bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Scaffolding of long read assemblies using long range contact information

Jay Ghurye [1], Mihai Pop [1], Sergey Koren[2] and Chen-Shan Chin[3,*]

[1] Department of Computer Science, University of Maryland, College Park, 20740, USA
[2] National Human Genome Research Institute, Bethesda, USA 21702 and
[3] Pacific Biosciences, Menlo Park, 94025 ,USA

* To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Long read technologies have made a revolution in *de novo* genome assembly by generating contigs of size orders of magnitude more than that of short read assemblies. Although the assembly contiguity has increased, it still does not span a chromosome or an arm of the chromosome, resulting in an unfinished chromosome level assembly. To address this problem, we develop a scalable and computationally efficient scaffolding method that can boost the contiguity of the assembly by a large extent using genome wide chromatin interaction data such as Hi-C. Particularly, we demonstrate an algorithm that uses Hi-C data for longer-range scaffolding of *de novo* long read genome assemblies.
Results: We tested our methods on two long read assemblies of different organisms. We compared our method with previously developed method and show that our approach performs better in terms of accuracy of scaffolding.
Availability: The software is available for free use and can be downloaded from here: https://github.com/machinegun/hi-c-scaffold
Contact: jchin@pacificbiosciences.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The advent of massively parallel sequencing technologies has made generation of billions of reads possible at a very low cost per sequenced base. Despite the progress made in *de novo* assembly algorithms, the quality of short read assemblies is far from the quality necessary for effective further analysis due to fundamental limit- the read length is shorter than repeat lengths for the majority of repeat classes (Nagarajan and Pop, 2009; Bresler *et al.*, 2013). For example, a short read *de novo* assemblies of human genome are highly fragmented compared to the chromosomes of the *H.sapien* reference (Gnerre *et al.*, 2011; Li *et al.*, 2010). Thus, high throughput sequencing technology has reached a point where increasing the number of short reads does not significantly improve assembly quality.

Recent advances in single-molecule sequencing technologies have provided reads almost 100 times longer than second generation methods (Schatz *et al.*, 2010). Most prominently, Pacific Biosciences' single molecule real time (SMRT®) sequencing delivers reads of lengths up to 50 Kbp (Eid *et al.*, 2009) whereas Oxford Nanopore's nanopore sequencing can deliver read lengths greater than 10 Kbp (Lee *et al.*, 2014). Such read lengths drastically reduce the complexity caused by repeats during the assembly process. However, these long reads suffer from low accuracy which requires new algorithms for assembly. Despite the higher error rates, SMRT sequencing has random error model (Koren *et al.*, 2012; Ono *et al.*, 2013) due to which near perfect assembly is possible (Lam *et al.*, 2014). Hence by sampling the genome at sufficient coverage, SMRT sequencing has been used to produce assemblies of unprecedented continuity (Koren *et al.*, 2013; Chin *et al.*, 2013; Ribeiro *et al.*, 2012; Berlin *et al.*, 2015).

Various strategies have been explored to increase the continuity of *de novo* genome assemblies. Some of these strategies are end sequencing of fosmid clones (Gnerre *et al.*, 2011), fosmid clone dilution pool sequencing (Chinwalla *et al.*, 2002), optical mapping (Schwartz *et al.*, 1993; Dong *et al.*, 2013; Shelton *et al.*, 2015; English *et al.*, 2015), linked-read sequencing (Zheng *et al.*, 2016; Zook *et al.*, 2016) and synthetic long reads (McCoy *et al.*, 2014; Koren and Phillippy, 2015; Madoui *et al.*, 2015). Some of the newer technologies like Hi-C use proximity ligation and

© The Author 2016. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

massively parallel sequencing to probe the three-dimensional structure of chromosomes within the nucleus, with interacting regions captured by pair-ended reads (Lieberman-Aiden *et al.*, 2009; Duan *et al.*, 2010). In the data generated by Hi-C protocol, the probability of intrachromosomal contacts is on an average much higher than the that of interchromosomal contacts (Burton *et al.*, 2013; Kaplan and Dekker, 2013). Another important property is that the probability of interaction decays rapidly with increasing genomic distance, even regions separated by several hundred megabases on the same chromosome are more likely to interact than the regions on different chromosomes (Lieberman-Aiden *et al.*, 2009).

In this work, we make use of the genome-wide chromatin interaction data sets generated by the Hi-C protocol to linearly arrange the pre-assembled contigs along entire chromosomes. We develop a scaffolding tool SALSA (**S**imple **A**ssemb**L**y **S**c**A**ffolder) based on a computational method that exploits the information of genomic proximity in Hi-C data sets for long range scaffolding of *de novo* genome assemblies. We tested SALSA on human and goat genome assembly. Our method can produce centromere to telomere scaffolds of chromosomes in most cases and telomere to telomere scaffolds in best cases.

2 Related Work

Several efforts have been made to use Hi-C data to scaffold the 'draft stage' short read assemblies. (Burton *et al.*, 2013) developed a computational approached in their tool LACHESIS that combined Hi-C data with pair ended sequencing data to generate chromosome level scaffolds. They used their methods to scaffold *de novo* assemblies of human, mouse and *Drosophilia Melanogaster*. LACHESIS uses the alignment of Hi-C reads to contigs to clusters contigs into 1 cluster per chromosome with hierarchical clustering. It then orients and orders the contigs in each cluster to obtain final scaffolds. One of the drawbacks of this method is that it needs the number of clusters to be pre-specified. This can not be applied to scaffolding contigs of genomes when the number of chromosomes in the organisms are unknown. (Kaplan and Dekker, 2013) developed a method for scaffolding based on statistical techniques. Their method uses the hierarchical clustering method similar to LACHESIS, but it predicts the number of clusters by itself. The major drawback of their method is that they do not orient the contigs in each cluster, thereby not providing complete information needed for scaffolding. Their method also works with an assumption that all contigs are of the same size, which does not hold true in the case of long read assemblies since there can be contigs which are tens of Mb long whereas there can be shorter contigs generated by the assembler. Since both of these methods rely on hierarchical clustering, it is expensive to compute all vs all link scores for all the contigs thereby causing the scalability issues. Another drawback of both methods is that they do not provide the way to detect and correct misassemblies generated by the assembler. If such errors are not corrected, then that would result in erroneous scaffolds in the end and may also propagate errors across multiple scaffolds causing misjoins.

In our work, we address the issues in the previous methods. Our method does not need the number of clusters to be pre-specified. We also provide an option to detect certain types of misassemblies in contigs before scaffolding them.

3 Methods

3.1 Aligning Hi-C Reads

Hi-C reads were aligned to SMRT read assemblies using BWA (Li and Durbin, 2009) with default parameters. Reads with mapping quality < 30, which included the reads mapped more than once were removed from further analysis. Also, only the read pairs with both reads in the pair aligned to contigs are considered for scaffolding.

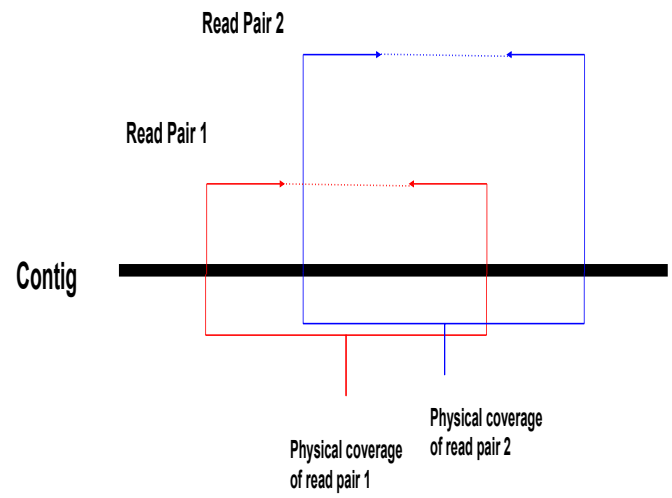

Fig. 1. Physical coverage induced by Hi-C read pair. The solid arrows denote the read pair and dotted line denote the gap between the reads in the pair.

3.2 Detection of Mis-assemblies

Contigs obtained from assemblers may contain mis-assemblies (Phillippy *et al.*, 2008). We provide a method to detect and correct these misassemblies similar to the one described in (Putnam *et al.*, 2016) using the mapping of Hi-C data to the assembled contigs. For each read pair, it's physical coverage is defined as the total bases spanned by the sequence of reads and the gap between the two reads when mapped to contigs (Figure 1). We also define, per base physical coverage for each base in the contig as the number of read pairs' physical coverage it is part of. Using these definitions, we compute the physical coverage for each base of all the contigs in the assembly. The misassembly can be detected by the sudden drop in per-base physical coverage in a contig. A particular threshold below which if per base physical coverage falls for contiguous regions in the genome, we call it a misassembly and break contigs at that point. To do this efficiently, we use a variation of Kadane's algorithm for maximum sum subarray problem(Simon and Kadane, 1975). We find the subarray in the array of physical coverage where coverage is consistently low compared to the adjacent regions and use that as the signal for misassembly.

3.3 Graph Construction and Link Scoring

We use an idea similar to the string graph formulation in (Myers, 2005) to construct the scaffold graph. The scaffold graph $G(V, E, W)$ consists of nodes V which represent the end of contigs, edgs E representing the edges implied by Hi-C read pairs between ends of two contigs and weight function W to assign weight to each edge. The ends of each contig are annotated by two tags, B and E where B stands for the beginning of the contig and E stands for the end of the contig. Using this concept of node, there are 4 types of edges in the graph, BE joining beginning of first contig to end of second, EB joining end of the first contig to beginning of second, BB joining beginning of the first contig to beginning of second contig and EE joining end of the first contig to end of second (Figure 2). Using raw counts of Hi-C read pairs shared between ends of two contigs is not the correct way to score the edges because of length biases since longer contigs with a large genomic distance between them tend to share

bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

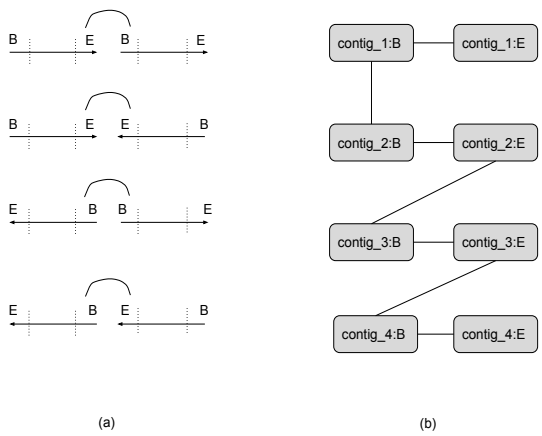

Fig. 2. (a) Type of edges present in the scaffold graph. We consider Hi-C read pairs originating from the end of contigs. Since a contig can be reverse complemented, there are 4 possible links. (b) Scaffold Graph G. There are two nodes in G corresponding to each contig. There is always an edge between 'B' node and 'E' nodes of same contig.

more read pairs compared to two short or one long and one short contig with much lesser genomic distance between them. To address this issue, we define an edge weight function in such a way that it reduces such length biases. We define a length cutoff l and consider the read pairs mapped in the region of length l at both ends (B and E) of contigs. Our normalization of the edge weight is based on how many times the restriction enzyme used in Hi-C protocol cuts the region of length l and divide the counts of read pairs by this number. Putting all this together, the edge weight function is expressed as:

$$W(E) = \frac{links(C1, C2)}{RE(C1) + RE(C2)}$$

where $C1$ and $C2$ are the contigs yielding the edge, $links(C1, C2)$ is the number of hi-c links present in region of length l at the end of contigs and $RE(C1)$ and $RE(C2)$ is the number of sites cut by the restriction enzyme in region l at the end of $C1$ and $C2$. This gives us the normalized edge weights which we use for scaffolding.

Once we calculate the edge weights, we construct graph G as follows. We first sort all the edges in decreasing order of their weights given by W. After this, we remove all the edges which have very low number of read pairs shared between them which account for sequencing errors. Once edges are sorted and filtered, we construct G according to Algorithm 1. We greedily add edges to G only if both the nodes corresponding to the edge are not present in G. In the end, we add edges between B and E ends of same contigs to G which completes the graph construction. G can contain some cycles due to the following reason: we can add an edge between both B and E ends of two contigs (BB and EE edges) and later add edges between B and E ends of same contigs. However, this kind of cycle can easily be removed by removing the lower cost edge among BB and EE edges. Once we remove cycles from G, we get final scaffold graph which we use for further analysis (Figure 2).

3.4 Scaffold Construction

Before explaining the scaffold construction algorithm, we prove following lemmas to understand the properties of G.

Lemma 3.1. *G has no nodes with degree greater than 2.*

Algorithm 1 Scaffold Graph Construction

procedure Construct(L) ▷ L is the list of links sorted by their scores
  $G = ()$ ▷ Scaffold Graph to be returned
  while L is not empty **do**
    l is the current link
    $u, v, w = l.source, l.target, l.weight$
    if $G.has_node(u) == False$ and $G.has_node(v) == False$ **then**
      $G.add_edge(u, v, w)$
    $L.remove(l)$
  for $c \in C$ **do** ▷ C is the set of all contigs
    $G.add_edge(c_B, c_E)$
  return G

Proof: While constructing G, we add edges at most twice for each node. First when we have no edge associated to that node and second when we add an edge between B and E ends of the contig associated with that node. If some node has degree greater than 2, it would mean that we added an edge to that node apart from the cases described previously , which is a contradiction.

Lemma 3.2. *Each connected component of G has exactly two nodes of degree 1.*

Proof: We know from the construction of G that G has no cycles. We can prove this for some connected component C of G and the argument can be applied to all connected components. Since G has no cycles, C will have at least one node with degree 1. In the first case, suppose C has exactly one node of degree 1. This implies that we have at least two edges originating from all other nodes in C. It would mean that there exists at least one node in C with a degree at least 3. This is a contradiction because of lemma 3.1. In the second case, suppose C has more than two nodes of degree 1. It would mean that there exists at least one node in C with degree 3. This is again a contradiction due to lemma 3.1.

Knowing these properties about G we construct scaffolds as described in Algorithm 2. First, a threshold th is decided for a scaffold to qualify as a seed scaffold. If a scaffold has a number of contigs greater than th, it is marked as seed scaffold. For each connected component of G, we first find out two nodes u and v with degree 1. We will always find such nodes due to lemma 3.2. After this, the path connecting u and v is found in the connecting component. Since all the nodes in the connected component have degree either 1 or 2, there will always be just 1 path connecting u and v. If this path has the number of contigs greater than th, we mark this path as seed scaffold, otherwise, it is marked as the small scaffold.

Even after edge weight normalization, there can still be some length biases resulting in the omission of smaller contigs from the seed scaffolds. To account for this, we develop a method to insert the contigs in small scaffolds into seed scaffolds. First, for each contig in small scaffolds, exactly one seed scaffold is assigned to it based on the total edge weight of the edges in the original scaffold graph connecting this contig to all the contigs in seed scaffold. After this, each contig is tested for insertion into its corresponding seed scaffolds in both the orientations at all possible position. It is inserted at the position where it maximizes the total weight of the scaffold. Once all the contigs are inserted into seed scaffolds, it leaves us with the final scaffolds. The algorithm is sketched in Algorithm 2.

4 Dataset

For NA 12878, a human genome used in 1000 genomes project (Siva, 2008), we used the assembly generated at Icahn School of Medicine (Genebank assembly accession GCA_001013985.1). This assembly was

bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

Algorithm 2 Scaffold Construction

```
procedure Get-Seed-Scaffold(G)                    ▷ G is scaffold graph
    C = Connected_Components(G)
    S = List of seed scaffolds
    T = List of small scaffolds
    for c ∈ C do
        u, v = nodes with degree 1 in c
        path = c.find_path(u, v)
        if path.length ≥ th then
            S.add(path)
        else
            T.add(path)
    return S, T
procedure Generate-Scaffold(G)                    ▷ G is scaffold graph
    S, T = Get-Seed-Scaffold(G)
    for t ∈ T do
        Assign seed scaffold from S for contigs in t
    for t ∈ T do
        for c ∈ t do                              ▷ c is contig in t
            S_c = seed scaffold of c
            Place c on S_c so that sum of edges in of S_c is maximized
    Return updated seed scaffolds as final scaffolds
```

performed using Celera Assembler (Koren *et al.*, 2012) and had 21235 contigs with N50 of 1.55 Mb. For the scaffolding of NA 12878 assembly, Hi-C data produced from human ESCs (hESCs) (Dixon *et al.*, 2012) was used. The hESC replicates 1 and 2 were used (NCBI SRA Accession: GSM862723, GSM892306), consisting of total 734M read-pairs. For *Capra hircus* genome we used the data presented in (Bickhart *et al.*, 2016). *Capra hircus* is a domestic goat of San Clemente breed. The assembly performed using long reads with CA PBcR pipeline had contig NG50 of 4.15 Mb. These contigs were scaffolded using Irys optical mapping data, resulting in increased scaffold size. All the sequencing data and intermediate assemblies are publically available [1]. We used Hi-C data (NCBI SRA Accession: SRX1910977) with both contig assembly and Irys optical map scaffolds to perform scaffolding and compared the extended scaffolds with the scaffolds given by LACHESIS.

5 Results

5.1 Contact Probability of Hi-C data

We aligned Hi-C reads for NA12878 to GRCh38 human reference using BWA mem (Li and Durbin, 2009) with default parameters. If both mates in the read pair are aligned to the same chromosome, it provides an evidence that Hi-C data gives valuable information about the intrachromosomal contact. For each chromosome, we count how many read pair have both mates mapped to that chromosome and how many reads have just one of the mates mapped to that chromosome. Using this information, we compute the probability of intrachromosomal and interchromosomal contact for each chromosome. It can be seen from Figure 3 that the probability of intrachromosomal contact is much higher than that of interchromosomal contact.

To further understand the contact probability at contig level, we aligned reads to NA12878 contigs. We also aligned contigs to GRCh38 reference using Mummer (Delcher *et al.*, 1999). For each contig that mapped to chromosome 1, we calculated its average interaction frequency to contigs belonging to the all other chromosomes. Figure 4 shows the box plot of the distribution of average interaction frequency of all contigs of chromosome 1 among themselves and with the contigs in all other chromosomes. We found that the average interaction frequency strongly separates inter from intra-chromosomal interactions.

[1] Assemblies and sequencing data information can be found here: https://gembox.cbcb.umd.edu/goat/index.html

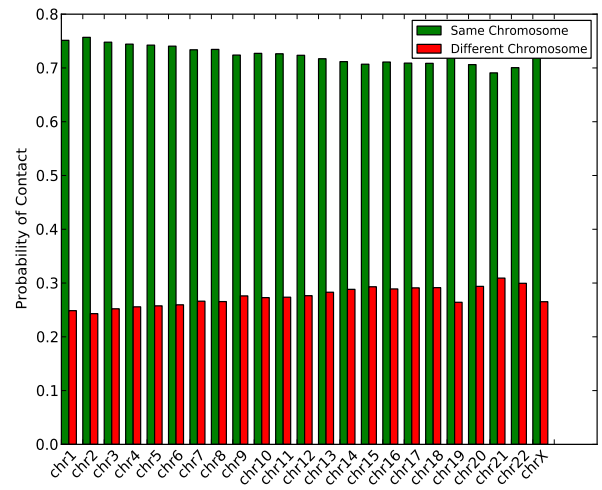

Fig. 3. The probability of contact calculated based on read mapping to GRCh38 reference

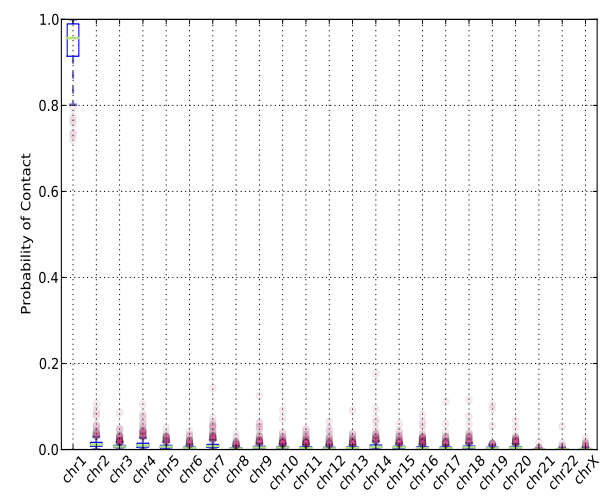

Fig. 4. The probability of contact of all the contigs in chromosome 1 to all other contigs.

Metric	NA12878	*Capra hircus*
Number of Contigs	21235	33767
Contig N50	1.55 Mb	3.86 Mb
Number of Scaffolds	1555	127
Scaffold NG50	60.02 Mb	58.64 Mb
Total Bases	2.77 Gb	2.94 Gb

Table 1. Contig and Scaffold statistics for NA 12878 and Capra hirrus

bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

5.2 Scaffolding assemblies of two genomes

We tested the effectiveness of our approach for the chromosome scale *de novo* assembly of two genomes . We started with two assemblies, one for NA 12878 with contig N50 of 1.55 Mb and the other for goat genome with contig N50 of 4.15 Mb. After aligning Hi-C read pairs to these assemblies, we used our algorithm to construct the scaffold graph and later to orient and order contigs. For NA12878 assembly, it gave 1555 scaffolds with NG50 of 60.02 Mb. For *Capra hircus* assembly, it gave 127 scaffolds with NG50 of 68.64 Mb. Table 1 shows the statistics of the contigs and scaffolds.

5.3 Comparison with LACHESIS

To evaluate the quality of the scaffolds, we aligned NA12878 scaffolds to human GRCh38 reference. For goat scaffolds, we aligned them to recently published goat reference genome (BioProject PRJNA290100) (Bickhart *et al.*, 2016) using *nucmer* program (parameters : -c 1000) in MUMmer package (Delcher *et al.*, 1999). The quality of alignments was assessed using *dnadiff* program (Phillippy *et al.*, 2008), which evaluates the draft assemblies by comparing with the reference genome based on a set of metrics. We particularly focus on four metrics. First one is a number of breakpoints which is defined as the number of alignments which are not end to end with reference to a particular scaffold. A second metric is a number of relocations which is the number of breaks in the alignment where adjacent aligned sequences are in the same sequence but not the correct order. This accounts of ordering errors in the scaffold construction. A third metric is a number of translocations, which counts the number of breaks in the alignment where adjacent sequences are in different chromosomes. This accounts for the inter-chromosomal join errors in the scaffold. A last metric is a number of inversions, which is the number of breaks in the alignment where adjacent sequences are inverted with respect to each other. This accounst for the orientation errors caused due to scaffolding algorithm.

Table 2 shows the comparison of scaffolds generated by SALSA and LACHESIS for NA12878 assembly. For the scaffolds generated by SALSA, 80.82% bases could align to reference whereas only 40.70% bases from the scaffolds generated by LACHESIS could align to the reference. Although the number of bases aligned to the reference is significantly low in LACHESIS scaffolds, the relocation, translocation and inversion errors are much higher in LACHESIS compared to SALSA. Particularly, the number of relocations is significantly high in LACHESIS implying that LACHESIS performs a lot of errors in ordering contigs belonging to a particular chromosome(Bickhart *et al.*, 2016).

Table 3 shows the comparison of scaffolds generated by SALSA and LACHESIS for *Capra hircus* assembly. In this case, both the scaffolds have an almost similar percent of aligned bases to reference. Even in this case, LACHESIS performs a lot of orientation and ordering errors compared to SALSA. SALSA produces 67 and 105 orientation and ordering errors in the scaffolds, which is significantly lesser than 374 and 439 orientation and ordering errors produced by LACHESIS. In addition to this, SALSA produces lesser inter-chromosomal joins(213) compared to LACHESIS(601).

Figure 5 and Figure 6 show the alignment dotplot for human and goat scaffolds respectively when aligned to respective reference genomes. In the case of NA12878 scaffolds, it can be seen that (Figure 5 A) the scaffolds generated by LACHESIS are very fragmented and lack continuity. The scaffolds generated by SALSA (Figure 5 B) are continuous, contain lesser orientation and ordering errors and much more consistent with the reference than LACHESIS scaffolds. Alignment dot plots for each chromosome can be found in the supplementary material (Supplementary Figure S1). In the case of goat scaffolds, it can be seen that although LACHESIS produces continuous scaffolds, it incurs a lot of large scale orientation and ordering errors (Figure 6 A). In contrast, SALSA is able to produce the continuous scaffolds with lesser amount of orientation and ordering errors compared to LACHESIS thereby producing scaffolds that are consistent with the reference to a large extent.

Metric	SALSA	LACHESIS
Number of Scaffolds	1555	23
Total Bases	2.92 Gb	2.79 Gb
% Aligned Bases	80.82%	40.70%
Breakpoints	14010	6820
Relocations	19	76
Translocations	4	9
Inversions	42	65

Table 2. Evaluation of scaffolds generated by SALSA and LACHESIS for the human NA12878 assembly

Metric	SALSA	LACHESIS
Number of Scaffolds	127	990
Total Bases	2.44 Gb	2.62 Gb
% Aligned Bases	99.88%	99.85%
Breakpoints	8514	14035
Relocations	67	374
Translocations	213	601
Inversions	105	439

Table 3. Evaluation of scaffolds generated by SALSA and LACHESIS for the Capra hircus assembly

Metric	SALSA	LACHESIS
Number of Scaffolds	90	596
Total Bases	2.22 Gb	2.61 Gb
NG50	46.64 Mb	87.34 Mb
% Aligned Bases	97.00%	99.91%
Breakpoints	10718	14741
Relocations	118	167
Translocations	185	407
Inversions	130	126

Table 4. Evaluation of scaffolds generated by SALSA and LACHESIS for Capra hircus assembly generated using optical map data

5.4 Scaffolding Optical Map Scaffolds

(Bickhart *et al.*, 2016) also made scaffolds of goat assembly generated using optical map data available publically. We used these scaffolds as a starting point for our scaffolding method and did further scaffolding using Hi-C data. The initial scaffold assembly had 1575 scaffolds with N50 of 23.08 Mb. After scaffolding Hi-C data with SALSA, it gave 90 scaffolds, and scaffolding with LACHESIS gave 596 scaffolds. We evaluated these scaffold using the metrics described before and the results are shown in Table 4. It can be observed that even though the initial N50 of the input assembly is high, LACHESIS is still prone to ordering errors compared to SALSA. However, in this case, SALSA produced 3.17% more orientation errors than LACHESIS. The alignment dotplots for both the scaffolds are shown in supplementary material (Supplementary Figure S2).

6 Conclusion and Discussion

In this work, we show that genome-wide interaction data sets like Hi-C are a good source of information to scaffold the pre-assembled contigs into scaffolds. Since we use long read assemblies at the start which mostly span through highly repetitive regions in the genome, we do not need to

bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

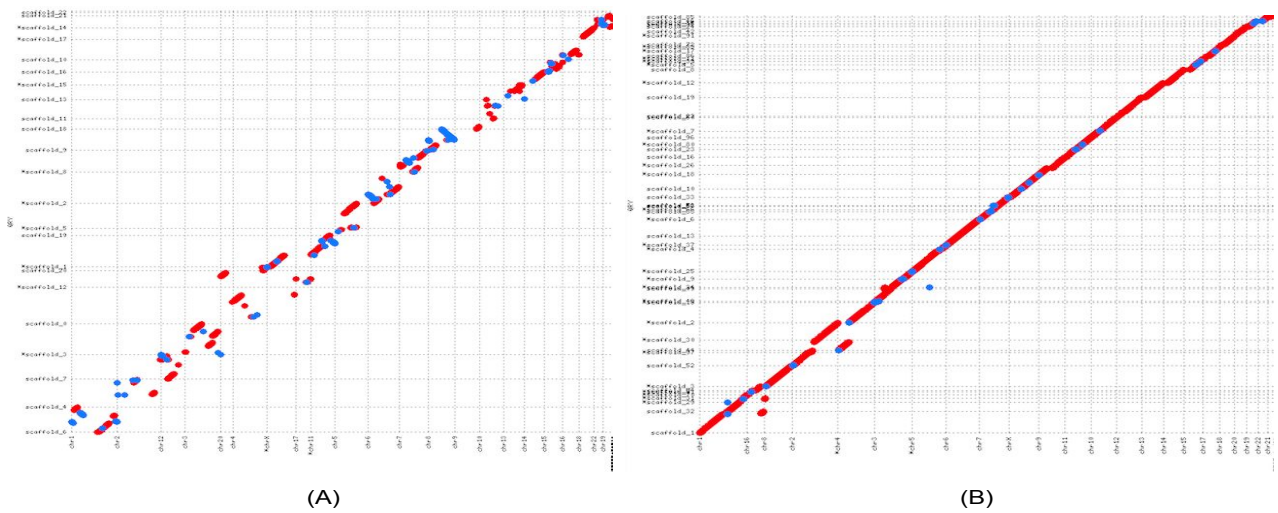

Fig. 5. (A) The alignment dotplot of Lachesis scaffolds for NA 12878.(B) The alignment dotplot for NA12878 scaffolds generated by SALSA.

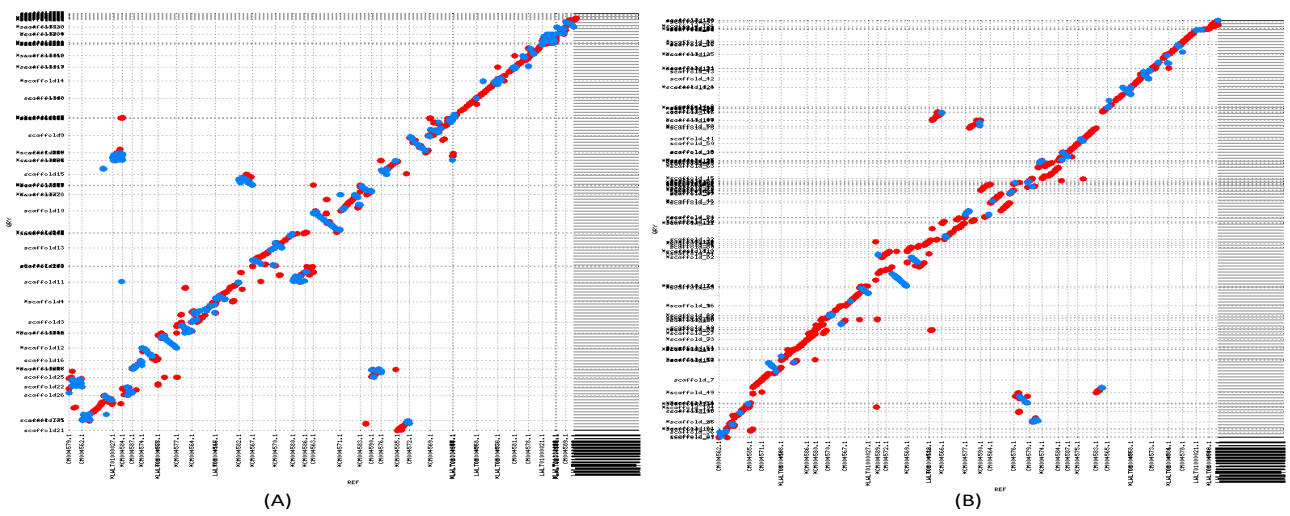

Fig. 6. (A) The alignment dotplot of Lachesis scaffolds for Capra hircus.(B) The alignment dotplot for Captra hircus scaffolds generated by SALSA.

do repeat masking as previous methods and hence derive more continuous scaffolds. We derive a weight function to normalize the scores of Hi-C links which reduces the length biases caused by long contigs. Since long read assemblies have non-uniform contig lengths, edge weight normalization plays an important role in deriving correct scaffolds. We provide a method to correct misassemblies in the starting assembly so that these errors do not propagate through the further scaffolding process. We used string graph-based approach along with some greedy heuristics to derive scaffolds. We tested our methods on assemblies of two different organisms namely human and goat. Our method showed significant improvements on the same dataset over previously used method. Since our method does not need the number of chromosomes to be specified apriori, it can be applied for scaffolding the assemblies of organisms where we don't know the number of chromosomes beforehand. However, designing a clustering method that clusters the contigs without knowing the actual number of desired clusters is needed to estimate the number of chromosomes in an unknown organism. To avoid errors in clustering, we can first orient and order the contigs with respect to each other and cluster the resulting scaffolds as we would have larger and more continuous parts of the genome to cluster compared to contigs which are smaller portions of the genome. Most of the orientation and ordering errors in our method were in the repetitive regions near centromeres and telomeres. One potential solution to overcome this problem is to do all pairwise alignment of contigs and trim the contigs so that these repetitive regions are masked. However, there is a serious computational bottleneck associated to this.

There are still several open questions. Our method needs some parameter tuning to get good scaffolds. We plan to incorporate optimal parameter detection at runtime so that the onus of parameter tuning is taken away from the user of the tool. There are other chromatin interaction datasets like Dovetail Chicago libraries (Putnam *et al.*, 2016) developed recently. We plan to extend our method so that it becomes generic to all such datasets and adapts to their chromosomal contact model.

bioRxiv preprint doi: https://doi.org/10.1101/083964; this version posted October 27, 2016. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under a CC-BY-NC-ND 4.0 International license.

Acknowledgements

We thank Ashton Trey Belew for helping with running LACHESIS tool for comparative analysis. We also thank Chris M. Hill for helpful discussions.

Funding

Sergey Koren was supported by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health. This study utilized the computational resources of the Biowulf system at the National Institutes of Health, Bethesda, MD (http://biowulf.nih.gov).

Competing Financial Interest

Chen-Shan Chin is an employee and shareholder of Pacific Biosciences, a company commercializing DNA sequencing technologies.

References

Berlin, K., Koren, S., Chin, C.-S., Drake, J. P., Landolin, J. M., and Phillippy, A. M. (2015). Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology*, **33**(6), 623–630.

Bickhart, D. M., Rosen, B. D., Koren, S., Sayre, B. L., Hastie, A. R., Chan, S., Lee, J., Lam, E. T., Liachko, I., Sullivan, S. T., Burton, J. N., Huson, H. J., Kelley, C. M., Hutchison, J. L., Zhou, Y., Sun, J., Crisa, A., Ponce de Leon, F. A., Schwartz, J. C., Hammond, J. A., Waldbieser, G. C., Schroeder, S. G., Liu, G. E., Dunham, M. J., Shendure, J., Sonstegard, T. S., Phillippy, A. M., Van Tassell, C. P., and Smith, T. P. (2016). Single-molecule sequencing and conformational capture enable de novo mammalian reference genomes. *bioRxiv*.

Bresler, G., Bresler, M., and Tse, D. (2013). Optimal assembly for high throughput shotgun sequencing. *BMC bioinformatics*, **14**(5), 1.

Burton, J. N., Adey, A., Patwardhan, R. P., Qiu, R., Kitzman, J. O., and Shendure, J. (2013). Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nature biotechnology*, **31**(12), 1119–1125.

Chin, C.-S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., Clum, A., Copeland, A., Huddleston, J., Eichler, E. E., *et al.* (2013). Nonhybrid, finished microbial genome assemblies from long-read smrt sequencing data. *Nature methods*, **10**(6), 563–569.

Chinwalla, A. T., Cook, L. L., Delehaunty, K. D., Fewell, G. A., Fulton, L. A., Fulton, R. S., Graves, T. A., Hillier, L. W., Mardis, E. R., McPherson, J. D., *et al.* (2002). Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**(6915), 520–562.

Delcher, A. L., Kasif, S., Fleischmann, R. D., Peterson, J., White, O., and Salzberg, S. L. (1999). Alignment of whole genomes. *Nucleic acids research*, **27**(11), 2369–2376.

Dixon, J. R., Selvaraj, S., Yue, F., Kim, A., Li, Y., Shen, Y., Hu, M., Liu, J. S., and Ren, B. (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, **485**(7398), 376–380.

Dong, Y., Xie, M., Jiang, Y., Xiao, N., Du, X., Zhang, W., Tosser-Klopp, G., Wang, J., Yang, S., Liang, J., *et al.* (2013). Sequencing and automated whole-genome optical mapping of the genome of a domestic goat (capra hircus). *Nature biotechnology*, **31**(2), 135–141.

Duan, Z., Andronescu, M., Schutz, K., McIlwain, S., Kim, Y. J., Lee, C., Shendure, J., Fields, S., Blau, C. A., and Noble, W. S. (2010). A three-dimensional model of the yeast genome. *Nature*, **465**(7296), 363–367.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., *et al.* (2009). Real-time dna sequencing from single polymerase molecules. *Science*, **323**(5910), 133–138.

English, A. C., Salerno, W. J., Hampton, O. A., Gonzaga-Jauregui, C., Ambreth, S., Ritter, D. I., Beck, C. R., Davis, C. F., Dahdouli, M., Ma, S., *et al.* (2015). Assessing structural variation in a personal genomeâŁ"towards a human reference diploid genome. *BMC genomics*, **16**(1), 1.

Gnerre, S., MacCallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., Sharpe, T., Hall, G., Shea, T. P., Sykes, S., *et al.* (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences*, **108**(4), 1513–1518.

Kaplan, N. and Dekker, J. (2013). High-throughput genome scaffolding from in vivo dna interaction frequency. *Nature biotechnology*, **31**(12), 1143–1147.

Koren, S. and Phillippy, A. M. (2015). One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly. *Current opinion in microbiology*, **23**, 110–120.

Koren, S., Schatz, M. C., Walenz, B. P., Martin, J., Howard, J. T., Ganapathy, G., Wang, Z., Rasko, D. A., McCombie, W. R., Jarvis, E. D., *et al.* (2012). Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature biotechnology*, **30**(7), 693–700.

Koren, S., Harhay, G. P., Smith, T. P., Bono, J. L., Harhay, D. M., Mcvey, S. D., Radune, D., Bergman, N. H., and Phillippy, A. M. (2013). Reducing assembly complexity of microbial genomes with single-molecule sequencing. *Genome biology*, **14**(9), 1.

Lam, K.-K., Khalak, A., and Tse, D. (2014). Near-optimal assembly for shotgun sequencing with noisy reads. *BMC bioinformatics*, **15**(Suppl 9), S4.

Lee, H., Gurtowski, J., Yoo, S., Marcus, S., McCombie, W. R., and Schatz, M. (2014). Error correction and assembly complexity of single molecule sequencing reads. *BioRxiv*, page 006395.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z., Li, Y., Li, S., Shan, G., Kristiansen, K., *et al.* (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome research*, **20**(2), 265–272.

Lieberman-Aiden, E., Van Berkum, N. L., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., Amit, I., Lajoie, B. R., Sabo, P. J., Dorschner, M. O., *et al.* (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science*, **326**(5950), 289–293.

Madoui, M.-A., Engelen, S., Cruaud, C., Belser, C., Bertrand, L., Alberti, A., Lemainque, A., Wincker, P., and Aury, J.-M. (2015). Genome assembly using nanopore-guided long and error-free dna reads. *BMC genomics*, **16**(1), 1.

McCoy, R. C., Taylor, R. W., Blauwkamp, T. A., Kelley, J. L., Kertesz, M., Pushkarev, D., Petrov, D. A., and Fiston-Lavier, A.-S. (2014). Illumina truseq synthetic long-reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. *PloS one*, **9**(9), e106689.

Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics*, **21**(suppl 2), ii79–ii85.

Nagarajan, N. and Pop, M. (2009). Parametric complexity of sequence assembly: theory and applications to next generation sequencing. *Journal of computational biology*, **16**(7), 897–908.

Ono, Y., Asai, K., and Hamada, M. (2013). Pbsim: Pacbio reads simulatorâŁ"toward accurate genome assembly. *Bioinformatics*, **29**(1), 119–121.

Phillippy, A. M., Schatz, M. C., and Pop, M. (2008). Genome assembly forensics: finding the elusive mis-assembly. *Genome biology*, **9**(3), 1.

Putnam, N. H., O'Connell, B. L., Stites, J. C., Rice, B. J., Blanchette, M., Calef, R., Troll, C. J., Fields, A., Hartley, P. D., Sugnet, C. W., *et al.* (2016). Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. *Genome research*, **26**(3), 342–350.

Ribeiro, F. J., Przybylski, D., Yin, S., Sharpe, T., Gnerre, S., Abouelleil, A., Berlin, A. M., Montmayeur, A., Shea, T. P., Walker, B. J., *et al.* (2012). Finished bacterial genomes from shotgun sequence data. *Genome research*, **22**(11), 2270–2277.

Schatz, M. C., Delcher, A. L., and Salzberg, S. L. (2010). Assembly of large genomes using second-generation sequencing. *Genome research*, **20**(9), 1165–1173.

Schwartz, D. C., Li, X., Hernandez, L. I., Ramnarain, S. P., Huff, E. J., and Wang, Y.-K. (1993). Ordered restriction maps of saccharomyces cerevisiae chromosomes constructed by optical mapping. *Science*, **262**(5130), 110–114.

Shelton, J. M., Coleman, M. C., Herndon, N., Lu, N., Lam, E. T., Anantharaman, T., Sheth, P., and Brown, S. J. (2015). Tools and pipelines for bionano data: molecule assembly pipeline and fasta super scaffolding tool. *BMC genomics*, **16**(1), 734.

Simon, H. A. and Kadane, J. B. (1975). Optimal problem-solving search: All-or-none solutions. *Artificial Intelligence*, **6**(3), 235–247.

Siva, N. (2008). 1000 genomes project. *Nature biotechnology*, **26**(3), 256–256.

Zheng, G. X., Lau, B. T., Schnall-Levin, M., Jarosz, M., Bell, J. M., Hindson, C. M., Kyriazopoulou-Panagiotopoulou, S., Masquelier, D. A., Merrill, L., Terry, J. M., *et al.* (2016). Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nature biotechnology*.

Zook, J. M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., Weng, Z., Liu, Y., Mason, C. E., Alexander, N., *et al.* (2016). Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data*, **3**.