Resolving multicopy duplications *de novo* using polyploid phasing

Mark J. Chaisson[1*], Sudipto Mukherjee[2*], Sreeram Kannan[2**], and Evan E. Eichler[1**]

[1] Department of Genome Sciences, University of Washington, Seattle, WA 98195, USA
mchaisso@gs.washington.edu, [†]eee@gs.washington.edu
[2] Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA
sudipm@uw.edu, ksreeram@uw.edu

Abstract. While the rise of single-molecule sequencing systems has enabled an unprecedented rise in the ability to assemble complex regions of the genome, long segmental duplications in the genome still remain a challenging frontier in assembly. Segmental duplications are at the same time both gene rich and prone to large structural rearrangements, making the resolution of their sequences important in medical and evolutionary studies. Duplicated sequences that are collapsed in mammalian *de novo* assemblies are rarely identical; after a sequence is duplicated, it begins to acquire *paralog specific variants*. In this paper, we study the problem of resolving the variations in multicopy long-segmental duplications by developing and utilizing algorithms for polyploid phasing. We develop two algorithms: the first one is targeted at maximizing the likelihood of observing the reads given the underlying haplotypes using *discrete matrix completion*. The second algorithm is based on *correlation clustering* and exploits an assumption, which is often satisfied in these duplications, that each paralog has a sizable number of paralog-specific variants. We develop a detailed simulation methodology, and demonstrate the superior performance of the proposed algorithms on an array of simulated datasets. We measure the likelihood score as well as reconstruction accuracy, i.e., what fraction of the reads are clustered correctly. In both the performance metrics, we find that our algorithms dominate existing algorithms on more than 93% of the datasets. While the discrete matrix completion performs better on likelihood score, the correlation clustering algorithm performs better on reconstruction accuracy due to the stronger regularization inherent in the algorithm. We also show that our correlation-clustering algorithm can reconstruct on an average 7.0 haplotypes in 10-copy duplication data-sets whereas existing algorithms reconstruct less than 1 copy on average.

---

[*]Joint first authorship
[**]Joint last authorship
[†]Corresponding author

1 Introduction

Advances in single-molecule sequencing (SMS) by Pacific Biosciences (Menlo Park, CA), and Oxford Nanopore (Cambridge, UK) have recently enabled the assembly of draft *de novo* mammalian genomes [35, 21] nearing the quality of the original release of the human genome. The goal of *de novo* fragment assembly is to estimate the sequence of a genome given overlaps of relatively short sequencing reads, and is a well-studied problem. While there are multiple formulations of the fragment assembly problem [27, 31], the common challenge is that repeats in the genome longer than the length of sequenced DNA fragments make a unique reconstruction of the genome impossible [30]. Reads produced by SMS are advantageous for *de novo* assembly because the read length is at least two orders of magnitude greater than other high-throughput sequencing methods, so that genome order may be uniquely resolved when repeats are small.

SMS reads are characterized by a raw read accuracy between 75% and 90% with read lengths that follow a log-normal distribution. Initial development in *de novo* assembly of SMS reads focused on efficient methods to detect overlaps between long but noisy reads [28, 6]. Consistent with information theory [26], regions of genomes without sufficiently long repeats are contiguously assembled [24] with SMS reads. A type of repeat not well represented in human and other *de novo* SMS assemblies are *segmental duplications*: sequences 1 to 400 kilobases in length that are duplicated with at least 90% identity [18]. Segmental duplications are at the same time both gene rich and prone to large structural rearrangements, making the resolution of their sequences important in medical and evolutionary studies [37]. Comparing an SMS-based assembly of a Yoruban individual [38] to the human reference (GRCh38) reveals that only 64.2% of known segmentally duplicated bases in the human genome are present in the assembly. Due to the low raw-read accuracy of SMS sequences, reads from different duplication paralogs are frequently merged together into the same sequence in an assembly. As a result, human assemblies of SMS reads contain large contigs with correctly resolved unique sequence, and shorter contigs containing the collapse of multiple copies of a duplication into one sequence.

Segmental duplications that are collapsed in real *de novo* assemblies are rarely identical; after a sequence is duplicated, over generations it begins to acquire *paralog specific variants* (PSVs): single-nucleotide variants that distinguish different duplication paralogs. To put this in an evolutionary context, sequences that have duplicated shortly after the human-chimpanzee divergence (6 million years ago) have acquired up to roughly one PSV per thousand bases [17]. Although the ultimate goal of *de novo* assembly is to completely resolve the sequence of a genome, an intermediate goal is to resolve the individual sequences that are collapsed in the assembly. We propose resolving sequences by estimating the number of duplications collapsed into an individual sequence in an assembly, and determining the PSVs belonging to each duplication.

Given S segmental-duplication paralogs of the same length containing V variants, one may represent all paralogs as an $S \times V$ matrix P with entries in $\{0, 1\}$, where each entry $P(i, j)$ is 0 if the repeat paralog i is in the consensus

state at site j, or 1 if it is a PSV. The set of N reads from all repeat paralogs may be aligned to the consensus sequence, and represented as an $N \times V$ read-fragment matrix X with entries in $\{0, 1, -\}$ corresponding to consensus, variant, or absent (since reads only give information about certain positions). The goal is to reconstruct the paralog matrix P given only the read matrix X, where there are also sequencing errors creating erroneous entries in X. Let us assume that the error probability is ϵ at any position, i.e., with probability $1-\epsilon$, the location is read correctly and with probability ϵ, the location is read incorrectly (0 is read as a 1 and vice-versa).

For $S = 2$, this problem is identical to haplotype phasing of a diploid genome [25, 4, 29]. Defining a read conflict as two overlapping reads that are non-gap and disagreeing at a site, haplotype phasing with error-free reads may be determined by grouping all conflict-free reads. To handle sequencing errors, a common formulation for haplotype phasing is Minimal Error Correction (MEC), where a minimal number of base changes are applied to reads so that they may be partitioned into two conflict-free sets. For $S = 2$, there has also been an exact information theoretic characterization of when it is possible to phase the genome correctly [36, 13], along with efficient algorithms. This is based on connections to a problem called "community detection" [20] where the goal is to cluster users into communities based on positive or negative interactions between individuals.

When $S > 2$ this corresponds to the much less studied problem of *polyploid phasing*, which was discussed in pioneering work by Aguiar and Istrail [1]. Beginning with Hapcompass [1], there has been some work on polyploid phasing using algorithms based on branch-and-extend [5], belief propagation [32] and semi-definite programming [14]. In a recent theoretical work [7], the hardness of optimizing the MEC for $S > 2$ has also been proven, indicating that algorithms for this problem need to be necessarily approximate or tailored to some assumptions. A major drawback of existing works is that they consider only $S = 3, 4$ and none have been developed, optimized, or tested for the high ploidy that is encountered in segmental duplications, where S can be potentially larger than 10, and to the low error-rate in Illumina sequencers. Thus algorithms that are robust to the high error rates and can handle the high poly-ploidy are imperative in solving the segmental duplication problem, and in this paper, we will design such algorithms.

In particular, we propose two algorithms for solving the problem. The first approach is based on a discrete matrix completion paradigm where the goal is to maximize the likelihood of the observed data given the underlying haplotypes. The second approach is based on a correlation-clustering framework with an inherent assumption that each haplotype has a paralog-specific variant (which holds in many types of segmental duplications). By performing detailed simulations, we demonstrate the superior performance of the proposed algorithms over existing algorithms, especially in the high ploidy regime.

2 Haplotype phasing via Discrete Matrix Completion

2.1 A probabilistic model

In order to represent the matrices in real-valued arithmetic, we adopt the following mapping f: $\{0, 1, -\} \to \{-1, 1, 0\}$, i.e., we represent the consensus allele as -1, variant as 1 and undisclosed locations as 0. To model the read matrix X, we first consider an idealized matrix M, which does not contain any noise nor does it contain any undisclosed position. If read n is sampled from the s-th paralog, then the n-th row of this matrix M is given by the s-th row of the paralog matrix, i.e., $M_{n\cdot} = f(P_{s\cdot})$. The disclosed locations of the matrix are represented by a set Ω which comprises of the set of tuples (n, v) where read n contains information about variant v. Given M and Ω, the matrix X is not a deterministic function since there are independent read errors, which convert a 1 into a -1 with probability ϵ and vice versa. The probability of observing X given M and Ω is therefore given as follows,

$$\begin{aligned}
\log \mathbb{P}(X \mid M, \Omega) &= \sum_{(n,v)\in\Omega} \log \mathbb{P}(X_{n,v} \mid M, \Omega) \\
&= \sum_{(n,v)\in\Omega} \log\left((1-\epsilon)\mathbb{1}_{X_{n,v}=M_{n,v}}\right) + \log\left(\epsilon \mathbb{1}_{X_{n,v}\neq M_{n,v}}\right) \\
&= d_H(X, M) * \log(\epsilon) + (|\Omega| - d_H(X, M)) * \log(1-\epsilon) \\
&= -d_H(X, M) * \log(\frac{1-\epsilon}{\epsilon}) + (|\Omega|) * \log(1-\epsilon),
\end{aligned}$$

where $d_H(X, M)$ is the Hamming distance between the two matrices X and M in the locations Ω, i.e., where $X \neq 0$. Different haplotype assembly algorithms have sought to minimize varied objective criteria in order to obtain the correct clustering of reads belonging to the respective haplotypes [34]. Some of the noteworthy objectives are minimum edge removal (MER), minimum SNP removal (MSR) and minimum error correction (MEC). The quantity $d_H(X, M)$ is called the error criterion, and in our approach, maximizing the likelihood is equivalent to minimizing this error criterion referred to as MEC.

We observe that the ideal matrix M has repeated rows, since all rows sampled from the same paralog are identical. This implies that the matrix M has low-rank. Indeed the matrix M can be factorized as the product of two matrices $M = A \cdot B$, where $A \in \mathbb{R}^{N\times S}$ with $A_{ij} \in \{0, 1\}\ \forall i, j$ and $B \in \mathbb{R}^{S\times V}$ with $B_{ij} \in \{-1, 1\}\ \forall i, j$.Each row of A is an elementary vector of length S denoting which paralog the read is from and matrix B is identical to $f(P)$ (represented in $\{-1, 1\}$).

The observed matrix X is a noisy partial observation of the low-rank matrix M, and the goal is to reconstruct the matrices A and B given X. If each read spanned the entire segmental duplication, the problem would be trivial, since similar reads can be grouped together and taking a consensus inside clusters

reveals the segmental duplications. The difficulty is posed by the fact that read lengths are much smaller and do not span all variant positions.

Each read only provides partial phasing information. The resulting X matrix is thus sparse, and our goal can be formally stated as follows:

$$\text{argmin}_{A,B} d_H(X, A \cdot B). \tag{1}$$

Real-valued versions of this problem has received much attention and is called the matrix completion problem. While this problem has a rich history, there is a significant difference in our setting, since the matrices A and B have structure (i.e., A has only elementary row vectors and B has binary entries) and the matrix X is ternary. We therefore have to develop new algorithms that exploit the discrete structure of the problem.

The problem of finding missing entries in a matrix arises in diverse research domains. One of the most illustrative examples is the Netflix challenge where users rate a small fraction of movies at random and the task is to predict user preferences for an unrated movie; a key assumption in this domain is that the true matrix of preferences is low-rank. While low-rank matrix-completion problem is known to be NP-Hard, there are methods that can give provably correct reconstruction under probabilistic rather than worst-case assumptions [10, 33]. Popular techniques for this problem include convex relaxation of the rank to nuclear norm [33], singular value thresholding [9] and alternating minimization [22], all of which have theoretical guarantees as well. The key difference between these works and our problem is that they consider real-valued matrix-completion, whereas, in this paper, we adapt and extend the algorithms to the discrete setting inherent to the phasing problem.

In a recent paper [8], Cai *et al.* formulate haplotype phasing as a low rank matrix completion problem and use structure constrained alternating minimization for obtaining the haplotypes. In the paper, they demonstrate improved performance over HapCompass for diploid and simulated polyploid data (with $S = 3, 4$). We show in this paper that while that method has good performance with small S, the performance starts deteriorating with higher S. The main reason for the deteriorating performance is the inability of the algorithm to exploit the discrete structure of the problem (for example, the algorithm does not use the fact that the B matrix is binary, instead treating it as a real-valued matrix). We alleviate this problem in the present paper by proposing an algorithm that explicitly exploits this fact.

2.2 Iterative Two Stage Matrix Completion

Our problem stated in (1) is a hard combinatorial problem. One can design alternating minimization based techniques for this problem, where A and B are optimized alternatively while keeping the other variable fixed. While such methods monotonically increase likelihood, they are not guaranteed to find the global optimum of the problem and display high sensitivity to initial conditions. The key idea in our approach is to first neglect the discrete nature of our problem,

Algorithm 1 Iterative Matrix Completion

Input : Noisy incomplete Matrix X, Rank Estimate S
Initialize $A_{init} \in \mathbb{R}^{N \times S}$ and $B_{init} \in \mathbb{R}^{S \times V}$ with sign corrected SVD.
$e \leftarrow$ Error rate
$k \leftarrow S$
while $k \geq 2$ and MEC Score decreases **do**
    $B_{\text{est}} \leftarrow \text{RealMatCom}(A_{\text{init}}, B_{\text{init}}, X, k)$
    $A_{\text{est}}, B_{\text{est}} \leftarrow \text{DiscreteMatCom}(B_{\text{est}}, X, e)$
    Choose the best segment based on individual scores
    $A_{\text{init}} \leftarrow A_{\text{est}}$
    $B_{\text{init}} \leftarrow B_{\text{est}}$
    $k \leftarrow k - 1$
end while
Output : Estimated Haplotypes B_{est}

Algorithm 2 Real Valued Matrix Completion

procedure RealMatCom($A_{\text{init}}, B_{\text{init}}, X, k$)
    $A \leftarrow A_{\text{init}}$
    $B \leftarrow B_{\text{init}}$
    while stopping criterion not satisfied **do**
        Minimize A using projected gradient descent
        Minimize $B_{1:k}$ using projected gradient descent
    end while
    return sign(B)
end procedure

Algorithm 3 Discrete Valued Matrix Completion

procedure DiscreteMatCom(B_{est}, X, e)
    while MEC Score decreases **do**
        for each row i of X **do**
            for each segment s of B_{est} **do**
                $d(i, s) \leftarrow$ Hamming distance of X_i and $B_{\text{est},s}$ for known entries
                $W_i \leftarrow$ Window size of revealed entries of X_i
                $A_{\text{est},is} \leftarrow (1 - e)^{W_i - d(i,s)} \cdot e^{d(i,s)}$
            end for
            Update overall MEC score and score for each individual segment
            Normalize $A_{\text{est},i}$ to be a probability distribution
        end for
        Initialize $B_{\text{est},new} \in \mathbb{R}^{S \times V}$ with zeros
        for each row i of X **do**
            $B_{\text{est,new}} \leftarrow B_{\text{est,new}} + \mathcal{P}_{\Omega}(A^T_{est,i} \cdot X_i)$
        end for
        $B_{\text{est}} \leftarrow \text{sign}(B_{\text{est,new}})$
    end while
    return $A_{\text{est}}, B_{\text{est}}$
end procedure

and view it as a real-valued matrix completion problem. We then "round" the results obtained from this real valued matrix completion to obtain a feasible solution for the discrete problem. This rounded solution then becomes the initial value of a discrete matrix completion routine designed based on the alternative minimization technique. While this method already has superior performance compared to existing approaches, we found that in the regime when the ploidy is high, the algorithm is able to extract some dominant haplotypes correctly while being incorrect on the other haplotypes. In order to overcome this barrier, in iteration i, we only fix the best $i-1$ haplotypes based on the current MEC, and optimize for the rest. A schematic representation of this algorithm is depicted in Fig. 1, and the detailed pseudocode is in Algorithm 1, Algorithm 2 and Algorithm 3.

A standard approach in combinatorial optimization is to relax the integer constraints in the problem in order to get a real-valued optimization problem, and then to round the obtained results to get a feasible solution. We follow a similar approach here by relaxing our discrete problem to a continuous optimization problem, and along with it, we relax the objective too. Instead of optimizing according to the Hamming distance objective with the discrete constraints on A, B (see (1)), we instead minimize the Frobenius norm of the difference while at the same time assuming that A and B are real valued.

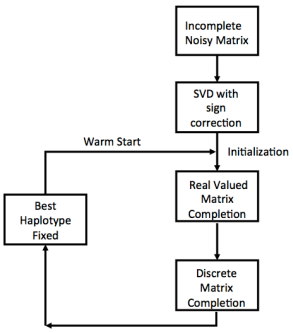

Fig. 1. The initialization and iteration workflow for MCP

The noisy low rank matrix completion can be formally stated as an optimization problem.

$$\min_{A,B} \frac{1}{2}\|\mathcal{P}_\Omega(A \cdot B - X)\|_F^2$$

The objective function is a squared sum of errors over all the known entries of X. $\mathcal{P}_\Omega(\cdot)$ is the projection operator and Ω is the set of known incides of X. So, $\mathcal{P}_\Omega(Z_{ij}) = Z_{ij}$ if $(i,j) \in \Omega$ and 0 otherwise. While we relax the integer constraints of the problem, we assume the following linear constraints to hold.

$$0 \leq A_{ij} \leq 1 \quad \forall\, i \in [N], j \in [S] \tag{2}$$
$$-1 \leq B_{ij} \leq 1 \quad \forall\, i \in [S], j \in [V] \tag{3}$$

Since the optimization is over unknown matrices A and B in a product form, the problem is non-convex. However, alternating minimization algorithms are known to have guaranteed reconstruction performance in certain regimes [22] and therefore we resort to using such algorithms. Thus we first solve the optimization over A, keeping B fixed, which makes the problem convex in A and vice-versa.

2.3 Projected Gradient Descent

The alternating minimization for our problem therefore can be stated as follows:

$$\min_{A} \frac{1}{2}\|\mathcal{P}_{\Omega}(A \cdot B - X)\|_F^2$$
$$s.t. \quad 0 \leq A_{ij} \leq 1 \; \forall \, i, j$$

and similarly

$$\min_{B} \frac{1}{2}\|\mathcal{P}_{\Omega}(A \cdot B - X)\|_F^2$$
$$s.t. \quad -1 \leq B_{ij} \leq 1 \; \forall \, i, j$$

To incorporate the constraints on the variables, we use a projected gradient descent to minimize each of the convex formulations.

2.4 Initialization

Since the overall problem is non-convex, it is required to choose a suitable initialization for better performance. For this purpose, we use the singular value decomposition (SVD). This is a factorization of a $m \times n$ rectangular matrix of rank r in the form $\mathcal{U}\Sigma\mathcal{V}^T$, where $\mathcal{U}$ is a $m \times r$ unitary matrix, Σ is a $r \times r$ diagonal matrix with non-negative diagonal entries and $\mathcal{V}$ is a $n \times r$ unitary matrix. The columns of $\mathcal{U}$ and $\mathcal{V}$ are called the left and right singular vectors respectively. Prior theoretical results [22] suggest taking the S singular vectors of $\mathcal{P}_{\Omega}(X)$ as the initial guess for A and B. While this is a reasonable initialization, the signs of the singular vectors obtained from SVD decomposition may not be consistent with our problem since we require the entries of A to be strictly non-negative. We note that the signs of the singular vectors can be swapped without affecting the SVD. Therefore, in our algorithm, in order to ensure this sign consistency, we reverse the signs of certain rows of B to ensure that all columns of A have a positive sum.

$$\mathcal{P}_{\Omega}(X) = \mathcal{U} \cdot \Sigma \cdot \mathcal{V}^T \quad \Gamma = \text{sign}(\mathbb{1}^T\mathcal{U}) \quad A_{\text{init}} = \mathcal{U} * \text{diag}(\Gamma) \quad B_{\text{init}} = (\mathcal{V} * \text{diag}(\Gamma))^T$$

For details of the projected gradient descent, we refer the reader to Appendix I.

2.5 Discrete Matrix Completion

We round the output of the real-valued matrix completion to satisfy the discrete constraints of A and B and utilize this to run a discrete alternating minimization algorithm to solve (1). The optimization of A given a fixed B is easy to solve: the basic idea is to assign each read to the segment which minimizes the Hamming distance with the read. To optimize B given a fixed A, we find the consensus of all the reads which are informative about a given position. In our algorithm, instead of having A to be a hard decision of which segment a given read belongs

to, each row i of A encodes the probability that read i belongs to segment j. Therefore, while optimizing over B, we utilize the weighted consensus rather than the plain consensus of the read assignments. This procedure of refinement comes under the purview of a broader class of algorithms called the Expectation Maximization (EM) algorithm [16] as well as Variational Bayes [40]. The matrix A can be viewed as hidden variables encoding the membership of read fragments to duplication copies and B as the parameters for the exact underlying segments. We refer the reader to Algorithm 3 for a detailed description of the algorithm.

2.6 Choosing the best segment and Effective Rank Reduction

As pointed out earlier, the algorithm as stated above works well with small polyploid instances; however, in the presence of higher ploidy, the algorithm returns only the top few haplotypes correctly. For example, consider the cascading topology of repeats in Figure 2, it is easier to resolve segment 7 but the other segments are more easily confused. Therefore, we propose an iterative algorithm, where in each iteration, the best haplotype is fixed and then the algorithm is run to optimize over possibilities of the other haplotypes. Thus in order to do matrix completion with S haplotypes, the algorithm is iterated over $S-1$ times. Such algorithms have a precedent even in real-valued matrix completion. For example, stagewise alternating minimization is shown to have better theoretical guarantees in [22]. In our implementation, at iteration i, the best $i-1$ haplotypes are chosen as the ones which have minimum Hamming distance from their assigned reads.

3 Haplotype phasing with correlation clustering

One limitation of the MEC objective function and therefore of the discrete matrix completion algorithm is that the ploidy must be known *a priori* or estimated. Since the MEC objective itself decreases monotonically with ploidy, it is not possible to estimate the ploidy using the MEC objective. This can be potentially remedied using regularized alternatives that account for model complexity like AIC, BIC or MDL. We propose an alternative algorithm here that can jointly estimate the ploidy while estimating the haplotypes themselves. This algorithm is based on a key assumption, distinct from the assumptions of the discrete matrix completion problem: that each of the haplotypes have uniquely identifying variants. While this assumption is stronger, it can lead to stronger regularization of the problem by restricting the search space and therefore leads to better estimates, especially when the ploidy is high.

The basic idea of the algorithm is the following: each locus is represented as a vertex and reads that straddle multiple vertices create edges between the vertices that have either positive or negative weight based on whether reads share the variant or not. The goal is then to cluster the nodes into groups which share the same variant, with each cluster representing a haplotype and each locus (node) in the cluster representing a haplotype-specific variant.

To formally define our algorithm, we begin with an alternative formulation for polyploid phasing through *correlation clustering* [3], with the premise that a metric defines how similar or dissimilar two objects are, and clusters maximize the amount of similarity within each cluster and dissimilarity between clusters. Importantly, in correlation clustering the number of clusters is discovered as a result of clustering and not as a parameter.

We use an augmented form of the SNP conflict graph $\mathcal{G}_{\mathcal{S}}$ introduced in [25], denoted $\mathcal{G}_{\text{PSV}} = (V, E), E = \{E^+, E^-\}$. The construction of $\mathcal{G}_{\text{PSV}}$ requires the fragment matrix M, and some data-dependent parameters: the expected range of coverage per haplotype c_{min} and c_{max}, and a distance d that is the maximum distance reads are expected to overlap variants. A vertex exists for each of the columns (sites) in the fragment matrix M, connected by an edge $(u, v) \in E^+$ if u and v are overlapped by between c_{min} and c_{max} reads that are variant (e.g., 1) at both sites, or an edge $(u, v) \in E^-$ if the sites corresponding u and v are within d bases and $(u, v) \notin E^+$. A weight $W(u, v)$ is assigned to each edge.

Correlation clustering on $\mathcal{G}_{\text{PSV}}$ corresponds to finding clusters $C = c_1, \ldots, c_n$ that minimize the sum of weighted negative edges within each cluster and weighed positive edges between clusters:
$\text{Score}_{\text{CC}} = \sum_{c_i}(\sum_{(u,v)\in c_i,(u,v)\in E^-} w(u, v)+\sum_{(u\in c_i,v\notin c_i),(u,v)\in E^+} w(u, v))$, where $w(u, v)$ may reflect certainty of clustering, or more simply $w(u, v) = 1$ to count edges. Each cluster defines a set of sites that belong to a haplotype. This was shown to be APX-hard [15, 12, 19], and approximations based on linear programming (LP) were described in [15, 12]. We developed an implementation of the LP approach that was successful at clustering smaller datasets, however the number of constraints grows with $|E|^2$, and $|E|$ grows by p^2v^2, for ploidy p and number of paralog specific variants v, which requires excessive resources for larger datasets.

To evaluate correlation clustering on larger datasets, we developed a simple randomized heuristic to search similar to the method of [2] for clusters that provide acceptable values for Score_{CC} that follows the steps:

1. Define clusters likely to represent repeat paralogs through a random search.
2. Merge clusters with sufficient overlap, and assign nodes to unique clusters.
3. Optimize clusters by swapping vertices from adjacent clusters.

Define the *neighbor similarity* $\text{Sim}(u, v)$ of two vertices to be the number of neighbors shared between u and v connected by edges in E^+, and $\text{Score}(V, E, c)$ to be the Score_{CC} of a single cluster c assuming all vertices $V \setminus c$ are in a separate cluster. First, clusters are formed by iteratively adding vertices neighboring a cluster as long as the neighbor similarity is sufficient and addition of the vertex decreases Score_{CC}, described in 4.

Given parameters for neighbor similarity s, a maximal number of search iterations max_search_it and swap iterations max_swap_it, and fraction cluster overlap f^{ovp}, the method FindCluster is used to find a set of clusters C by first initializing $C = \emptyset$, and iteratively selecting a vertex $v_i \notin C$, and adding the result of $\text{FindCluster}(V, v_i, E, s)$ to C until C contains all vertices in V or max_it iterations are reached. The resulting clusters in C are not disjoint, and so any

Algorithm 4 Find cluster

```
procedure FindCluster(V, v_i, E, s)
    c ← v_i
    repeat
        for all v ∈ c do
            for all n ∈ Neighbors(v) ∉ c do
                if Sim(v, n) ≥ s and Score(V, E, c ∪ n) < Score(V, E, c) then
                    c ← c ∪ n
                end if
            end for
        end for
    until c has not grown
    return c
end procedure
```

cluster c_i with a fraction of vertices overlapping with a cluster $c_j > f^{\text{ovp}}$ is first merged into c_j, then remaining vertices belonging to more than one cluster are assigned to the largest cluster for which they are a member. Finally, the clusters are further optimized by selecting edges $(u, v) \in E^+$ where $u \in c_i$ and $v \in c_j$ and swapping u and v if this improves Score_{CC} for up to max_swap_it iterations.

4 Results

We benchmarked our methods on a dataset of simulated collapsed segmental duplications. It is difficult to simulate the complex mosaic architecture of segmental duplications [23], and so we elected to use a simplified model of 100kbp non-mosaic duplications. While this lacks the complexity of mosaic duplication architecture, the length is greater than the average duplication unit (∼30kbp), ensuring evaluation on challenging problems. Starting with an ancestral sequence, sequences are duplicated according to a specified tree topology T and mutation rate r, where each child node is a copy of a parent node mutated at a rate of r random SNV mutations per base. In real data, duplications arise with many complex histories [17, 39]. To capture the complexity of evolution, we used two classes of trees: 12 simulations from well defined topologies such as flat, bifurcating, and cascading resulting in four to eight duplicated sequences, and 50 simulations from random tree topologies that have 10 duplicated sequences. Examples of the duplication toplogies are shown in Figure 2. The mutation rate was varied from between 0.01, 0.005, and 0.001, and 0.0005 mutations per base to simulate various ages of duplications. For each set of duplications we simulated 50× read coverage using the Alchemy SMS read simulator [11], a model based simulator that emulates a sequencing run by Pacific Biosciences, and mapped reads back to the ancestral sequence. PSV sites are detected as sites that contain between 25 and 60 non-ancestral bases.

For each of the simulated topologies and mutation rates, we evaluated the Discrete Matrix Completion (DMP), Correlation Clustering (CC), and Structure

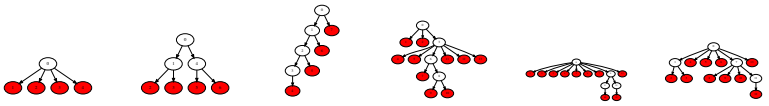

Fig. 2. Examples of topologies of duplication simulations. In total there are 12 structured trees and 50 random topologies. The divergence between any two simulated duplications is given by the mutation rate $r\times$ the shortest path between the duplications in the tree.

Constrained Gradient Descent (SCGD). The SCGD method has been shown to outperform other previously developed methods in polyploid phasing [8].

We report Minimum error correction (MEC) values by computing the sum of Hamming distance between each read and the consensus sequence for each haplotype. For CC, we assign reads to each haplotype according to the minimal Hamming distance from the read to each haplotype. For duplications simulated under models of high mutation rates (0.01 and 0.005), the DMP method is able to obtain a lower MEC than the other methods. Out of the 128 datasets for which every method is able to run within the time constraint set on our server, we compare the performance. We observe that CC obtains the best MEC score 11% of the datasets, DMP 85%, and SCGD 3.9%.

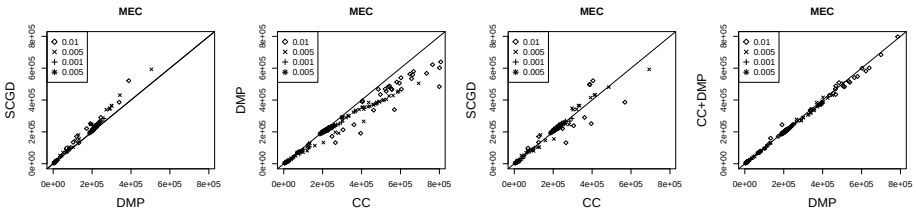

Fig. 3. MEC scores for the DMP, CC, SCGD, and CC+DMP methods. The DMP method is shown to produce haplotpyes with lower MEC than either method, particularly for the high mutation rate simulations. Lower MEC score is better.

For each haplotype we count the number of reads in the haplotype that are shared with the reads simulated in each duplication, and define a *matching* statistic as the sum of number of reads in the maximally matched duplication divided by the total number of reads. This statistic ranges between 1 for perfect reconstruction of haplotypes down to a $1/p$ when all of them are collapsed into a single reconstructed haplotype. The results are shown in Figure 4. CC had the greatest matching score 67.7% of the datasets, DMP 26.1% of the datasets, and SCGD on 6.1% of the datasets. Interestingly, while the CC method has a higher MEC statistic, it has a greater number of correctly partitioned reads when compared with the ground truth. We reason that this is because the CC

method exploits the assumption that the positions are variant specific explicitly resulting in stronger regularization, so that even though the likelihood score is somewhat lower for CC method than other methods, it is able to fit the data more accurately. The other methods DMP and SCGD are unable to exploit this assumption and therefore overfit more severely to the data. The DMP method is sensitive to the initialization conditions for B_{est}, and so we used a solution derived by CC as initial conditions for DMP. We measured improvements on this combination (CC+DMP) relative to DMP on MEC (Figure 3, *right*), and CC for matching score. While the MEC score was largely unchanged, 220 of the 224 simulations where both CC and CC+DMP had a solution had a greater matching score in CC+DMP (Figure 4).

We also measure a more stringent quality of reconstruction accuracy: we ask for what fraction of the true haplotypes there is a reconstructed cluster into which 90% of the correct reads are assigned. Formally, for each simulated duplication we determined which haplotype had the most reads overlapping with the reads simulated from that duplication, and counted how many such haplotypes had at least 90% of the reads from that haplotype reciprocally assigned to that duplication. This gives an indication of the number of copies of a segmental duplication that would be correcly assembled given the phased haplotypes. For the 48 simulations with duplication copy number between 3 and 8, the CC method resolves 70% of duplication copies, while the DMP and SCGD methods resolve 66% and 26%, respectively (Figure 5). For duplications of ploidy 10, the CC method resolves on average 7.0 copies of each duplication, whereas the DMP and SCGD methods resolve on average 3.3 and 0.03 copies, respectively. The CC+DMP combined method resolved 80% of duplications for simulations of copy number between 3 and 8. However for copy number 10 duplications this provided marginal improvements over CC alone, providing solutions for 24 fewer simulations than CC, resolving on average 7.1 duplications per simulation.

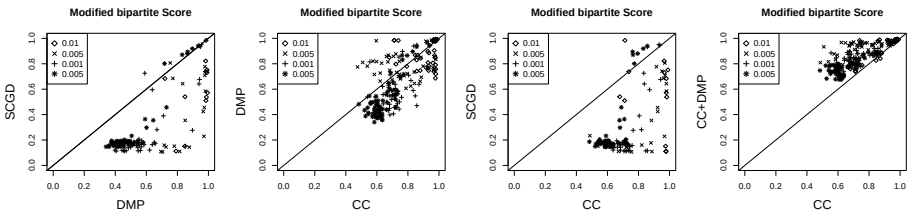

Fig. 4. Matching statistics for the DMP, CC, SCGD, and CC+DMP methods. A perfect reconstruction of haplotypes shows a score of 1, while a random assignment will score 1/ ploidy. Higher matching score is better.

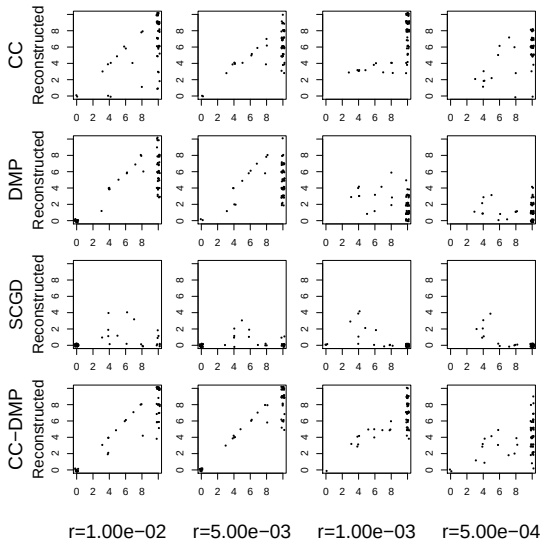

Fig. 5. Correctly assembled haplotypes for the DMP, CC, SCGD, and CC+DMP methods. Each point is the number of correctly phased genotypes per simulation, with points jittered for display

5 Conclusions

The resolution of segmental duplications remains problematic in *de novo* assemblies. Deviating from the typical formulations of *de novo* assembly, we present a new formulation and two novel algorithms for resolving high copy collapsed duplications that rely on polyploid phasing. We demonstrated that while it is possible to optimize for minimal error correction, methods that focus on resolving clusters with unique paralog specific variants actually resolve more duplications despite having a higher minimal error correction value, perhaps due to less over-fitting of results to variants present in ancestral copies of a duplication. In future work we hope to improve the the rank estimation for the discrete matrix completion method, possibly leveraging the clusters discovered by correlation clustering, and characterizing the conditions under which correlation clustering converges to the correct clusters. Finally we plan on applying these methods to resolving duplications in published human assemblies [35, 38].

References

1. Derek Aguiar and Sorin Istrail. Haplotype assembly in polyploid genomes and identical by descent shared tracts. *Bioinformatics*, 29(13):i352–i360, 2013.
2. Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

3. Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.
4. Vikas Bansal and Vineet Bafna. Hapcut: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24(16):i153–i159, 2008.
5. Emily Berger, Deniz Yorukoglu, Jian Peng, and Bonnie Berger. Haptree: A novel bayesian framework for single individual polyplotyping using ngs data. *PLoS Comput Biol*, 10(3):e1003502, 2014.
6. Konstantin Berlin, Sergey Koren, Chen-Shan Chin, James P Drake, Jane M Landolin, and Adam M Phillippy. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology*, 33(6):623–630, 2015.
7. Paola Bonizzoni, Riccardo Dondi, Gunnar W Klau, Yuri Pirola, Nadia Pisanti, and Simone Zaccaria. On the minimum error correction problem for haplotype assembly in diploid and polyploid genomes. *Journal of Computational Biology*, 2016.
8. Changxiao Cai, Sujay Sanghavi, and Haris Vikalo. Structured low-rank matrix factorization for haplotype assembly. *J. Sel. Topics Signal Processing*, 10(4):647–657, 2016.
9. Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
10. Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Commun. ACM*, 55(6):111–119, 2012.
11. Mark J Chaisson. https://github.com/mchaisso/blasr.
12. Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 524–533. IEEE, 2003.
13. Yuxin Chen, Govinda Kamath, Changho Suh, and David Tse. Community recovery in graphs with locality. *arXiv preprint arXiv:1602.03828*, 2016.
14. Shreepriya Das and Haris Vikalo. Sdhap: Haplotype assembly for diploids and polyploids via semi-definite programming. *BMC Genomics*, 16(1), 4 2015.
15. Erik D Demaine and Nicole Immorlica. Correlation clustering with partial information. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 1–13. Springer, 2003.
16. Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
17. Megan Y Dennis, Xander Nuttle, Peter H Sudmant, Francesca Antonacci, Tina A Graves, Mikhail Nefedov, Jill A Rosenfeld, Saba Sajjadian, Maika Malig, Holland Kotkiewicz, et al. Evolution of human-specific neural srgap2 genes by incomplete segmental duplication. *Cell*, 149(4):912–922, 2012.
18. Evan E Eichler. Recent duplication, domain accretion and the dynamic mutation of the human genome. *TRENDS in Genetics*, 17(11):661–669, 2001.
19. Dotan Emanuel and Amos Fiat. Correlation clustering–minimizing disagreements on arbitrary weighted graphs. In *European Symposium on Algorithms*, pages 208–220. Springer, 2003.
20. Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
21. David Gordon, John Huddleston, Mark JP Chaisson, Christopher M Hill, Zev N Kronenberg, Katherine M Munson, Maika Malig, Archana Raja, Ian Fiddes,

LaDeana W Hillier, et al. Long-read sequence assembly of the gorilla genome. *Science*, 352(6281):aae0344, 2016.

22. Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 665–674, New York, NY, USA, 2013. ACM.
23. Zhaoshi Jiang, Haixu Tang, Mario Ventura, Maria Francesca Cardone, Tomas Marques-Bonet, Xinwei She, Pavel A Pevzner, and Evan E Eichler. Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. *Nature genetics*, 39(11):1361–1368, 2007.
24. Sergey Koren, Brian P Walenz, Konstantin Berlin, Jason R Miller, and Adam M Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *bioRxiv*, page 071282, 2016.
25. Giuseppe Lancia, Vineet Bafna, Sorin Istrail, Ross Lippert, and Russell Schwartz. Snps problems, complexity, and algorithms. In *European symposium on algorithms*, pages 182–193. Springer, 2001.
26. Abolfazl Motahari, Kannan Ramchandran, David Tse, and Nan Ma. Optimal dna shotgun sequencing: Noisy reads are as good as noiseless reads. *arXiv preprint arXiv:1304.2798*, 2013.
27. Eugene W Myers. Toward simplifying and accurately formulating fragment assembly. *Journal of Computational Biology*, 2(2):275–290, 1995.
28. Gene Myers. Efficient local alignment discovery amongst noisy long reads. In *International Workshop on Algorithms in Bioinformatics*, pages 52–67. Springer, 2014.
29. Murray Patterson, Tobias Marschall, Nadia Pisanti, Leo van Iersel, Leen Stougie, Gunnar W Klau, and Alexander Schönhuth. Whatshap: Haplotype assembly for future-generation sequencing reads. In *International Conference on Research in Computational Molecular Biology*, pages 237–249. Springer, 2014.
30. Pavel A. Pevzner. Dna physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13(1-2):77–105, 1995.
31. Pavel A Pevzner, Haixu Tang, and Michael S Waterman. An eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753, 2001.
32. Zrinka Puljiz and Haris Vikalo. Decoding genetic variations: Communications-inspired haplotype assembly. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(3):518–530, 2016.
33. Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, August 2010.
34. Russell Schwartz et al. Theory and algorithms for the haplotype assembly problem. *Communications in Information & Systems*, 10(1):23–38, 2010.
35. Jeong-Sun Seo, Arang Rhie, Junsoo Kim, Sangjin Lee, Min-Hwan Sohn, Chang-Uk Kim, Alex Hastie, Han Cao, Ji-Young Yun, Jihye Kim, et al. De novo assembly and phasing of a korean human genome. *Nature*, 2016.
36. Hongbo Si, Haris Vikalo, and Sriram Vishwanath. Haplotype assembly: An information theoretic view. In *Information Theory Workshop (ITW), 2014 IEEE*, pages 182–186. IEEE, 2014.
37. Pawel Stankiewicz and James R Lupski. Genome architecture, rearrangements and genomic disorders. *TRENDS in Genetics*, 18(2):74–82, 2002.

38. Karyn Meltz Steinberg, Tina Graves-Lindsay, Valerie A Schneider, Mark JP Chaisson, Chad Tomlinson, John L Huddleston, Patrick Minx, Milinn Kremitzki, Derek Albrecht, Vincent Magrini, et al. High-quality assembly of an individual of yoruban descent. *bioRxiv*, page 067447, 2016.
39. Christina L Usher, Robert E Handsaker, Tõnu Esko, Marcus A Tuke, Michael N Weedon, Alex R Hastie, Han Cao, Jennifer E Moon, Seva Kashin, Christian Fuchsberger, et al. Structural forms of the human amylase locus and their relationships to snps, haplotypes and obesity. *Nature genetics*, 47(8):921–925, 2015.
40. Max Welling and Kenichi Kurihara. Bayesian k-means as a maximization-expectation algorithm, 2007.

6 Appendix

After each gradient step, the resultant matrix is projected onto the box. The updates for A and B are as follows :

$$\tilde{A}^{(t+1)} \leftarrow A^{(t)} - \alpha_A \nabla_A f(A)$$

$$\text{Then } A_{ij}^{(t+1)} = \begin{cases} 0, & \text{if } \tilde{A}_{ij}^{(t+1)} < 0 \\ \tilde{A}_{ij}^{(t+1)}, & \text{if } 0 \leq \tilde{A}_{ij}^{(t+1)} \leq 1 \\ 1, & \text{if } \tilde{A}_{ij}^{(t+1)} > 1 \end{cases}$$

$$\tilde{B}^{(t+1)} \leftarrow B^{(t)} - \alpha_B \nabla_A f(B)$$

$$\text{Then } B_{ij}^{(t+1)} = \begin{cases} -1, & \text{if } \tilde{B}_{ij}^{(t+1)} < -1 \\ \tilde{B}_{ij}^{(t+1)}, & \text{if } -1 \leq \tilde{A}_{ij}^{(t+1)} \leq 1 \\ 1, & \text{if } \tilde{A}_{ij}^{(t+1)} > 1 \end{cases}$$

where $f(\cdot)$ is the objective function. The projected gradient descent allows us to incorporate additional constraints on the problem as well. If we further enforce that the sum of each row of A equals 1, then we would have the projection as $A_{ij}^{(t+1)} = \max\{0, \tilde{A}_{ij}^{(t+1)} - \nu_i\}$ where ν_i can be computed for each row i using the equality

$$\sum_{j=1}^{S} \max\{0, \tilde{A}_{ij}^{(t+1)} - \nu_i\} = 1$$

We allow a maximum of 50 iteration steps for minimizing each of A and B, and 100 iteration steps for alternating minimization. We exit the iterations if the change in norm is insignificant ($1e-02$) or if the objective value change is below a tolerance ($1e-04$). The learning rate values have to be computed in order to ensure that gradient steps do not diverge. Our choices of learning rates have been

$$\alpha_A = C \frac{\|\nabla f(A^{(t)})\|_F^2}{\|\mathcal{P}_\Omega(\nabla f(A^{(t)}) \cdot B^{(t)})\|_F^2}$$

and

$$\alpha_B = C \frac{\|\nabla f(B^{(t)})\|_F^2}{\|\mathcal{P}_\Omega(A^{(t)} \cdot \nabla f(B^{(t)}))\|_F^2}$$

where $C \in (0, 1)$.