BMC Bioinformatics

RESEARCH

Open Access

GCphase: an SNP phasing method using a graph partition and error correction algorithm

Junwei Luo[1], Jiayi Wang[1], Haixia Zhai[1] and Junfeng Wang[1*]

*Correspondence:
wangjunfeng@hpu.edu.cn

[1] School of Software, Henan Polytechnic University, Jiaozuo 454003, China

Abstract

Background: The utilization of long reads for single nucleotide polymorphism (SNP) phasing has become popular, providing substantial support for research on human diseases and genetic studies in animals and plants. However, due to the complexity of the linkage relationships between SNP loci and sequencing errors in the reads, the recent methods still cannot yield satisfactory results.

Results: In this study, we present a graph-based algorithm, GCphase, which utilizes the minimum cut algorithm to perform phasing. First, based on alignment between long reads and the reference genome, GCphase filters out ambiguous SNP sites and useless read information. Second, GCphase constructs a graph in which a vertex represents alleles of an SNP locus and each edge represents the presence of read support; moreover, GCphase adopts a graph minimum-cut algorithm to phase the SNPs. Next, GCpahse uses two error correction steps to refine the phasing results obtained from the previous step, effectively reducing the error rate. Finally, GCphase obtains the phase block. GCphase was compared to three other methods, WhatsHap, HapCUT2, and LongPhase, on the Nanopore and PacBio long-read datasets. The code is available from https://github.com/baimawjy/GCphase.

Conclusions: Experimental results show that GCphase under different sequencing depths of different data has the least number of switch errors and the highest accuracy compared with other methods.

Keywords: Haplotype assembly, SNP phasing, Graph minimum-cut algorithm, Error correction

Background

For diploid organisms such as humans, all autosomes (nonsex chromosomes) have their corresponding homologous chromosomes. The variation between homologous chromosomes are minimal (approximately 99% of the base pairs being identical). These variant sites across the genome contribute to the genetic diversity among human individuals. Single nucleotide polymorphisms (SNPs) refer to the variation of a single nucleotide in the genome, and they are the most common form of genetic variation. SNPs can occur at any position in the genome and are widely present in the human genome. Haplotype, short for haploid genotype, refers to a specific set of genotypes composed of alleles from

multiple SNP loci located on the same chromosome. The process of reconstructing the two alleles of an SNP to their respective haplotype is called phasing. The acquisition of haplotypes helps in studying the correlation of human genetic variations and in assessing the risk of genetic diseases [1, 2]. By analysing the distribution, frequency and the length of haplotypes, the genetic structure and the evolutionary history of human populations can be revealed. Established initiatives such as the 1000 Genomes Project aim to unravel variations and diversity in the human genome [3], including SNPs and structural variations. Additionally, there are reference genomes generated for different populations, which can be utilized for downstream data analysis purposes [4].

Due to the limitations of current sequencing technologies, genome sequencing generally does not distinguish between haplotypes. For example, in Nanopore sequencing technology [5], all DNA is directly processed, such as through specific enzymatic cleavage, to obtain DNA molecules suitable for Nanopore sequencing. These DNA fragments are then sequenced by using a Nanopore sequencer, resulting in sequencing data that contain a mixture of DNA sequences from all homologous chromosomes.

Therefore, to obtain phased results of target haplotypes with lower financial and time costs, the method based on alignment for obtaining haplotypes has become mainstream. The accurate assignment of the two alleles of SNP loci to their corresponding haplotypes in polynomial time is currently a major challenge.

For current long read sequencing technologies, such as Nanopore and PacBio sequencing, in the generated sequencing data, the bases within the same read can be determined to originate from the same chromosome [6–9]. Therefore, the alleles of SNP loci contained within that read can be assigned to the same haplotype. Therefore, in an ideal scenario, despite the limited number of SNP loci contained in each read and the limited phase information between obtained SNP loci, it is still possible to link all the alleles together rapidly and effectively by extending the overlaps between different reads. However, in actual sequencing processes, not all bases in each read are accurate. Due to the limitations of sequencing technologies, there are still significant sequencing errors present in the reads. The sequencing error rate for short reads usually are approximately 0.1% [10]. The sequencing error rate of Nanopore Long reads and Pacbio Long reads are approximately 10–25% [11, 12]. However, the sequencing error rate of Pacbio HiFi read is approximately 1% [12]. These sequencing errors make the process of obtaining SNP phasing an NP-hard problem [13]. Therefore, addressing the impact of sequencing errors on the phasing process has become a primary challenge that must be resolved.

Current read-based SNP phasing methods can be broadly classified into three categories [14]. First is the method based on minimum error correction (MEC). MEC is a commonly used objective function optimization method that selects the optimal SNP phasing results by minimizing the number of error corrections. Representative methods include SCGDhap [15], Poly-Harsh [16], HapTree [17], GTIHR [18], and SDhaP [19]. This method is effective in handling polyploid genome phasing problems, but it heavily relies on read quality, and in genomes with low heterozygosity, the limited phase information due to a small number of SNP loci leads to a significant decrease in predictive accuracy. Second is the method based on group division. This method represents the two alleles of reads or SNP loci as vertices in a graph, where edges represent the similarity between two vertices. The graph is then partitioned into k subsets based on genome

ploidy. Representative methods, such as WhatsHap [20], Hap10 [21] and ComHapDet [22], are known for their fast speed and high accuracy. HapCUT2 [23] is a SNP phasing tool based on the maximum likelihood method. It has good adaptability to different sequencing technologies or protocols, making it a powerful tool for studying complex genomic variations and disease associations. However, due to the requirement of reads containing more SNP information, these methods are mostly applicable when using long reads for phasing. Third is the method based on clustering. These methods generally cluster reads or SNPs, which are adjacent in genomic location, together into sets using various algorithms. They calculate the consensus for each set and then perform clustering again, forming consensus based on the genome ploidy. For example, H-Pop represents reads as a matrix and partitions the matrix to form haplotypes [24]. Phasebook selects reads with the most overlaps as seed reads and assigns similar reads to the same cluster [25]. This process is repeated iteratively by using the consensus of the final partitioned set as the output. LongPhase extends phase information between adjacent SNP loci to the entire haplotype level [26]. These methods effectively utilize various clustering approaches to generate haplotype outputs. Unlike graph partitioning methods, they do not construct a graph but perform clustering by iteratively increasing the size of the sets. Despite the existence of numerous phasing methods, the accuracy of phasing results remains limited by the constraints of sequencing technologies. Short-read sequencing offers high accuracy but suffers from limited read length and coverage of SNP information. Long read sequencing provides longer reads at a lower cost but with higher error rates. PacBio HiFi sequencing combines the advantages of both short reads and long reads. However, its high cost limits its application. The primary goal in current SNP phasing methods remains achieving low-cost, high-quality, and widely applicable phasing solutions.

If the impact of a high long-read error rate can be overcome, this goal can be achieved. The group division algorithm with high fault tolerance can effectively address this issue, as the neighbouring vertices of each vertex can serve as a basis for phasing that vertex, allowing individual point errors to be accommodated by the remaining vertices. However, the group division algorithm has the drawback of local optima. Therefore, after using the group division algorithm for phasing, additional error correction steps are still needed to improve accuracy.

In this paper, we present a method called GCphase that can phase SNPs more accurately. In the experiments, GCphase achieves the lowest error rate compared to that of other methods at different sequencing depths. It can control the error rate to approximately 0.015% and can even achieve an error rate of 0 on certain chromosomes.

Methods

The algorithm flow of GCphase can be roughly divided into six steps (Fig. 1): (i) Data Preprocessing. GCphase preprocesses the input VCF and BAM files to transform them into a format suitable for the algorithm, facilitating subsequent computations. (ii) Selecting SNPs and long reads. GCphase filters the obtained SNPs and reads, removing ambiguous SNPs and reads that do not cover any SNP site. (iii) Constructing the graph. The alleles of SNP loci are represented as vertices of an undirected graph, and the reads supporting two alleles are constructed as edges. (iv) Partitioning Graph. GCphase divides

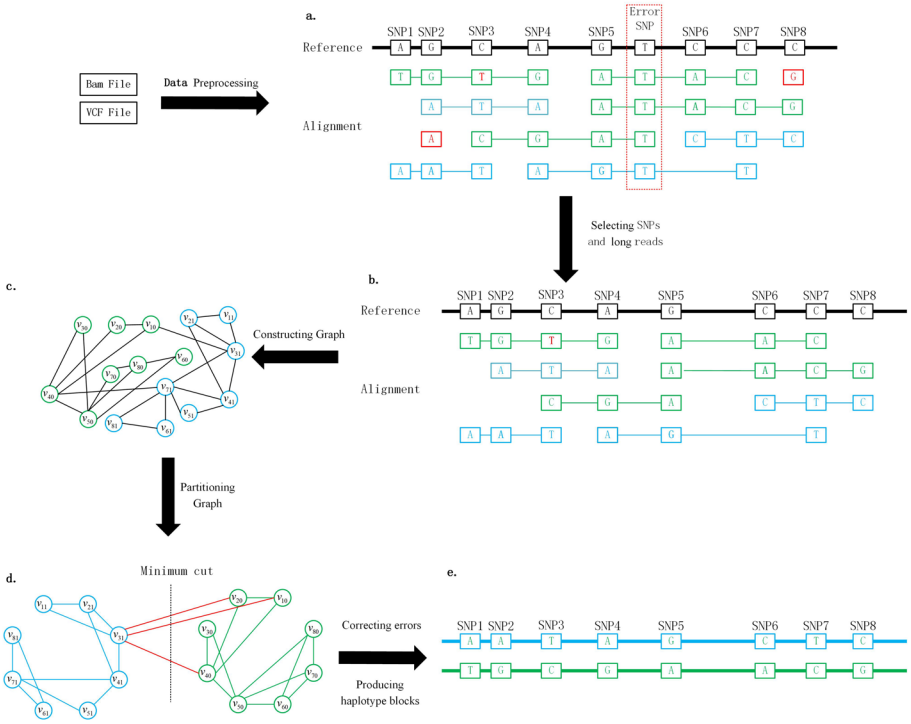

Fig. 1 Overview of the GCphase workflow. **a** Data preprocessing. GCphase simplifies the reads into a format that contains only SNP information. **b** Selecting SNPs and long reads. SNP loci with disproportionately large allele ratios (the number of reads supporting the major allele accounts for more than 85% of the total number of reads) and reads with insufficient SNP information (indicated by red borders in the graph) were removed. **c** Constructing the Graph. The two alleles of SNP loci are represented as vertices in the graph, and the reads supporting two alleles are represented as edges. **d** Partitioning Graph. The graph is partitioned into two sets with the smallest intersection using the minimum-cut algorithm. **e** Correcting errors and Producing haplotype blocks. After undergoing two error correction steps, the algorithm traverses the maximal connected components in the graph to generate haplotype blocks as the output

the graph into two sets as disjoint as possible by utilizing the minimum-cut algorithm on the graph. (v) Correcting Errors. GCphase employs two error correction steps to improve the accuracy of phasing results. (vi) Producing haplotype blocks. GCphase outputs each maximal connected component in the undirected graph as a haplotype block. Now, let us delve into the specific phasing process.

Data preprocessing

GCphase (version 1.0) utilizes minimap2 to align long reads to the reference genome. The resulting alignment in SAM format was then converted to BAM format by using samtools [27, 28]. The sorted and indexed BAM file serves as the input of GCphase and uses pysam to extract information from bam files [29]. The SNP location and its corresponding allele are provided as input in the form of a variant call format (VCF) file, which is a standard format for storing genetic variations.

The long reads with optimal alignment are retained, and others are filtered. For a given long read, GCphase first identifies all the SNP loci it contains. The *i-th* SNP in the long read is represented as a two-tuple $m_i = (v_i, k)$. vi is the SNP identity, k is a binary value, and when $k = 0$, it means that the base in the long read is the same as the major allele; when $k = 1$, it means that the base is the same as the minor allele. Finally, the long read

can be transformed into a vector (m_1, m_2... m_n). Ultimately, a single read can be simplified into a representation of SNPs along with the alleles supported by the read at those loci, convenient for subsequent processing.

Selecting SNPs and long reads

After GCphase obtains the information of SNPs and reads, not all the information can be utilized in the phasing process. For each SNP locus to be phased, if the proportion of the allele with the highest read support is greater than or equal to 85% of the total read support for both alleles, this SNP locus is removed. Let vi0 and vi1 represent the two alleles of SNP v_i. Let n_1 and n_2 denote the counts of reads supporting v_{i0} and v_{i1}, respectively. If $\max(n_1,n_2)/(n_1+n_2) \geq 0.85$, that is, if the proportion of one allele is excessively high, it is considered a fuzzy locus for SNP v_i, indicating unclear allele expression. Phasing at this locus is not feasible, and there is a higher likelihood of introducing errors. Therefore, the locus is removed. After iterating through all the SNP loci to be phased and removing all the SNP with imbalanced heterozygosity, it is also necessary to update the read information by removing these fuzzy loci from the previously obtained reads.

Since the core algorithm of phasing relies on the extension and expansion of the relative positions between SNP loci, only the relative positions between multiple (greater than or equal to 2) SNP loci can provide effective phase information for the phasing process. For each long reads, if a read ultimately provides less than two SNPs, it cannot provide effective information for phasing and is considered an invalid read. Therefore, all information from that read was deleted (Fig. 2).

Constructing the graph

Let *G(V,E)* be an undirected graph, where each allele of the SNP loci to be phased is represented as a vertex in graph *G*. For the *i-th* SNP loci p_i, we construct two vertices

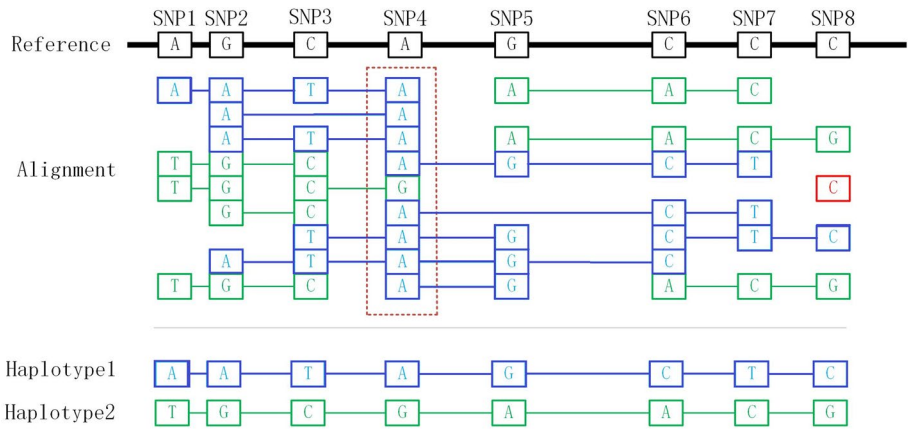

Fig. 2 Filtering criteria for SNP loci and edges. Alleles of the same colour connected by lines represent a single read, where different colours indicate different haplotypes. In the figure, at the fourth SNP (represented by red dashed lines), the major allele 'A' has a count of 9, while the minor allele 'G' has a count of 1. The proportion of the major allele reaches 90%, which exceeds the set threshold of 85%. Therefore, SNP4 is considered a fuzzy SNP, and phasing cannot be effectively performed. Thus, SNP4 is removed. The read only aligned to SNP8 and represented in red is shown in the figure. However, since they cover only one SNP locus, they cannot provide effective information for SNP phasing. Therefore, these reads were deleted

v_{i0} and v_{i1} in G, which represent the major and minor alleles of p_i, respectively. If there is an edge $e(v_{i0},v_{j1})$ between v_{i0} and v_{j1}, it indicates the reads simultaneously supporting the major allele of SNP locus p_i and the minor allele of SNP locus p_j. The weight $w(v_{i0},v_{j1})$ of edge $e(v_{i0},v_{j1})$ represents the count of reads that support both alleles simultaneously. After processing all the reads iteratively, an undirected graph can be constructed.

After constructing the undirected graph G, it is necessary to perform a filtering process on the edges of the graph. For any two SNP loci p_i and p_j, there are four vertices: v_{i0}, v_{i1}, v_{j0}, and v_{j1} in the graph. There can be up to four edges: $e(v_{i0},v_{j0})$, $e(v_{i0},v_{j1})$, $e(v_{i1},v_{j0})$, and $e(v_{i1},v_{j1})$ between them. Due to the mutually exclusive nature of the two alleles at the same SNP locus, if allele (v_{i0},v_{j0}) is present in one haplotype, the corresponding (v_{i1},v_{j1}) should be present in the other haplotype. Similarly, if allele (v_{i0},v_{j1}) is present in one haplotype, the corresponding (v_{i1},v_{j0}) should be present in the other haplotype. Therefore, the four edges can be classified into two sets: $N_1=\{e(v_{i0},v_{j0}), e(v_{i1},v_{j1})\}$ and $N_2=\{e(v_{i0},v_{j1}), e(v_{i1},v_{j0})\}$, where the weight of each set is the sum of the weights of the edges it contains. The phase relationship between the alleles of the two SNP loci is determined by the weight bias between these two sets. If the weight $w(N_1)$ is greater than the weight $w(N_2)$, the set N_1 is correct, and vice versa. If $\max(w(N_1),w(N_2))$-$\min(w(N_1),w(N_2)) \leq 1$, it indicates that the possibilities of N_1 and N_2 are converging, suggesting an ambiguous phase between SNP loci p_i and p_j. Phasing these two SNP loci cannot be effectively performed based on the edges between them, and there is a higher likelihood of introducing errors in the phasing process. In that case, all edges between the alleles about p_i and p_j are removed.

Partitioning graph

The minimum cut of a graph refers to dividing the vertices of the graph into two parts in such a way that the sum of the weights of the edges between the two parts is minimized. For the two alleles of an SNP locus, since they belong to two different haplotypes, the weight between alleles within the same haplotype is maximum, while the weight between alleles from different haplotypes is minimum. Therefore, applying the minimum-cut algorithm on the graph involves partitioning the vertices into two subsets that are as disjoint as possible, such that the sum of the edge weights within each subset is maximized, and the sum of the edge weights between the subsets is minimized. For the graph *G(V, E)* constructed previously, $V=\{v_{00}, v_{01}, v_{10}, v_{11}...v_{i0}, v_{i1}...,v_{m0}, v_{m1}\}$, there are m SNP loci and $2 \times m$ vertices. By using the minimum-cut algorithm, V is partitioned into two sets, V_0 and V_1. For SNP locus p_i, if *vi0* belongs to V_0, v_{i1} should belong to V_1. The optimization objective is shown below.

$$Obj = \min \sum_{i=0;j=0}^{m} e\left(v_{ik}, v_{jk'}\right);$$

where k and $k' \in \{0, 1\}$, $k \neq k'$, and $v_{ik} \in V_0$ and $v_{ik'} \in V_1$.

The Fiduccia-Mattheyses (FM) algorithm is a minimum-cut algorithm for graphs, primarily used to divide a graph into two equally sized parts with the minimum cut value between them [30] (Fig. 3). GCphase revised FM for partitioning V into V_0 and V_1. The specific implementation process is shown as follows.

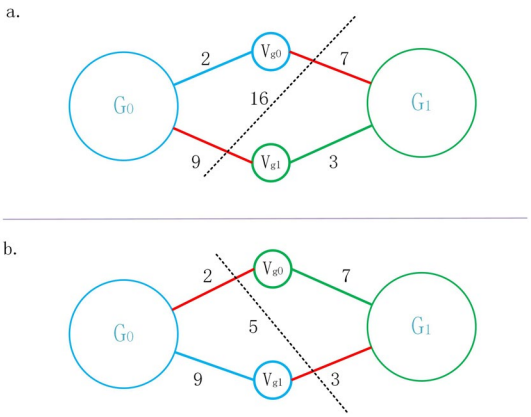

Fig. 3 For the initial set partitioning G_0 and G_1 of the initial SNP to be phased, there are edges connecting the two alleles v_{g0} and v_{g1} of any SNP locus v_g with the points in both sets. The cut value is the sum of the weights of the edges connecting the allele vertex with all vertices in its complementary set. Therefore, when the two alleles belong to different sets, different cut values exist. **a** When $v_{g0} \in G_0$, $v_{g1} \in G_1$, the cut value is the sum of the weights of the red edges, which is equal to 16. **b** When $v_{g1} \in G_0$, $v_{g0} \in G_1$, the cut value is the sum of the weights of the red edges, which is equal to 5. The cut value (16) of the allocation method in (**a**) is greater than the cut value (5) of the allocation method in (**b**). Therefore, the allocation method for the two alleles v_{g0} and v_{g1} of SNP tends to be the allocation method in (**b**), which is $v_{g1} \in G_0$, $v_{g0} \in G_1$

Step 1: Initialization. Due to the mutually exclusive nature of the two alleles at SNP loci, both alleles cannot be assigned to the same set. During initialization, GCphase selects the major allele of all the SNP loci into V_0, while the minor allele is inevitably assigned to V_1. Therefore, only the partitioning of the major alleles needs to be saved.

Step 2: We calculate the cut value of vertices and determine whether to perform allele grouping exchanges. For an SNP locus p_t with two alleles v_{g0} and v_{g1}, first, we assume that v_{g0} belongs to G_0 and v_{g1} belongs to G_1.

The cut value of v_{g0} is calculated by the following formula.

$$cut(v_{g0}) = \sum_{j=0}^{m} e(v_{g0}, v_{jp});$$

where v_{jp} belongs to G_1 and $p \in \{0,1\}$.

The cut value of v_{g1} is calculated by the following formula.

$$cut(v_{g1}) = \sum_{j=0}^{m} e(v_{g1}, v_{jq});$$

where v_{jq} belongs to G_0 and $q \in \{0,1\}$. Then, we can obtain the weights $cut_{01}(v_{g0},v_{g1}) = cut(v_{g0}) + cut(v_{g1})$.

Next, we exchange v_{g0} and v_{g1}, which means v_{g0} belongs to G_1 and v_{g1} belongs to G_0. We recalculate the cut values of v_{g0} and v_{g1} and obtain a new value $cut_{10}(v_{g0},v_{g1})$.

If $cut_{01}(v_{g0},v_{g1}) > cut_{10}(v_{g0},v_{g1})$, we assign v_{g0} to G_1 and v_{g1} to G_0.

Step 3: All the SNP loci are processed iteratively by Step 2 until no SNP locus undergoes an allele exchange operation.

After these steps, preliminary grouping results are obtained.

Correcting errors

The initial phasing results obtained from the revised FM algorithm may still contain a few errors. As the revised FM algorithm is a heuristic algorithm, it can produce good partitioning results but does not guarantee finding the global optimum. Therefore, GCphase applies two error correction steps to refine the initial phasing results and to obtain final results. In the error correction step, first, each SNP is sorted in ascending order based on its position in the reference sequence, and then, it is corrected based on the information from its connected vertices.

Step 1 of error correction: We determine the presence of switch errors between adjacent SNPs.

For the four alleles (v_{i0},v_{i1},v_{j0},v_{j1}) of two SNP loci v_i and v_j, there are two phasing results for these four alleles: $set1(v_i, v_j) = \{(v_{i0},v_{j0}) \in V_0;(v_{i1},v_{j1}) \in V_1\}$ and $set2(v_i, v_j) = \{(v_{i0},v_{j1}) \in V_0;(v_{i1},v_{j0}) \in V_1\}$. The weight of $set1(v_i, v_j)$ is $w(v_{i0},v_{j0}) + w(v_{i1}, v_{j1})$, and the weight of $set2(v_i, v_j)$ is $w(v_{i0},v_{j1}) + w(v_{i1}, v_{j0})$.

When $[w(v_{i0},v_{j0}) + w(v_{i1}, v_{j1})]—[w(v_{i0},v_{j1}) + w(v_{i1}, v_{j0})] \geq 2$, GCphase selects *set1* as the final phasing result. When $[w(v_{i0},v_{j1}) + w(v_{i1}, v_{j0})]—[w(v_{i0},v_{j0}) + w(v_{i1}, v_{j1})] \geq 2$, GCphase selects *set2* as the final phasing result. The final phasing results about (v_{i0},v_{i1},v_{j0},v_{j1}) are represented as $OptimizedSet(v_{i0},v_{i1},v_{j0},v_{j1})$;

For v_i and its adjacent v_{i+1}, if edges exist between their alleles, the phasing result about their alleles obtained by the revised FM algorithm (Sect. 2.4) is assumed to be $FM(v_{i0},v_{i1},v_{(i+1)0},v_{(i+1)1})$. When $FM(v_{i0},v_{i1},v_{(i+1)0},v_{(i+1)1}) \neq OptimizedSet(v_{i0},v_{i1},v_{(i+1)0},v_{(i+1)1})$, there is a switch error at SNP v_{i+1}. If there is no edge connecting the alleles v_i and v_{i+1}, GCphase skips v_{i+1} and considers the next locus v_{i+2}. If $FM(v_{i0},v_{i1},v_{(i+2)0},v_{(i+2)1}) \neq OptimizedSet$ ($v_{i0},v_{i1},v_{(i+2)0},v_{(i+2)1}$), there exists the following calculation for all SNP loci v_k that are connected to v_{i+2}:

$$Fs(v_{k0}, v_{k1}, v_{(i+2)0}, v_{(i+2)1}) = \frac{k-(i+2)}{|k-(i+2)|} * \left(w(v_{k0}, v_{(i+2)0}) + w(v_{k1}, v_{(i+2)1})\right);$$

when $(v_{k0},v_{(i+2)0}) \in V_0$ and $(v_{k1},v_{(i+2)1}) \in V_1$;

$$Fs(v_{k0}, v_{k1}, v_{(i+2)0}, v_{(i+2)1}) = \frac{k-(i+2)}{|k-(i+2)|} * \left(w(v_{k0}, v_{(i+2)1}) + w(v_{k1}, v_{(i+2)0})\right);$$

when $(v_{k0},v_{(i+2)1}) \in V_0$ and $(v_{k1},v_{(i+2)0}) \in V_1$;

$Fs(v_{k0},v_{k1},v_{(i+2)0},v_{(i+2)1})$ is the support degree of v_k for v_{i+2}. If $\sum_k Fs(v_k, v_{i+2}) \leq 0$, it indicates that the support for locus v_{i+2} from the upstream loci is less than the support from the downstream loci. In this case, it is also considered to have a switch error at locus v_{i+1}, and locus v_{i+1} is labelled as an error locus.

Following the given procedure, error detection is performed for each SNP locus to be phased. For the error loci, we consider a high probability of switch errors at those loci. Therefore, GCphase partitions the entire set of SNP loci into consecutive reversal blocks, with the error loci as the first loci in each reversal block. The reversal blocks are numbered from 0 in the order of their positions in the reference sequence. We reverse the alleles of SNP loci within the reversal blocks with odd numbers, which means exchanging the grouping of the two alleles.

Step 2 of error correction: We determine the support level for each SNP.

For a SNP locus v_i and a locus v_k located before it, there exist edges for the four alleles (v_{i0}, v_{i1}, v_{k0}, v_{k1}). We calculate the sum of the weights $Fn(v_k, v_i)$ of edges between alleles assigned to the same set, that is:

$Fn(v_k, v_i) = w(v_{k0}, v_{i0}) + w(v_{k1}, v_{i1})$; when $(v_{k0}, v_{i0}) \in V_0$ and $(v_{k1}, v_{i1}) \in V_1$;

$Fn(v_k, v_i) = w(v_{k0}, v_{i1}) + w(v_{k1}, v_{i0})$; when $(v_{k0}, v_{i1}) \in V_0$ and $(v_{k1}, v_{i0}) \in V_{1;}$

If $\sum_k Fn(v_k, v_i) = 0$, it is considered that the posterior support of SNP v_i is greater than the anterior support, and it is considered to have a switch error at locus v_i.

With these two error correction steps, the switch error and Hamming distance of the initial phasing results can be reduced.

Producing haplotype blocks

Since GCphase utilizes a graph-based approach for phasing SNPs, each maximum connected subgraph in the graph represents a haplotype block. This means that there are no edges connecting the two haplotype blocks. By performing a breadth-first search to traverse the entire undirected graph, all the maximum connected subgraphs can be obtained. Sorting the SNP loci within each subgraph in ascending order based on their positions on the reference genome, GCphase can obtain all the haplotype blocks.

Results

GCphase utilizes Python as the programming language and performs phasing of the entire human genome on a 24-core (Intel) CPU. However, the runtime of the program may vary depending on the length of the reads and changes in coverage depth. We compared the sequencing depths ($20\times$, $30\times$, $50\times$) of PacBio and Nanopore sequencing data on the human public genome HG002 by using three SNP phasing software [31], namely, LongPhase, WhatsHap, and HapCUT2, all focusing solely on SNP phasing. To compare the performance of GCphase on data with different sequencing accuracies, we conducted tests on the high-accuracy PacBio HiFi dataset (27x).

Data availability and command lines

First, the PacBio data were obtained by downloading the PacBio (CCS 15 kb_20 kb Chemistry2) dataset from the Genome in a Bottle (GIAB) consortium. The Nanopore data were downloaded from the NCBI official website, specifically the SRR23215364 and SRR23447694 datasets. Subsequently, sequencing data with coverage depths of $50\times$, $30\times$, and $20\times$ were generated through downsampling. The PacBio HiFi data were downloaded from NCBI as the human HG002 whole-genome sequencing data (SRR19020573). The SNP loci information was provided by the HG002 standard set from the Genome in a Bottle (GIAB) consortium.

The compare command line of WhatsHap is: WhatsHap compare–names truth, WhatsHap–tsv-pairwise eval_mine.tsv SNP.vcf. The command line of LongPhase is: longphase phase -s SNP.vcf -b sort.bam -t 1–pb = o phasedprefix. The command line of HapCUT2 is: extractHAIRS–bam sort.bam–VCF SNP.vcf–out fragment_file HapCUT2–fragments fragment_file–VCF SNP.vcf–output haplotype_output_file. The command line of WhatsHap is: WhatsHap phase–ignore-read-groups–reference = hg38.fa -o phased-WhatsHap.vcf SNP.vcf sort.bam.

Evaluation of SNP phasing by GCphase in Nanopore and PacBio sequencing data

In Nanopore sequencing, GCphase was compared with WhatsHap, HapCUT2, and LongPhase by using the whole-genome sequencing data of the human HG002 sample downloaded from NCBI at coverage depths of 20x, 30x, and 50x. The evaluation software employed the "compare" method of WhatsHap. The VCF files containing phasing information generated by GCphase and the other three methods were compared against the standard set of the human HG002 sample provided by GIAB, resulting in the final evaluation results (Table 1). In the Nanopore data at the three coverage depths, GCphase has little differences in the number of SNPs identified compared to those of the other three methods. For example, in Nanopore Long reads 50 × sequencing data, the percentage of SNP sites in GCphase phasing is 96.71%, the percentage of SNP sites in HapCUT2 phasing is 95.70%, the percentage of SNP sites in LongPhase phasing is 96.64%, and the percentage of SNP sites in WhatsHap phasing is 96.01%. In terms of Hamming distance, the performance of the four methods is similar to the number of SNPs phased, with comparable Hamming distances observed. Only in the case of 30 × coverage is there an approximately twofold difference between the maximum Hamming distance of LongPhase (10,175) and the minimum Hamming distance of WhatsHap (5187). As shown in Tables 1 and 2, in the number of blocks, there is little difference between the four methods, but we see that the performance of GCPhase on high coverage data is better than that of low coverage data. In terms of accuracy, GCphase exhibits significantly fewer switch errors than the other three methods. Across all three coverage depths, GCphase consistently has the lowest switch error rate. Additionally, the switch error rate of GCphase at 20 × is not significantly different from that at 50x.

In PacBio sequencing, GCphase was compared with WhatsHap, HapCUT2, and LongPhase by using the whole-genome sequencing data of the human HG002 sample provided by GIAB (PacBio CCS 15 kb_20 kb chemistry2) at coverage depths of 20x, 30x, and 50x. Similar to the Nanopore experiments, the evaluation was conducted by using the "compare" method of WhatsHap. The VCF files containing phasing information obtained from GCphase and the other three methods were compared against the standard set of the human HG002 sample provided by GIAB. This comparison resulted in

Table 1 Evaluation results based on Nanopore datasets

Sample	Method	Block num	SNP num (%)	SW	SW rate (%)	HMD	HMD rate (%)
Nanopore 50 ×	GCphase	14,593	96.71	262	0.30	6003	7.26
	HapCUT2	14,586	95.70	618	0.70	4087	5.33
	LongPhase	15,454	96.64	450	0.50	6386	8.07
	WhatsHap	15,096	96.01	620	0.71	5061	6.34
Nanopore 30 ×	GCphase	16,690	96.22	352	0.40	8535	9.16
	HapCUT2	16,930	95.46	627	0.69	5984	6.05
	LongPhase	18,031	96.27	569	0.63	10,175	10.55
	WhatsHap	17,661	95.82	639	0.73	5187	5.97
Nanopore 20 ×	GCphase	19,366	95.89	312	0.35	3800	3.20
	HapCUT2	19,456	95.81	740	0.82	3711	4.79
	LongPhase	20,215	95.74	479	0.56	4841	7.47
	WhatsHap	20,095	96.27	661	0.78	3722	4.73

SW: switch error; HMD: Hamming distance

Table 2 Evaluation results based on Pacbio datasets

Sample	Method	Block num	SNP num (%)	SW	SW rate(%)	HMD	HMD rate (%)
Pacbio 50 ×	GCphase	12,689	96.89	132	0.15	1247	1.14
	HapCUT2	13,716	95.85	451	0.51	1449	1.44
	LongPhase	14,198	96.92	177	0.19	1761	2.09
	WhatsHap	13,876	95.98	168	0.20	708	1.05
Pacbio 30 ×	GCphase	13,607	96.67	130	0.15	710	0.77
	HapCUT2	14,602	95.73	488	0.55	2162	2.30
	LongPhase	15,149	96.78	175	0.20	1741	1.89
	WhatsHap	14,814	95.88	232	0.27	1344	1.53
Pacbio 20 ×	GCphase	14,644	96.23	120	0.14	855	1.27
	HapCUT2	15,541	95.62	496	0.56	2061	1.97
	LongPhase	16,142	96.64	175	0.19	1318	1.59
	WhatsHap	15,820	95.8	242	0.30	1047	1.20

SW: switch error; HMD: Hamming distance

Table 3 Evaluation results based on Pacbio HiFi dataset

Sample	Method	Block num	SNP num (%)	SW	SW rate (%)	HMD	HMD rate (%)
Pacbio HiFi	GCphase	18,252	96.41	153	0.18	2284	3.10
	HapCUT2	18,493	95.59	421	0.49	849	0.8
	LongPhase	18,987	96.56	170	0.19	1345	1.5
	WhatsHap	18,634	95.70	259	0.3	633	0.6

SW: switch error; HMD: Hamming distance

the final evaluation results (Table 2). In terms of the number of phased SNPs, there was still no significant difference among the four methods. In the number of blocks, there is little difference among the four methods, but GCphase has the smallest number of blocks among the three depths (50 ×, 30 ×, 20 ×). In terms of the Hamming distance, no significant difference is observed among the four methods. However, GCphase consistently outperforms the other methods in two coverage depths (30x, 20x) and ranks in the second position in the 50 × coverage depth. In terms of accuracy, due to the higher accuracy of PacBio sequencing data compared to that of the Nanopore sequencing data, the switch error performance of each program is better in PacBio sequencing data than that in Nanopore sequencing data. However, similar to the Nanopore sequencing data, GCphase still outperforms the other methods in all three coverage depths, with switch error rates of approximately 0.15%.

The performance of GCphase on high-accuracy PacBio HiFi data

In this study, experiments were conducted on high-accuracy PacBio HiFi data (27 × coverage) to compare the evaluation indicators (Block num, Switch error, SNP num, and so on) of these four methods (Table 3). Due to the high accuracy of PacBio HiFi data, all four methods exhibit low error rates. In terms of switch error, GCphase has a significantly lower count (153) compared to that of the other methods. In the number of blocks, GCphase has the smallest number of blocks. In terms of the Hamming distance,

GCphase does not perform well here. Regarding the number of phased SNPs, all four methods had similar counts with no significant differences.

From the above experiments, we observe that all four methods, whether in PacBio or Nanopore sequencing data, demonstrate similar performance in terms of Hamming distance and the number of phased SNPs. In terms of accuracy, GCphase exhibits the lowest number of switch errors, making it more reliable in SNP phasing. Consequently, GCphase can provide more accurate and effective data support for related experiments.

In this experiment, the running time of GCphase is 100 min and the memory usage is 8 GB. The running time of LongPhased is 35 min and the memory usage is 10.2 GB. The running time of HapCUT2 is 44 min and the memory usage is 6 GB. The running time of WhatsHap is 693 min and the memory usage is 2.7 GB.

Discussion

In this paper, GCphase first constructs a graph to represent the linkage among alleles of SNPs, and utilizes the minimum cut algorithm to perform phasing. Through some experiments, the evaluation indicators shows some advantage compared with other methods, which also demonstrate graph theory can be utilized for phasing. However, due to the limitations of the graph min-cut algorithm, when dealing with highly heterozygous genomes, the performance of GCphase such as accuracy and running time will decrease significantly, because the construction of the graph will be too complex.

Conclusion

In this article, we propose a new SNP phasing program called GCphase. GCphase utilizes the minimum-cut algorithm on a graph to perform initial phasing of SNP loci and then applies two additional correction steps to refine the phasing at each locus, ensuring more accurate phasing results. By comparing GCphase with three SNP phasing software programs, WhatsHap, HapCUT2, and LongPhase, on different datasets, GCphase achieves the lowest switch error rates. Furthermore, through comparison on the high-accuracy PacBio HiFi dataset, GCphase outperforms the other four methods in terms of switch error rates. These results demonstrate that GCphase exhibits high applicability across various datasets while maintaining accuracy.

Acknowledgements

Thank you very much to the anonymous reviewers and editors for their valuable comments on improving this work.

Author contributions

JWL, JYW and JFW participated in the design of the study and the analysis of the experimental results. JYW and HXZ performed the implementation. JYW and JWL prepared the tables and figures. JWL and JFW summarized the results of the study and checked the format of the manuscript. All authors read and approved the final manuscript.

Funding

This research was supported by the National Natural Science Foundation of China (Grant No. 61972134, 62372156), Young Backbone Teachers of Henan Province (Grant No. 2020GGJS050), Doctoral Fund of Henan Polytechnic University (Grant No. B2018-36), Innovative Research Team of Henan Polytechnic University (Grant No. T2021-3), and Henan Provincial Department of Science and Technology Research Project (Grant No. 232102211046).

Availability of data and materials

The download links for the Nanopore sequencing data are as follows: https://www.ncbi.nlm.nih.gov/sra/?term = SRR23215364 https://www.ncbi.nlm.nih.gov/sra/?term = SRR23447694 The download links for the Pacbio sequencing data are as follows: https://sra-pub-src-2.s3.amazonaws.com/SRR10382245/m64011_190830_220126.fastq.1 https://sra-pub-src-2.s3.amazonaws.com/SRR10382244/m64011_190901_095311.fastq.1 https://sra-pub-src-2.s3.amazonaws.com/SRR10382249/m64012_190920_173625.fastq.1 https://sra-pub-src-2.s3.amazonaws.com/SRR10382248/m64012_190921_234837.fastq.1 https://sra-pub-src-2.s3.amazonaws.com/SRR10382247/m64015_190920_185703.fastq.1 https://sra-pub-src-2.s3.amazonaws.com/SRR10382246/m64015_190922_010918.fastq.1

The download links for the Pacbio HiFi sequencing data are as follows: https://www.ncbi.nlm.nih.gov/sra/?term = SRR8858432 The source code is available from GitHub at https://github.com/baimawjy/GCphase.

Declarations

Competing interests
The authors declare no competing interests.

Received: 15 November 2023 Accepted: 14 August 2024
Published online: 19 August 2024

References

1. Lan W, Lai D, Chen Q, Wu X, Chen B, Liu J, Chen YPP. LDICDL: LncRNA-disease association identification based on collaborative deep learning. IEEE/ACM Trans Comput Biol Bioinf. 2020;19(3):1715–23.
2. Chaisson MJP, Sanders AD, Zhao X, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. Nat Commun. 2019;10(1):1–16.
3. 1000 Genomes Project Consortium (2015) A global reference for human genetic variation. Nature, 526(7571), 68
4. Garg S. Computational methods for chromosome-scale haplotype reconstruction. Genome Biol. 2021;22(1):1–24.
5. Simpson JT, Workman RE, Zuzarte PC, David M, Dursi LJ, Timp W. Detecting DNA cytosine methylation using nanopore sequencing. Nat Methods. 2017;14(4):407–10.
6. van Dijk EL, Jaszczyszyn Y, Naquin D, Thermes C. The third revolution in sequencing technology. Trends Genet. 2018;34:666–81.
7. Du H, Liang C. Assembly of chromosome-scale contigs by efficiently resolving repetitive sequences with long reads. Nat Commun. 2019;10:5360.
8. Vollger MR, et al. Improved assembly and variant detection of a haploid human genome using single-molecule, high-fidelity long reads. Ann Hum Genet. 2019. https://doi.org/10.1111/ahg.12364.
9. Vollger MR, et al. Long-read sequence and assembly of segmental duplications. Nat Methods. 2019;16:88–94.
10. Victoria Wang X, Blades N, Ding J, Sultana R, Parmigiani G. Estimation of sequencing error rates in short reads. BMC Bioinf. 2012;13:1–12.
11. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Loose M. Nanopore sequencing and assembly of a human genome with ultra-long reads. Nat Biotechnol. 2018;36(4):338–45.
12. Wenger AM, Peluso P, Rowell WJ, Chang PC, Hall RJ, Concepcion GT, Hunkapiller MW. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. Nat Biotechnol. 2019;37(10):1155–62.
13. Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Wang J. De novo assembly of human genomes with massively parallel short read sequencing. Genome Res. 2010;20(2):265–72.
14. Abou Saada O, Friedrich A, Schacherer J. Towards accu-rate, contiguous and complete alignment-based polyploid phasing al-gorithms. Genomics. 2022;114(3):110369.
15. Schrinner SD, Mari RS, Ebler J, Rautiainen M, Seillier L, Reimer JJ, Klau GW. Haplotype threading: accurate polyploid phasing from long reads. Genome Biol. 2020;21(1):1–22.
16. He D, Saha S, Finkers R, Parida L. Efficient algorithms for polyploid haplotype phasing. BMC Genomics. 2018;19(2):171–80.
17. Berger E, Yorukoglu D, Peng J, Berger B. HapTree: a novel Bayesian framework for single individual polyplotyping using NGS data. PLoS Comput Biol. 2014;10(3):e1003502.
18. Wu J, Chen X, Li X. Haplotyping a single triploid individual based on genetic algorithm. Biomed Mater Eng. 2014;24:3753–62.
19. Das S, Vikalo H. SDhaP: haplotype assembly for diploids and polyploids via semidefinite programming. BMC Genomics. 2015;16:260.
20. Patterson M, Marschall T, Pisanti N, Van Iersel L, Stougie L, Klau GW, Schönhuth A. WhatsHap: weighted haplotype assembly for future-generation sequencing reads. J Comput Biol. 2015;22(6):498–509.
21. Majidian S, Kahaei MH, De Ridder D. Hap10: reconstructing accurate and long polyploid haplotypes using linked reads. BMC Bioinf. 2020;21(1):1–18.
22. Sankararaman A, Vikalo H, Baccelli F. ComHapDet: a spatial community detection algorithm for haplotype assembly. BMC Genomics. 2020;21:1–14.
23. Edge P, Bafna V, Bansal V. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome Res. 2017;27(5):801–12.
24. Xie M, Wu Q, Wang J, Jiang T. H-PoP and H-PoPG: heuristic partitioning algorithms for single individual haplotyping of polyploids. Bioinformatics. 2016;32(24):3735–44.
25. Luo X, Kang X, Schönhuth A. Phasebook: haplotype-aware de novo assembly of diploid genomes from long reads. Genome Biol. 2021;22(1):1–26.
26. Lin JH, Chen LC, Yu SC, Huang YT. LongPhase: an ultra-fast chromosome-scale phasing algorithm for small and large variants. Bioinformatics. 2022;38(7):1816–22.
27. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–100.
28. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, 1000 Genome Project Data Processing Subgroup (2009) The sequence alignment/map format and SAMtools. Bioinformatics, 25(16), 2078–2079.

29. Gilman P, Janzou S, Guittet D, Freeman J, DiOrio N, Blair N, Wagner M (2019) Pysam (python wrapper for system advisor model" sam") (No. PySAM). National Renewable Energy Lab.(NREL), Golden, CO (United States)
30. Kernighan BW, Lin S. An efficient heuristic procedure for partitioning graphs. Bell Syst Tech J. 1970;49(2):291–307.
31. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Salit M. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data. 2016;3(1):1–26.

Publisher's Note