi:10.1038/s41587-020-0719-5
277. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, et al. The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. 2010;20: 1297–1303. doi:10.1101/gr.107524.110

278. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, Van Baren MJ, et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol. 2010;28: 511–515. doi:10.1038/nbt.1621
279. Langmead B, Hansen KD, Leek JT. Cloud-scale RNA-sequencing differential expression analysis with Myrna. Genome Biol. 2010;11. doi:10.1186/gb-2010-11-8-r83
280. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9: 357–359. doi:10.1038/nmeth.1923
281. Li H, Homer N. A survey of sequence alignment algorithms for next-generation sequencing. Briefings in Bioinformatics. 2010. pp. 473–483. doi:10.1093/bib/bbq015
282. Schbath S, Martin V, Zytnicki M, Fayolle J, Loux V, Gibrat JF. Mapping reads on a genomic sequence: An algorithmic overview and a practical comparative analysis. J Comput Biol. 2012;19: 796–813. doi:10.1089/cmb.2012.0022
283. Shang J, Zhu F, Vongsangnak W, Tang Y, Zhang W, Shen B. Evaluation and comparison of multiple aligners for next-generation sequencing data analysis. Biomed Res Int. 2014;2014. doi:10.1155/2014/309650
284. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215: 403–410. doi:10.1016/S0022-2836(05)80360-2
285. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981;147: 195–197. doi:10.1016/0022-2836(81)90087-5
286. Li R, Li Y, Kristiansen K, Wang J. SOAP: Short oligonucleotide alignment program. Bioinformatics. 2008;24: 713–714. doi:10.1093/bioinformatics/btn025
287. Jiang H, Wong WH. SeqMap: Mapping massive amount of oligonucleotides to the genome. Bioinformatics. 2008;24: 2395–2396. doi:10.1093/bioinformatics/btn429
288. Li H, Ruan J, Durbin R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. 2008;18: 1851–1858. doi:10.1101/gr.078212.108
289. Smith AD, Xuan Z, Zhang MQ. Using quality scores and longer reads improves accuracy of Solexa read mapping. BMC Bioinformatics. 2008;9. doi:10.1186/1471-2105-9-128
290. Chen Y, Souaiaia T, Chen T. PerM: Efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. Bioinformatics. 2009;25: 2514–2521. doi:10.1093/bioinformatics/btp486
291. Lin H, Zhang Z, Zhang MQ, Ma B, Li M. ZOOM! Zillions of oligos mapped. Bioinformatics. 2008;24: 2431–2437. doi:10.1093/bioinformatics/btn416
292. Rumble SM, Lacroute P, Dalca A V., Fiume M, Sidow A, Brudno M. SHRiMP: Accurate mapping of short color-space reads. PLoS Comput Biol. 2009;5. doi:10.1371/journal.pcbi.1000386
293. Weese D, Emde AK, Rausch T, Döring A, Reinert K. RazerS - Fast read mapping with sensitivity control. Genome Res. 2009;19: 1646–1654. doi:10.1101/gr.088823.108
294. Abouelhoda MI, Kurtz S, Ohlebusch E. Replacing suffix trees with enhanced suffix arrays. J Discret Algorithms. 2004;2: 53–86. doi:10.1016/S1570-8667(03)00065-0
295. Ferragina P, Manzini G. Opportunistic data structures with applications. Annual Symposium on Foundations of Computer Science - Proceedings. 2000. pp. 390–398. doi:10.1109/sfcs.2000.892127
296. Burrows M, Wheeler D. A block-sorting lossless data compression algorithm. Algorithm, Data Compression. 1994; 18. doi:10.1.1.37.6774
297. Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C, et al. Versatile and open software for comparing large genomes. Genome Biol. 2004;5. doi:10.1186/gb-2004-5-2-r12
298. Meek C, Patel JM KS. OASIS: an online and accurate technique for local-alignment searches on biological sequences. Proc 29th Int Conf Very Large Data Bases (VLDB 2003). 2003; (pg. 910-21).
299. Hoffmann S, Otto C, Kurtz S, Sharma CM, Khaitovich P, Vogel J, et al. Fast mapping of short sequences with mismatches, insertions and deletions using index structures. PLoS Comput Biol. 2009;5. doi:10.1371/journal.pcbi.1000502

300. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009;10. doi:10.1186/gb-2009-10-3-r25
301. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics. 2009;25: 1754–1760. doi:10.1093/bioinformatics/btp324
302. Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, et al. SOAP2: An improved ultrafast tool for short read alignment. Bioinformatics. 2009;25: 1966–1967. doi:10.1093/bioinformatics/btp336
303. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment / Map format and SAMtools. Bioinformatics. 2009;25: 2078–2079. doi:10.1093/bioinformatics/btp352
304. The SAM/BAM Format Specification Working Group. Sequence Alignment/Map Format Specification. Available: https://github.com/samtools/hts-specs
305. Bonfield JK, Marshall J, Danecek P, Li H, Ohan V, Whitwham A, et al. HTSlib: C library for reading/writing high-throughput sequencing data. Gigascience. 2021;10. doi:10.1093/gigascience/giab007
306. Li H. Minimap2: Pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34: 3094–3100. doi:10.1093/bioinformatics/bty191
307. Kuhn RM, Haussler D, James Kent W. The UCSC genome browser and associated tools. Brief Bioinform. 2013;14: 144–161. doi:10.1093/bib/bbs038
308. Ng PC, Henikoff S. SIFT: Predicting amino acid changes that affect protein function. Nucleic Acids Res. 2003;31: 3812–3814. doi:10.1093/nar/gkg509
309. Altmann A, Weber P, Bader D, Preuß M, Binder EB, Müller-Myhsok B. A beginners guide to SNP calling from high-Throughput DNA-sequencing data. Human Genetics. 2012. pp. 1541–1554. doi:10.1007/s00439-012-1213-z
310. Abel HJ, Duncavage EJ. Detection of structural DNA variation from next generation sequencing data: A review of informatic approaches. Cancer Genetics. 2013. doi:10.1016/j.cancergen.2013.11.002
311. Escaramís G, Docampo E, Rabionet R. A decade of structural variants: Description, history and methods to detect structural variation. Brief Funct Genomics. 2015;14: 305–314. doi:10.1093/bfgp/elv014
312. Guan P, Sung W-K. Structural variation detection using next-generation sequencing data: A comparative technical review. Methods. 2016;102. doi:http://dx.doi.org/10.1016/j.ymeth.2016.01.020
313. Mahmoud M, Gobet N, Cruz-Dávalos DI, Mounier N, Dessimoz C, Sedlazeck FJ. Structural variant calling: The long and the short of it. Genome Biology. 2019. doi:10.1186/s13059-019-1828-7
314. Zverinova S, Guryev V. Variant calling: Considerations, practices, and developments. Human Mutation. 2021. doi:10.1002/humu.24311
315. Fan X, Abbott TE, Larson D, Chen K. BreakDancer: Identification of genomic structural variation from paired-end read mapping. Curr Protoc Bioinforma. 2014. doi:10.1002/0471250953.bi1506s45
316. Zhenxian Zheng, Shumin Li, Junhao Su, Amy Wing-Sze Leung, Tak-Wah Lam RL. Symphonizing pileup and full-alignment for deep learning-based long-read variant calling. bioRxiv. 2021. doi:https://doi.org/10.1101/2021.12.29.474431
317. Suzuki S, Yasuda T, Shiraishi Y, Miyano S, Nagasaki M. ClipCrop: a tool for detecting structural variations with single-base resolution using soft-clipping information. BMC Bioinformatics. 2011;12 Suppl 1. doi:10.1186/1471-2105-12-S14-S7
318. Suvakov M, Panda A, Diesh C, Holmes I, Abyzov A. CNVpytor: a tool for copy number variation detection and analysis from read depth and allele imbalance in whole-genome sequencing. Gigascience. 2021;10. doi:10.1093/gigascience/giab074
319. Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G. De novo assembly and genotyping of variants using colored de Bruijn graphs. Nat Genet. 2012;44: 226–232. doi:10.1038/ng.1028
320. Poplin R, Chang PC, Alexander D, Schwartz S, Colthurst T, Ku A, et al. A universal snp and small-indel variant caller using deep neural networks. Nature Biotechnology. 2018. p. 983. doi:10.1038/nbt.4235

321. Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, Korbel JO. DELLY: Structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics. 2012;28. doi:10.1093/bioinformatics/bts378
322. Albers CA, Lunter G, MacArthur DG, McVean G, Ouwehand WH, Durbin R. Dindel: Accurate indel calls from short-read data. Genome Res. 2011;21: 961–973. doi:10.1101/gr.112326.110
323. Li H. FermiKit: Assembly-based variant calling for Illumina resequencing data. Bioinformatics. 2015;31: 3694–3696. doi:10.1093/bioinformatics/btv440
324. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. arXiv Prepr. 2012. doi:arXiv:1207.3907 [q-bio.GN]
325. Sindi SS, Önal S, Peng LC, Wu HT, Raphael BJ. An integrative probabilistic model for identification of structural variation in sequencing data. Genome Biol. 2012;13. doi:10.1186/gb-2012-13-3-r22
326. Poplin R, Ruano-Rubio V, DePristo MA, Fennell TJ, Carneiro MO, Auwera GA Van der, et al. Scaling accurate genetic variant discovery to tens of thousands of samples. bioRxiv. 2017; 201178. doi:10.1101/201178
327. Cameron DL, Schröder J, Penington JS, Do H, Molania R, Dobrovic A, et al. GRIDSS: Sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. Genome Res. 2017;27: 2050–2060. doi:10.1101/gr.222109.117
328. Quinlan AR, Clark RA, Sokolova S, Leibowitz ML, Zhang Y, Hurles ME, et al. Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. Genome Res. 2010;20: 623–635. doi:10.1101/gr.102970.109
329. Ratan A, Olson TL, Loughran TP, Miller W. Identification of indels in next-generation sequencing data. BMC Bioinformatics. 2015;16. doi:10.1186/s12859-015-0483-6
330. Au CH, Leung AYH, Kwong A, Chan TL, Ma ESK. INDELseek: detection of complex insertions and deletions from next-generation sequencing data. BMC Genomics. 2017;18: 16. doi:10.1186/s12864-016-3449-9
331. Layer RM, Chiang C, Quinlan AR, Hall IM. LUMPY: A probabilistic framework for structural variant discovery. Genome Biol. 2014;15. doi:10.1186/gb-2014-15-6-r84
332. Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Källberg M, et al. Manta: Rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics. 2016;32: 1220–1222. doi:10.1093/bioinformatics/btv710
333. Mohiyuddin M, Mu JC, Li J, Bani Asadi N, Gerstein MB, Abyzov A, et al. MetaSV: An accurate and integrative structural-variant caller for next generation sequencing. Bioinformatics. 2015;31: 2741–2744. doi:10.1093/bioinformatics/btv204
334. Cooke DP, Wedge DC, Lunter G. A unified haplotype-based method for accurate and comprehensive variant calling. Nat Biotechnol. 2021;39: 885–892. doi:10.1038/s41587-021-00861-3
335. Korbel JO, Abyzov A, Mu XJ, Carriero N, Cayting P, Zhang Z, et al. PEMer: A computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. Genome Biol. 2009;10. doi:10.1186/gb-2009-10-2-r23
336. de Araújo Lima L, Wang K. PennCNV in whole-genome sequencing data. BMC Bioinformatics. 2017;18. doi:10.1186/s12859-017-1802-x
337. Rimmer A, Phan H, Mathieson I, Iqbal Z, Twigg SRF, Wilkie AOM, et al. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. Nat Genet. 2014;46: 912–918. doi:10.1038/ng.3036
338. Ye K, Schulz MH, Long Q, Apweiler R, Ning Z. Pindel: A pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics. 2009;25: 2865–2871. doi:10.1093/bioinformatics/btp394
339. Fang H, Bergmann EA, Arora K, Vacic V, Zody MC, Iossifov I, et al. Indel variant analysis of short-read sequencing data with Scalpel. Nat Protoc. 2016;11: 2529–2548. doi:10.1038/nprot.2016.150

340. Yang R, Nelson AC, Henzler C, Thyagarajan B, Silverstein KAT. ScanIndel: A hybrid framework for indel detection via gapped alignment, split reads and de novo assembly. Genome Med. 2015;7. doi:10.1186/s13073-015-0251-2
341. Bartenhagen C, Dugas M. Robust and exact structural variation detection with paired-end and soft-clipped alignments: SoftSV compared with eight algorithms. Brief Bioinform. 2016;17: 51–62. doi:10.1093/bib/bbv028
342. Wala JA, Bandopadhayay P, Greenwald NF, O'Rourke R, Sharpe T, Stewart C, et al. SvABA: Genome-wide detection of structural variants and indels by local assembly. Genome Res. 2018;28: 581–591. doi:10.1101/gr.221028.117
343. Zeitouni B, Boeva V, Janoueix-Lerosey I, Loeillet S, Legoix-né P, Nicolas A, et al. SVDetect: A tool to identify genomic structural variations from paired-end and mate-pair sequencing data. Bioinformatics. 2010;26: 1895–1896. doi:10.1093/bioinformatics/btq293
344. Gillet-Markowska A, Richard H, Fischer G, Lafontaine I. Ulysses: Accurate detection of low-frequency structural variations in large insert-size sequencing libraries. Bioinformatics. 2015;31: 801–808. doi:10.1093/bioinformatics/btu730
345. Lai Z, Markovets A, Ahdesmaki M, Chapman B, Hofmann O, Mcewen R, et al. VarDict: A novel and versatile variant caller for next-generation sequencing in cancer research. Nucleic Acids Res. 2016;44. doi:10.1093/nar/gkw227
346. Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, et al. VarScan: Variant detection in massively parallel sequencing of individual and pooled samples. Bioinformatics. 2009;25: 2283–2285. doi:10.1093/bioinformatics/btp373
347. Kronenberg ZN, Osborne EJ, Cone KR, Kennedy BJ, Domyan ET, Shapiro MD, et al. Wham: Identifying Structural Variants of Biological Consequence. PLoS Comput Biol. 2015;11. doi:10.1371/journal.pcbi.1004572
348. O'Rawe J, Jiang T, Sun G, Wu Y, Wang W, Hu J, et al. Low concordance of multiple variant-calling pipelines: Practical implications for exome and genome sequencing. Genome Med. 2013;5. doi:10.1186/gm432
349. Spencer DH, Abel HJ, Lockwood CM, Payton JE, Szankasi P, Kelley TW, et al. Detection of FLT3 internal tandem duplication in targeted, short-read-length, next-generation sequencing data. J Mol Diagnostics. 2013;15: 81–93. doi:10.1016/j.jmoldx.2012.08.001
350. Depristo MA, Banks E, Poplin R, Garimella K V., Maguire JR, Hartl C, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet. 2011;43: 491–501. doi:10.1038/ng.806
351. Narzisi G, O'Rawe JA, Iossifov I, Fang H, Lee YH, Wang Z, et al. Accurate de novo and transmitted indel detection in exome-capture data using microassembly. Nat Methods. 2014;11: 1033–1036. doi:10.1038/nmeth.3069
352. Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, et al. BreakDancer: An algorithm for high-resolution mapping of genomic structural variation. Nat Methods. 2009;6: 677–681. doi:10.1038/nmeth.1363
353. Cai L, Wu Y, Gao J. DeepSV: Accurate calling of genomic deletions from high-throughput sequencing data using deep convolutional neural network. BMC Bioinformatics. 2019;20. doi:10.1186/s12859-019-3299-y
354. Popic V, Rohlicek C, Cunial F, Hajirasouliha I, Meleshko D, Garimella K, et al. Cue: a deep-learning framework for structural variant discovery and genotyping. Nat Methods. 2023. doi:10.1038/s41592-023-01799-x
355. Shrestha AMS, Yoshikawa N, Asai K. Combining probabilistic alignments with read pair information improves accuracy of split-alignments. Bioinformatics. 2018;34: 3631–3637. doi:10.1093/bioinformatics/bty398
356. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, et al. The variant call format and VCFtools. Bioinformatics. 2011. doi:10.1093/bioinformatics/btr330

357. The Global Alliance for Genomics & Health. The Variant Call Format Specification. Available: https://github.com/samtools/hts-specs
358. Talwalkar A, Liptrap J, Newcomb J, Hartl C, Terhorst J, Curtis K, et al. SMaSH: A benchmarking toolkit for human genome variant calling. Bioinformatics. 2014. doi:10.1093/bioinformatics/btu345
359. Jacobs K. Variant Graph Comparison Tool. Github; Available: https://github.com/bioinformed/vgraph
360. Cleary JG, Braithwaite R, Gaastra K, Hilbush BS, Inglis S, Irvine SA, et al. Comparing Variant Call Files for Performance Benchmarking of Next-Generation Sequencing Variant Calling Pipelines. bioRxiv. 2015. doi:10.1101/023754
361. Garrison E. Vcflib, a C++ library for parsing and manipulating VCF files. In: GitHub [Internet]. 2016. Available: https://github.com/vcflib/vcflib
362. Krishnan V, Utiramerur S, Ng Z, Datta S, Snyder MP, Ashley EA. Benchmarking workflows to assess performance and suitability of germline variant calling pipelines in clinical diagnostic assays. BMC Bioinformatics. 2021;22. doi:10.1186/s12859-020-03934-3
363. Yen JL, Garcia S, Montana A, Harris J, Chervitz S, Morra M, et al. A variant by any name: Quantifying annotation discordance across tools and clinical databases. Genome Med. 2017;9. doi:10.1186/s13073-016-0396-7
364. Wildeman M, Van Ophuizen E, Den Dunnen JT, Taschner PEM. Improving sequence variant descriptions in mutation databases and literature using the mutalyzer sequence variation nomenclature checker. Hum Mutat. 2008;29: 6–13. doi:10.1002/humu.20654
365. Lefter M, Vis JK, Vermaat M, den Dunnen JT, Taschner PEM, Laros JFJ. Mutalyzer 2: next generation HGVS nomenclature checker. Bioinformatics. 2021;37: 2811–2817. doi:10.1093/bioinformatics/btab051
366. McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GRS, Thormann A, et al. The Ensembl Variant Effect Predictor. Genome Biol. 2016;17. doi:10.1186/s13059-016-0974-4
367. Cingolani P, Platts A, Wang LL, Coon M, Nguyen T, Wang L, et al. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. Fly (Austin). 2012;6: 80–92. doi:10.4161/fly.19695
368. Wang K, Li M, Hakonarson H. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res. 2010;38. doi:10.1093/nar/gkq603
369. Freeman PJ, Hart RK, Gretton LJ, Brookes AJ, Dalgleish R. VariantValidator: Accurate validation, mapping, and formatting of sequence variation descriptions. Hum Mutat. 2018;39: 61–68. doi:10.1002/humu.23348
370. Hart RK, Rico R, Hare E, Garcia J, Westbrook J, Fusaro VA. A Python package for parsing, validating, mapping and formatting sequence variants using HGVS nomenclature. Bioinformatics. 2015;31: 268–270. doi:10.1093/bioinformatics/btu630
371. Neuman JA, Isakov O, Shomron N. Analysis of insertion-deletion from deep-sequencing data: Software evaluation for optimal detection. Brief Bioinform. 2013;14: 46–55. doi:10.1093/bib/bbs013
372. Liu X, Han S, Wang Z, Gelernter J, Yang BZ. Variant Callers for Next-Generation Sequencing Data: A Comparison Study. PLoS One. 2013. doi:10.1371/journal.pone.0075619
373. Fang H, Wu Y, Narzisi G, O'Rawe JA, Barrón LTJ, Rosenbaum J, et al. Reducing INDEL calling errors in whole genome and exome sequencing data. Genome Med. 2014;6. doi:10.1186/s13073-014-0089-z
374. Ghoneim DH, Myers JR, Tuttle E, Paciorkowski AR. Comparison of insertion/deletion calling algorithms on human next-generation sequencing data. BMC Res Notes. 2014;7. doi:10.1186/1756-0500-7-864
375. Kim BY, Park JH, Jo HY, Koo SK, Park MH. Optimized detection of insertions/deletions (INDELs) in whole-exome sequencing data. PLoS One. 2017;12. doi:10.1371/journal.pone.0182272

376. Sandmann S, De Graaf AO, Karimi M, Van Der Reijden BA, Hellström-Lindberg E, Jansen JH, et al. Evaluating Variant Calling Tools for Non-Matched Next-Generation Sequencing Data. Sci Rep. 2017;7. doi:10.1038/srep43169
377. Hasan MS habbi., Wu X, Zhang L. Performance evaluation of indel calling tools using real short-read data. Hum Genomics. 2015;9: 20. doi:10.1186/s40246-015-0042-2
378. Laurie S, Fernandez-Callejo M, Marco-Sola S, Trotta JR, Camps J, Chacón A, et al. From Wet-Lab to Variations: Concordance and Speed of Bioinformatics Pipelines for Whole Genome and Whole Exome Sequencing. Hum Mutat. 2016;37: 1263–1271. doi:10.1002/humu.23114
379. Li D, Kim W, Wang L, Yoon KA, Park B, Park C, et al. Comparison of INDEL calling tools with simulation data and real short-read data. IEEE/ACM Trans Comput Biol Bioinforma. 2019;16: 1635–1644. doi:10.1109/TCBB.2018.2854793
380. Hwang KB, Lee IH, Li H, Won DG, Hernandez-Ferrer C, Negron JA, et al. Comparative analysis of whole-genome sequencing pipelines to minimize false negative findings. Sci Rep. 2019;9. doi:10.1038/s41598-019-39108-2
381. Chen J, Guo J tao. Comparative assessments of indel annotations in healthy and cancer genomes with next-generation sequencing data. BMC Med Genomics. 2020;13. doi:10.1186/s12920-020-00818-6
382. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, et al. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. Nat Biotechnol. 2014;32: 246–251. doi:10.1038/nbt.2835
383. Cornish A, Guda C. A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference. Biomed Res Int. 2015;2015. doi:10.1155/2015/456479
384. Hwang S, Kim E, Lee I, Marcotte EM. Systematic comparison of variant calling pipelines using gold standard personal exome variants. Sci Rep. 2015;5. doi:10.1038/srep17875
385. Supernat A, Vidarsson OV, Steen VM, Stokowy T. Comparison of three variant callers for human whole genome sequencing. Sci Rep. 2018. doi:10.1038/s41598-018-36177-7
386. Chen J, Li X, Zhong H, Meng Y, Du H. Systematic comparison of germline variant calling pipelines cross multiple next-generation sequencers. Sci Rep. 2019;9. doi:10.1038/s41598-019-45835-3
387. Zhao S, Agafonov O, Azab A, Stokowy T, Hovig E. Accuracy and efficiency of germline variant calling pipelines for human genome data. Sci Rep. 2020;10. doi:10.1038/s41598-020-77218-4
388. Barbitoff YA, Abasov R, Tvorogova VE, Glotov AS, Predeus A V. Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery. BMC Genomics. 2022;23. doi:10.1186/s12864-022-08365-3
389. Kosugi S, Momozawa Y, Liu X, Terao C, Kubo M, Kamatani Y. Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. Genome Biol. 2019;20. doi:10.1186/s13059-019-1720-5
390. Pei S, Liu T, Ren X, Li W, Chen C, Xie Z. Benchmarking variant callers in next-generation and third-generation sequencing analysis. Brief Bioinform. 2021;22. doi:10.1093/bib/bbaa148
391. Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, Getz G, et al. Integrative genomics viewer. Nature Biotechnology. 2011. pp. 24–26. doi:10.1038/nbt.1754
392. Cunningham F, Allen JE, Allen J, Alvarez-Jarreta J, Amode MR, Armean IM, et al. Ensembl 2022. Nucleic Acids Res. 2022;50: D988–D995. doi:10.1093/nar/gkab1049
393. Zhang P, Boisson B, Stenson PD, Cooper DN, Casanova JL, Abel L, et al. SeqTailor: A user-friendly webserver for the extraction of DNA or protein sequences from next-generation sequencing data. Nucleic Acids Res. 2019;47: W623–W631. doi:10.1093/nar/gkz326
394. San lucas FA, Wang G, Scheet P, Peng B. Integrated annotation and analysis of genetic variants from next-generation sequencing studies with variant tools. Bioinformatics. 2012;28: 421–422. doi:10.1093/bioinformatics/btr667
395. Benson G. Tandem Repeats Finder. Nucleic Acids Res. 1999;27: 573–580.
396. Smith A, Hubley R, Green P. RepeatMasker Open-4.0. RepeatMasker Open-40. 2013.

397. Kofler R, Schlötterer C, Lelley T. SciRoKo: A new tool for whole genome microsatellite search and investigation. Bioinformatics. 2007. pp. 1683–1685. doi:10.1093/bioinformatics/btm157
398. Du L, Zhang C, Liu Q, Zhang X, Yue B. Krait: An ultrafast tool for genome-wide survey of microsatellites and primer design. Bioinformatics. 2018;34: 681–683. doi:10.1093/bioinformatics/btx665
399. Avvaru AK, Sowpati DT, Mishra RK. PERF: An exhaustive algorithm for ultra-fast and efficient identification of microsatellites from large DNA sequences. Bioinformatics. 2018;34: 943–948. doi:10.1093/bioinformatics/btx721
400. Delucchi M, Näf P, Bliven S, Anisimova M. TRAL 2.0: Tandem Repeat Detection With Circular Profile Hidden Markov Models and Evolutionary Aligner. Front Bioinforma. 2021;1. doi:10.3389/fbinf.2021.691865
401. Das G, Ghosh I. Benchmarking tools for DNA repeat identification in diverse genomes. bioRxiv. 2021; 1–23. Available: https://www.biorxiv.org/content/10.1101/2021.09.10.459798v1%0Ahttps://www.biorxiv.org/content/10.1101/2021.09.10.459798v1.abstract
402. Jeffrey Niu, Danielle Denisko MMH. The Browser Extensible Data (BED) format. 2022. Available: https://samtools.github.io/hts-specs/BEDv1.pdf
403. Pang KC, Frith MC, Mattick JS. Rapid evolution of noncoding RNAs: Lack of conservation does not mean lack of function. Trends in Genetics. 2006. pp. 1–5. doi:10.1016/j.tig.2005.10.003
404. Ingolia NT. Ribosome profiling: New views of translation, from single codons to genome scale. Nature Reviews Genetics. 2014. pp. 205–213. doi:10.1038/nrg3645
405. Liu S, Zhao X, Zhang G, Li W, Liu F, Liu S, et al. Predlnc-gfstack: A global sequence feature based on a stacked ensemble learning method for predicting lncrnas from transcripts. Genes (Basel). 2019;10. doi:10.3390/genes10090672
406. Tong X, Liu S. CPPred: Coding potential prediction based on the global description of RNA sequence. Nucleic Acids Res. 2019;47. doi:10.1093/nar/gkz087
407. Yang S, Wang Y, Zhang S, Hu X, Ma Q, Tian Y. NCResNet: Noncoding Ribonucleic Acid Prediction Based on a Deep Resident Network of Ribonucleic Acid Sequences. Front Genet. 2020;11. doi:10.3389/fgene.2020.00090
408. Sun L, Luo H, Bu D, Zhao G, Yu K, Zhang C, et al. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. Nucleic Acids Res. 2013;41. doi:10.1093/nar/gkt646
409. Kang YJ, Yang DC, Kong L, Hou M, Meng YQ, Wei L, et al. CPC2: A fast and accurate coding potential calculator based on sequence intrinsic features. Nucleic Acids Res. 2017;45: W12–W16. doi:10.1093/nar/gkx428
410. Fickett JW, Tung C shung. Assessment of protein coding measures. Nucleic Acids Res. 1992;20: 6441–6450. doi:10.1093/nar/20.24.6441
411. Wang L, Park HJ, Dasari S, Wang S, Kocher JP, Li W. CPAT: Coding-potential assessment tool using an alignment-free logistic regression model. Nucleic Acids Res. 2013;41. doi:10.1093/nar/gkt006
412. Fan XN, Zhang SW, Zhang SY, Ni JJ. Lncrna_mdeep: An alignment-free predictor for distinguishing long non-coding rnas from protein-coding transcripts by multimodal deep learning. Int J Mol Sci. 2020;21: 1–11. doi:10.3390/ijms21155222
413. Zheng H, Talukder A, Li X, Hu H. A systematic evaluation of the computational tools for lncRNA identification. Brief Bioinform. 2021;22. doi:10.1093/bib/bbab285
414. D'Lima NG, Ma J, Winkler L, Chu Q, Loh KH, Corpuz EO, et al. A human microprotein that interacts with the mRNA decapping complex. Nat Chem Biol. 2017;13: 174–180. doi:10.1038/nchembio.2249
415. Huang JZ, Chen M, Chen D, Gao XC, Zhu S, Huang H, et al. A Peptide Encoded by a Putative lncRNA HOXB-AS3 Suppresses Colon Cancer Growth. Mol Cell. 2017;68: 171–184.e6. doi:10.1016/j.molcel.2017.09.015

416. Liu Y, Guo J, Hu G, Zhu H. Gene prediction in metagenomic fragments based on the SVM algorithm. BMC Bioinformatics. 2013;14. doi:10.1186/1471-2105-14-S5-S12
417. Yang C, Yang L, Zhou M, Xie H, Zhang C, Wang MD, et al. LncADeep: An ab initio lncRNA identification and functional annotation tool based on deep learning. Bioinformatics. 2018;34: 3825–3834. doi:10.1093/bioinformatics/bty428
418. Sun L, Liu H, Zhang L, Meng J. lncRScan-SVM: A tool for predicting long non-coding RNAs using support vector machine. PLoS One. 2015;10. doi:10.1371/journal.pone.0139654
419. Baek J, Lee B, Kwon S, Yoon S. LncRNAnet: Long non-coding RNA identification using deep learning. Bioinformatics. 2018;34: 3889–3897. doi:10.1093/bioinformatics/bty418
420. Han S, Liang Y, Ma Q, Xu Y, Zhang Y, Du W, et al. LncFinder: An integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property. Brief Bioinform. 2019;20: 2009–2027. doi:10.1093/bib/bby065
421. Chen XG, Liu S, Zhang W. Predicting Coding Potential of RNA Sequences by Solving Local Data Imbalance. IEEE/ACM Trans Comput Biol Bioinforma. 2022;19: 1075–1083. doi:10.1109/TCBB.2020.3021800
422. Hu L, Xu Z, Hu B, Lu ZJ. COME: A robust coding potential calculation tool for lncRNA identification and characterization based on multiple features. Nucleic Acids Res. 2017;45. doi:10.1093/nar/gkw798
423. Wucher V, Legeai F, Hédan B, Rizk G, Lagoutte L, Leeb T, et al. FEELnc: A tool for long non-coding RNA annotation and its application to the dog transcriptome. Nucleic Acids Res. 2017;45. doi:10.1093/nar/gkw1306
424. Platon L, Zehraoui F, Bendahmane A, Tahi F. IRSOM, a reliable identifier of ncRNAs based on supervised self-organizing maps with rejection. Bioinformatics. 2018. pp. i620–i628. doi:10.1093/bioinformatics/bty572
425. Sun K, Chen X, Jiang P, Song X, Wang H, Sun H. iSeeRNA: Identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data. BMC Genomics. 2013;14. doi:10.1186/1471-2164-14-S2-S7
426. Han S, Liang Y, Li Y, Du W. Lncident: A Tool for Rapid Identification of Long Noncoding RNAs Utilizing Sequence Intrinsic Composition and Open Reading Frame Information. Int J Genomics. 2016;2016. doi:10.1155/2016/9185496
427. Zhao J, Song X, Wang K. LncScore: Alignment-free identification of long noncoding RNA from assembled novel transcripts. Sci Rep. 2016;6. doi:10.1038/srep34838
428. Schneider HW, Raiol T, Brigido MM, Walter MEMT, Stadler PF. A Support Vector Machine based method to distinguish long non-coding RNAs from protein coding transcripts. BMC Genomics. 2017;18. doi:10.1186/s12864-017-4178-4
429. Hill ST, Kuintzle R, Teegarden A, Merrill E, Danaee P, Hendrix DA. A deep recurrent neural network discovers complex biological rules to decipher RNA protein-coding potential. Nucleic Acids Res. 2018;46: 8105–8113. doi:10.1093/nar/gky567
430. Li A, Zhang J, Zhou Z. PLEK: A tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. BMC Bioinformatics. 2014;15. doi:10.1186/1471-2105-15-311
431. Camargo AP, Sourkov V, Pereira GAG, Carazzolle MF. RNAsamba: neural network-based assessment of the protein-coding potential of RNA sequences. NAR Genomics Bioinforma. 2020;2. doi:10.1093/nargab/lqz024
432. Cox DR. The Regression Analysis of Binary Sequences. J R Stat Soc Ser B. 1959;21: 238–238. doi:10.1111/j.2517-6161.1959.tb00334.x
433. Cortes C, Vapnik V. Support-Vector Networks. Mach Learn. 1995;20: 273–297. doi:10.1023/A:1022627411411
434. Ho TK. Random decision forests. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. 1995. pp. 278–282. doi:10.1109/ICDAR.1995.598994

435. Ho TK. The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell. 1998;20: 832–844. doi:10.1109/34.709601
436. Alam T, Al-Absi HRH, Schmeier S. Deep learning in lncrnaome: Contribution, challenges, and perspectives. Non-coding RNA. 2020. pp. 1–23. doi:10.3390/ncrna6040047
437. Hinton GE. How neural networks learn from experience. Sci Am. 1992;267: 145–151. doi:10.1038/scientificamerican0992-144
438. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323: 533–536. doi:10.1038/323533a0
439. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015. pp. 436–444. doi:10.1038/nature14539
440. Hinton G. Deep belief networks --Scholarpedia. Scholarpedia. 2009;4: 5947.
441. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process. 1997;45: 2673–2681. doi:10.1109/78.650093
442. Han S, Liang Y, Li Y, Du W. Long noncoding RNA identification: Comparing machine learning based tools for long noncoding transcripts discrimination. Biomed Res Int. 2016;2016. doi:10.1155/2016/8496165
443. Zhang Y, Huang H, Zhang D, Qiu J, Yang J, Wang K, et al. A Review on Recent Computational Methods for Predicting Noncoding RNAs. BioMed Research International. 2017. doi:10.1155/2017/9139504
444. Antonov I V., Mazurov E, Borodovsky M, Medvedeva YA. Prediction of lncRNAs and their interactions with nucleic acids: Benchmarking bioinformatics tools. Brief Bioinform. 2019;20: 551–564. doi:10.1093/bib/bby032
445. Costa Negri T Da, Rossi Paschoal A, Luz Alves WA. Comparison tools for lncRNA identification: Analysis among plants and humans. 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020. 2020. doi:10.1109/CIBCB48159.2020.9277716
446. DUAN Y, ZHANG W, CHENG Y, SHI M, XIA XQ. A systematic evaluation of bioinformatics tools for identification of long noncoding RNAs. RNA. 2021;27: 80–98. doi:10.1261/rna.074724.120
447. Amin N, McGrath A, Chen Y-PP. Evaluation of deep learning in non-coding RNA classification. Nat Mach Intell. 2019;1: 246–256. doi:10.1038/s42256-019-0051-2
448. Huang W, Li L, Myers JR, Marth GT. ART: A next-generation sequencing read simulator. Bioinformatics. 2012. doi:10.1093/bioinformatics/btr708
449. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data. 2016;3. doi:10.1038/sdata.2016.25
450. Justin Wagner, Nathan D. Olson, Lindsay Harris, Ziad Khan, Jesse Farek, Medhat Mahmoud, Ana Stankovic, Vladimir Kovacevic, Byunggil Yoo, Neil Miller, Jeffrey A. Rosenfeld, Bohan Ni, Samantha Zarate, Melanie Kirsche, Sergey Aganezov, Michael C. Schatz, Giu JMZ. Benchmarking challenging small variants with linked and long reads. Cell Genomics. 2022;2. doi:https://doi.org/10.1016/j.xgen.2022.100128
451. Neil A.Hanchard AC. 1000 Genomes Project phase 4: The gift that keeps on giving. Cell. 2022;158. doi:https://doi.org/10.1016/j.cell.2022.08.001
452. Byrska-Bishop M, Evani US, Zhao X, Basile AO, Abel HJ, Regier AA, et al. High-coverage whole-genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios. Cell. 2022;185. doi:https://doi.org/10.1016/j.cell.2022.08.004
453. Zheng-Bradley X, Streeter I, Fairley S, Richardson D, Clarke L, Flicek P. Alignment of 1000 Genomes Project reads to reference assembly GRCh38. GigaScience. 2017. doi:10.1093/gigascience/gix038
454. Lowy-Gallego E, Fairley S, Zheng-Bradley X, Ruffier M, Clarke L, Flicek P. Variant calling on the GRCh38 assembly with the data from phase three of the 1000 Genomes Project. Wellcome Open Res. 2019;4: 50. doi:10.12688/wellcomeopenres.15126.2

455. Landrum MJ, Lee JM, Benson M, Brown G, Chao C, Chitipiralla S, et al. ClinVar: Public archive of interpretations of clinically relevant variants. Nucleic Acids Res. 2016;44: D862–D868. doi:10.1093/nar/gkv1222
456. Volders PJ, Anckaert J, Verheggen K, Nuytens J, Martens L, Mestdagh P, et al. Lncipedia 5: Towards a reference set of human long non-coding rnas. Nucleic Acids Res. 2019;47: D135–D139. doi:10.1093/nar/gky1031
457. Ghoneim DH, Myers JR, Tuttle E, Paciorkowski AR. Comparison of insertion/deletion calling algorithms on human next-generation sequencing data. BMC Res Notes. 2014;7. doi:10.1186/1756-0500-7-864
458. Lim KG, Kwoh CK, Hsu LY, Wirawan A. Review of tandem repeat search tools: A systematic approach to evaluating algorithmic performance. Briefings in Bioinformatics. 2013. pp. 67–81. doi:10.1093/bib/bbs023
459. Dashnow H, Lek M, Phipson B, Halman A, Sadedin S, Lonsdale A, et al. STRetch: Detecting and discovering pathogenic short tandem repeat expansions. Genome Biol. 2018;19. doi:10.1186/s13059-018-1505-2
460. Sawaya S, Bagshaw A, Buschiazzo E, Kumar P, Chowdhury S, Black MA, et al. Microsatellite Tandem Repeats Are Abundant in Human Promoters and Are Associated with Regulatory Elements. PLoS One. 2013;8. doi:10.1371/journal.pone.0054710
461. Gardner PP, Paterson JM, McGimpsey S, Ashari-Ghomi F, Umu SU, Pawlik A, et al. Sustained software development, not number of citations or journal choice, is indicative of accurate bioinformatic software. Genome Biol. 2022;23. doi:10.1186/s13059-022-02625-x
462. Hébrant A, Froyen G, Maes B, Salgado R, Le Mercier M, D'haene N, et al. The Belgian next generation sequencing guidelines for haematological and solid tumours. Belg J Med Oncol. 2017;11: 56–67.
463. Matthijs G, Souche E, Alders M, Corveleyn A, Eck S, Feenstra I, et al. Guidelines for diagnostic next-generation sequencing. European Journal of Human Genetics. 2016. pp. 2–5. doi:10.1038/ejhg.2015.226
464. Zook JM, Hansen NF, Olson ND, Chapman L, Mullikin JC, Xiao C, et al. A robust benchmark for detection of germline large deletions and insertions. Nat Biotechnol. 2020;38: 1347–1355. doi:10.1038/s41587-020-0538-8
465. Willems T, Zielinski D, Yuan J, Gordon A, Gymrek M, Erlich Y. Genome-wide profiling of heritable and de novo STR variations. Nat Methods. 2017;14: 590–592. doi:10.1038/nmeth.4267
466. Zook JM, McDaniel J, Olson ND, Wagner J, Parikh H, Heaton H, et al. An open resource for accurately benchmarking small variant and reference calls. Nat Biotechnol. 2019;37: 561–566. doi:10.1038/s41587-019-0074-6
467. Xing J, Liu H, Jiang W, Wang L. LncRNA-Encoded Peptide: Functions and Predicting Methods. Frontiers in Oncology. 2021. doi:10.3389/fonc.2020.622294