PLOS ONE

RESEARCH ARTICLE

A chaotic viewpoint-based approach to solve haplotype assembly using hypergraph model

Mohammad Hossein Olyaee[1], Alireza Khanteymoori[2]*, Khosrow Khalifeh[3,4]

1 Faculty of Engineering, Department of Computer Engineering, University of Gonabad, Gonabad, Iran, **2** Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg im Breisgau, Germany, **3** Department of Biology, Faculty of Sciences, University of Zanjan, Zanjan, Iran, **4** Department of Biotechnology, Research Institute of Modern Biological Techniques, University of Zanjan, Zanjan, Iran

* khanteymoori@gmail.com

Citation: Olyaee MH, Khanteymoori A, Khalifeh K (2020) A chaotic viewpoint-based approach to solve haplotype assembly using hypergraph model. PLoS ONE 15(10): e0241291. https://doi.org/10.1371/journal.pone.0241291

Editor: Zechen Chong, University of Alabama at Birmingham, UNITED STATES

Received: May 3, 2020

Accepted: October 12, 2020

Published: October 29, 2020

Data Availability Statement: The Geraci's dataset is available via email (contact via filippo.geraci@iit.cnr.it). The real dataset is available for download (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/). The source code is available from GitHub (https://github.com/mholyaee/HRCH).

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Decreasing the cost of high-throughput DNA sequencing technologies, provides a huge amount of data that enables researchers to determine haplotypes for diploid and polyploid organisms. Although various methods have been developed to reconstruct haplotypes in diploid form, their accuracy is still a challenging task. Also, most of the current methods cannot be applied to polyploid form. In this paper, an iterative method is proposed, which employs hypergraph to reconstruct haplotype. The proposed method by utilizing chaotic viewpoint can enhance the obtained haplotypes. For this purpose, a haplotype set was randomly generated as an initial estimate, and its consistency with the input fragments was described by constructing a weighted hypergraph. Partitioning the hypergraph specifies those positions in the haplotype set that need to be corrected. This procedure is repeated until no further improvement could be achieved. Each element of the finalized haplotype set is mapped to a line by chaos game representation, and a coordinate series is defined based on the position of mapped points. Then, some positions with low qualities can be assessed by applying a local projection. Experimental results on both simulated and real datasets demonstrate that this method outperforms most other approaches, and is promising to perform the haplotype assembly.

Introduction

Improving the high-throughput DNA sequencing technologies dramatically decreased the costs of genome sequencing methods. This achievement help researchers to understand the variation of individual's genomic data and pave the way toward individualized strategies for diagnostic or therapeutic decision-making [1]. The most frequent type of genetic variation is the single nucleotide polymorphisms (SNPs). Each SNP is just a mutation over similar distinctive positions on the DNA sequences of homologous pair of chromosomes in an individual, and among the corresponding DNA sequences of the whole population. Similarly, the term "allele" refers to different forms of a gene at one loci. Accordingly, four different alleles are possible for a given SNP site. Nonetheless, most SNPs are bi-allelic containing only two kinds of

alleles, which can be simply denoted by '0' and '1' [2]. Each SNP contains valuable information about genomic alternations. Experimental studies revealed that SNPs have been clustered across the human genome and are not randomly distributed [3]. In line with this assumption, linkage disequilibrium (LD), demonstrates that there are correlations and spatial dependencies among neighboring SNPs. Different SNPs on the string of DNA is known as a haplotype. In other words, a haplotype could be considered as the combinations of marker alleles which are positioned closely together on the same strand of DNA, and tend to be inherited together from parents to offspring [4]. It has been shown that some diseases such as sickle-cell anemia [5], cystic fibrosis [6] and hemochromatosis [7] are more common in specific ethnic populations due to unique genetic mutations in their genomes; but they are rarely found in others. There are also reports indicating that different populations may have various responses to drugs [8–10]. These findings demonstrate that haplotypes in human genomics data could be a useful and informative tool in mapping genes that are involves in representative diseases, as well as personalized medicine [11]. Haplotypes can also be used to investigate the pattern of inheritance over evolution, human migration, and the genetically aspects of populations [12–14]. Genetic association analysis for gene mapping can also be improved by haplotype analysis [15]. Also, it is possible to detect errors and missing sequencing data in experimental sequencing of DNA sequences using the information of haplotypes [16].

It is worth mentioning that the experimental analysis of haplotypes is labor-intensive and expensive. Moreover, it can be used only for constructing local haplotypes. In other words, human haplotypes are provided as sequencing reads or fragments. It is a vital task to obtain haplotype information from the numerous fragments due to its profound impacts on different aspects of medicine and molecular biology [15, 17–19]. However, the detection of genetic variations has critical limitations compared with the molecular approaches. According to the type of input data, the existing methods of haplotype reconstruction are divided into two main categories, including single individual haplotyping (SIH) and haplotype inference. SIH methods receive several fragments that have been sequenced from a given chromosome. It is to be noted that most of the fragments contain gaps, and are usually disrupted by noise. To cope with these problems, the input fragments are clustered based on their similarities. Then, the haplotypes can be reconstructed using the center of each cluster [4]. The haplotype inference methods receive genotype information of several individuals as input data and infer their related haplotype sequences [20]. It is worth noting that each genotype represents a combination of haplotypes on the homologous chromosomes.

With increasing the size of data, a growing number of researchers have tried to solve haplotype assembly problem. Moreover, several computational models, including minimum fragment removal (MFR), minimum error correction (MEC), minimum SNP removal (MSR), and the longest haplotype reconstruction (LHR), have been developed to cope with the SIH problem. The MEC is one of the most popular and successful algorithms compared with the models as mentioned above [4, 21–28]. This model attempts to cluster the input fragments, such that all the fragments belonging to a specified cluster to be compatible. Otherwise, they will be compatible by applying the minimum alternations. The current approaches can be divided into exact and heuristic methods. Since finding the optimal minimum error correction is NP-Hard, the exact approaches have exponential complexity [21]. Among exact solutions, WhatsHap [29] is regarded as a pioneering method, which is dynamic programming-based and utilizes a weighted variant of the MEC. The experimental results demonstrate that it can process long reads at coverage up to 20×. In [30], the authors proposed a parallel version of WhatsHap which is able to process higher coverages up to 25×. AROHap [24] is a recently published evolutionary-based method that exploits the asexual reproduction optimization algorithm to solve the SIH problem. In this method, the fitness function is designed based on

the MEC model. In [26], a heuristic method, namely, Fasthap was developed, where it makes a weighted fuzzy conflict graph based on the MEC model. Furthermore, the constructed graph is used to cluster the input fragments. Fuzzy C-means (FCM) approach has been applied in [25] to enhance the performance of the proposed method in clustering the fragments. However, this method obtains low performance in dealing with noisy fragments. Some popular methods, including MCMC [31], HapCUT [27], and HapCUT2 [32], have differently construct the graph. These methods start with a set of arbitrary sequences as initial haplotypes, and improve it step by step concerning the input fragments. They make a similar weighted graph in their distinctive model. However, instead of fragments, SNPs are used as vertices of the graph. Each pair of SNPs is connected if they are covered by at least one input fragment. The weight of each edge determines the amount of consistency with their corresponding positions in the current haplotypes. Although this model efficiently determines the consistency of the current haplotype with the input fragments, the existing gaps and noise lead to a loss of accuracy in determining the weight of edges. In [33]. It has been proved that the hypergraph can precisely describe the distance of input fragments.

Although, various methods have been developed to solve the SIH problem, most of them can only be applied to diploid organisms, and fail to consider polyploid organisms. It should be noted that the haplotype reconstruction in polyploid type is more complicated than a diploid one. Suppose that P is the number of ploids, and m is the length of haplotype sequences. In this case, there are at least $2^{m-1}(P-1)^m$ different solutions for phasing the haplotypes [23]. Recently, several studies, such as [23, 34–36], have been conducted on the polyploid organism. Althap [23] and SCGD [36] are two recently developed methods based on matrix factorization to solve the SIH problem. H-PoP [34] is a heuristic method that divides the input fragments into P clusters. Therefore, the members of each cluster have the minimum distance with each other and are entirely far from the fragments of other clusters. Belief propagation (BP) [35] is another method addressing the SIH problem by mapping the MEC model to a decoding mechanism. It involves a message transmission in a noisy channel. In this context, it has been reported that the haplotype's blocks with proper lengths can exhibit chaotic behavior. This feature has been recently used to improve the reconstruction rate in the single individual haplotyping problem [37].

Considering the chaotic nature of haplotype sequences, in this paper, an iterative algorithm is proposed to reconstruct the haplotypes using the hypergraph model. The method includes two main steps. Firstly, an iterative mechanism is applied due to the SNP matrix to construct the haplotype set, and the consistency between SNPs is modeled based on the hypergraph. Then, the corrected parts of the haplotypes are determined by partitioning the hypergraph.

This step is followed by transforming the obtained haplotypes into a line using the chaos game representation, where a coordinate series is defined based on the position of the mapped points. Also, a local projection (LP) method is applied to refine the remaining ambiguous measures and increasing the quality of the reconstructed haplotypes.

The significant contributions of the proposed method are as follows:

- The similarity measurement between the input fragments can be described more accurately by utilizing the hypergraph model. Moreover, it helps to overcome challenges originated from the huge amount of gaps and sequencing errors.
- The quality score for each position of the reconstructed haplotypes can be calculated to predict the remaining error measures.
- The chaotic nature hypothesis is used to refine the reconstructed haplotypes. To this end, we only concentrate on the neighboring dependencies between SNPs.
- The proposed method could be applied effectively for both diploid and polyploid organisms.

The rest of the paper is organized as follows. Section 2 provides a brief review of the problem statement. In section 3, the proposed method is described in detail. Experimental results are presented in section 4. Finally, the conclusion is arrived at section 5.

Preliminaries and assumptions

The challenge of the SIH problem in the polyploid organisms includes the reconstruction of the whole set $H = \{h_1, h_2, \ldots, h_P\}$ containing P haplotype sequences. It is based on the available aligned input fragments. Similar to diploid case, the input fragments can be represented as a standard form. Let X be the SNP matrix in which each row corresponds to an input fragment, and each column indicates a specified SNP. In binary allelic haplotypes, it is assumed that $x_{ij} \in \{0,1,'-'\}$ indicating the obtained allele in a specified fragment f_i at SNP s_j. Also, each haplotype h_i ($i = 1,2, \ldots, P$) equals to $\{1,0\}^N$. In diploid case, there are some positions called homozygote sites in which h_{1k} equals to h_{2k}. On the other hand, the sites with different measures are called heterozygote positions. Homozygote sites are usually removed from the input matrix, as they do not provide useful information for the haplotype assembly problem. It is worth noting that the $'-'$ sign indicates missing information during the sequencing process. For two fragments which are originated from different haplotypes, it is expected that there are some dissimilarities between them. Several relations have been developed to describe the differences between the two fragments. Hamming distance (HD) is the most practical approach, which can be used to calculate the differences between two input fragments f_i and f_j as follows:

$$HD(f_i, f_j) = \sum\nolimits_{l=1} d(f_i[l], f_j[l]) \tag{1}$$

Where d is defined as follows:

$$d(x, y) = \begin{cases} 1 & x \neq y \text{ and, } x \neq '-' \text{and } y \neq '-' \\ 0 & \text{else} \end{cases} \tag{2}$$

In the case where the SNP matrix is error-free, two fragments that were sequenced from the same haplotype are compatible, as their distance equals to zero. On the other hand, in dealing with the noisy SNP matrix, for two arbitrary fragments f_i, f_j, it is not possible to simply interpret the dissimilarity between two fragments, as they can be originated from the existing noise or have been sequenced from different haplotypes. In the error-free case, the fragments can be clustered in P clusters, such that the members of each cluster are compatible with each other.

Fig 1 represents an example of the SIH problem in the ploidy level. The rows of matrix X indicate sequenced fragments, and the rows of matrix H contain the obtained haplotypes.

In diploid case, several models have been proposed to solve the SIH problem based on the input fragments.

Extending the models to solve the SIH problem in polyploidy form is a difficult task [38]. Recently, several MEC-based approaches have been developed to solve this problem. In this regard, the input fragments are organized in P clusters, and the haplotypes are considered as the centers of constructed clusters. In fact, each cluster involves the fragments which have the same provenance. The optimized result of the clustering algorithm can be obtained by minimizing the following Eq.:

$$MEC(X, H) = \sum_{i=1}^{P} \sum_{f \in C_i} HD(f, H_i) \tag{3}$$

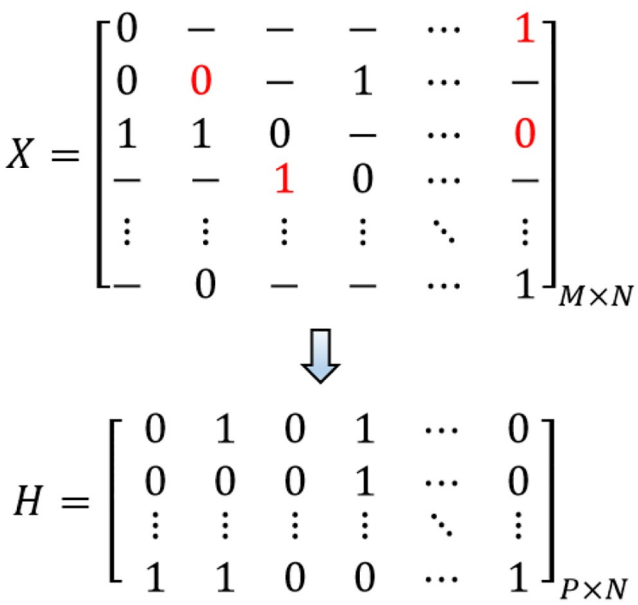

Fig 1. An example of SNP matrices *X* and *H* relevant to the resulting haplotypes. The red measures in *X* indicate sequencing errors. Each row of *H* demonstrates a specified haplotype sequence.

https://doi.org/10.1371/journal.pone.0241291.g001

In the optimal case, if the SNP matrix is error-free, then the MEC measurement equals zero, and each fragment f belonging to C_i is compatible with H_i. However, in dealing with the noisy SNP matrix, it is expected that some fragments to be in conflict with their corresponding haplotypes. It should be noted that finding the optimal MEC measure is an NP-hard problem. On the other hand, the huge amount of gaps in the input fragments does negatively affect the distance measurement between pairs of input fragments. Therefore, the current work aims to address these challenges by a better description of the similarity measurement between the input fragments. This was done by a heuristic method with a favorable runtime based on the hypergraph model.

The proposed method

This section presents a Haplotype Reconstruction approach based on the Chaotic viewpoint and Hypergraph model (HRCH). The proposed method is briefly described below.

(i) a set of haplotype sequences is randomly generated;(ii) the input fragments are assigned to the haplotype sequences based on their similarities;(iii) a weighted SNP hypergraph is built, using the similarity measure between haplotype sequences and the assigned input fragments; (iv) the constructed hypergraph is used to find a set called CutSet, containing the SNPs which should be modified. This procedure is repeated for a predefined number of iterations to minimize the MEC score. Next, by considering the existence of chaotic properties of haplotype sequences, the results are improved. A high-level overview of the method is demonstrated in Fig 2.

Data preprocessing

As described in the preliminaries sections, $X_{M\times N}$ is a matrix containing M reads with length N. It is essential to note that homozygote columns can be ignored in diploid cases. Removing the homozygote positions was performed as described by [33] such that the most frequent

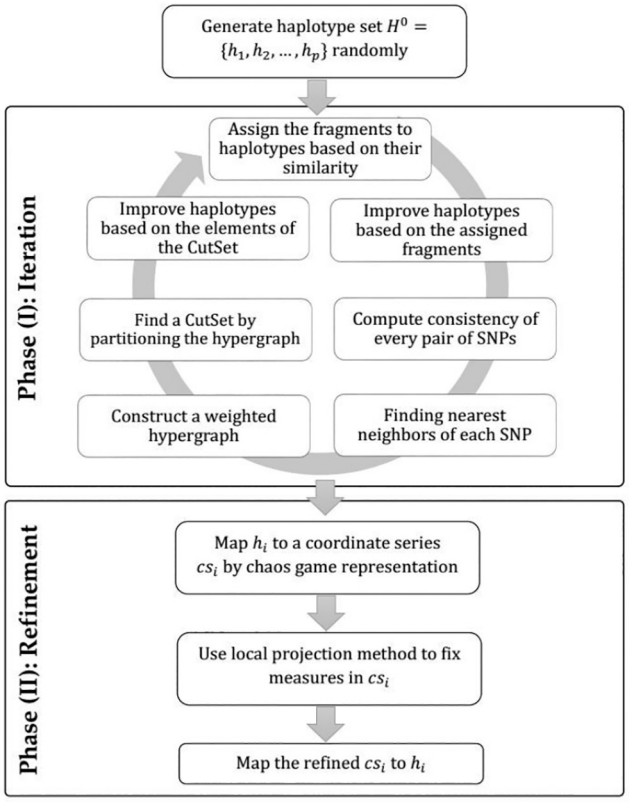

Fig 2. The workflow of proposed method.

https://doi.org/10.1371/journal.pone.0241291.g002

measure for each column could be found. If the frequency is higher than 0.8, the column is identified as a homozygote site. Thus, the output of this step is a matrix with M fragments and N' columns; where $N' \leq N$. Finally, $H^0 = \{h_1, h_2, \ldots, h_p\}$, as an initial set of haplotypes is randomly generated.

Pair-SNP consistency

Let $\bowtie$ be a binary operator which provides the concatenation of two variables. For example, if a and b are two variables with measures '0' and '1', respectively, $a \bowtie b$ equals to '01'. Given two variables, $c, d \in \{'00', '01', '10', '11'\}$ the operator $\oplus$ is defined as follows:

$$c \bigoplus \mathrm{d} = \begin{cases} -1 & \text{if } \mathrm{c} = \mathrm{d} \\ 1 & \text{if } \mathrm{c} \neq \mathrm{d} \end{cases} \tag{4}$$

Definition 1 (Pair-SNP consistency). Given matrix $X_{M\times N}$ involving the input fragments, pair-SNP consistency, ω_{ij} is defined between s_i and s_j as two arbitrary SNPs which are covered by f_k $(k = 1,2, \ldots, M)$ as follows.

$$\omega_{ij} = \frac{1}{T_{ij}} \sum\nolimits_{f_k \in cov(s_i, s_j)} [f_k(i) \bowtie f_k(j)] \bigoplus \left[\mathrm{h}_{\mathrm{c}(\mathrm{f_k})}(\mathrm{i}) \bowtie \mathrm{h}_{\mathrm{c}(\mathrm{f_k})}(\mathrm{j}) \right] \tag{5}$$

Where T_{ij} is the number of fragments covering both SNPs s_i and s_j. By applying this measure, ω_{ij} is normalized such that its value ranges between -1 and +1 (*i.e.*, $-1 \leq \omega_{ij} \leq +1$). Moreover,

$cov(s_i, s_j)$ includes the fragments which cover the SNPs s_i and s_j. Finally, as mentioned above, $c(f_k)$ identifies the origin of f_k.

The Pair-SNP consistency metric is used to evaluate the compatibility between each pair of SNPs with the current haplotype H^t. The intuition behind the Eq 5 is as follows. For given SNPs s_i and s_j, ω_{ij} describes the amount of similarity between the pair SNPs and their corresponding measures in H^t. This measure equals to -1 if the current haplotype is entirely identical with the covered fragments in columns i and j. On the other hand, it takes 1 if they are completely different in those columns. Otherwise, ω_{ij} equals to 0, when the SNPs are not covered by any fragment. It is noticeable that, for high measures of ω_{ij}, it is expected that the SNPs, s_i and s_j, are considered to belong to different clusters upon partitioning. The complexity of this step is $O(MN^2)$, where M and N are the number of fragments and SNPs, respectively.

Hypergraph construction

To construct the weighted hypergraph based on the achieved ω matrix, for each SNP s_i, its K nearest neighbors is found using the following Eq.:

$$KNN(s_i) = \{s_j | i \neq j, \omega_{ij} \leq \omega_{il}\} \tag{6}$$

Where l is index of the K^{th} nearest SNP of s_i. $KNN(s_i)$ is a set containing the index of K nearest neighbors of s_i. More specifically, each set represents the K SNPs, which have the most consistent relationship with s_i. In this case, each hyperedge can connect more than two vertices. Applying K nearest neighbors is a common approach to determine the hyperedges. However, it is necessary to specify the hyperedges more precisely due to the existing noise and sparsity of the SNP matrix. Therefore, the connectivity of vertices is defined by finding frequent itemsets. In other words, the hyperedges are determined as the shared K nearest neighbors, which can be defined as follows:

$$SKNN(E) = \cap_{e \in E} KNN(e) \tag{7}$$

In Eq 7, E contains several SNPs, and $SKNN(E)$ provides a set of SNPs which are shared between all nearest neighbors of E. If the number of shared KNNs is more than a predefined threshold, called minimum support count (sc), then E can be defined as a frequent itemset. In the proposed model, each frequent itemset is defined as a specified hyperedge e_i, and the number of shared KNN is assigned as its weight measure w_i. Among the existing methods, frequent pattern (FP)-growth [39] has been gaining much attention due to the ability to find frequent itemsets. FP-growth is a tree-based method which uses a depth-first strategy to mine frequent itemsets. Accordingly, the database is modeled as a prefix tree, and the depth-first search is recursively applied to generate all maximal frequent itemsets. The runtime of this algorithm increases linearly, and it depends on the number of SNPs [40].

Improving H^t by partitioning the hypergraph

As can be seen in Fig 3, in the constructed hypergraph, the SNPs correspond with vertices, and each hyperedge equals with an obtained frequent itemset. In other words, it contains a set of SNPs that has more consistency with the corresponding position in H^t. It is noteworthy that hyperedges with higher weights indicate the higher similarity between the constituent's SNPs and their relevant positions in H^t. The vertices can be divided into two clusters via partitioning the hypergraph. The objective of the partitioning is to minimize the sum of the weights of the hyperedges located between the clusters. To this end, hmetis as a popular algorithm was used. The algorithm includes three steps: (i) a number of small hypergraphs in several layers are

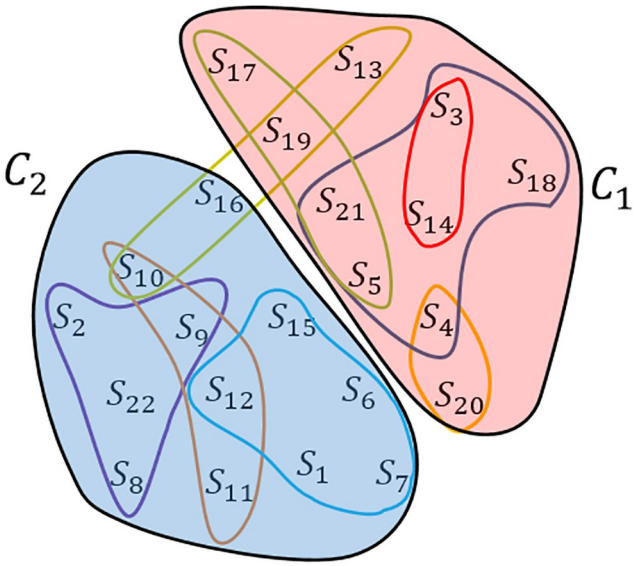

Fig 3. An example of constructing and partitioning the hypergraph. S_i corresponds with the i[th] SNP, and the curves demonstrate the hyperedges. C_1 and C_2 denote the clusters which are obtained by hypergraph partitioning.

https://doi.org/10.1371/journal.pone.0241291.g003

built; (ii) the hypergraph in the lowest level is partitioned; (iii) the resulted partitions are extended to the upper levels through a successive mapping.

The computational complexity of the algorithm is $O(|E|)$, where E is the set of hyperedges. Suppose that C_1 and C_2 are two clusters obtained by the hmetis algorithm. As can be seen in Fig 4, H_1^t and H_2^t are partial haplotypes originating from the resulting clusters.

In the diploid case, as can be seen in Fig 5 like the HapCUT method, improving H^t is performed as follows. First, C_1 or C_2 is selected as a CutSet. Next, H^{t+1} is obtained from H^t by flipping the measures of the SNPs in the CutSet.

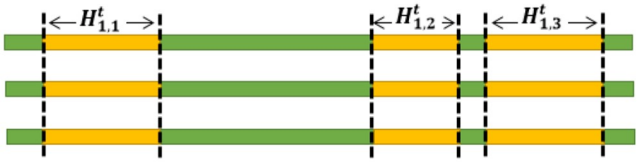

Fig 4. Partitioning H^t of a three ploid genome. The yellow parts indicate H_1^t and the green parts demonstrate H_2^t. It must be pointed out that $H_1^t = H_{1,1}^t \cup H_{1,2}^t \cup H_{1,3}^t$.

https://doi.org/10.1371/journal.pone.0241291.g004

$$H^t = \begin{bmatrix} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & \dots & S_{18} & S_{19} & S_{20} & S_{21} & S_{22} \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\Downarrow$$

$$CutSet = \{3,4,5,13,14,17,18,19,20,21\}$$

$$\Downarrow$$

$$H^{t+1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Fig 5. An example of updating the current haplotype based on the partitioning of the hypergraph.

https://doi.org/10.1371/journal.pone.0241291.g005

Algorithm 1. HRCH Improving H^t

Require: H_1^t, H_2^t, and G as genotype
Select H_1^t or H_2^t as $CutSet$ randomly
FOR each combination of $CutSet$ elements **DO**
  Let H_{new}^t is the resulted haplotype
  IF MEC(H_{new}^t) < MEC(H^t) **THEN**
    $H^t = H_{new}^t$
  END-IF
Let nc is a percent of SNPs involved in $CutSet$
Let $GenSet \subset G$ involves genotypes related to $CutSection$
$counter = 0$
WHILE ($counter \leq nc \times size\ of\ CutSet$) **DO**
  Let G_i and G_j are two genotypes selected from $GenSet$ randomly
  Let $GComb$ is a combination of G_i and G_j
  Let $tempH$ is resulted by replacing $GComb$ in H^t
  IF MEC($tempH$)<MEC(H^t) **THEN**
    $H^t = tempH$
  END-IF
  $counter = counter + 1$
END-WHILE

Fig 6. The algorithm of improving H^t.

https://doi.org/10.1371/journal.pone.0241291.g006

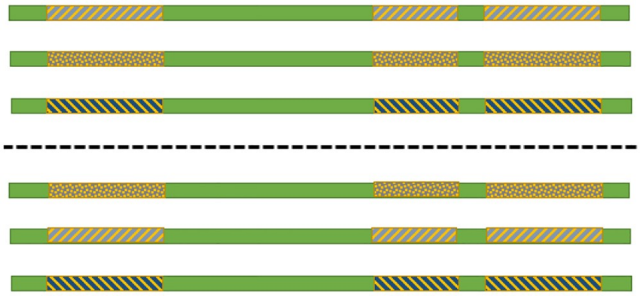

Fig 7. Two combinations of six possible combinations of the CutSet in three ploid form.

https://doi.org/10.1371/journal.pone.0241291.g007

For polyploid, improving H^t is accomplished based on the algorithm which is shown in Fig 6. In the first step, a partial haplotype (i.e., H_1^t or H_2^t) is randomly assigned to CutSet. This set involves some parts of H^t that should be corrected.

As shown in Fig 7, all combinations of the CutSet are evaluated to find a new set of haplotypes (H_{new}^t) with lower MEC score.

Moreover, in order to evaluate more allelic combinations of SNPs, for a predefined percent of SNPs belonging to the CutSet, in each time two arbitrary SNPs are nominated. Then one of its various genotype's combinations is randomly selected, and is replaced at corresponding positions in H^t. This step repeats for a predefined percent of SNPs.

Since H^t has randomly generated, in the early iterations, its MEC score is poor. Therefore, finding the hyperedges with lower weights is not a difficult task. But, by improving the quality of H^t and increasing the consistencies between SNPs, MEC measure will be decreased slowly.

Refinement of H^t

Computing confidence score. Upon performing the iterative procedure of the proposed method, the haplotype $H = \{h_1, h_2, \ldots, h_p\}$ will be obtained. It is possible to define a confidence

measure for each loci of the reconstructed haplotypes. For diploid case, we used the emission probability $P(X_j|h_j, R_j)$ that has been defined in [41], which is used to identify errors in the reconstructed haplotype. This measure is calculated for each position j as follows:

$$P(X_j|h_j, R_j) = \prod_{i:j\in PO(i)} p(x_{ij}|h_j, f_i) \tag{8}$$

Where h is a haplotype sequence belongs to H, $h_j \in \{0,1\}$ denotes an allele in position j, and $PO(i)$ contains the columns which have been covered by f_i as the i-th fragment. Furthermore, R_j is a set which includes fragments such as f_i covering a position j. Finally, $p(x_{ij} \mid h_j, f_i)$ is calculated as follows:

$$p(x_{ij}|h_j, f_i) = \begin{cases} Q_{ij} & if \ \ x_{ij} \neq h_j(f_i) \\ 1 - Q_{ij} & else \end{cases} \tag{9}$$

Where Q is an $M \times N$ matrix; for each element $x_{ij} \in X$, Q_{ij} includes the probability of sequencing error and h_j (f_i) as the j-th loci of the reconstructed haplotype is computed based on the following Eq.:

$$h_j(f_i) = \begin{cases} h_j & if \ f_i \in C_1 \\ \overline{h_j} & if \ f_i \in C_2 \end{cases} \tag{10}$$

Where C_1 and C_2 are the obtained clusters containing similar fragments that indicate the provenance of f_i. Eq (10) provides more information in each loci, and is based on the fact that $h_1 = \overline{h_2}$. Therefore, the confidence score could be calculated more precisely. On the other hand, there is no relationship between the haplotype sequences in the polyploid form. Hence, applying Eq (8) is not applicable. In this case, we used genotype information. Suppose that g_i is the genotype information in position i and H_i is the reconstructed measure in this position. The sorted measure of g_i and H_i are compared, and the position i will be selected for refinement if the two sets are not equivalent.

Applying chaos game representation. Chaos game representation (CGR) is a graphical tool which maps an arbitrary sequence to a 2-dimensional form.

This map is reversible and all the information of the sequence is preserved. Moreover, it depicts the hidden dependencies among the letters. CGR was initially introduced by Barnsley [42] to evaluate random sequences. Afterwards, Jeffrey [43] developed the method for visualizing genomic sequences. For this aim, according to the number of distinct letters constructing the input sequence, a regular polygon can be considered. For example since DNA sequences are constructed from four nucleotides 'a', 't', 'c', and 'g', a square with unit length is considered and each distinct letter is assigned to one vertex. Each letter of the given sequence is iteratively mapped as a point inside the square. The process is started by locating the first point half-way between the center of the square and the corner related to the occurrence of the first letter. The method continues such that the i-th point is placed half-way between the previous point and the vertex related to the i-th letter. Using this procedure, many attempts have been made with the purpose of extracting novel features from biological sequences by exploiting CGR [44–48].

Recently, CGR was used to reveal the chaotic properties of haplotypes [37]. Since haplotypes are represented in binary form, the achieved map will be a dotted line which its vertices are named by 0 and 1, respectively. In this step, in order to improve the reconstructed haplotypes, CGR is utilized as follows.

For loci's which their qualities are less than θ, as a predefined threshold, their measures may be disrupted by noise or missing information. Therefore, it is refined based on the existing dependencies between SNPs. For this purpose each $h_i \in H$ is mapped to a line by applying

CGR. The places assigned to each point construct a coordinate series, namely cs_i. The route of chaos helps to refine the ambiguous measures in the low-quality positions. To this end, the points in cs_i that are correspond with the alleles with low confidences, are shown by '-'. Then, the measure of ambiguous positions can be determined by applying a local projection (LP) method. After filling the removed measures based on the LP method, the refined coordinate series called $\widehat{cs}_i$, are transformed into the final haplotype known as $\widehat{h}_i$. It must be indicated that extracting the *cs* and applying the LP method are accomplished in linear time. The conversion is calculated according to Eq 11:

$$\widehat{h}_i(j) = \begin{cases} 0 & if\ \ h_i(j) = '-'\ \ and\ \ \widehat{cs}_i(j) \leq 0.5, \\ 1 & if\ \ h_i(j) = '-'\ \ and\ \ \widehat{cs}_i(j) > 0.5, \\ h_i(j) & Otherwise \end{cases} \tag{11}$$

Results

In the following section, the performance of the proposed method is compared with several state-of-the-art approaches in diploid and polyploid forms. The method was implemented in MATLAB, and all the results were obtained on a Windows 10 PC with 3.6 GHz CPU and 16 G Ram. The parameters of the algorithm are defined as $t = 100$, $k = 5$, $sc = 2$, and $nc = 20\%$. Reconstruction rate (RR) [4] as a conventional metric was used to evaluate the quality of the obtained haplotypes. In diploid case, RR is defined as follows:

$$RR = 1 - \frac{1}{2 \times N}\left(\min(HD(h_1, \widehat{h}_1) + HD(h_2, \widehat{h}_2), HD(h_1, \widehat{h}_2) + HD(h_2, \widehat{h}_1))\right) \tag{12}$$

Here, *HD* denotes hamming distance between h_i and $\widehat{h}_j$ which are the target and the reconstructed haplotype, respectively and $i, j = 1,2$. For polyploid case, this formula is written in the form of Eq 13:

$$RR = 1 - \frac{1}{N \times P}\left(\min_{\mathcal{M}} \sum_{i=1}^{P} \sum_{j=1}^{N} d(\mathcal{M}(\widehat{H})_{ij}, H_{ij})\right) \tag{13}$$

Where $\mathcal{M}$ is a one-to-one mapping from the set of reconstructed haplotypes to the set of target haplotypes.

Diploid case

The experiments have been carried out on two widely used and well-known datasets including Geraci's dataset [49] and a dataset from the 1000 genome project that are prime examples of the simulated and experimental datasets, respectively.

Simulated data. The Geraci's dataset involves three parameters: Coverage $c = \{3,5,8,10\}$, length of haplotypes $l = \{100,350,700\}$, and error rate $e = \{0,0.1,0.2,0.3\}$. For each combination of these parameters, there are 100 samples. The output of the proposed method was compared with a set of state-of-the-art and well-known methods including; SCGD [36], H-pop [34], ARO [24], HG [33], FCM [25], FastHap [26], DGS [50], SHR [51], MLF [52], HapCut [27], 2d [22], Fast [53], and SPH [54]. All of these methods were run with their default parameter settings. In accordance with the existing methods, the reconstruction rate (RR) was also used to assess the result of the current method. Tables 1–3 comparatively show the reconstruction rate of the proposed method with those described for haplotype blocks with length 100, 350, and 700. Note that in the last column of each table, the highest measures are boldfaced. Moreover,

Table 1. Average of reconstruction rate for haplotypes with length 100.

e	C	SCGD	H-pop	SPH	Fast	2d	Cut	MLF	SHR	DGS	Fasthap	FCM	HG	ARO	HRCH
0%	3	1.000	1.000	0.999	0.999	0.990	1.000	0.973	0.816	1.000	0.916	1.000	0.999	0.992	**1.000**
	5	0.999	1.000	1.000	0.999	0.997	1.000	0.992	0.861	1.000	0.953	1.000	1.000	1.000	0.999
	8	0.999	1.000	1.000	1.000	1.000	1.000	0.997	0.912	1.000	0.956	1.000	1.000	1.000	0.999
	10	1.000	1.000	1.000	1.000	1.000	1.000	0.998	0.944	1.000	1.000	1.000	1.000	1.000	0.997
10%	3	0.918	0.921	0.895	0.913	0.911	0.928	0.889	0.696	0.930	0.823	0.882	0.941	0.844	**0.957**
	5	0.944	0.919	0.967	0.964	0.951	0.920	0.969	0.738	0.985	0.917	0.948	0.989	0.922	0.987
	8	0.948	0.900	0.989	0.993	0.983	0.901	0.985	0.758	0.989	0.955	0.971	0.994	0.945	0.991
	10	0.959	0.892	0.990	0.998	0.988	0.892	0.995	0.762	0.997	0.926	0.972	0.997	0.920	0.995
20%	3	0.806	0.836	0.623	0.715	0.738	0.782	0.725	0.615	0.725	0.806	0.739	0.752	0.711	**0.851**
	5	0.825	0.865	0.799	0.797	0.793	0.838	0.836	0.655	0.813	0.834	0.772	0.899	0.736	**0.926**
	8	0.861	0.873	0.852	0.881	0.873	0.864	0.918	0.681	0.878	0.849	0.793	0.966	0.760	0.941
	10	0.886	0.878	0.865	0.915	0.894	0.871	0.938	0.699	0.917	0.899	0.835	0.981	0.788	0.956
30%	3	0.671	0.717	0.480	0.617	0.623	0.602	0.618	0.557	0.611	0.578	0.629	0.621	0.627	0.695
	5	0.676	0.784	0.637	0.639	0.640	0.629	0.653	0.599	0.647	0.711	0.648	0.698	0.638	**0.798**
	8	0.740	0.835	0.667	0.661	0.675	0.673	0.697	0.632	0.663	0.700	0.664	0.790	0.649	**0.861**
	10	0.798	0.855	0.676	0.675	0.678	0.709	0.715	0.632	0.688	0.732	0.675	0.856	0.653	**0.881**

https://doi.org/10.1371/journal.pone.0241291.t001

the second-highest measures are highlighted. The results, demonstrate that current method has an acceptable level of performance and outperforms in most of the cases.

The performance of the refinement phase has been considered in Table 4. Since evaluating the chaotic feature is limited to the long coordinate series, this phase can only be performed for sequences with length 700. For this purpose, the LP method is applied for each coordinate series with embedding dimensions (*em*) equal to 1 and 2, individually. It should be noted that the first column demonstrates the quality of the obtained haplotypes after terminating the first phase. The next two columns involve the rate of reconstruction for em equals to 1 and 2,

Table 2. Average of reconstruction rate for haplotypes with length 350.

e	C	SCGD	H-pop	SPH	Fast	2d	Cut	MLF	SHR	DGS	Fasthap	FCM	HG	ARO	HRCH
0%	3	0.999	1.000	0.999	0.989	0.965	1.000	0.864	0.830	1.000	0.985	1.000	0.996	0.999	0.999
	5	0.999	1.000	1.000	0.999	0.993	1.000	0.929	0.829	1.000	0.983	1.000	0.997	1.000	0.999
	8	1.000	1.000	1.000	1.000	0.998	1.000	0.969	0.895	1.000	0.983	1.000	0.998	1.000	0.996
	10	1.000	1.000	1.000	1.000	0.999	1.000	0.981	0.878	1.000	0.998	1.000	1.000	1.000	0.999
10%	3	0.941	0.921	0.819	0.871	0.839	0.930	0.752	0.682	0.926	0.872	0.873	0.939	0.844	0.939
	5	0.945	0.912	0.959	0.945	0.913	0.913	0.858	0.724	0.978	0.927	0.919	0.979	0.892	**0.983**
	8	0.950	0.896	0.984	0.985	0.964	0.896	0.933	0.742	0.996	0.977	0.934	0.988	0.908	0.991
	10	0.952	0.889	0.984	0.995	0.978	0.888	0.962	0.728	0.998	0.947	0.935	0.995	0.910	0.994
20%	3	0.813	0.813	0.439	0.684	0.675	0.771	0.642	0.591	0.691	0.763	0.671	0.712	0.659	**0.813**
	5	0.817	0.860	0.729	0.746	0.728	0.831	0.728	0.632	0.769	0.811	0.719	0.905	0.691	0.897
	8	0.832	0.871	0.825	0.853	0.791	0.862	0.798	0.670	0.842	0.912	0.728	0.899	0.709	**0.922**
	10	0.838	0.873	0.855	0.877	0.817	0.867	0.831	0.668	0.878	0.923	0.733	0.907	0.719	**0.937**
30%	3	0.637	0.629	0.251	0.590	0.593	0.565	0.581	0.548	0.578	0.575	0.597	0.602	0.595	**0.640**
	5	0.661	0.744	0.578	0.602	0.606	0.582	0.606	0.557	0.609	0.720	0.614	0.632	0.609	0.737
	8	0.690	0.830	0.629	0.626	0.623	0.621	0.634	0.604	0.628	0.790	0.626	0.675	0.628	0.788
	10	0.700	0.850	0.638	0.644	0.634	0.664	0.641	0.619	0.641	0.833	0.631	0.742	0.635	0.821

https://doi.org/10.1371/journal.pone.0241291.t002

Table 3. Average of reconstruction rate for haplotypes with length 700.

e	C	SCGD	H-pop	SPH	Fast	2d	Cut	MLF	SHR	DGS	Fasthap	FCM	HG	ARO	HRCH
0%	3	1.000	1.000	0.999	0.988	0.946	1.000	0.782	0.781	1.000	0.992	1.000	0.983	1.000	0.986
	5	1.000	1.000	1.000	0.999	0.976	1.000	0.854	0.832	1.000	0.993	1.000	0.989	1.000	0.999
	8	1.000	1.000	1.000	1.000	0.992	1.000	0.919	0.868	1.000	0.994	1.000	0.994	1.000	0.999
	10	1.000	1.000	1.000	0.999	0.997	1.000	0.933	0.898	1.000	0.991	1.000	1.000	1.000	**1.000**
10%	3	0.934	0.919	0.705	0.829	0.786	0.927	0.698	0.668	0.931	0.917	0.834	0.934	0.801	0.928
	5	0.951	0.923	0.947	0.941	0.880	0.916	0.809	0.716	0.977	0.872	0.881	0.990	0.862	0.972
	8	0.956	0.945	0.985	0.986	0.948	0.896	0.863	0.743	0.987	0.945	0.883	0.987	0.899	0.983
	10	0.973	0.951	0.986	0.995	0.965	0.889	0.884	0.726	0.997	0.983	0.996	0.997	0.912	0.992
20%	3	0.796	0.811	0.199	0.652	0.647	0.753	0.624	0.591	0.669	0.703	0.652	0.677	0.644	0.797
	5	0.829	0.854	0.681	0.712	0.697	0.825	0.682	0.617	0.741	0.681	0.672	0.910	0.662	0.869
	8	0.832	0.868	0.801	0.808	0.751	0.856	0.747	0.653	0.818	0.916	0.686	0.884	0.695	0.885
	10	0.86	0.869	0.813	0.872	0.778	0.861	0.765	0.675	0.861	0.896	0.746	0.894	0.698	**0.900**
30%	3	0.652	0.600	0.095	0.581	0.583	0.552	0.570	0.536	0.573	0.627	0.592	0.592	0.588	0.602
	5	0.659	0.733	0.523	0.591	0.596	0.555	0.594	0.562	0.595	0.682	0.599	0.621	0.598	0.699
	8	0.662	0.804	0.616	0.615	0.613	0.597	0.614	0.611	0.614	0.741	0.606	0.646	0.613	0.729
	10	0.714	0.844	0.627	0.616	0.622	0.645	0.625	0.625	0.622	0.805	0.606	0.696	0.618	0.759

https://doi.org/10.1371/journal.pone.0241291.t003

respectively. The obtained results demonstrate that the inclusion of the chaotic nature of haplotype sequences can significantly improve the reconstruction rate.

Experimental dataset. The second dataset which is used for evaluation of the proposed algorithm involves experimental data which was provided by 1000 genome project. The gathered data belongs to an individual NA12878 which often is used to analyze the performance of the existing haplotype assembly methods. The sample was provided by using 454 sequencing method. According to the overlapping of the obtained fragments, they are represented in multiple matrices. In this experiment, for each chromosome, the first 500 matrices have been selected. In each matrix, the length of each row is ~90 in average and cover the genome at a depth of ~ × **3**. Furthermore, the trio-phased variant calls from the GATK resource bundle [55] was used as the target haplotypes. The obtained reconstruction rates of the proposed method are compared to those of H-pop [34], SCGD [36], HG [33], ARO [24], and FCM [25]

Table 4. The effect of refinement phase for haplotypes with length 700 in diploid case.

E	C	Hypergraph	em = 1	em = 2
10%	3	0.916	0.918	0.928
	5	0.966	0.968	0.972
	8	0.980	0.981	0.983
	10	0.989	0.991	0.992
20%	3	0.786	0.787	0.797
	5	0.854	0.857	0.869
	8	0.870	0.880	0.885
	10	0.886	0.896	0.900
30%	3	0.600	0.601	0.602
	5	0.688	0.689	0.699
	8	0.716	0.720	0.729
	10	0.748	0.754	0.759

https://doi.org/10.1371/journal.pone.0241291.t004

Table 5. The reconstruction rate and running time for the proposed method, H-pop, SCGD, HG, ARO and FCM applied to the experimental dataset NA12878 dataset provided by 1000 genome project.

Chr	H-pop		SCGD		HG		ARO		FCM		HRCH	
	RR	t(sec)	RR	t(sec)	RR	t(sec)	RR	t(sec)	RR	t(sec)	RR	t(sec)
1	0.957	5.22	0.925	3.62	0.937	1.54	0.935	20.28	0.913	1.09	0.954	10.40
2	0.956	5.65	0.926	4.41	0.929	1.30	0.943	18.03	0.908	1.04	0.943	12.34
3	0.912	6.99	0.919	3.40	0.928	1.17	0.940	18.45	0.913	1.91	**0.944**	12.75
4	0.970	5.24	0.927	5.47	0.923	1.20	0.949	18.06	0.923	1.68	0.960	13.07
5	0.966	4.67	0.939	3.54	0.932	1.24	0.942	15.09	0.912	1.27	0.952	14.98
6	0.952	4.93	0.930	8.70	0.935	1.22	0.948	15.60	0.929	1.04	**0.958**	13.58
7	0.924	4.24	0.935	3.95	0.925	1.26	0.951	16.34	0.904	1.03	**0.954**	12.53
8	0.947	4.14	0.907	2.18	0.906	1.25	0.934	16.62	0.903	1.07	**0.949**	13.03
9	0.910	3.36	0.971	2.94	0.901	1.30	0.966	15.25	0.937	1.04	0.921	12.63
10	0.945	3.67	0.926	2.56	0.940	1.21	0.945	15.73	0.913	1.28	**0.954**	13.14
11	0.915	3.71	0.932	2.95	0.939	1.17	0.942	14.34	0.923	1.18	**0.963**	10.46
12	0.903	3.46	0.923	2.03	0.945	1.19	0.935	14.26	0.908	1.14	**0.954**	11.33
13	0.941	2.89	0.970	3.31	0.930	1.22	0.935	15.72	0.925	1.43	0.946	14.12
14	0.971	2.54	0.911	1.36	0.917	1.52	0.934	15.42	0.932	1.11	0.949	14.03
15	0.974	2.40	0.991	1.21	0.920	1.02	0.937	16.65	0.905	1.04	0.951	12.24
16	0.935	2.47	0.930	1.79	0.932	1.11	0.946	15.27	0.924	1.35	**0.962**	11.01
17	0.911	1.98	0.967	2.61	0.931	1.25	0.951	15.86	0.920	1.11	0.963	11.35
18	0.976	2.51	0.903	1.16	0.924	1.86	0.949	15.66	0.919	1.01	0.954	13.02
19	0.978	1.82	0.972	3.25	0.949	1.60	0.942	14.58	0.923	1.40	0.960	10.46
20	0.950	2.00	0.968	1.38	0.945	1.90	0.946	15.49	0.922	1.12	**0.957**	11.31
21	0.970	1.70	0.943	0.63	0.933	1.52	0.941	15.12	0.915	1.08	0.960	12.77
22	0.983	1.44	0.941	0.74	0.951	1.16	0.941	14.34	0.914	1.33	0.964	9.64
Mean	0.948	3.50	0.939	2.87	0.931	1.33	0.943	16.00	0.918	1.22	**0.953**	12.28

https://doi.org/10.1371/journal.pone.0241291.t005

approaches. The results for all 22 homologous chromosomes are listed in Table 5. The results show that in most cases the proposed method achieved higher reconstruction rates compared to the others. The last row of the table demonstrates the mean of RR values of the comparing methods for all of the chromosomes. According to the obtained results, it can be concluded that this method completely outperforms the other approaches.

Polyploid case

Here, the proposed method is compared with three recent approaches that have been developed to solve haplotype assembly in polyploid form including Althap [23], H-POP [34] and SCGD [36]. The source codes of all comparing methods are available. To investigate the quality of reconstructed haplotypes, reconstruction rate (RR), and MEC measure of the methods have compared.

Indeed, the benchmark dataset is provided by simulation. For this aim, we have used the source code, which is available upon request by [23]. Its input parameters are coverage (c), error rate (e), and length of haplotypes (l). In this experiment, we defined $c \in \{5,10,15,20\}$, $e \in \{0.1,0.2,0.3\}$ and $l \in \{100,350,700\}$. For each combination of those parameters 10 samples have generated. Each sample contains an SNP matrix with a huge amount of gaps. As can be seen in Tables 6–8 the proposed method is compared with RR and MEC-based algorithms.

The results demonstrate that the proposed method outperforms the other approaches in most cases considering both RR and MEC parameters. Similar to the previous section, to

Table 6. Average of reconstruction rate for haplotypes with length 100.

e	C	SCGD		H-pop		AltHap		HRCH	
		RR	MEC	RR	MEC	RR	MEC	RR	MEC
10%	5	0.609	1289	0.745	269	0.736	315	**0.830**	**260**
	10	0.567	2917	0.813	534	0.783	598	**0.846**	**523**
	15	0.567	4282	0.828	805	0.754	1095	**0.859**	**773**
	20	0.564	5846	**0.844**	**1004**	0.747	1864	0.839	1020
20%	5	0.596	1367	0.667	467	0.657	478	**0.717**	**447**
	10	0.548	2862	0.692	1009	0.730	1050	**0.768**	**942**
	15	0.549	3047	0.706	1539	0.666	2241	**0.797**	**1443**
	20	0.547	5894	0.740	2016	0.631	3559	**0.781**	**1920**
30%	5	0.587	1373	0.596	554	0.630	548	**0.661**	**538**
	10	0.550	2857	0.599	1244	0.633	1338	**0.689**	**1190**
	15	0.548	4267	0.619	1916	0.596	2640	**0.730**	**1849**
	20	0.550	5916	0.633	2591	0.572	4051	**0.730**	**2512**

https://doi.org/10.1371/journal.pone.0241291.t006

Table 7. Average of reconstruction rate for haplotypes with length 350.

e	C	SCGD		H-pop		AltHap		HRCH	
		RR	MEC	RR	MEC	RR	MEC	RR	MEC
10%	5	0.585	4925	0.596	1236	**0.746**	**1016**	0.737	1048
	10	0.559	10059	0.698	2225	**0.835**	**2055**	0.809	2146
	15	0.546	15717	0.670	3611	0.724	6583	**0.844**	**2922**
	20	0.547	19824	0.727	4381	0.686	11103	**0.786**	**4307**
20%	5	0.576	4819	0.517	1771	0.656	1716	**0.661**	**1700**
	10	0.552	10159	0.565	3784	0.651	5344	**0.696**	**3450**
	15	0.539	15078	0.588	5776	0.602	9952	**0.751**	**5201**
	20	0.538	20874	0.589	7801	0.592	14345	**0.755**	**6962**
30%	5	0.555	4845	0.470	2020	0.588	2016	**0.631**	**1998**
	10	0.542	10108	0.508	4455	0.560	6208	**0.646**	**4317**
	15	0.540	15164	0.511	6891	0.558	10841	**0.665**	**6631**
	20	0.538	21403	0.518	9330	0.546	15422	**0.658**	**8946**

https://doi.org/10.1371/journal.pone.0241291.t007

Table 8. Average of reconstruction rate for haplotypes with length 700.

e	C	SCGD		H-pop		AltHap		HRCH	
		RR	MEC	RR	MEC	RR	MEC	RR	MEC
10%	5	0.641	9259	0.702	2549	**0.772**	**2282**	0.720	2392
	10	0.582	20313	0.736	5438	**0.924**	**3971**	0.751	4850
	15	0.514	30996	0.795	8062	**0.948**	**6086**	0.823	6860
	20	0.514	41555	0.817	10205	**0.889**	10257	0.870	**8389**
20%	5	0.614	8987	0.634	3488	0.656	3409	**0.698**	**3281**
	10	0.520	20500	0.650	7781	**0.722**	7280	0.711	**7107**
	15	0.511	30936	0.659	11785	**0.788**	13672	0.763	**10705**
	20	0.517	40781	0.701	15833	**0.768**	23738	0.764	**14509**
30%	5	0.578	9547	0.594	3882	0.600	3864	**0.681**	**3813**
	10	0.523	20494	0.600	8770	0.607	8680	**0.679**	**8442**
	15	0.514	31211	0.614	**13737**	0.566	17385	**0.692**	15393
	20	0.514	41155	0.611	18620	0.545	27813	**0.699**	**17841**

https://doi.org/10.1371/journal.pone.0241291.t008

Table 9. The effect of refinement phase for haplotypes with length 700 in polyploid case.

e	C	Hypergraph	em = 1	em = 2
10%	5	0.712	0.715	0.720
	10	0.749	0.750	0.751
	15	0.823	0.823	0.823
	20	0.870	0.870	0.870
20%	5	0.678	0.687	0.698
	10	0.705	0.708	0.711
	15	0.762	0.763	0.763
	20	0.763	0.764	0.764
30%	5	0.643	0.662	0.681
	10	0.664	0.672	0.679
	15	0.681	0.688	0.692
	20	0.692	0.697	0.699

https://doi.org/10.1371/journal.pone.0241291.t009

emphasize the efficiency of the refinement phase, the RRs of haplotypes with length 700 have been considered, as provided in Table 9. Similar to the diploid case, the improvement of RRs reveals the role of chaotic viewpoint to efficiently decrease the amount of remaining noise in the constructed haplotypes. Obviously, the proposed method is slower than the competitors, because it starts from a random measure and is iterative. However, it can solve the problem in a reasonable amount of time.

Since sequencing coverage of the used benchmark datasets were relatively low, we further evaluated the performance of HRCH by dealing with high coverage data. For this aim, by using the provided source code in [23], for diploid and polyploid cases, several samples were generated individually. For each combination of l = 500, e = 0.3, and c = {30,40,50}, 10 samples were generated. As shown in Fig 8, the reconstruction rates of the proposed method are compared to those of AltHap [23], SCGD [36], and H-pop [34]. The obtained results demonstrate that HRCH provides encouraging accuracy as compared to the competing schemes in diploid and polyploid forms.

Conclusion

The high amounts of noise, as well as existing gaps in the input fragments, are the main challenges in solving the SIH problem. In this study, we established a sampling-based method that starts from an initial set of haplotypes and iteratively proceeds to improve the input data by correcting the SNPs with wrong measures. The proposed method involves two main steps. First, it utilizes the hypergraph model to conquer the sparsity and high amount of noise, and

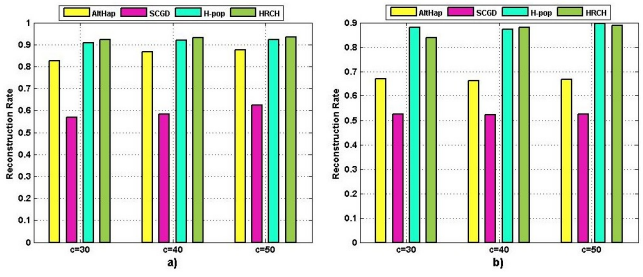

Fig 8. Comparison of reconstruction rate of the methods over high coverage data a) Diploid b) Polyploid.

https://doi.org/10.1371/journal.pone.0241291.g008

reconstructs haplotypes iteratively. Positions with low confidence are then rectified by mapping haplotype sequences to the coordinate series and applying a chaotic viewpoint. The proposed method has the capability to manipulate genomic data of both diploid and polyploid organisms. The promising results for diploid and polyploid data highlight that the method is comparable with the existing approaches, and they have complementary roles to each other. Finally, the source code of the proposed method is available at https://github.com/mholyaee/HRCH.

Acknowledgments

The authors wish to thank Dr. Jamshid Pirgazi and Dr. Omid AbbasZadeh for their valuable suggestions and discussions. We also thank Dr. Sajad Ahmadian and Dr. Sina Majidian for their constructive comments.

Author Contributions

Conceptualization: Mohammad Hossein Olyaee, Alireza Khanteymoori.

Formal analysis: Mohammad Hossein Olyaee, Alireza Khanteymoori, Khosrow Khalifeh.

Investigation: Mohammad Hossein Olyaee, Alireza Khanteymoori, Khosrow Khalifeh.

Methodology: Alireza Khanteymoori, Khosrow Khalifeh.

Supervision: Alireza Khanteymoori, Khosrow Khalifeh.

Validation: Mohammad Hossein Olyaee, Alireza Khanteymoori, Khosrow Khalifeh.

Writing – original draft: Mohammad Hossein Olyaee, Alireza Khanteymoori, Khosrow Khalifeh.

Writing – review & editing: Mohammad Hossein Olyaee, Alireza Khanteymoori, Khosrow Khalifeh.

References

1. Browning SR, Browning BL (2011) Haplotype phasing: existing methods and new developments. Nature Reviews Genetics 12: 703. https://doi.org/10.1038/nrg3054 PMID: 21921926
2. Group ISMW (2001) A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. Nature 409: 928. https://doi.org/10.1038/35057149 PMID: 11237013
3. Lee C-Y (2016) A model for the clustered distribution of SNPs in the human genome. Computational Biology and Chemistry 64: 94–98. https://doi.org/10.1016/j.compbiolchem.2016.06.003 PMID: 27318295
4. Wang R-S, Wu L-Y, Li Z-P, Zhang X-S (2005) Haplotype reconstruction from SNP fragments by minimum error correction. Bioinformatics 21: 2456–2462. PMID: 15731204
5. Loggetto SR (2013) Sickle cell anemia: clinical diversity and beta S-globin haplotypes. Revista brasileira de hematologia e hemoterapia 35: 155–157. PMID: 23904799
6. Rohlfs EM, Zhou Z, Heim RA, Nagan N, Rosenblum LS, et al. (2011) Cystic fibrosis carrier testing in an ethnically diverse US population. Clinical chemistry 57: 841–848. PMID: 21474639
7. McLaren GD, Gordeuk VR (2009) Hereditary hemochromatosis: insights from the hemochromatosis and iron overload screening (HEIRS) study. ASH Education Program Book 2009: 195–206. https://doi.org/10.1182/asheducation-2009.1.195 PMID: 20008199
8. Wilson JF, Weale ME, Smith AC, Gratrix F, Fletcher B, et al. (2001) Population genetic structure of variable drug response. Nature genetics 29: 265. https://doi.org/10.1038/ng761 PMID: 11685208
9. Exner DV, Dries DL, Domanski MJ, Cohn JN (2001) Lesser response to angiotensin-converting–enzyme inhibitor therapy in black as compared with white patients with left ventricular dysfunction. New England Journal of Medicine 344: 1351–1357. https://doi.org/10.1056/NEJM200105033441802 PMID: 11333991

10. Varner RV, Ruiz P, Small DR (1998) Black and white patients response to antidepressant treatment for major depression. Psychiatric Quarterly 69: 117–125. https://doi.org/10.1023/a:1024762503100 PMID: 9627929

11. Glusman G, Cox HC, Roach JC (2014) Whole-genome haplotyping approaches and genomic medicine. Genome medicine 6: 73. https://doi.org/10.1186/s13073-014-0073-7 PMID: 25473435

12. Lawson DJ, Hellenthal G, Myers S, Falush D (2012) Inference of population structure using dense haplotype data. PLoS genetics 8: e1002453. https://doi.org/10.1371/journal.pgen.1002453 PMID: 22291602

13. Green RE, Krause J, Briggs AW, Maricic T, Stenzel U, et al. (2010) A draft sequence of the Neandertal genome. science 328: 710–722. https://doi.org/10.1126/science.1188021 PMID: 20448178

14. Sabeti PC, Varilly P, Fry B, Lohmueller J, Hostetter E, et al. (2007) Genome-wide detection and characterization of positive selection in human populations. Nature 449: 913. https://doi.org/10.1038/nature06250 PMID: 17943131

15. Liu N, Zhang K, Zhao H (2008) Haplotype-association analysis. Advances in genetics 60: 335–405. https://doi.org/10.1016/S0065-2660(07)00414-2 PMID: 18358327

16. Roach JC, Glusman G, Hubley R, Montsaroff SZ, Holloway AK, et al. (2011) Chromosomal haplotypes by genetic phasing of human families. The American Journal of Human Genetics 89: 382–397. https://doi.org/10.1016/j.ajhg.2011.07.023 PMID: 21855840

17. Douglas JA, Boehnke M, Gillanders E, Trent JM, Gruber SB (2001) Experimentally-derived haplotypes substantially increase the efficiency of linkage disequilibrium studies. Nature genetics 28: 361.

18. Ruano G, Kidd KK, Stephens JC (1990) Haplotype of multiple polymorphisms resolved by enzymatic amplification of single DNA molecules. Proceedings of the National Academy of Sciences 87: 6296–6300. https://doi.org/10.1073/pnas.87.16.6296 PMID: 1974719

19. Ruano G, Kidd KK (1989) Direct haplotyping of chromosomal segments from multiple heterozygotes via allele-specific PCR amplification. Nucleic acids research 17: 8392. PMID: 2573038

20. Tininini L, Bertolazzi P, Godi A, Lancia G (2010) CollHaps: a heuristic approach to haplotype inference by parsimony. IEEE/ACM transactions on computational biology and bioinformatics 7: 511–523. https://doi.org/10.1109/TCBB.2008.130 PMID: 20671321

21. Rhee J-K, Li H, Joung J-G, Hwang K-B, Zhang B-T, et al. (2016) Survey of computational haplotype determination methods for single individual. Genes & Genomics 38: 1–12.

22. Wang Y, Feng E, Wang R (2007) A clustering algorithm based on two distance functions for MEC model. Computational biology and chemistry 31: 148–150. https://doi.org/10.1016/j.compbiolchem.2007.02.001 PMID: 17363329

23. Hashemi A, Zhu B, Vikalo H (2018) Sparse tensor decomposition for haplotype assembly of diploids and Polyploids. BMC genomics 19: 191. https://doi.org/10.1186/s12864-018-4551-y PMID: 29589554

24. Olyaee M-H, Khanteymoori A (2018) AROHap: An effective algorithm for single individual haplotype reconstruction based on asexual reproduction optimization. Computational biology and chemistry 72: 1–10. https://doi.org/10.1016/j.compbiolchem.2017.12.005 PMID: 29289750

25. Olyaee M-H, Khanteymoori AR (2019) Single Individual Haplotype Reconstruction Using Fuzzy C-Means Clustering with Minimum Error Correction. Bioinformatics and Biocomputational Research 3.

26. Mazrouee S, Wang W (2014) FastHap: fast and accurate single individual haplotype reconstruction using fuzzy conflict graphs. Bioinformatics 30: i371–i378. PMID: 25161222

27. Bansal V, Bafna V (2008) HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. Bioinformatics 24: i153–i159. PMID: 18689818

28. Wang T-C, Taheri J, Zomaya AY (2012) Using genetic algorithm in reconstructing single individual haplotype with minimum error correction. Journal of biomedical informatics 45: 922–930. https://doi.org/10.1016/j.jbi.2012.03.004 PMID: 22465411

29. Patterson M, Marschall T, Pisanti N, Van Iersel L, Stougie L, et al. (2015) WhatsHap: Weighted haplotype assembly for future-generation sequencing reads. Journal of Computational Biology 22: 498–509. https://doi.org/10.1089/cmb.2014.0157 PMID: 25658651

30. Bracciali A, Aldinucci M, Patterson M, Marschall T, Pisanti N, et al. (2016) PWHATSHAP: efficient haplotyping for future generation sequencing. BMC Bioinformatics 17: 342. https://doi.org/10.1186/s12859-016-1170-y PMID: 28185544

31. Bansal V, Halpern AL, Axelrod N, Bafna V (2008) An MCMC algorithm for haplotype assembly from whole-genome sequence data. Genome research 18: 1336–1346. https://doi.org/10.1101/gr.077065.108 PMID: 18676820

32. Edge P, Bafna V, Bansal V (2017) HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome research 27: 801–812. https://doi.org/10.1101/gr.213462.116 PMID: 27940952

33. Chen X, Peng Q, Han L, Zhong T, Xu T (2014) An effective haplotype assembly algorithm based on hypergraph partitioning. Journal of theoretical biology 358: 85–92. https://doi.org/10.1016/j.jtbi.2014.05.034 PMID: 24954019

34. Xie M, Wu Q, Wang J, Jiang T (2016) H-PoP and H-PoPG: Heuristic partitioning algorithms for single individual haplotyping of polyploids. Bioinformatics 32: 3735–3744. PMID: 27531103

35. Puljiz Z, Vikalo H (2016) Decoding genetic variations: Communications-inspired haplotype assembly. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 13: 518–530. https://doi.org/10.1109/TCBB.2015.2462367 PMID: 27295635

36. Cai C, Sanghavi S, Vikalo H (2016) Structured low-rank matrix factorization for haplotype assembly. IEEE Journal of Selected Topics in Signal Processing 10: 647–657.

37. Olyaee MH, Khanteymoori A, Khalifeh K (2019) Application of Chaotic Laws to Improve Haplotype Assembly Using Chaos Game Representation. Scientific reports 9. https://doi.org/10.1038/s41598-019-46844-y PMID: 31316124

38. Mazrouee S, Wang W (2018) PolyCluster: Minimum Fragment Disagreement Clustering for Polyploid Phasing. IEEE/ACM transactions on computational biology and bioinformatics. https://doi.org/10.1109/TCBB.2018.2858803 PMID: 30040655

39. Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data mining and knowledge discovery 8: 53–87.

40. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. ACM sigmod record 29: 1–12.

41. Kuleshov V (2014) Probabilistic single-individual haplotyping. Bioinformatics 30: i379–i385. PMID: 25161223

42. Barnsley MF (2014) Fractals everywhere: Academic press.

43. Jeffrey HJ (1990) Chaos game representation of gene structure. Nucleic Acids Research 18: 2163–2170. PMID: 2336393

44. Olyaee MH, Yaghoubi A, Yaghoobi M (2016) Predicting protein structural classes based on complex networks and recurrence analysis. Journal of theoretical biology 404: 375–382. https://doi.org/10.1016/j.jtbi.2016.06.018 PMID: 27320678

45. Hoang T, Yin C, Yau SS-T (2016) Numerical encoding of DNA sequences by chaos game representation with application in similarity comparison. Genomics 108: 134–142. https://doi.org/10.1016/j.ygeno.2016.08.002 PMID: 27538895

46. Ge L, Liu J, Zhang Y, Dehmer M (2019) Identifying anticancer peptides by using a generalized chaos game representation. Journal of mathematical biology 78: 441–463. https://doi.org/10.1007/s00285-018-1279-x PMID: 30291366

47. Zheng K, Wang L, You Z-H (2019) CGMDA: An Approach to Predict and Validate MicroRNA-Disease Associations by Utilizing Chaos Game Representation and LightGBM. IEEE Access 7: 133314–133323.

48. Anitas EM, Slyamov A (2017) Structural characterization of chaos game fractals using small-angle scattering analysis. PloS one 12. https://doi.org/10.1371/journal.pone.0181385 PMID: 28704515

49. Geraci F (2010) A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem. Bioinformatics 26: 2217–2225. PMID: 20624781

50. Levy S, Sutton G, Ng PC, Feuk L, Halpern AL, et al. (2007) The diploid genome sequence of an individual human. PLoS biology 5: e254. https://doi.org/10.1371/journal.pbio.0050254 PMID: 17803354

51. Chen Z, Fu B, Schweller R, Yang B, Zhao Z, et al. (2008) Linear time probabilistic algorithms for the singular haplotype reconstruction problem from SNP fragments. Journal of Computational Biology 15: 535–546. https://doi.org/10.1089/cmb.2008.0003 PMID: 18549306

52. Zhao Y-Y, Wu L-Y, Zhang J-H, Wang R-S, Zhang X-S (2005) Haplotype assembly from aligned weighted SNP fragments. Computational Biology and Chemistry 29: 281–287. https://doi.org/10.1016/j.compbiolchem.2005.05.001 PMID: 16051522

53. Panconesi A, Sozio M. Fast hare: A fast heuristic for single individual SNP haplotype reconstruction; 2004. Springer. pp. 266–277.

54. Genovese LM, Geraci F, Pellegrini M (2008) SpeedHap: an accurate heuristic for the single individual SNP haplotyping problem with many gaps, high reading error rate and low coverage. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 5: 492–502. https://doi.org/10.1109/TCBB.2008.67 PMID: 18989037

55. DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, et al. (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nature genetics 43: 491. https://doi.org/10.1038/ng.806 PMID: 21478889