Towards accurate, contiguous and complete alignment-based polyploid phasing algorithms

Omar Abou-Saada, Anne Friedrich, Joseph Schacherer

Genomics 114 (2022) 110369

Contents lists available at ScienceDirect

Genomics

journal homepage: www.elsevier.com/locate/ygeno

Review

Towards accurate, contiguous and complete alignment-based polyploid phasing algorithms

Omar Abou Saada [a], Anne Friedrich [a], Joseph Schacherer [a,b,*]

[a] Université de Strasbourg, CNRS, GMGM UMR, 7156 Strasbourg, France
[b] Institut Universitaire de France (IUF), Paris, France

ARTICLE INFO

Keywords:
Population genomics
Heterozygosity
Phased genome
Polyploid
Algorithms

ABSTRACT

Phasing, and in particular polyploid phasing, have been challenging problems held back by the limited read length of high-throughput short read sequencing methods which can't overcome the distance between heterozygous sites and labor high cost of alternative methods such as the physical separation of chromosomes for example. Recently developed single molecule long-read sequencing methods provide much longer reads which overcome this previous limitation. Here we review the alignment-based methods of polyploid phasing that rely on four main strategies: population inference methods, which leverage the genetic information of several individuals to phase a sample; objective function minimization methods, which minimize a function such as the Minimum Error Correction (MEC); graph partitioning methods, which represent the read data as a graph and split it into *k* haplotype subgraphs; cluster building methods, which iteratively grow clusters of similar reads into a final set of clusters that represent the haplotypes. We discuss the advantages and limitations of these methods and the metrics used to assess their performance, proposing that accuracy and contiguity are the most meaningful metrics. Finally, we propose the field of alignment-based polyploid phasing would greatly benefit from the use of a well-designed benchmarking dataset with appropriate evaluation metrics. We consider that there are still significant improvements which can be achieved to obtain more accurate and contiguous polyploid phasing results which reflect the complexity of polyploid genome architectures.

1. Introduction

An organism's genome is the haploid (1n) set of its chromosomes. An organism is diploid (2n) if it has two copies of its genome, and polyploid if it has more than two (>2n). Many yeasts such as the model organism *Saccharomyces cerevisiae* can survive as haploids, humans are a well-known diploid species and important crops such as *Solanum tuberosum* (4n) are polyploid. The current processes to sequence a genome all involve a step which fragments the DNA molecules. To obtain an accurate and complete view of the genome, these DNA fragments, also called "reads", must then be pieced back together into the original chromosomes. For heterozygous organisms, this fragmentation makes it difficult to know which SNPs (Single Nucleotide Polymorphisms) co-occur on the same chromosome. Obtaining accurate haplotype information is biologically relevant, as it can provide more accurate reference genomes [1], uncover allele-specific functions such as heterosis [2,3], allele-specific expression [4] and compound heterozygosity [5]. Haplotype information can also be leveraged in GWAS studies [6], or used to dissect the evolution of polyploids [7,8], and the origins of hybrids [9]. The challenge of determining the original sequences of the chromosomes, known as haplotypes, is the phasing problem. For a heterozygous diploid, solutions to the problem can exploit an obvious symmetry: finding the sequence of one haplotype necessarily leads to knowing the sequence of the other. The polyploid phasing problem, however, does not display this symmetry, which greatly increases its complexity. Knowing the sequence of one haplotype still leaves uncertainty over the two or more remaining haplotypes.

Solutions to the polyploid phasing problem can be categorized into three main strategies: Physical separation methods, *de novo* haplotype assembly, and alignment-based phasing. Briefly, physical separation methods attempt to only sequence one chromosome at a time, sidestepping the polyploid phasing problem by sequencing individual chromosomes [10]. *De novo* haplotype assembly methods ambitiously attempt to simultaneously recreate the different haplotypes and resolve the structure of the genome, typically relying on long-range sequencing methods such as Hi-C [11]. Alignment-based phasing methods map the

* Corresponding author at: Université de Strasbourg, CNRS, GMGM UMR, 7156 Strasbourg, France.
E-mail address: schacherer@unistra.fr (J. Schacherer).

https://doi.org/10.1016/j.ygeno.2022.110369
Received 31 October 2021; Received in revised form 9 March 2022; Accepted 11 April 2022
Available online 26 April 2022

sequencing reads to a reference sequence and identify variable positions, which are then used as input to a phasing algorithm that outputs predicted haplotypes.

In this review, we discuss the different paradigms in the field of alignment-based polyploid phasing methods and how the performance of these methods is evaluated. We also propose that it would greatly benefit the field to standardize the performance metrics used to evaluate proposed methods, including the generation of gold standard datasets to systematically benchmark against.

2. Trends in polyploid phasing solutions

All alignment-based phasing methods share the same pre-processing steps. First, a reference sequence must be chosen or assembled *de novo*, to serve as a guide. Then, the sequenced reads are mapped to this reference sequence and variable positions are identified. Finally, the dataset of reads, reduced to their variable, phase-informative positions, is used as input for the phasing method (Fig. 1). The methods proposed as solutions to the polyploid phasing problem are highly varied in their approaches and mathematical and conceptual underpinnings. To provide a coherent framework for this review, we delineate the development and usage of different strategies, identifying four major trends:

Population inference methods, which leverage mapped reads of known genotypes in related individuals or in a population to infer the haplotypes in a sample.

Objective function optimization methods, which typically represent the mapped reads as a matrix and seek to minimize an objective function which typically represents the amount of discrepancies between the predicted haplotypes and the observed sequencing data.

Graph partitioning methods, which convert the mapped reads to a graph and seek to split the graph into subgraphs that correspond to the haplotype predictions.

Cluster building methods, which rely on the similarity between mapped reads to group them into clusters that correspond to predicted haplotypes.

We discuss these four paradigms, their implementations and limitations.

2.1. Population inference

To solve the polyploid phasing problem, population inference methods rely on the availability of significant amounts of genomic data. Rather than attempt to phase each genome individually, these methods leverage the genetic information of several individuals to inform the phasing (Fig. 2). The choice of population is important to the strategy, and can range from large, non-specific populations of individuals of the same species [12–16], to highly specific, smaller populations such as parents or siblings [17–19].

The first such methods, SATlotyper [12] and polyHap [13], used large populations of unrelated individuals, while later methods such as TriPoly [17], PopPoly [18] and mapPoly [19] exploit pedigree information to inform their predictions. The methods employed to leverage population data for phasing are highly varied: SATlotyper casts the polyploid phasing problem as a boolean satisfiability problem [12], polyHap [13] and mapPoly [19] both use Hidden Markov Models to leverage the statistical information in populations of individuals, superMASSA [14] frames it as a graphical Bayesian problem, SHEsisPlus [15] developed a formulation of the Expectation Maximization algorithm to predict the most likely haplotypes, and TriPoly [17] and PopPoly [18] both leverage pedigree information and Mendelian laws of inheritance to phase haplotypes. Finally, while Poly-Harsh [16] is not fully a population inference algorithm, its authors describe a clustering algorithm using population inference to connect fragmented phase blocks, improving the contiguity of phasing.

Population inference methods are particularly powerful when it comes to extending the reach of short read sequencing using statistical information. This has a significant effect on contiguity without requiring the use of other sequencing methods. The public availability of a significant amount of sequencing data for various organisms is an invaluable resource for this method, though applying it to less studied organisms can prove more costly than other strategies presented here. One of the notable limitations inherent to population inference methods is the requirement of a sequenced population. For the methods which require large populations, the material and labor cost of obtaining and sequencing a large number of individuals can be a significant limiting factor. For those which require fewer but related individuals, the difficulty can lay in the existence or availability of such individuals. This renders these methods inappropriate for situations with limited resources, such as any study of a single individual, particularly if it is an individual of a species which is not extensively studied and sequenced.

The choice of the reference sequence against which to map the population is also a crucial one for these methods. The mapping and variant calling operations can be computationally expensive, and their quality is dependent on the quality of the reference sequence in use. Here, a seemingly intractable problem is apparent for some applications of population inference methods. Any species with a propensity for structural variation would be difficult to phase with these methods, as the architecture of their genomes does not lend itself well to using the same reference for all individuals of the population. This makes it impossible to pick a reference sequence which accurately represents the population, and difficult to obtain sufficiently many distinct individuals with the same genomic architecture. Not all organisms have extensive structural variations within their population, however, and for populations which maintain highly similar genomic architectures, this strategy remains appropriate.

2.2. Objective function optimization

The objective function optimization strategy seeks to solve the phasing problem for single individuals. This method defines an objective function, which it then seeks to minimize algorithmically (Fig. 3). The objective function is typically a measurement of how well the predicted haplotypes correspond to the reads in the dataset. For example, for MEC (Minimum Error Correction) optimization, the objective function counts

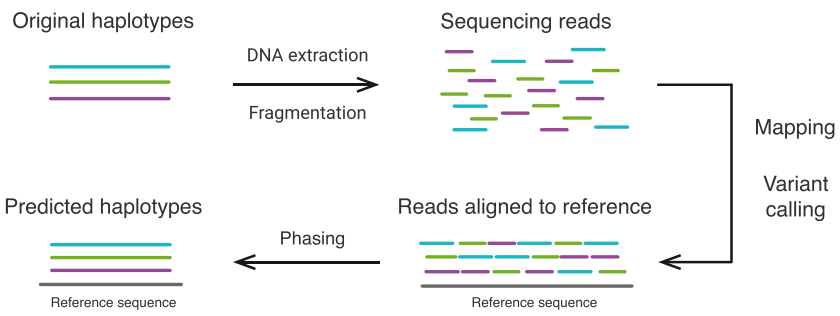

Fig. 1. Alignment-based phasing.
Alignment-based phasing methods invariably require the following steps: DNA sequencing of the sample, which fragments the DNA into sequenced reads. The reads are then mapped to a reference sequence and heterozygous sites are identified by variant calling. The dataset of reads associated with their variable positions is then input to a phasing method and predicted haplotypes are output. These predicted haplotypes therefore conform to the structure of the reference sequence that was aligned to initially.

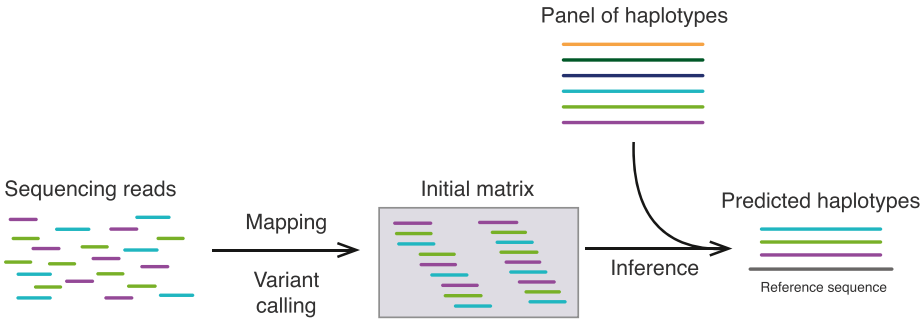

Fig. 2. Population inference strategy.
Population inference methods typically cast the mapped reads to a matrix and compare them to a panel composed of haplotype information obtained from sequencing either a large population of individuals, or a smaller group of individuals related to the sample. Haplotypes are predicted through statistical inference based on the frequency of jointly observed genotypes.

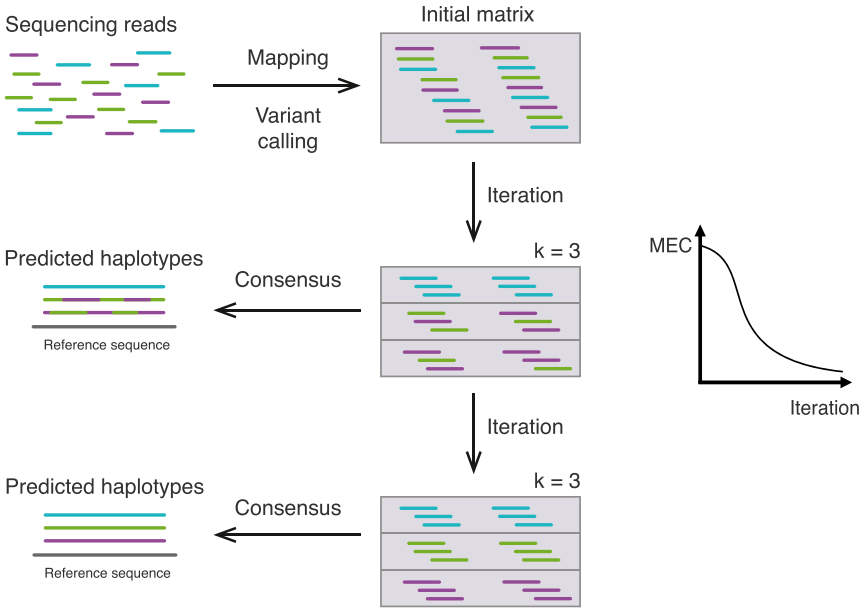

Fig. 3. Objective function optimization strategy.
Objective function minimization strategies define a function which has a high score when the sample is not phased, and an increasingly lower score as the phasing improves. In theory such a function should lead to increasingly accurate haplotypes, until finally reaching a good haplotype prediction when minimized. In this figure we used the dominant MEC function as an example, though other functions can be used in this strategy. The objective function minimization strategy treats the polyploid phasing problem as an optimization problem which splits the matrix into *k* submatrices and applies various optimization methods to solve it. Each submatrix is then converted to a haplotype prediction through consensus of the reads.

how many mismatches there are between the predicted haplotypes and the set of mapped reads. The intuition is that a low MEC score implies a highly accurate phasing. Another variant of this method is the MFR (Minimum Fragment Removal) method, in which the objective function is minimized when the predicted haplotypes and the set of mapped reads are in perfect agreement after the removal of as few reads as possible. Typically, but not always, objective function optimization methods cast the dataset of reads as a matrix, and implement known or novel algorithms and heuristics intended to minimize the chosen objective function in the matrix.

Objective function optimization methods showcase a variety of heuristics and statistical methods. The first full application of the objective function optimization method for higher ploidies is found in HapTree [20], which uses a relative likelihood function to phase polyploid genomes. However, the most common objective function is the MEC [16,21–26]. The first polyploid application which optimizes the MEC function, GTIHR [21], uses a genetic algorithm which only applies to triploids. It was followed by SDhaP [22], whose authors developed a novel convex optimization method to minimize the MEC for higher ploidies. SCGDhap [23], BFBP [24], AltHap [25] and Poly-Harsh [16] all also use the MEC function and attempt to optimize it using various approaches, such as BFBP's belief propagation algorithm derived from communication theory and Poly-Harsh's Gibbs sampling method. EHTLD [26] extends the MEC function by applying additional genetic constraints, naming it the MEC with Genotype Information (MEC/GI), but it only applies to triploids. Finally, HaplotypeAssembler [27] uses the MFR objective function and optimizes it using integer linear programming.

The approach of objective function optimization is dominated by the MEC function, yet remains varied in the methods implemented to solve it. In contrast with the preceding population inference strategy, these methods aim to phase individual genomes, relying solely on the mapped reads to inform the reconstruction of the haplotypes. This, however, puts the objective function optimization and other strategies at a disadvantage when the sequencing data is not sufficiently informative to overcome low levels of phasing information. This would be the case of genomes with particularly low levels of heterozygosity (<0.1%, or an average of 1 heterozygous SNP per kb) or datasets in which the sequencing data consists of reads that are shorter than the distance between heterozygous positions, inevitably leading to fragmented haplotypes. Long reads are particularly interesting for phasing applications due to how phase-informative they are. Each long read can contain significantly more heterozygous positions than its short read counterparts. However, none of the objective function optimization methods cited here take long reads into account. The intuition behind the optimization of an objective function is typically guided by the notion that the predicted haplotypes must conform in some way to the information present in the set of mapped reads. This assumption holds fairly well only if the read dataset is known to be of high quality and not error-prone. These methods are more appropriate for relatively error-free reads.

Objective function minimization strategies, like the graph

partitioning and cluster building strategies presented below, do not rely on population data and therefore don't suffer from the same issues with complex genomic architectures as the population inference methods. However, they all coerce the reads into a selectable ploidy *k*, which is incompatible with the biological reality that a polyploid genome of ploidy *n* does not necessarily have *n* haplotypes throughout its genome. For example, it may have an extra copy of one of its chromosomes, with its own unique haplotype. Alternatively, it may have the exact same haplotype for a large region of two of its chromosomes, effectively presenting only *n*-1 haplotypes for that region. An algorithm that coerces exactly *k* haplotypes on the entire genome will provide erroneous results if these edge cases are not considered and explicitly handled in some way. For polyploid genomes with simpler genomic architectures, where the ploidy and number of haplotypes remain stable, these methods remain appropriate.

2.3. Graph partitioning

The graph partitioning strategy casts the dataset of mapped reads as a graph. Typically, each mapped read is a node and each edge represents how similar two nodes are. The goal is then to determine the optimal way to split the graph into subgraphs that represent the different predicted haplotypes (Fig. 4). It departs from the objective function strategy by seeking to group similar mapped reads together, away from dissimilar mapped reads, rather than seeking to optimize for coherence of the predicted haplotypes with the set of mapped reads. It achieves this through the use of the graph model and its associated mathematical tools and algorithms. To this end, graph partitioning algorithms are implemented or developed and applied, outputting subgraphs which are then converted to haplotype sequences, usually through majority voting.

Typical graph partitioning solutions to the polyploid phasing problem cast the mapped reads as nodes, and give weights to overlapping nodes which penalize differences between them. Then a graph partitioning algorithm is applied to the graph in order to obtain the subgraphs which correspond to the haplotype predictions. In HapColor [28], the weight between mapped reads corresponds to the number of mismatches between them. It then applies the DSatur (Degree of saturation) algorithm, obtaining a high number of subgraphs, which it then iteratively merges until only *k* subgraphs remain. For PolyCluster [29], Hap10 [30], ComHapDet [31] and WhatsHap Polyphase [32], the nodes are also mapped reads, and the weights are negative if there are many mismatches between reads, and positive if there are many matches. This then encourages their respective graph partitioning algorithm to cut the graph along the lines of negatively weighted disagreement. Hap10 and WhatsHap Polyphase distinguish themselves through their use of long reads. Hap10 uses 10× linked reads and applies a max-k-cut algorithm, while WhatsHap Polyphase uses PacBio and Oxford Nanopore long reads and applies heuristics to solve the cluster editing problem. Notably, the initial cluster editing step of WhatsHap Polyphase is ploidy agnostic, meaning it is not biased towards a specific ploidy. However, WhatsHap Polyphase still coerces a specific ploidy, but it does so while explicitly taking into account the edge case of local regions of similarity between haplotypes in a process it terms haplotype threading. Finally, the recently published flopp [33] uses Uniform Tree Partitioning to partition reads by similarity into *k* subgraphs based on their newly defined objective function, the UPEM (uniform probabilistic error minimization) score.

There have been two other graph partitioning methods which cast the mapped reads to a graph in a different way. The first application of graph partitioning methods to the polyploid phasing problem was an extension to HapCompass [34] which made it applicable to polyploids. Under the HapCompass model, each node is a SNP, and the mapped reads are edges. The use of SNPs as nodes is uncommon, but observed again recently with HRCH [35], another non-standard example of a graph partitioning method. HRCH uses a weighted SNP hypergraph, which it then partitions into predicted haplotypes using the hypergraph partitioning algorithm hMETIS.

The graph partitioning strategy relies on the notion that reads which derive from the same haplotype will be similar to each other, and dissimilar to reads derived from other haplotypes. They should then naturally form tightly connected graphs if attributed weights which correspond to their similarity (or dissimilarity). This strategy leverages well-established algorithms which efficiently split graphs into well-connected components. WhatsHap Polyphase's application of a graph partitioning strategy to long read datasets and its handling of part of the complexity brought on by the variability in genomic architectures is encouraging for the handling of the more complex problems of polyploid phasing. However, most graph partitioning algorithms, and all methods

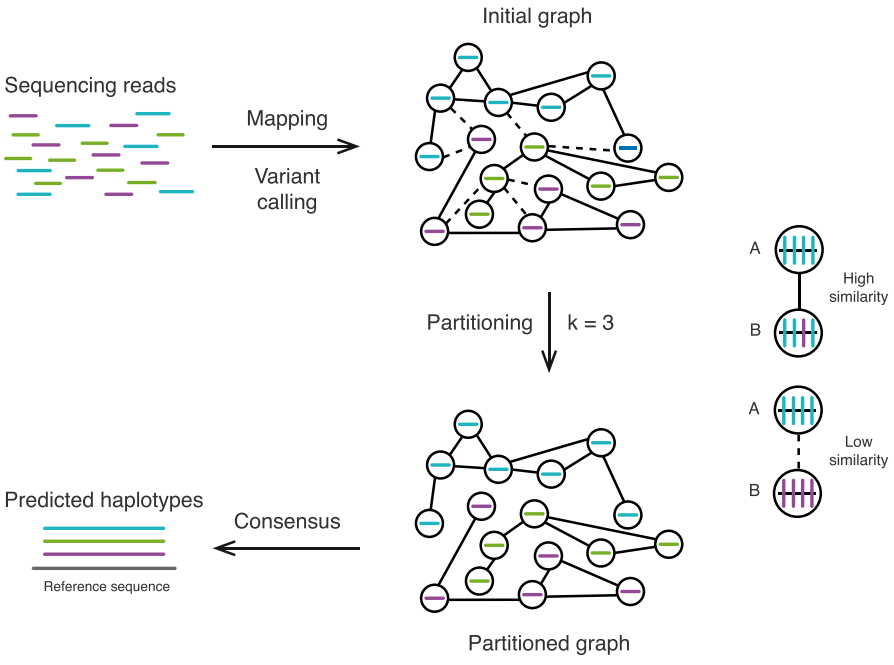

Fig. 4. Graph partitioning strategy.
Graph partitioning strategies cast the mapped reads to a graph in which typically the reads are nodes and the edges between them correspond to a measure of how similar or dissimilar the reads are to each other based on the variants they carry. The goal is to identify *k* subgraphs of reads derived from the same haplotype, and to that end various graph partitioning methods are applied. Each subgraph is then converted to a haplotype prediction through consensus of the reads.

presented here, coerce the graph into *k* subgraphs. This leads to the same pitfalls as discussed for the objective function strategy, notably with structural variants and aneuploidies. While it may be possible to handle all edge cases in post-processing steps, careful consideration should be placed upon the sample being studied and the limitations and biases inherent to the phasing algorithm being used. There may be an existing or yet to be developed graph partitioning method which is intrinsically capable of resolving complex genomes containing aneuploidies, structural variants and a variable number of local haplotypes, however this has not yet been shown. This strategy may prove to be the model of choice for resolving complex polyploid genomes, particularly when combined with long reads.

2.4. Cluster building

The cluster building strategy groups methods which do not appear to have a favored way of representing the data. Instead, these methods iteratively create and extend clusters of similar reads using heuristics (Fig. 5). These methods are related to the graph partitioning methods in that they establish a way to cluster similar reads together, and dissimilar reads apart. However, they either do not explicitly cast the mapped reads to a graph, or they do not use graph partitioning algorithms. Another notable aspect of the methods in this strategy is the interest displayed in leveraging long reads to improve phasing quality.

H-Pop and H-PopG [36] represent the read data as a matrix and seek to split the matrix into *k* parts, with each part corresponding to a group of reads with maximal similarity. Each group then represents a different haplotype, and it therefore introduces a diversity measure, which seeks to maximize the difference between the *k* groups, or predicted haplotypes. Similarly, Ranbow [37] uses a seed and extend paradigm to locally, iteratively cluster reads together based on similarity and dissimilarity measures. While it does coerce *k* haplotypes, it also handles the edge case where the number of haplotypes is less than *k*. While Ranbow is described only for short reads, its authors express interest in extending it to use long reads.

All of the cluster building methods which do use long reads are ploidy agnostic, meaning they do not coerce a specific ploidy. Ploidy agnostic methods seek to cluster similar reads together and prevent dissimilar clusters from merging, which naturally results in *n'* clusters which are ideally equal to the *n* haplotypes present in the data. Chaisson et al., *2018* propose a correlation clustering method to solve the polyploid phasing problem using long reads, however it is designed to only phase parts of the genome, intended to resolve multicopy duplications, and no tool was released [38]. This is the first ploidy agnostic phasing method applied to part of a genome. In an unnamed method [9], Fay et al., *2019* describe a custom phasing algorithm they developed in order to analyze admixed polyploid yeasts. Using mapped long reads, they score similar reads positively, and dissimilar ones negatively, then proceed to iteratively merge long reads together for three rounds. This is the first example of a ploidy agnostic method applied to entire genomes, though it is not compared to other methods or released as a tool for the community to use. Finally, nPhase [39], a method we recently developed, solves the polyploid phasing problem by iteratively clustering similar reads together until only unique haplotypes remain. It is the first ploidy agnostic phasing method applicable to entire genomes to be released as a tool.

The cluster building strategy shares the same intuition that drives the graph partitioning strategy. Reads derived from the same haplotype will resemble each other and be different from reads derived from another haplotype. However, in contrast with the graph partitioning strategy, these methods do not cast the set of mapped reads to a graph. Instead, the cluster building methods are defined by the strategy of iteratively growing clusters of reads while maintaining the diversity of the clusters. Interestingly, this strategy has led to three ploidy agnostic phasing methods, all of which leverage long reads. Ranbow handles the edge case where the number of haplotypes is locally lower than the ploidy, and the ploidy agnostic methods in theory adapt to the shape of the genomic architecture. While it should be expected that ploidy agnostic methods are capable of handling aneuploidies and local changes in the number of haplotypes, they do not provide any handling of other structural variants such as heterozygous inversions and translocations. This is partly a consequence of the nature of all of these strategies as alignment-based phasing methods, since they are limited to the genomic architecture imposed by the haploid reference sequence. However, long reads can provide a significant amount of information about structural variants, notably through the analysis of individual reads which map to distant genomic regions, sometimes on different chromosomes, known as split reads. No method of polyploid phasing attempts to use split reads to resolve heterozygous structural variation. The development of such a method would be a significant step towards complete polyploid phasing methods. For complex genomes, cluster building methods, and in particular ploidy agnostic phasing methods are appropriate. However, one major drawback of ploidy agnostic methods is the interpretability of the results. It is less straight-forward to handle ploidy agnostic phasing results than phasing results which neatly fit an expectation of *k* haplotypes.

3. Overview

The four strategies we described attempt to solve the same problem, and there are large interfaces between them. The way a problem is modeled influences the solution space that is intuitive and the

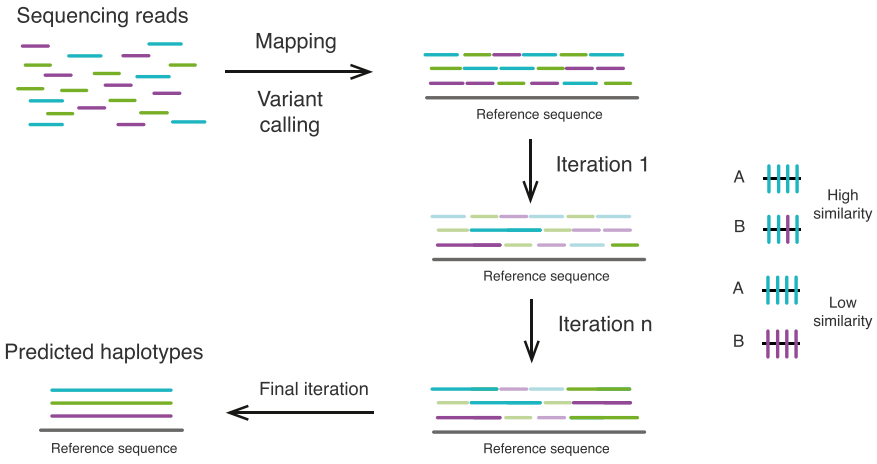

Fig. 5. Cluster building strategy.
Cluster building strategies do not appear to have a favored model to which to cast the set of mapped reads. These methods typically score the similarity and dissimilarity between overlapping reads and iteratively build local clusters from the most similar pairs of reads. This strategy has led to ploidy agnostic methods, which cluster reads until the remaining clusters are too dissimilar rather than cluster them until the remaining clusters fit *k* haplotype predictions.

mathematical tools which are at our disposal to solve it. We find that the field of alignment-based polyploid phasing algorithms has evolved to tackle increasingly complex formulations of the problem, using increasingly sophisticated strategies and tools, yet still has significant room for improvement. In particular, long reads are under-exploited despite representing a significant tool to obtain large amounts of phase information. The polyploid phasing problem also needs to explicitly tackle and resolve the problems of heterozygous structural variants, aneuploidies and local variations in the number of haplotypes. The ploidy agnostic methods tackle some of the complexity of genomic architecture, but not all. For brevity, we did not discuss whether or not each method phases only biallelic SNPs, or also phases indels and multiallelic SNPs. However, it is clear that the majority of methods limit themselves to only phasing biallelic SNPs, sometimes also multiallelic SNPs, and indels seem to only be phased by Ranbow. We also discussed the importance of the chosen reference sequence, and it may become common practice to perform a collapsed *de novo* assembly to generate an appropriate reference for each sample prior to alignment-based phasing. However, this also entails having to generate a new genome annotation for downstream analyses and can unnecessarily complicate comparisons between samples. Overall, there is still room for improvement in the field of polyploid phasing algorithms and recommended practices.

4. Validation datasets and performance metrics

Once a polyploid phasing method has been developed, its performance must be evaluated. To that end, a validation dataset which corresponds to a set of reads obtained from a polyploid must be given as input to the phasing method, and the output haplotype predictions must be evaluated by performance metrics.

The validation dataset can be simulated or real. In the case of simulated datasets, it is possible to know the optimal phasing result, which allows for the use of detailed metrics to better understand the performance of the polyploid phasing algorithm. A validation dataset can be fully simulated, such as in Haptree [20], which randomly generates haplotypes and simulates reads derived from these haplotypes. Validation datasets can also be partially simulated, or reconstructed. This is the case for WhatsHap Polyphase and nPhase, which both merge real sequencing reads of organisms with known haplotypes. WhatsHap Polyphase combines human datasets with known haplotypes, while nPhase combines *S. cerevisiae* datasets of haploid and homozygous diploid individuals. Fully simulated datasets have a high degree of control over all characteristics of the genome, which allows them to test the effects of different ploidy levels, heterozygosity levels, genome architectures, coverage levels. However, these methods are highly dependent on the accuracy of their simulations of genomes and sequencing results. Partially simulated datasets are more faithful simulations of real haplotype phasing scenarios as they use real genomes, with real SNPs and real sequencing reads. However, these genomes are still artificially produced, typically presenting relatively uniform distance between haplotypes, and there is less control over their characteristics, which limits the testing space. Some parameters, such as the effects of coverage level and heterozygosity rate, can still be queried by downsampling the number of reads or the variable positions input to the phasing algorithm, however this process is less straight-forward than it is for a fully simulated dataset.

For all simulated datasets, the ground truth is known and can be used to evaluate the predicted haplotypes. A variety of metrics have been implemented, here we discuss those most commonly used in the field.

The MEC score is not only an objective function used in a number of phasing methods, but also a metric which has been routinely used as evidence of good phasing. Individual reads are, barring sequencing errors, considered to be naturally phased sequences. It can therefore appear intuitive that comparing the phases of predicted haplotypes with the phases of individual reads can be a useful proxy for phasing quality. Another desirable property of the MEC score as a performance metric is that it can be calculated even when the true phase is unknown, bypassing any need for a more complex validation process. Despite these qualities, this metric has received some criticism in the context of the polyploid phasing problem. In their paper on Ranbow, Moeinzadeh et al. note that the MEC metric is incomplete, only considering sequencing errors [37], while in their paper for WhatsHap Polyphase, Schrinner et al. point out that MEC scores can be lowered by strategies which lead to objectively worse phasing results [32]. This is because in polyploids, some haplotypes can be identical over large regions, which they term “collapsed regions”. The MEC score can be lowered in these collapsed regions by clustering the reads of identical haplotypes together, as one haplotype, and create a new haplotype which contains noisy reads, which will then no longer cause an increase of the MEC score. It is also trivial to obtain a perfect MEC score by not clustering reads together at all, and simply comparing the input set of reads to itself. Due to the significantly higher error rate of long read sequencing, any method relying on these reads will necessarily obtain worse MEC scores despite the obvious advantages of long reads, further limiting the usefulness of this metric for the evaluation of polyploid phasing methods, which exploit different input read types (such as comparisons between short read and long read methods). Finally, the MEC score is necessarily dependent on the error rate of the sequencing and base-calling technology, the read coverage used, and therefore gives no direct, quantitative indication on the trustworthiness of a prediction. It can only qualitatively be used to compare different datasets to each other. On its own, the MEC score is not a straight-forward metric for phasing quality and should be interpreted carefully in conjunction with other metrics, such as the average phasing block length and number of haplotype blocks predicted.

Due to these flaws in the MEC as a performance metric, it does not appear optimal (when the ground truth is known) to use the MEC score to validate a phasing method, or to compare it to other methods, instead of assessing phasing quality directly by calculating the exact accuracy of the predictions made.

The Switch Error Rate (SWER), also described as the Vector Error Rate (VER), measures how frequently the predicted haplotype switches between true phases (Fig. 6A). Optimization of this metric does not necessarily lead to improved phasing accuracy, as a single vector error can reduce the accuracy by half. In a real use case, the presence of a switch error has a much more significant consequence than the presence of a few point errors. As we argued in our paper on nPhase [39], the interpretability of the SWER is further complicated by the fact that the presence of more switch errors is not incompatible with significantly better phasing results, rendering the metric fundamentally unpredictable. The use of this metric is no doubt motivated by the observation that it is possible to phase several SNPs correctly, yet a single switch error can reduce the accuracy by up to 50%. Hence methods which produce longer phase blocks, more susceptible to switch errors, may appear to have worse accuracy despite having large stretches of correctly phased blocks. However, this metric remains flawed and does not behave predictably. Some possible replacement metrics would be to report the mean length of unbroken phase blocks, or the minimal unbroken phased block length to cover 90% of the SNPs.

The accuracy, also described as the Reconstruction Rate or Hamming distance measures how accurate the phasing is globally. Accuracy can be defined in two forms. The first is the prediction accuracy, which at 99% can state that for every 100 SNP predictions it makes, on average 1 SNP will be in the wrong phase. The second is the reconstruction accuracy, which at 99% states that for every 100 SNPs in the genome, on average 1 SNP will be in the wrong phase or not phased. The latter is more stringent by taking the missing rate into account. In both cases, the accuracy metric gives an important notion of how accurate the predictions are, making it a crucial performance metric to evaluate. By contrast with the MEC and SWER, accuracy metrics provide users and developers of polyploid phasing methods with a clear indication of how closely they came to reconstructing the original haplotypes. It also provides them

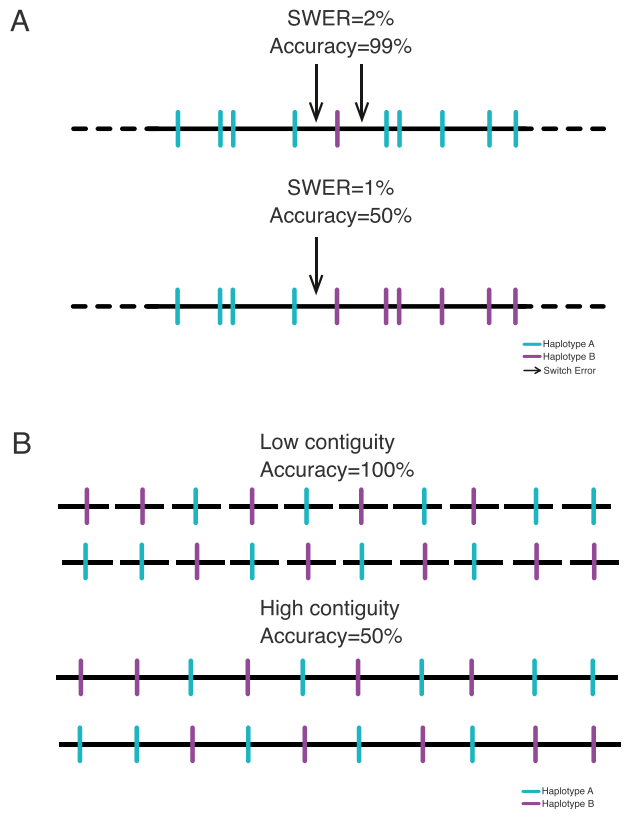

Fig. 6. Behavior of the SWER and contiguity performance metrics.
A We illustrate the unpredictable nature of the SWitch Error Rate (SWER) metric with two examples. In both cases we suppose we have a haplotype prediction of 100 variants. In the first, top case, two consecutive switch errors lead to a 2% SWER, but due to being consecutive the accuracy is at a very high 99%. In the second, lower case, there is only one switch error, giving a better SWER score of 1%, however the accuracy is reduced by half to 50% due to it occurring in the middle of the prediction. This behavior of the SWER metric makes it unpredictable and unreliable. B We illustrate the importance of contiguity to the interpretation of accuracy results with two examples. In both cases we show haplotype predictions for a diploid sequence. In the first, low contiguity example, we illustrate how it can be trivial to obtain extremely accurate predictions if they are sufficiently fragmented. Through the second, high contiguity example, we show how the accuracy of the previous example could dramatically decrease by increasing contiguity.

with the information of how reliable the results are. The expectation when phasing using a method which reliably phases similar samples with 90% accuracy is that only around 1 in 10 SNPs won't be in the correct phase. There is no straight-forward or useful expectation when phasing using a method which reliably phases similar samples with a SWER of 0.1% or an MEC score of 2000.

Accuracy on its own, however, is not a sufficient marker of how informative a phasing result is. Without an indication of how contiguous the results are, the accuracy metric (like the MEC or SWER) can also be highly misleading (Fig. 6B). It is trivial to obtain very good results by making highly fragmented predictions which avoid making any predictions about distant SNPs, therefore a good haplotype prediction must be both accurate and contiguous. However, the definition of contiguity is not straight-forward. Definitions based on the number of phased blocks per chromosome can be used to compare methods to each other, but due to the variability of genome sizes they do not provide an intuitive understanding of how good the phasing is. Taking inspiration from the metrics used to assess the contiguity of genome assemblies, contiguity could be defined as the minimum number or length of haplotype blocks to cover 50% or 90% of the SNPs. This is done by some methods, such as Hap10 which determined the N50 haplotype block length [30].

Another option is to represent contiguity with the following formula, where the number of haplotigs is the number of predictions made, and the number of haplotypes is the true number of distinct sequences in the sample (a triploid with four chromosomes would therefore have a total of twelve haplotypes):

Contiguity = 1/(Number of haplotigs/Number of haplotypes).

With this definition of contiguity, in the optimal case that there are exactly as many predictions as there are haplotypes the score will be exactly 1, and more fragmented predictions will have lower scores. However, this definition would necessarily be highly organism- and technology-dependent. Highly different standards of what constitutes a good contiguity should be expected when comparing performance on organisms with different genome sizes due to the difficulty of phasing long sequences, or when comparing short and long read methods due to the ability of long reads to phase distant SNPs.

These metrics are all applicable when the ground truth is known, which is the case for simulated datasets. However, it is less straightforward to evaluate the performance of these methods with real polyploid data due the absence of a ground truth. A few proxies have been developed to tackle this problem. We have already discussed the MEC metric, which is one of the main metrics used to evaluate performance on real polyploids. Ranbow [37] phases the sweet potato, *Ipomoea batatas*, and uses long, accurate Roche 454 reads to validate its haplotype predictions. WhatsHap Polyphase and nPhase both phase the autotetraploid potato plant, *S. tuberosum*, and show qualitatively that its genes appear well-phased [32,39].

In their paper [40], Motazedi et al. develop haplosim, a simulation pipeline which can generate simulated haplotypes and associated reads. This tool has been used by several polyploid phasing methods for their validation steps, such as Hap10 [30] and Ranbow [37]. However, there is no widely used benchmarking dataset which can systematically be compared against, and haplosim does not appear to have been updated in the past three years to reflect the significant improvements in quality achieved in long read sequencing methods. A well-maintained gold standard benchmark would be of benefit to the field of polyploid phasing. It would be interesting for such a resource to carefully consider the performance metrics to evaluate, the diversity of read sequencing methods and the effects of variable ploidy, genome architecture, heterozygosity level, genome size, structural variation, indels, polyallelic sites and local variations in the number of haplotypes.

One prevalent set of metrics which we do not discuss here is that of the technical, computational aspects of the methods used, such as the total amount of memory and computational time used by the algorithm.

5. Call for a community benchmarking project

Alignment-based polyploid phasing algorithms have significantly evolved since their inception. We consider the graph partitioning and cluster building strategies to be the most promising for future methods, in particular for the relative ease with which long reads can be leveraged. More nuanced understanding of the wide variety in the genome architectures of polyploids will have to be considered and taken into account in the design of future strategies. Additionally, it will be important not to ignore indels and polyallelic sites, and to seek to obtain accurate and contiguous phasing results.

We believe that the polyploid phasing field would greatly benefit from the development of a common gold standard dataset of simulated, partially simulated, or real polyploids, with carefully selected performance metrics. Systematic benchmarks against the same well-designed datasets would be beneficial in several ways by reducing the effort required to demonstrate the performance of a new phasing method and standardizing the metrics being used, allowing a user to more easily compare two polyploid phasing methods. It would also help show the conditions under which some algorithms perform highly, or meet their limitations.

In the interest of furthering this goal, we developed a toolkit to

benchmark alignment-based polyploid phasing strategies and tested it on a previously described dataset. We hope that this can serve as a starting point for the community to develop into a robust gold standard benchmarking process and associated datasets.

This Phasing Toolkit is a series of scripts which perform all of the necessary steps for polyploid benchmarking (Fig. 7). It is an open source project which welcomes contributions and strives to become a useful resource first for benchmarking but also potentially for real use cases of phasing tools and subsequent analysis of the results. We will present here the current state of the toolkit and how we have used it to generate an initial example benchmark of three tools on one dataset: flopp, nPhase and WhatsHap polyphase.

The first step of the benchmarking process is to obtain the data we will be testing different phasing methods on. For simplicity, we reused the virtual polyploid dataset described in the nPhase paper [39]. We provide the downloadData.py script with the names and NCBI accession codes of the sequencing datasets we want to download. In this case, we provide the accession codes for the short and long read sequencing data of four strains of *S. cerevisiae* which are either haploid or homozygous diploid. We also need to download the reference sequence of *S. cerevisiae* by providing the downloadData.py script with its taxid and taxonomic group according to the ncbi (559,292, fungi).

The downloadData.py script uses sra-downloader (https://github.com/s-andrews/sradownloader) to download sequencing data and ncbi-genome-download (https://github.com/kblin/ncbi-genome-download) to download reference sequences.

In the second step, we launch the processReads.py script and the short reads are mapped to the reference with bwa-mem [41], sorted with samtools [42] and variant called using GATK [43]. This provides us with a ground truth dataset which will later be used in the performance evaluation.

For the third step, we use the hybridGenerator.py script to generate virtual polyploids, in this case a 2n, 3n and 4n dataset of short and long reads. These datasets are obtained by merging the reads of individual samples and can be set to a custom coverage level (set to 20× in our example).

We can then run the processReads.py script to process the short reads as described before to obtain variant calls for the short reads. In order to evaluate the effect of heterozygosity rate on the performance of phasing tools, we can also automatically subset these variants to obtain specific heterozygosity rates. In our example we subset the variants to 1%, 0.5%, 0.1% and 0.05% heterozygosity rate. We also indicate that we want to obtain variant calling results with and without indels, to be able to evaluate how well indels are phased. Finally, this script will also process the long reads by mapping them to the reference using NGM-LR [44].

We now have a dataset of a 2n, 3n and 4n sample, each at 1%, 0.5%, 0.1% and 0.05% heterozygosity rate, with and without indels, for a total of 3*4*2 = 24 datasets ready to be phased, and the corresponding ground truth. We then run the phaseToolRunner.py script on each dataset and select the tools we want to test. For this initial release of the Phasing Toolkit we have only implemented flopp, nPhase and Whatshap polyphase, which we all test on our benchmarking dataset. The phaseToolRunner.py script will keep track of runtime and memory usage for each tool and phase each dataset with every tool selected.

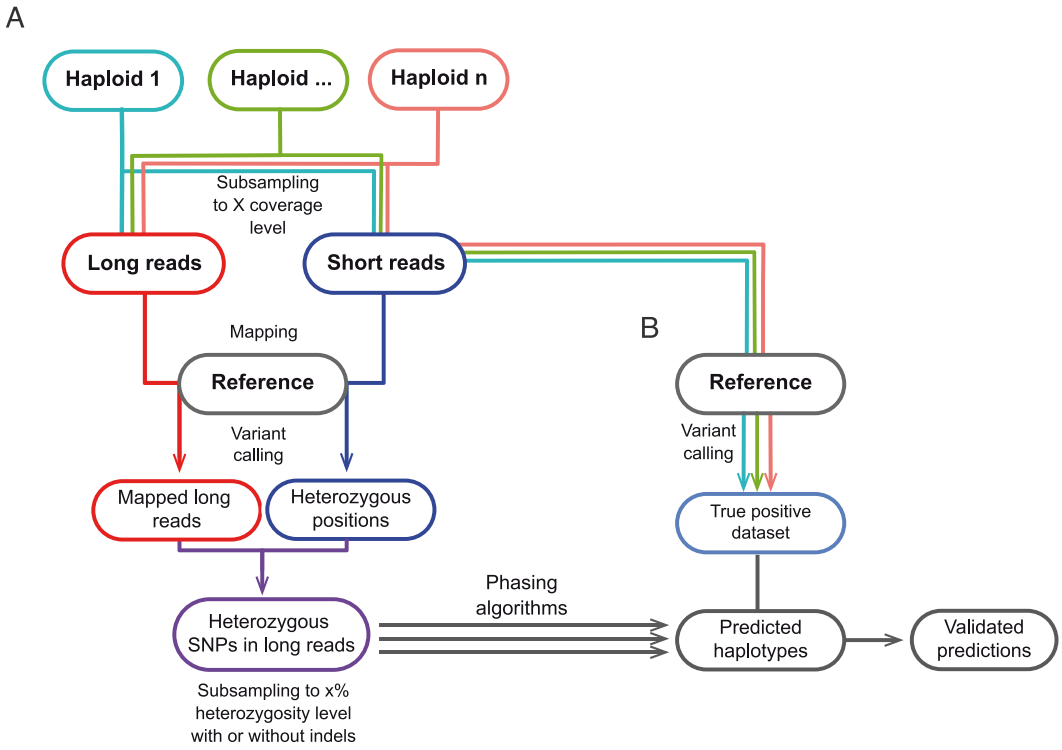

Fig. 7. Phasing toolkit benchmarking strategy.
The benchmarking strategy we describe here consists in obtaining long and short read data of individual haploids of the same species and using them to construct virtual polyploids which, once phased, will be compared to the ground truth obtained from the original samples.
A We generate a virtual polyploid's long and short read datasets by combining the long and short read datasets of n haploid individuals. At this step we can subsample the reads to reach a coverage level of X to evaluate their effects on phasing quality. We then map and variant call these reads. Once variant called, we can subsample the variants to a heterozygosity level of x% to evaluate its effect on phasing quality and choose to include or exclude indels for the same reason. The final dataset is then phased using the available phasing algorithms.
B In parallel, we use the accurate short read sequencing data of the original individuals to generate a ground truth dataset by mapping and variant calling to the same reference. We can then compare this ground truth to the predictions of different phasing algorithms obtained in A, and calculate various performance metrics in order to then evaluate and compare phasing tools.

Finally, the results are analyzed by the accuracyCalculator.py script which will output, for each test, the true positive rate, false positive rate, completeness, number of haplotigs, and contiguity.

The true positive rate is the number of correctly phased elements divided by the best possible score (all elements phased perfectly). The false positive rate is the rate of incorrectly phased elements. Completeness is the number of phased elements divided by the total number of elements to phase. Contiguity is 1/(Number of haplotigs/Number of haplotypes). We expect that the Phasing Toolkit will evolve to provide different, more informative and precise performance metrics.

With this initial benchmarking test, we were able to show that on this dataset, flopp performs very well, demonstrating that for higher heterozygosity rates it is capable of delivering very high accuracy and perfect contiguity, while using very little resources. Deeper analysis of the results of this initial benchmark would warrant a thorough and detailed analysis which falls outside the scope of this review, however the full results, including runtime and memory usage, are available in Supplementary Table S1.

The Phasing Toolkit is available on github (https://github.com/OmarOakheart/Phasing-Toolkit).

6. Perspectives

We look forward to more extensive benchmarking being made convenient and straight-forward with the Phasing Toolkit and granting us a better understanding of how flopp and other tools perform on different datasets, such as individuals with more complex genomic architectures such as aneuploidy, large regions of highly similar sequences between haplotypes, or simply much larger genomes. We believe that indels are an important and still overlooked source of heterozygosity with non-negligible genetic impacts and anticipate their explicit handling by future polyploid phasing methods. Finally, we hope that the Phasing Toolkit will be a useful platform for the community to discuss, develop and implement gold standard polyploid phasing benchmarking datasets, performance metrics, and generate actionable information on the strengths and limitations of available methods.

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ygeno.2022.110369.

Acknowledgements

We thank Detlef Weigel and Ken Wolfe for helpful comments and suggestions. This work was supported by an European Research Council (ERC) Consolidator grant (772505). J.S. is a fellow of the University of Strasbourg Institute for Advanced Study (USIAS) and a member of the Institut Universitaire de France.

References

[1] J.O. Kitzman, et al., Haplotype-resolved genome sequencing of a Gujarati Indian individual, Nat. Biotechnol. 29 (2011) 59–63.
[2] M.J. Roach, et al., Population sequencing reveals clonal diversity and ancestral inbreeding in the grapevine cultivar chardonnay, PLoS Genet. 14 (2018), e1007807.
[3] J. Yang, et al., Incomplete dominance of deleterious alleles contributes substantially to trait variation and heterosis in maize, PLoS Genet. 13 (2017), e1007019.
[4] J.A. Benitez, S. Cheng, Q. Deng, Revealing allele-specific gene expression by single-cell transcriptomics, Int. J. Biochem. Cell Biol. 90 (2017) 155–160.
[5] J.S. Sanjak, A.D. Long, K.R. Thornton, A model of compound heterozygous, loss-of-function alleles is broadly consistent with observations from complex-disease GWAS datasets, PLoS Genet. 13 (2017), e1006573.
[6] K. Hamazaki, H. Iwata, RAINBOW: haplotype-based genome-wide association study using a novel SNP-set method, PLoS Comput. Biol. 16 (2020), e1007663.
[7] N.D. Wagner, L. He, E. Hörandl, Phylogenomic relationships and evolution of Polyploid Salix species revealed by RAD sequencing data, Front. Plant Sci. 11 (2020).
[8] J.S. Eriksson, et al., Allele phasing is critical to revealing a shared allopolyploid origin of *Medicago arborea* and M. strasseri (Fabaceae), BMC Evol. Biol. 18 (2018) 9.
[9] J.C. Fay, et al., A polyploid admixed origin of beer yeasts derived from European and Asian wine populations, PLoS Biol. 17 (2019), e3000147.
[10] R.-N. Zhou, Z.-M. Hu, The development of chromosome microdissection and microcloning technique and its applications in genomic research, Curr. Genomics 8 (2007) 67–72.
[11] X. Zhang, S. Zhang, Q. Zhao, R. Ming, H. Tang, Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data, Nat. Plants 5 (2019) 833–845.
[12] J. Neigenfind, et al., Haplotype inference from unphased SNP data in heterozygous polyploids based on SAT, BMC Genomics 9 (2008) 356.
[13] S.-Y. Su, J. White, D.J. Balding, L.J. Coin, Inference of haplotypic phase and missing genotypes in polyploid organisms and variable copy number genomic regions, BMC Bioinform. 9 (2008) 513.
[14] O. Serang, M. Mollinari, A.A.F. Garcia, Efficient exact maximum a posteriori computation for Bayesian SNP genotyping in polyploids, PLoS One 7 (2012), e30906.
[15] J. Shen, et al., SHEsisPlus, a toolset for genetic studies on polyploid species, Sci. Rep. 6 (2016) 24095.
[16] D. He, S. Saha, R. Finkers, L. Parida, Efficient algorithms for polyploid haplotype phasing, BMC Genomics 19 (2018) 110.
[17] E. Motazedi, et al., TriPoly: haplotype estimation for polyploids using sequencing data of related individuals, Bioinformatics 34 (2018) 3864–3872.
[18] E. Motazedi, C. Maliepaard, R. Finkers, R. Visser, D. de Ridder, Family-based haplotype estimation and allele dosage correction for polyploids using short sequence reads, Front. Genet. 0 (2019).
[19] M. Mollinari, A.A.F. Garcia, Linkage analysis and haplotype phasing in experimental autopolyploid populations with high ploidy level using hidden Markov models, G3 GenesGenomesGenetics 9 (2019) 3297–3314.
[20] E. Berger, D. Yorukoglu, J. Peng, B. Berger, HapTree: a novel Bayesian framework for single individual Polyplotyping using NGS data, PLoS Comput. Biol. 10 (2014), e1003502.
[21] J. Wu, X. Chen, X. Li, Haplotyping a single triploid individual based on genetic algorithm, Biomed. Mater. Eng. 24 (2014) 3753–3762.
[22] S. Das, H. Vikalo, SDhaP: haplotype assembly for diploids and polyploids via semi-definite programming, BMC Genomics 16 (2015) 260.
[23] C. Cai, S. Sanghavi, H. Vikalo, Structured low-rank matrix factorization for haplotype assembly, IEEE J. Sel. Top. Signal Process. 10 (2016) 647–657.
[24] Z. Puljiz, H. Vikalo, Decoding genetic variations: communications-inspired haplotype assembly, IEEE/ACM Trans. Comput. Biol. Bioinform. 13 (2016) 518–530.
[25] A. Hashemi, B. Zhu, H. Vikalo, Sparse tensor decomposition for haplotype assembly of diploids and Polyploids, BMC Genomics 19 (2018) 191.
[26] J. Wu, Q. Zhang, A fast and accurate enumeration-based algorithm for haplotyping a triploid individual, Algorithms Mol. Biol. 13 (2018) 10.
[27] E. Siragusa, N. Haiminen, R. Finkers, R. Visser, L. Parida, Haplotype assembly of autotetraploid potato using integer linear programing, Bioinformatics 35 (2019) 3279–3286.
[28] S. Mazrouee, W. HapColor Wang, A graph coloring framework for polyploidy phasing, in: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2015, pp. 105–108, https://doi.org/10.1109/BIBM.2015.7359663.
[29] S. Mazrouee, W. Wang, PolyCluster: minimum fragment disagreement clustering for polyploid phasing, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (2020) 264–277.
[30] S. Majidian, M.H. Kahaei, D. de Ridder, Hap10: reconstructing accurate and long polyploid haplotypes using linked reads, BMC Bioinform. 21 (2020) 253.
[31] A. Sankararaman, H. Vikalo, F. Baccelli, ComHapDet: a spatial community detection algorithm for haplotype assembly, BMC Genomics 21 (2020) 586.
[32] S.D. Schrinner, et al., Haplotype threading: accurate polyploid phasing from long reads, Genome Biol. 21 (2020) 252.
[33] J. Shaw, Y.W. Yu, Flopp: extremely fast Long-read Polyploid haplotype phasing by uniform tree partitioning, J. Comput. Biol. 29 (2022) 195–211.
[34] D. Aguiar, S. Istrail, Haplotype assembly in polyploid genomes and identical by descent shared tracts, Bioinformatics 29 (2013) i352–i360.
[35] M.H. Olyaee, A. Khanteymoori, K. Khalifeh, A chaotic viewpoint-based approach to solve haplotype assembly using hypergraph model, PLoS One 15 (2020), e0241291.
[36] M. Xie, Q. Wu, J. Wang, T. Jiang, H-PoP and H-PoPG: heuristic partitioning algorithms for single individual haplotyping of polyploids, Bioinformatics 32 (2016) 3735–3744.
[37] M.-H. Moeinzadeh, et al., Ranbow: a fast and accurate method for polyploid haplotype reconstruction, PLoS Comput. Biol. 16 (2020), e1007843.
[38] Chaisson, M. J., Mukherjee, S., Kannan, S. & Eichler, E. E. Resolving multicopy duplications de novo using polyploid phasing. in Research in Computational Molecular Biology (ed. Sahinalp, S. C.) 117–133 (Springer International Publishing, 2017). doi:https://doi.org/10.1007/978-3-319-56970-3_8.
[39] O. Abou Saada, A. Tsouris, C. Eberlein, A. Friedrich, J. Schacherer, nPhase: an accurate and contiguous phasing method for polyploids, Genome Biol. 22 (2021) 126.
[40] E. Motazedi, R. Finkers, C. Maliepaard, D. de Ridder, Exploiting next-generation sequencing to solve the haplotyping puzzle in polyploids: a simulation study, Brief. Bioinform. 19 (2018) 387–403.
[41] H. Li, Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM, ArXiv13033997 Q-Bio (2013), https://doi.org/10.48550/arXiv.1303.3997.

[42] P. Danecek, et al., Twelve years of SAMtools and BCFtools, GigaScience 10 (2021) giab008.
[43] R. Poplin, et al., Scaling Accurate Genetic Variant Discovery to Tens of Thousands of Samples 201178, 2018, https://doi.org/10.1101/201178.
[44] F.J. Sedlazeck, et al., Accurate detection of complex structural variations using single-molecule sequencing, Nat. Methods 15 (2018) 461–468.