Kmer2SNP: reference-free SNP calling from raw reads based on matching

Yanbo Li
Research School of Computer Science
Australian National University
Canberra, ACT, Australia

Hardip Patel
John Curtin School of Medical Research
Australian National University
Canberra, ACT, Australia

Yu Lin*
Research School of Computer Science
Australian National University
Canberra, ACT, Australia

***Abstract*—SNP calling is a fundamental problem of genetic analysis and has many applications, such as gene-disease diagnosis, drug design, and ancestry inference. Prior approaches either require high-quality reference genome, or suffer from low recall/precision or high runtime. We develop a reference-free algorithm Kmer2SNP to call SNP directly from raw reads, an approach that models SNP calling into a maximum weight matching problem. We benchmark Kmer2SNP against reference-free methods including hybrid (assembly-based) and assembly-free methods on both simulated and real datasets. Experimental results show that Kmer2SNP achieves better SNP calling quality while being an order of magnitude faster than the state-of-the-art methods. Kmer2SNP shows the potential of calling SNPs only using k-mers from raw reads without assembly. The source code is freely available at https://github.com/yanboANU/Kmer2SNP.**

***Index Terms*—SNP calling, Reference-free, K-mer analysis, Maximum-weight matching.**

I. Introduction

Whole-genome sequencing provides reads originated from two homologous sets of chromosomes (*i.e.*, haplotypes) from a single individual, thus makes it amenable to call heterozygous SNPs[1]. Thanks to its high throughput, moderate cost and low error rate, next-generation sequencing (NGS) is gaining popularity in SNP calling with many applications in population genetics and biomedical research, such as gene-disease diagnosis, drug design and ancestry inference [4].

Most existing approaches for SNP calling rely on the alignment between raw reads and a reference genome, such as SOAPsnp [5], GATK [6] and SAMtools [7]. When a high-quality reference genome is available, these reference-based approaches achieve the best SNP calling result. However, many species may not have high-quality reference genomes for read mapping.

There is a strong need to develop reference-free algorithms for SNP calling. Without the reference genome, such algorithms are limited to identify heterozygous SNPs (two different alleles on two haplotypes) rather than homozygous SNPs (the same allele on two haplotypes but differ from the reference). Moreover, the inferred SNPs cannot be assigned to a known genomic position as the reference genome is unknown. In hybrid approaches, raw reads are assembled into long contigs or scaffolds and SNPs can be identified by aligning raw reads to assembled contigs and assigned to positions on these contigs (instead of a reference genome). Such hybrid approaches suffer not only from misalignment and errors of raw reads but also from incompleteness and errors in assemblies [3]. Existing hybrid approaches are limited to call SNPs in organisms with a small genome size, and result in lower recall and precision rates comparing to the reference-based approaches [8]. Another type of reference-free algorithms call SNPs without explicitly assembling raw reads into contigs [8]. For example, de Bruijn graphs are built from raw reads and then SNPs are detected and represented as specific patterns (*e.g.*, bubble) in de Bruijn graphs [1], [9], [10]. However, detecting patterns becomes challenging when de Bruijn graphs get complicated due to various repeats in genomes and errors in raw reads. Moreover, EBWT2SNP [2] does not build a de Bruijn graph and uses extended Burrows-Wheeler Transform(eBWT) from reads to call SNPs as pairs of k-mers. However, the eBWT positional clustering may become ambiguous due to sequencing errors in raw reads or highly repetitive genomic regions, resulting in unstable SNP calling performance. All the above reference-free algorithms are not very scalable because it is time-consuming to build either de Bruijn graphs or eBWT indices from all raw reads.

In this paper, we propose a reference-free approach, Kmer2SNP for SNP calling directly from raw reads. Our work has three contributions: (1) we show that the k-mer frequency distribution provides the power to detect heterozygous k-mers covering SNPs in an unknown reference genome; (2) we propose a graph model to model heterozygous k-mers and reduce the SNP calling problem to finding a maximum weight matching in the heterozygous k-mer graph; (3) experimental results show that Kmer2SNP achieves better quality while being an order of magnitude faster, compared to the state-of-the-art approaches for SNP calling without references.

II. Methods

In this section, we define some general terminology. Kmer2SNP is based on the k-mer analysis from raw reads.

*Correspondence should be addressed to Y. L. (yu.lin@anu.edu.au).

[1]In this paper and other literature [1]–[3], when considering two haplotypes in an individual genome, SNP refers to single nucleotide variation (SNV) between two haplotypes without considering the occurrence percentage within a population.

A k-mer is a substring of length k from either the reference genome or raw reads. Let $\Sigma = \{A, C, G, T\}$ be the alphabet of nucleotides and Σ^k represents the set of all possible k-mers. The i^{th} nucleotide of a k-mer x is denoted as $x[i]$, i in $[1, ..., k]$. A segment of k-mer can be represented as $x[i, j] = x[i]x[i+1]\dots x[j]$, $1 \leq i \leq j \leq k$ and i, j are integers. A k-mer from raw reads is called *erroneous* if it does not appear in the reference genome. A k-mer from raw reads is called *genomic* if it appears in the reference genome. Genomic k-mers can be divided into two categories, *heterozygous* k-mers and *homozygous* k-mers. Heterozygous k-mers appear in only one of the two haplotypes while homozygous k-mers appear in both haplotypes. See Fig. 1 (A) for an example.

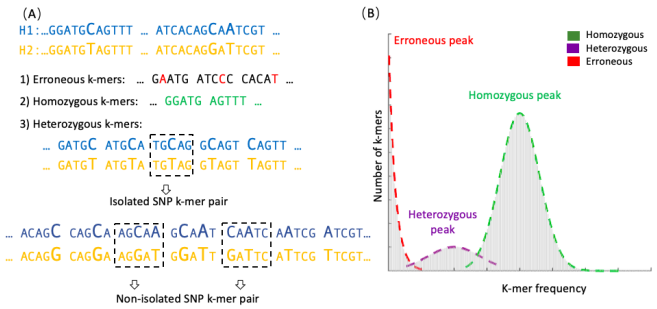

Fig. 1. Erroneous, homozygous and heterozygous k-mers as well as isolated and non-isolated SNP k-mer pairs. H_1 and H_2 are two haplotypes of one chromosome from one individual.

We define the frequency of a k-mer as the number of times this k-mer appears in all raw reads, and the frequency of k-mers (for sufficiently large k) can be used to distinguish erroneous, heterozygous and homozygous k-mers [11]. In an ideal case, erroneous k-mers have the lowest frequencies as sequencing errors are more or less random in raw reads. The frequencies of heterozygous k-mers are roughly half of the frequencies of homozygous k-mers as the former ones only appear in one of the two haplotypes. Fig. 1 (B) shows the k-mer frequency histogram where three peaks (from left to right), corresponding to erroneous, heterozygous and homozygous k-mers, respectively.

In the following, we focus on heterozygous k-mers in our work. A SNP can be represented by a pair of k-mers (from two haplotypes) that differ at the middle position. A SNP is called *isolated* if the above pair of k-mers differ only at the middle position and such a k-mer pair is called *isolated SNP* k-mer pair. A SNP is called *non-isolated* if the above pair of k-mers differ at least two positions (including the middle position) and such a k-mer pair is called *non-isolated SNP* k-mer pair. Fig. 1(A) shows examples of a isolated SNP and a non-isolated SNP along with their corresponding k-mer pairs.

A. Pipeline

We introduce a graph model of heterozygous k-mers for SNP calling. As shown in Fig. 2, Kmer2SNP takes raw reads as the input and starts from building a *heterozygous k-mer graph*, where vertices correspond to heterozygous k-mers and edges between a pair of heterozygous k-mers correspond to potential SNPs. Kmer2SNP further computes the weight for each edge using overlapping information between heterozygous k-mers. Kmer2SNP finally finds a maximum weight matching in the above graph and outputs the corresponding SNPs. We will explain each step in the following sections.

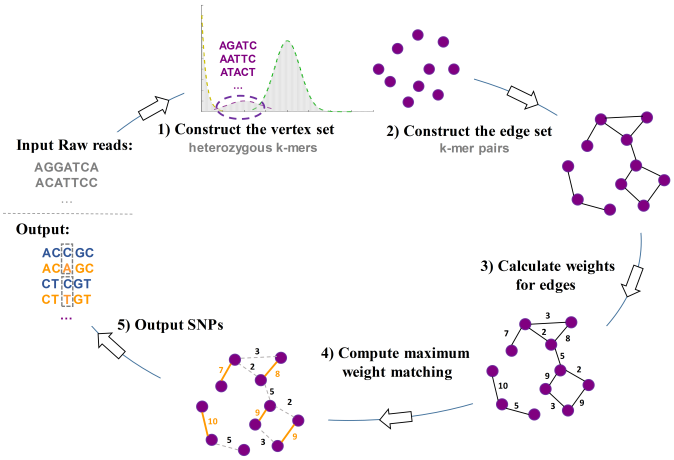

Fig. 2. The pipeline of Kmer2SNP.

B. Construct the vertex set – heterozygous k-mers

In the above heterozygous k-mer graph, each vertex is a heterozygous k-mer. Constructing the vertex set is to identify heterozygous k-mers. Kmer2SNP uses DSK [12] to count k-mer frequencies from raw reads and derive a corresponding k-mer histogram file. Then FindGSE [11] is used to find the frequency range of heterozygous k-mers. Using DSK and FindGSE, Kmer2SNP effectively retains most heterozygous k-mers and filters most erroneous k-mers and homozygous k-mers[2]. Kmer2SNP further uses k-mer pairing information to filter homozygous k-mers and erroneous k-mer in the next step.

C. Construct the edge set – k-mers pairs

After constructing the vertex set, Kmer2SNP constructs edges between k-mer pairs that correspond to SNPs. As defined in Section II, there are two types of k-mer pairs: isolated SNP k-mer pairs and non-isolated SNP k-mer pairs. Kmer2SNP thus introduces two types of edges, h1-edges and h2-edges to represent those k-mer pairs, respectively.

According to the definition of isolated SNP k-mer pairs, there is an $h1$-edge between two k-mers x and x' if and only if $x[\frac{(k+1)}{2}] \neq x'[\frac{(k+1)}{2}]$ and $h(x, x') = 1$, where k is odd and $h()$ computes the hamming distance. For each k-mer, Kmer2SNP removes its middle position and uses the remaining (k-1) positions as a key to compute its index in a hash table. Clearly, Kmer2SNP connects x and x' by an $h1$-edge if and only if they have the same index in the above hash table.

[2]Note that k-mers from paralogous sequence variants (PSVs, differences between duplicated regions in a genome) belong to homozygous k-mers and thus are filtered.

Similarly, there is an $h2$-edge between two k-mers x and x' if and only if $x[\frac{(k+1)}{2}] \neq x'[\frac{(k+1)}{2}]$ and $h(x, x') = 2$, where k is odd and h() computes the hamming distance. Clearly, if x and x' are connected by an h2-edge, either $x[1, \frac{(k-1)}{2}] = x'[1, \frac{(k-1)}{2}]$ or $x[\frac{(k+3)}{2}, k] = x'[\frac{(k+3)}{2}, k]$ holds. For each k-mer x, Kmer2SNP uses the prefix $x[1, \frac{(k-1)}{2}]$ and suffix $x[\frac{(k+3)}{2}, k]$ respectively as keys to compute its index in hash table. Two k-mers x and x' are connected by an h2-edge only if either their prefixes or suffixes have the same index. Note that Kmer2SNP needs to verify the hamming distances between k-mers with the same index to add h2-edges.

Note that most homozygous k-mers at low coverage regions have been successfully filtered because they are unlikely to form any k-mer pairs. Although k-mers in connected components of size 2 (i.e., k-mer pairs) naturally correspond to potential SNPs, there are still a significant number of vertices locating in connected components with at least 3 vertices. If a k-mer is attached to multiple edges, how could we assign this k-mer to an edge that most likely corresponds to a real SNP? In the next step, we show how to compute a weight for each edge that indicates the likelihood of the corresponding SNP being true.

D. Calculate weights for edges

In the heterozygous k-mer graph, each edge now corresponds to a potential SNP and the weight of an edge should indicate how likely this potential SNP is true. As shown Fig. 3 (a), an edge is constructed between an isolated SNP k-mer pair where these two k-mers only differ at the middle position. Note that there are other heterozygous k-mers which also cover this isolated SNP (not at the middle position) and may provide support to this potential SNP. Therefore, Kmer2SNP calculates a weight for each edge based on the presence or absence overlapping heterozygous k-mers that also support this SNP.

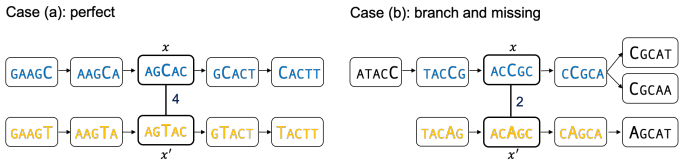

Fig. 3. Compute weights for edges. Case (a) is a perfect case and the weight of edge (x, x') is 4. Case (b), the left-side extension terminates when no left-overlapping pair of heterozygous k-mers is found and the right-side extension terminates when two pairs of right-overlapping heterozygous k-mers are found. Therefore, the weight of edge (x, x') is 2.

A pair of k-mers (y, y') is called *left-overlapping (right-overlapping)* with another pairs of k-mers (x, x') if $x[1, k-1] = y[2, k]$, $x'[1, k-1] = y'[2, k]$ and $y[1] = y'[1]$ ($x[2, k] = y[1, k-1]$, $x'[2, k] = y'[1, k-1]$ and $y[k] = y'[k]$). Kmer2SNP iteratively recruits left-overlapping and right-overlapping pairs of heterozygous k-mers to extend isolated SNP k-mer pairs on both sides. The extension on the left (right) side terminates if Kmer2SNP fails to find a unique pair of left-overlapping (right-overlapping) heterozygous k-mers which still covers the isolated SNP. The *left (or right) extendable length* of an isolated SNP k-mer pair (x, x'), $l(x, x')$ (or $r(x, x')$), is defined as the number of left-overlapping (right-overlapping) pairs of heterozygous k-mers recruited in the extension. For example, $l(x, x') = r(x, x') = 1$ in Fig. 3 (b). Finally, the weight of an edge between an isolated SNP k-mer pair (x, x') is defined as $l(x, x') + r(x, x')$ as a conservative estimate.

Similarly, we also introduce weights for edges between the corresponding non-isolated SNP k-mer pairs. We expect the True-Positive (TP) edges have higher weights than False-Positive (FP) edges, e.g., the mean weight of TP edges is 28.92 while the mean weight of FP edges is 5.31 in a simulated dataset from NA12878 Chromosome 22 (30x for each haplotype). In the next step, we thus use the maximum weight matching to select more confident SNPs.

E. Compute maximum weight matching and output SNPs

According to the previous section, the more weight that Kmer2SNP assigns to an edge, the more likely the corresponding SNP is true. Therefore, Kmer2SNP computes the *maximum weight matching* in the above heterozygous k-mer graph, where the *maximum weight matching* is a set of pairwise non-adjacent edges in which the sum of weights is maximized. Kmer2SNP identifies all the connected components and uses python package NetworkX [13] to compute the *maximum weight matching* for each component with at least three vertices. This can be done efficiently thanks to the relatively small size of connected components in the graph.

The edges in the maximum weight matching are then converted back to SNPs as pairs of k-mer. Kmer2SNP further filters edges if its weight is lower than a weight threshold (4 in default setting). As each non-isolated SNP corresponds to two pairs of non-isolated SNP k-mers (refer to Fig. 1) and this non-isolated SNP is included in the output only if both edges are selected in the final maximum weight matching.

III. Experimental Design and Results

We use the following datasets for our evaluations.

1) **HG-C$_N$**. Simulated datasets are generated from Chromosome N using trio-phased haplotypes of individual NA12878 [14]. Illumina Hiseq reads with different coverages on different chromosomes are simulated by ART [15] Version 2.5.8 (June 7, 2016).
2) **NA12878** and **NA24385**. NA12878 and NA24385 contain real 300X Illumina HiSeq reads aligned to different chromosomes of two individuals (NA12878 and NA24385) and are downloaded from NCBI [16], [17].
3) **Fungal**. The phased *Candida albicans SC5314* reference genome (version A22) is downloaded from Candida Genome Data [18]. The size of this fungal genome is 14.3 Mb and its heterozygous rate is 0.5% [19]. 100X Illumina Hiseq reads (50X for each haplotype) are simulated by ART [15].
4) **PucStrE137**. The Dikaryotic Wheat Stripe Rust Fungus *Puccinia striiformis f. sp. tritici (Strain: 104 E137 A-)* dataset consists of 108X HiSeq 2000 Illumina reads downloaded from NCBI (accession number

SRX3181917). The size of this genome is 83 Mb after manual curation and its heterozygous rate is estimated to be 1.2% [20].

A. Baseline Methods

We compare Kmer2SNP against two state-of-the-art reference-free SNP calling tools, DiscoSNP++ [1], [3] and EBWT2SNP [2], as they show the best results in recent benchmarking [1], [2]. As for hybrid approaches, we use SPAdes [21] and SGA [22] to assemble Illumina reads into contigs. We further use BWA [23] to align reads to the assembled contigs and apply two popular SNP calling pipelines, SAMtools (bcftools) [24] and GATK [25], to call SNPs. These *hybrid* approaches are called SPAdes+GATK, SGA+SAMtools, etc.

Note reference-free approaches cannot assign SNPs to known genomic positions. Following the evaluation metrices in [2], each ground-truth SNP is represented as a pair of heterozygous k-mers. For NA12878 and NA24385, we download the reference genomes along with SNP annotations from [14], [26], [27] to generate heterozygous k-mer pairs as ground-truth. For Fungal, we download two homologous sets of chromosomes from [18], and then use BLASR [28] to align them to obtain heterozygous k-mer pairs. For PucStrE137, heterozygous k-mer pairs are derived by two phased haplotypes derived from 100X PacBio reads assembled by FALCON-Unzip [20]. As different reference-free approaches for SNP calling may have different output formats, we convert all output SNPs into heterozygous k-mer pairs to make a fair comparison. Three metrics, recall, precision and F1-score, are used to evaluate the performance of SNP calling.

B. Results on HG-C_N datasets

We perform extensive experiments by varying the k-mer sizes and the sequencing coverages on different chromosomes. Among the many combinations of parameters tested, we show some representative results on HG-C_{22}. HG-C_{22} consists of simulated Illumina Hiseq reads with different coverages of Chromosome 22.

The choice of k-mer sizes The k-mer size is an important parameter for Kmer2SNP, DiscoSNP++ and EBWT2SNP. DiscoSNP++ shows that the k-mer size has a limited impact on the SNP calling quality [1], [3] and performs experiments by setting k=31. EBWT2SNP [2] chooses a sufficiently large k-mer (k=31 in all the experiments) such that a k-mer is expected to appear at most once in the genome. Kmer2SNP also achieves stable performance across different k-mer sizes on HG-C_{16} and HG-C_{22} dataset. Therefore, in the following experiments, the k-mer size is chosen to be 31 by default.

The effect of different coverages Fig. 4 shows that Kmer2SNP outperforms DiscoSNP++ and EBWT2SNP on recall and precision while being an order of magnitude faster for SNP calling on HG-C_{22} dataset. Moreover, with the increase of read coverages, the running time of DiscoSNP++ and EBWT2SNP increases significantly while Kmer2SNP still maintains low running time. Also, the memory usage of Kmer2SNP is also lower or comparable to DiscoSNP++ and EBWT2SNP.

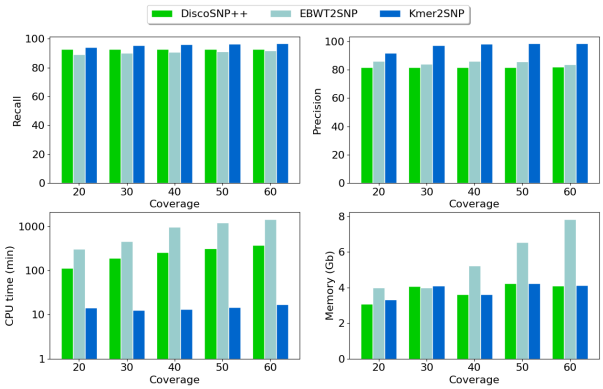

Fig. 4. SNP calling on different read coverage of HG-C_{22}.

C. Results on NA12878 and NA24385 datasets

The SNP calling results on NA12878 (Chromosome 22) and NA24385 (Chromosome 22) are shown in Table I and Table II, respectively. Both tables demonstrate that Kmer2SNP outperforms existing reference-free SNP calling tools in both quality and scalability. Consistent with previous results [3], although hybrid approaches achieve high recall rates, they suffer seriously from errors in assemblies of complex genomes, thus call many false positive SNPs and result in low precision rates.

TABLE I
PERFORMANCE OF SNP CALLING ON NA12878 (CHROMOSOME 22) [16].

Tools	Recall	Precision	F1-score	CPU time
SPAdes+GATK	**0.94**	0.43	0.59	80.53h
SPAdes+Samtools	0.94	0.47	0.63	64.98h
SGA+GATK	0.78	0.44	0.56	57.31h
SGA+Samtools	0.77	0.44	0.56	41.26 h
DiscoSNP++	0.77	0.76	0.77	8.88h
EBWT2SNP	0.85	0.56	0.68	47.16h
Kmer2SNP	0.89	**0.80**	**0.84**	**1.35h**

TABLE II
PERFORMANCE OF SNP CALLING OF NA24385 (CHROMOSOME 22) [17]

Tools	Recall	Precision	F1-score	CPU time
SPAdes+GATK	**0.95**	0.30	0.46	84.60h
SPAdes+SAMtools	0.95	0.36	0.52	67.27h
SGA+GATK	0.76	0.37	0.50	67.66h
SGA+SAMtools	0.76	0.37	0.50	48.50h
DiscoSNP++	0.65	0.62	0.64	9.40h
EBWT2SNP	0.83	0.50	0.62	41.75h
Kmer2SNP	0.80	**0.73**	**0.76**	**1.19h**

D. Results on Fungal dataset

Table III summarizes the SNP calling results of different tools for SNP calling on Fungal dataset. Similar to the above experiments results, Kmer2SNP outperforms existing reference-free approaches including DiscoSNP++,

EBWT2SNP and hybrid approaches. Note the precision of hybrid approaches improves as this genome does not contain complex repeat structures as in the human genome, however, the overall performance of hybrid approaches is still not as good as Kmer2SNP.

TABLE III
PERFORMANCE OF SNP CALLING ON THE FUNGAL DATASET

Tools	Recall	Precision	F1-score	CPU time
SPAdes+GATK	0.77	0.92	0.84	15.32h
SPAdes+SAMtools	0.77	0.92	0.84	10.09h
SGA+GATK	0.63	0.94	0.75	12.15h
SGA+SAMtools	0.63	0.93	0.75	6.92h
DiscoSNP++	0.76	0.97	0.85	2.41h
EBWT2SNP	0.69	**0.98**	0.81	4.91h
Kmer2SNP	**0.82**	0.97	**0.89**	**0.19h**

E. Results on PucStrE137 dataset

Table IV shows SNP calling performance on PucStrE137 dataset with 1.2% heterozygous rate [20]. Again, Kmer2SNP achieves the best F1-score comparing to other approaches. Note that all reference-free approaches have relatively low recall and precision rates due to the limitations in short reads, which indicates the importance of using long reads to assemble and phase haplotypes in genomes with relatively high heterozygous rate.

TABLE IV
PERFORMANCE OF SNP CALLING ON PUCSTRE137 DATASET[a]

Tools	Recall	Precision	F1-score	CPU time
SPAdes+GATK	0.40	0.50	0.44	155.45h
SPAdes+SAMtool	0.38	0.51	0.44	127.19h
DiscoSNP++	0.31	**0.67**	0.42	8.59h
EBWT2SNP	0.17	0.52	0.26	22.39h
Kmer2SNP	**0.47**	0.55	**0.51**	**1.62h**

[a]SGA+GATK, SGA+SAMtools are not included in this table because SGA discards more than 90% of reads and the assembly is incomplete.

IV. CONCLUSION AND DISCUSSION

Kmer2SNP shows the potential of calling SNPs directly from raw reads when the reference genome is not available. Kmer2SNP introduces a graph model on k-mers and the SNP calling problem is to find the maximum weight matching in the graph. Kmer2SNP outperforms other reference-free approaches in SNP calling quality while being an order of magnitude faster. Currently, Kmer2SNP has several limitations. First, the current implementation of Kmer2SNP only supports at most two non-isolated SNPs in a heterozygous k-mer pair for efficiency purposes. Second, Kmer2SNP is only applicable to diploid genomes of an individual and how to extend it to handle polyploid genomes and multiple individuals is worth future investigation. Last but not least, Kmer2SNP may benefit from varying the k-mer sizes in building the vertex set and introducing a probabilistic model in computing the maximum weight matching.

REFERENCES

[1] P. Peterlongo *et al.*, "Discosnp++: de novo detection of small variants from raw unassembled read set (s)," *BioRxiv*, p. 209965, 2017.

[2] N. Prezza *et al.*, "Snps detection by ebwt positional clustering," *Algorithms for Molecular Biology*, vol. 14, no. 1, p. 3, 2019.

[3] R. Uricaru, G. Rizk *et al.*, "Reference-free detection of isolated snps," *Nucleic acids research*, vol. 43, no. 2, pp. e11–e11, 2014.

[4] R. L. Strausberg *et al.*, "Sequence-based cancer genomics: progress, lessons and opportunities," *Nature Reviews Genetics*, vol. 4, no. 6, p. 409, 2003.

[5] R. Li, Y. Li *et al.*, "Snp detection for massively parallel whole-genome resequencing," *Genome research*, vol. 19, no. 6, pp. 1124–1132, 2009.

[6] M. A. DePristo *et al.*, "A framework for variation discovery and genotyping using next-generation dna sequencing data," *Nature genetics*, vol. 43, no. 5, p. 491, 2011.

[7] H. Li, B. Handsaker *et al.*, "The sequence alignment/map format and samtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[8] L. Wu *et al.*, "Direct comparison of performance of single nucleotide variant calling in human genome with alignment-based and assembly-based approaches," *Scientific reports*, vol. 7, no. 1, p. 10963, 2017.

[9] Z. Iqbal *et al.*, "De novo assembly and genotyping of variants using colored de bruijn graphs," *Nature genetics*, vol. 44, no. 2, p. 226, 2012.

[10] R. M. Leggett *et al.*, "Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de bruijn graphs," *PLoS One*, vol. 8, no. 3, p. e60058, 2013.

[11] H. Sun, J. Ding *et al.*, "findgse: estimating genome size variation within human and arabidopsis using k-mer frequencies," *Bioinformatics*, vol. 34, no. 4, pp. 550–557, 2017.

[12] G. Rizk, D. Lavenier, and R. Chikhi, "Dsk: k-mer counting with very low memory usage," *Bioinformatics*, vol. 29, no. 5, pp. 652–653, 2013.

[13] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[14] J. Duitama *et al.*, "Fosmid-based whole genome haplotyping of a hapmap trio child: evaluation of single individual haplotyping techniques," *Nucleic acids research*, vol. 40, no. 5, pp. 2041–2053, 2011.

[15] W. Huang, L. Li *et al.*, "Art: a next-generation sequencing read simulator," *Bioinformatics*, vol. 28, no. 4, pp. 593–594, 2011.

[16] NA12878, "1000 Genome Project," ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NIST_NA12878_HG001_HiSeq_300x, 2016.

[17] NA24385, "1000 Genome Project," ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946, 2016.

[18] "Candida genome database," http://www.candidagenome.org.

[19] D. Muzzey, K. Schwartz *et al.*, "Assembly of a phased diploid candida albicans genome facilitates allele-specific measurements and provides a simple model for repeat and indel structure," *Genome biology*, vol. 14, no. 9, p. R97, 2013.

[20] B. Schwessinger, J. Sperschneider *et al.*, "A near-complete haplotype-phased genome of the dikaryotic wheat stripe rust fungus puccinia striiformis f. sp. tritici reveals high interhaplotype diversity," *MBio*, vol. 9, no. 1, pp. e02 275–17, 2018.

[21] A. Bankevich *et al.*, "Spades: a new genome assembly algorithm and its applications to single-cell sequencing," *Journal of computational biology*, vol. 19, no. 5, pp. 455–477, 2012.

[22] J. T. Simpson and R. Durbin, "Efficient de novo assembly of large genomes using compressed data structures," *Genome research*, vol. 22, no. 3, pp. 549–556, 2012.

[23] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[24] V. Narasimhan, P. Danecek *et al.*, "Bcftools/roh: a hidden markov model approach for detecting autozygosity from next-generation sequencing data," *Bioinformatics*, vol. 32, no. 11, pp. 1749–1751, 2016.

[25] G. A. Van der Auwera *et al.*, "From fastq data to high-confidence variant calls: the genome analysis toolkit best practices pipeline," *Current protocols in bioinformatics*, vol. 43, no. 1, pp. 11–10, 2013.

[26] J. M. Zook *et al.*, "Integrating human sequence data sets provides a resource of benchmark snp and indel genotype calls," *Nature biotechnology*, vol. 32, no. 3, p. 246, 2014.

[27] J. M. Zook, D. Catoe *et al.*, "Extensive sequencing of seven human genomes to characterize benchmark reference materials," *Scientific data*, vol. 3, p. 160025, 2016.

[28] M. J. Chaisson and G. Tesler, "Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory," *BMC bioinformatics*, vol. 13, no. 1, p. 238, 2012.