UNIVERSITY OF CALIFORNIA
RIVERSIDE

Optical Map-Based Genome Scaffolding

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Weihua Pan

September 2019

Dissertation Committee:

Dr. Stefano Lonardi, Chairperson
Dr. Tao Jiang
Dr. Tamar Shinar
Dr. Wenxiu Ma

The Dissertation of Weihua Pan is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I was very lucky to join Ph.D program in Computer Science at University of California, Riverside. In the past five years, I met a lot of outstanding people and improved myself in almost all the aspects.

First and foremost, I am very grateful to my advisor, Dr. Stefano Lonardi, without whose help, I would not have been here. Stefano changed me from a person who thought himself be able to do research to a person who really can do research and like doing research. I will never forget the time I just joined UCR. I was extremely poor in English at that time, and not able to do almost any communication. He was always very patient to me and tried his best to explain every details carefully. When I failed in a few projects and felt frustrated, instead of blaming me, he always encouraged me and gave me confidence. Stefano is a open minded person and gave me even more freedom than I could imagine. He allowed me to do whatever interesting research and take whatever classes I wanted, and was always supportive for me obtaining a statistics master degree during Ph.D period. In addition to research, Stefano is also helpful in life. When I was involved in a car accident and felt helpless, he was the one who stood up and showed me how to solve the problems. Honestly speaking, he is a unique professor and very different from all the educators in my previous life.

I would like to thank Dr. Tao Jiang, Dr. Tamar Shinar, and Dr. Wenxiu Ma serving on my Ph.D dissertation committee. They provided insightful comments and valuable suggestions to help me complete a high quality thesis. I am especially grateful to Dr. Tao Jiang for his help and advice during the five years. He offered great help in the theoretical part of my research work, and always gave me important advice in both study and life.

Without him, I would not have decided to continue my future career in academia.

My thanks goes to all the professors that greatly contributed to my training, besides the ones mentioned above, especially to Dr. Neal Young, Dr. Christian Shelton, Dr. Eamon Keogh, Dr. Rajiv Gupta, Dr. Weixin Yao, Dr. Barry C. Arnold, Dr. Subir Ghosh, Dr. Daniel Jeske and Dr. Shujie Ma. The knowledge they provided me with and the enthusiasm they showed in teaching will affect my future career and life.

I am indebted to all members of Algorithms and Computational Biology Lab at UCR, especially to Dr. Anton Polishko, Dr. Seyed Hamid Mirebrahim, Dr. Rachid Ounit, Dr. Hind Alhakami, Dr. Ei-Wen Yang, Md. Abid Hasan, Abbas Roayaei, Qihua Liang, Dipankar Ranjan Baisya, Hao Chen, Ashraful Arefeen, Dipan Shaw, Miguel Coviello Gonzalez, Huong Luu, Parker Newton and Yugarshi Shashwat. Without them, I could not have enjoyed the five years happy time.

Last but not least, I would like to thank my family members for their love and encouragement. They were always very supportive during my whole study life, and it has always been their dream for me to obtain a Ph.D degree.

To all my family and friends.

ABSTRACT OF THE DISSERTATION

Optical Map-Based Genome Scaffolding

by

Weihua Pan

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, September 2019
Dr. Stefano Lonardi, Chairperson

De novo genome assembly is one of the most critical problems in computational biology. Due to the limitations of current sequencing technologies, the *de novo* assembly is typically carried out in two stages, namely contig (sequence) assembly and scaffolding. The scaffolding process can vastly improve the assembly contiguity and can produce chromosome-level assemblies. Despite significant algorithmic progress, the scaffolding problem can be challenging due to the high repetitive content of eukaryotic genomes, possible mis-joins in assembled contigs and the inaccuracies of the linkage information.

Different types of linkage information such as paired-end/mate-pair/linked/Hi-C reads or genome-wide maps (optical, physical or genetic) are used to carry out the scaffolding process. Optical maps (in particular Bionano Genomics maps) have been extensively used in many recent large-scale genome assembly projects (e.g., goat, apple, barley, maize, quinoa, sea bass, among others).

In this dissertation, we address some of the computational issues associated with genome scaffolding when optical maps are used. We propose novel algorithms for scaffolding,

chimeric detection, and assembly reconciliation. First, we introduce a novel chimeric removal tool called ChimericOgnizer. ChimericOgnizer takes advantage of one or more Bionano Genomics optical maps to accurately detect and correct chimeric contigs. Experimental results show that ChimericOgnizer is very accurate, and significantly better than the chimeric detection method offered by the Bionano Hybrid Scaffold pipeline. ChimericOgnizer can also detect and correct chimeric optical molecules.

Second, we describe a novel method called Novo&Stitch that can take advantage of optical maps to accurately carry out assembly reconciliation. Experimental results demonstrate that Novo&Stitch can double the contiguity (N50) of the input assemblies without introducing mis-joins or reducing genome completeness.

Third, we introduce a scaffolding algorithm called OMGS that for the first time can take advantages of multiple optical maps. OMGS solves several optimization problems to generate scaffolds with optimal contiguity and correctness. Extensive experimental results demonstrate that our tool outperforms existing methods when multiple optical maps are available, and produces comparable scaffolds using a single optical map.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The number of eukaryotic species on this planet is estimated to be about 8.7 million [52] but only a few thousand had their genomes sequenced. Prokaryotes are also likely to number in the millions. Obtaining the complete genome sequence for a species is a fundamental first step in understanding its cellular and molecular processes. However, the current sequencing technology does not allow life scientists to read each chromosome from the beginning to the end. Sequencing instruments can only read short DNA fragments, called *reads*.

There are three generations of sequencing technologies. The first-generation sequencing, also called *Sanger sequencing*, was used from the 1970s to the 1990s. Sanger sequencing produced a scientific revolution in biology and led to the Human Genome Project [87, 17]. Sanger sequencing generates ≈1000bp-long reads with low throughput, high cost, but good accuracy. Because of the limitations of Sanger sequencing, the Human Genome Project took 13 years to complete at a cost of almost US$3 billion. In the mid-to-late 2000s,

second-generation sequencing technologies (Solexa, Illumina, ABI Solid) quickly replaced Sanger sequencing, especially for re-sequencing applications (e.g., RNA-Seq). The second generation has much higher throughput and substantially lower cost [28]. However second-generation sequencing reads are much shorter (100–250bp) than Sanger's DNA sequence with higher error rate (about 1%). In recent years, the third-generation sequencing technologies emerged [72, 68, 47, 55, 86]. The third-generation sequencing technology includes single-molecule real time (SMRT) sequencing from Pacific Biosciences (PacBio) [67] and nanopore-based sequencing from Oxford Nanopore Technologies [36]. Third-generation sequencing technologies produce reads averaging 10kbp in length, with many reads over 100kb and some reaching over 1Mb [75]. However, the error rate of third-generation technologies is much higher (typically 15%) than the second generation.

Genome assembly is the computational problem of assembling the reads into a complete genome sequence. There are two "flavors" of this problem, namely reference-based genome assembly and *de novo* genome assembly. In reference-based genome assembly one has to assemble reads for an organism using a evolutionarily related genome as a reference; in *de novo* genome assembly the problem is to assemble reads of a new species from "scratch" using only the overlaps between reads.

Genome assembly is considered one of the most fundamental problems in computational biology. Due to the current limitations of sequencing instruments, the assembly process is typically carried out in two stages, namely contig (sequence) assembly and scaffolding. Contig assembly is the step assembling reads into longer DNA sequences called *contigs* according to the overlaps between reads. Existing methods for contig assembly can be

classified into two major categories: overlap graph based methods [30, 56, 57, 6, 35, 32, 5, 83] and *de Bruijn* graph based methods [64, 96, 79, 65, 34, 91, 44, 79, 11, 97, 49]. Overlap graph based algorithms assemble reads by first constructing the overlap graph. In the overlap graph each vertex represents a read, and each edge represents an overlap between reads. *De Bruijn* graph based algorithms assemble reads by constructing first a de Bruijn graph. In a de Bruijn graph, each vertex represents a length-k substring (called k-mer), and each edge connects consecutive k-mers in the input read (i.e., two k-mers overlapping $k-1$ bases). Both of the overlap graph based methods and *de Bruijn* graph based methods report maximal simple paths of vertices without branches as contigs [59, 98].

Scaffolds are arrangements of oriented contigs with gaps representing the estimated distance separating them. Gaps indicate genome regions not covered by any contig. Since eukaryotic genomes are very repetitive and repeats are hard (if not impossible) to assemble, assemblies often miss these repetitive regions which are represented as gaps. A *chimeric contig* is contig that has been incorrectly assembled from reads originating from non-adjacent regions of the genome. Irrespective on the type of sequencing technology or the contig assembly algorithms employed, mis-joins are hard to avoid.

The scaffolding process can vastly improve the assembly contiguity and can produce chromosome-level assemblies. Despite significant algorithmic progress, the scaffolding problem can be challenging due to the high repetitive content of eukaryotic genomes, possible mis-joins in assembled contigs and the inaccuracies of the linkage information.

Since contigs are not expected to overlap, scaffolding relies on additional linkage information. Several protocols have been developed to generate different kinds of linkage

information. The most common type of linkage information is in the form of paired-end/mate-pair reads. Paired-end reads are pair of sequenced reads from both ends of the same DNA molecule (tipically $\approx$ 500 base pairs). Mate-pair reads are paired-end reads for which the DNA molecule is much longer (1kbp to 100kbp). For the assembly of large eukaryotic genomes, multiple libraries of mate-pair reads (with different insert sizes) are used to span repetitive regions.

By carrying out sequence alignment, one can anchor paired-end/mate-pair reads to assembled contigs. Since the relative orientations and approximate distances of each pair of reads are known, two contigs anchored by paired-end/mate-pair reads can be oriented and the distance between the contigs can be estimated. However, the distance between each pair of reads is relatively small (hundreds of base pairs), which prevents one to scaffold contigs with large gaps [66, 41, 29, 70, 26, 22, 19, 9, 79, 78]. To solve this problem, long-range linkage information need to be used. Genetic maps provide the order and gentic distances of single-nucleotide polymorphism (SNP) sites, so that the contigs anchored by SNPs can be scaffolded [85]. Hi-C data provides the approximate spacial distances between each pair of regions of genome. Although theoretically two regions in large genomic distance could be very close to each other in space, in most situations, the spacial distance is a good estimation of genomic distance, so that the gaps between contigs can be estimated [10].

The optical map is another type of genome-wide map, which can provide accurate distances between genetic markers. Since its emergence over twenty years ago [74], optical mapping has undergone a transition from laboratory technique to commercially available data generation method. The optical mapping technologies currently on the market (e.g.,

Figure 1.1: BioNano Genomics optical mapping (source [28])

BioNano Genomics Irys systems, OpGen Argus) allow life scientists to produce genome-wide maps by fingerprinting long DNA molecules (up to 1 Mb), via nicking restriction enzymes. Linear DNA fragments are stretched on a glass surface or in a nano-channel array, then the locations of restriction sites are identified with the help of dyes or fluorescent labels (see Figure 1.1). The results are imaged and aligned to each other to map the locations of the restriction sites relative to each other. While the assembly process for optical molecules is highly reliable, there is clear evidence that a small fraction of the optical molecules is chimeric [38].

An *optical map* is composed by a set of optical map *molecules*, each of which is represented by an ordered set of positions for the restriction enzyme sites. By digesting *in silico* the assembled contig using the same restriction enzyme used to produce the optical map and matching the sequence of distances between adjacent sites, one can align assembled contigs to an optical map. High-quality alignments allow some of the contigs in the assembly to be anchored at specific coordinates on the optical map. In addition, contigs can be oriented with respect to each other. When multiple contigs align to the same optical map molecule, an estimate of the distance between them can be obtained. If the distance is positive, a gap is introduced and a *scaffold* can be formed [76, 84]. When the distance is negative (i.e., contigs are overlapping), it may be possible to stitch them. Optical maps can

also be used to detect and break chimeric contigs. When only a fraction of a contig aligns to a molecule, the "overhang" of the contig that is not aligned to molecule is likely to be improperly assembled [84] (see Figure 2.2 for an example).

The focus of this dissertation is to develop innovative algorithmic solutions for improving *de novo* genome scaffolding with the help of optical maps. Specifically, we provide new methods to generate scaffolds with higher contiguity and smaller number of errors (e.g., mis-joins) using BioNano optical maps.

In Chapter 2, we describe a novel chimeric removal tool called ChimericognizER. ChimericognizER takes advantage of one or more Bionano Genomics optical maps to accurately detect and correct chimeric contigs. Experimental results show that ChimericognizER is very accurate, and significantly better than the chimeric detection method offered by the Bionano Hybrid Scaffold pipeline. ChimericognizER can also detect and correct chimeric optical molecules.

In Chapter 3, we introduce a novel method called Novo&Stitch that can take advantage of optical maps to accurately carry out assembly reconciliation. Experimental results demonstrate that Novo&Stitch can double the contiguity (N50) of the input assemblies without introducing mis-joins or reducing genome completeness.

In Chapter 4, we describe a scaffolding algorithm called OMGS that for the first time can take advantages of multiple optical maps. OMGS solves several optimization problems to generate scaffolds with optimal contiguity and correctness. Extensive experimental results demonstrate that our tool outperforms existing methods when multiple optical maps are available, and produces comparable scaffolds using a single optical map.

Chapter 2

Chimericognizer: Accurate Detection of Chimeric Contigs via Bionano Optical Maps

In this chapter, we focus on the problem of detecting and correcting the chimeric contigs. A *chimeric contig* is contig that has been incorrectly assembled from reads originating from non-adjacent regions of the genome. Irrespective on the type of sequencing technology or the contig assembly algorithms employed, mis-joins are hard to avoid. Failing to recognize and correct chimeric contigs can have dramatic consequences in downstream steps in the assembly pipeline, e.g., scaffolding or construction of pseudo-molecules. Therefore, chimeric removal can been seen as a pre-processing step of scaffolding.

While most of the sequencing projects carry out a chimeric detection/correction step before the scaffolding step, it is clear that this step is carried out manually by visually

inspecting the alignments of the contigs on the optical map (e.g., using IrysView for BioNano maps). Our experience in carrying out this step many times for the cowpea (*Vigna unguiculata*) and potato late blight pathogen (*Phytophthora infestans*) genome projects currently under way at UC Riverside, is that manual chimeric detection/correction is tedious and error-prone. In response to this need, here we introduce Chimericognizer, a tool that can detect large-scale mis-joins in either assembled contigs or Bionano optical molecules. The presence of mis-joins induces conflicts in high-quality alignments between contigs and optical molecules [38] (see Figure 2.2). The quality of an alignment depends on the consistency of shared distances between adjacent restriction enzyme sites and the total length of the alignment. Due to the requirement for high-quality alignments, Chimericognizer can detect mis-joins only on assembled contigs that are sufficiently long to be reliably aligned, e.g., 50 Kbp or longer. Contigs produced from the assembly of third-generation sequencing data (e.g., PacBio and Oxford Nanopore) generally meet this criterion. In this case, the detection of chimeric contigs appears straightforward if one assumes that optical maps are error-free and all the alignment conflicts are caused by mis-joins in the contigs. Unfortunately, since optical maps are obtained via an assembly process similar to sequence assembly, optical molecules can also be chimeric. According to [38], in about *"7% of the (alignment) conflicts, the consensus map (optical map) was wrong"*. Mis-joins in optical molecules typically occur in repetitive regions of the genome, which induce long stretches of regularly-spaced restriction enzyme sites.

Chimericognizer depends on the availability of multiple assemblies and one (or more) Bionano optical map to accurately detect chimeric contigs and reduce the possibility

of incorrectly splitting non-chimeric contigs. Multiple assemblies can be obtained by either running several assembly tools or by using one assembler with multiple parameter settings on the same input data.

2.1 Methods

The algorithm used by ChimeriCognizer has three phases. The algorithm pipeline is illustrated in Figure 2.1. The first phase has three steps. In step 1, we concatenate all the available genome assemblies and *in silico*-digest them using the same restriction enzyme(s) used to produce the Bionano optical map(s). Then, we align digested contigs to their corresponding optical map using Bionano Genomics' RefAligner. In step 2, we remove alignments either i) when they have a confidence lower than a minimum threshold or ii) when there is another alignment between the same contig and the same molecule with higher confidence. In step 3, we unify the coordinates of alignments when multiple optical maps are available. Due to imprecisions in optical mapping, the distances between restriction enzyme sites in optical maps can be inflated. To compensate for the inflation, RefAligner has to amplify the distances of restriction enzyme sites on the contigs by a scaling factor so that accurate alignments can be produced. Since this scaling factor is different for each optical map, in order to make the coordinates comparable across maps, we have to normalize them by the appropriate scaling factor.

After pre-processing, we identify possible conflicts between contigs and molecules. For each alignment a between an optical molecule o and a contig c, we compute the left overhang l_o and right overhang r_o from o and the left overhang l_c and right overhang r_c from

c. The left-end of alignment a is declared a *conflict site* if i) both l_o and l_c are longer than some minimum length (default 50 kbp) and ii) at least one restriction enzyme sites appear in both l_o and l_c. A symmetric argument applies to the right-end of the alignment (which determines the values for r_o and r_c). The example in Figure 2.2A illustrates a conflicting alignment between an optical molecules (green) and an assembled contigs (blue). Observe that a) l_o is approximately 0.37 Mb and l_c is approximately 0.27 Mb and b) the green overhang and the blue overhang contain several restriction sites. Since conditions i) and ii) are satisfied, this is an alignment conflict. Once a conflict site is recognized, the location on the optical molecule and the contig are stored as a pair of *candidate chimeric sites* (red arrows in Figure 2.2A). Figure 2.2B illustrates a likely chimeric optical molecule, where again the candidate locations for splits are indicated by the red arrows (here l_o is the optical molecule left overhang, l_c is the contig left overhang, r_o is the optical molecule right overhang, and r_c is the contig right overhang). Observe that the 1.5 Mb-long region between the two red arrows contains regularly-spaced restriction enzyme sites, indicating a repetitive region of the genome. It is likely the the Bionano Assembler created a mis-join in the optical map in that region.

In the second phase, high-confidence chimeric sites are selected from the list of candidate sites. The *relevance* of each candidate site is first quantified, then a maximum parsimony strategy is applied. Among all the candidate sites, we find the subset with minimum total relevance which can resolve all the conflicts. We model this problem as a weighted vertex cover problem on a *conflict graph* in which a vertex represents a candidate site and an edge indicates that the two sites conflict with each other. Each vertex v in the *conflict*

graph is weighted by its *relevance* $\text{cov}(v)/\left(1+t(v)\right)$ where $t(v)=\sum_{u\in N(v)} q_{g(u)}/\sum_i q_i$, $\text{cov}(v)$ is the number of alignments covering the candidate chimeric site corresponding to v, $N(v)$ is the set of vertices connected to v, $g(u)$ is the optical molecule or contig corresponding to u, and q_i is the quality score for contig/molecule i. The variable i ranges from 1 to the sum of the number of contigs plus the number of optical molecules. Values q_i are provided by the users. By default all optical molecules are given quality 1.5 and all contigs are given quality 1. The value of $\text{cov}(v)$ is the main factor in deciding whether to cut the contigs or the molecule in order to resolve an alignment conflict. When $\text{cov}(v)$ is a tie, the denominator in the relevance formula breaks the tie based on the "trust" users have on the optical map vs. the assemblies.

While building the *conflict graph*, candidate chimeric sites which are close to each other (i.e., when their distance is smaller than a minimum threshold) are merged into the same vertex. Then, among the set of vertices which covers all the edges, we identify the subset with the smallest total weight. To speed up the process, we find the minimum vertex cover of each connected component of the conflict graph. We run the exhaustive (optimal) algorithm on small components and Clarkson's 2-approximation algorithm on larger components [16]. In the third phase, contigs and molecules are cut at the chimeric sites determined by the solution of the minimum vertex cover.

2.2 Experimental results

To assess the performance of ChimericOgnizer, we used real and synthetic datasets for cowpea (*Vigna unguiculata*) along with two Bionano Genomics optical maps.

We also tested CHIMERICOGNIZER on a fruit fly (*Drosophila melanogaster*) dataset [80], for which a high-quality reference genome is available. To the best of our knowledge, the BIONANO HYBRID SCAFFOLD pipeline is the only available tool that solves exactly the same problem addressed by CHIMERICOGNIZER. Other chimeric detection methods are available, but they either require additional data or focus on different types of mis-joins. For example, Missequel can detect mis-joins that are much shorter than our tool, but it requires short reads in addition to an optical map [53].

2.2.1 Experimental results on cowpea assemblies

We tested our tool on synthetic and real data of cowpea (*Vigna unguiculata*). Cowpea is a legume crop that is resilient to hot and drought-prone climates, and a primary source of protein in sub-Saharan Africa and other parts of the developing world. Cowpea is a diploid with a chromosome number 2n = 22 and an estimated genome size of 620 Mb. The genome has very low heterozygosity, in practice it can be consider haploid. We sequenced an elite African variety (IT97K-499-35) using single-molecule real-time sequencing (Pacific Biosciences RSII). A total of 87 SMRT cell yielded about 6M reads for a total of 56.84 Gbp (91.7x genome equivalent). The raw PacBio reads are available in the public domain at NCBI SRA sample SRS3721827 (study SRP159026).

To test CHIMERICOGNIZER we generated multiple assemblies from the PacBio data described above with a mix of parameters, polishing qualities and assembly tools. We used CANU [7, 42], FALCON [14] and ABRUIJN [46] to generate eight assemblies. CANU was run with different parameters to generate six of the eight assemblies (parameters shown in Table 2.1). CANU_4, CANU_5 and CANU_6 were polished with QUIVER. We used two Bionano

Genomics optical maps. The first optical map was obtained using the BspQI nicking enzyme (which recognizes "GCTCTTC"), while the second was obtained using the BssSI nicking enzyme (which recognizes "CACGAG"). The BspQI optical map had 508 assembled optical molecules with a molecule N50 of 1.62 Mb and a total length of 622.21 Mb. The BssSI optical map had 743 assembled optical molecules with a molecule N50 of 1.02 Mb and a total length of 577.76 Mb. Both optical maps were assembled at UC Davis using the Bionano IrysSolve Assembler. In all the experiments, Chimericognizer was run using default parameters (`-a 1.5 -b 1 -d 25 -e 50000 -h 50000 -r 80000`). Please refer to the README at `https://github.com/ucrbioinfo/Chimericognizer` for details about these parameters. Bionano Hybrid Scaffold was run using default parameters, i.e., we executed the script `hybridScaffold.pl` (v.4741) with the parameters in the XML file `hybridScaffold_config.xml`

Table 2.2 shows the assembly statistics after the removal of chimeric contigs via Chimericognizer compared to the manually-curated assemblies (carried out by an expert several months before we developed Chimericognizer). The manual curation involves detecting chimeric contigs by visually inspecting the alignments using Bionano IrisView. For a genome of the size of cowpea, it takes about three hours for each assembly. The process is tedious and error-prone. First, observe in Table 2.2 that there is almost no difference between Chimericognizer's statistics using one vs. two optical maps. We believe that the second optical map does not help in this case because the number of input assemblies is sufficiently high (experiments below seem to support this hypothesis). Second, note that the N50 is higher for Chimericognizer's assemblies compared to the manually-curated

assemblies, indicating that the expert was overly aggressive in splitting contigs. Since there is no "ground truth" on this dataset (i.e., no high-quality reference genome), we evaluated these results using other independent metrics. First, we mapped ≈200M paired-end Illumina reads using BWA. A comparative lower percentage of mapped reads (particularly properly-paired) would indicate an assembly that still contains chimeric contigs. Table 2.2 shows there is almost no difference between CHIMERICOGNIZER's and the expert's assemblies in terms of mapped reads. Second, we compared the assemblies against the high-density genetic map available from [54]. To evaluate whether the assemblies contained residual chimeric contigs, we BLASTed the 121bp-long sequence surrounding the 51,128 SNPs provided in [54] against each assembly, then we identified which contigs had SNPs mapped to them, and what linkage groups (chromosomes) of the genetic map those mapped SNPs belonged to. Chimeric contigs are revealed when their mapped SNPs belong to more than one linkage group. The last row of each panel in Table 2.2 reports the total size of contigs in each assembly for which i) they contain at least one SNPs and ii) all mapped SNPs belong to the same linkage group (i.e. likely to be non-chimeric). Observe in Table 2.2 that CHIMERICOGNIZER's assemblies have higher agreement with the genetic map than the expert's assemblies. Finally, CHIMERICOGNIZER determined that the expert missed 23/28 chimeric contigs in the eight assemblies using BspQI/BssSI, respectively, and 40 chimeric contigs when using both maps (some examples are shown in Figure 2.4).

We also test CHIMERICOGNIZER on synthetic datasets. To generate synthetic datasets with artificial chimeric contigs, we first used CHIMERICOGNIZER to remove and split possible chimeric contig from the eight assemblies described above. For each of the

eight chimeric-free assemblies, we injected artificial chimeric contigs by pairwise joining 2% of the contigs selected at random. We create mis-joins only for contigs longer than 500 Kbp. Results of these simulations for CHIMERICOGNIZER are reported in Table 2.3 (two optical maps) and Table 2.4 (one optical map). Results of these simulations for BIONANO HYBRID SCAFFOLD are reported in Table 2.5. To generate synthetic datasets with artificial chimeric optical molecules, we first used CHIMERICOGNIZER to remove and split possible chimeric molecules from the two optical maps described above. For each of the two chimeric-free optical maps, we created a corresponding synthetic optical map by pairwise joining 0.5% of the molecules selected at random. We created mis-joins only on molecules longer than 1 Mbp. These synthetic optical maps were given in input to CHIMERICOGNIZER along with the eight original cowpea assemblies. To produce a more realistic simulation we decided to use the original cowpea assemblies instead of chimeric-free assemblies. Results of these simulations are reported in Table 2.6. Then we used Chimericognizer and Bionano Hybrid Scaffold to detect these synthetic chimeric contigs. To evaluate the performance of CHIMERICOGNIZER and BIONANO HYBRID SCAFFOLD on the datasets containing synthetic chimeric contigs, we measured precision and recall by comparing its results to the "ground truth". The same approach was used to measure the performance of these tools on the datasets containing synthetic chimeric optical molecules. Figure 2.3 illustrates how we computed true positives, false negatives, false positives and true negatives. When a contig contains a known mis-join (TOP, condition positive), a tool may decide to cut it (true positive) or not (false negative). When a contig does not contain a mis-join (BOTTOM, condition negative), a tool may decide to cut it (false positive) or not (true negative). Precision is defined as

TP/(TP+FP). Sensitivity is defined as TP/(TP+FN). For Bionano Hybrid Scaffold the list of contigs classified as positives are those marked `cut` in the 7th and 8th column (corresponding to `ref_leftBkpt_toCut` and `ref_rightBkpt_toCut`, respectively) of output file `conflicts_cut_status.txt`. For Chimericognizer the list of contigs classified as positives are those that are listed in the output file `qry_cuts.txt`. Among these, we determined which ones are true positive by matching them against the "ground truth".

Experimental results for Chimericognizer are reported in Table 2.3 and 2.4, while the results for Bionano Hybrid Scaffold are summarized in Table 2.5. These are average values over ten synthetic datasets generated as described above. First, observe that Bionano Hybrid Scaffold missed all the chimeric contigs. In the case of Chimericognizer, using two optical maps the precision is very close to 100% while the sensitivity is always higher than 94%. The precision with one optical map is as good as two optical maps, but the sensitivity is worse (around 80%). We also generated a synthetic dataset in which we injected chimeric molecules in the optical map. Table 2.6 shows that the Chimericognizer's precision is 100% and the sensitivity varies between 77% and 93%. As said, the accuracy of Chimericognizer depends on the availability in multiple assemblies. To study Chimericognizer's performance as a function of the number of available assemblies, we randomly selected a subset of the assemblies then generated datasets containing synthetic chimeric contigs as described above. Table 2.7 and 2.8 report average values over ten synthetic datasets for each choice of the subset size. With one optical map and one assembly, Chimericognizer recognizes chimeric contigs and sites with relatively low precision (about 68%). The precision improves significantly (97-99%) when either two optical maps or two assemblies are used. Note that

the precision increases with the number of assemblies, while the sensitivity increases with the number of optical maps. Also observe that having more than one assembly is critical when Chimericognizer can only rely on one optical map.

2.2.2 Experimental results on fruit fly assemblies

We also tested the performance of Chimericognizer and Bionano Hybrid Scaffold on the *Drosophila melanogaster* (ISO) dataset from [80].

We downloaded three *D. melanogaster* assemblies generated in [80] (`https://github.com/danrdanny/Nanopore_ISO1`). The first assembly (295 contigs, total size = 141 Mb, N50 = 3 Mb) was generated using Canu [7, 42] on Oxford Nanopore (ONT) reads longer than 1kb. The second assembly (208 contigs, total size = 132 Mb, N50 = 3.9 Mb) was generated using MiniMap and MiniAsm [43] using only ONT reads. The third assembly (339 contigs, total size = 134 Mb, N50 = 10 Mb) was generated by Platanus [39] and DBG2OLC [94] using 67.4x of Illumina paired-end reads and the longest 30x ONT reads. The first and third assemblies were polished using nanopolish [48] and Pilon [90].

The Bionano Genomics optical for *D. melanogaster* map was provided by the authors of [80]. This optical map (363 molecules, total size = 246 Mb, N50 = 841 kb) was created using IrysSolve 2.1 from 78,397 raw Bionano molecules (19.9 Gb of data with a mean read length 253 kb). We used release 6.21 of the *D. melanogaster* genome, downloaded from FlyBase (`http://www.flybase.org`). Chimericognizer was run using parameters (`-a 0.5 -b 1.0 -d 25 -e 100000 -h 100000 -r 80000`). Please refer to the README at `https://github.com/ucrbioinfo/Chimericognizer` for details about these parameters. Bionano Hybrid Scaffold was run with using default parameters, i.e., we

executed the script `hybridScaffold.pl` (v.4741) with the parameters defined in the XML file `hybridScaffold_config.xml`

To evaluate the performance of CHIMERICOGNIZER and BIONANO HYBRID SCAFFOLD on *D. melanogaster* assemblies, we measured precision and sensitivity by comparing its results to the "ground truth" (reference genome). To determine which contigs were truly chimeric (i.e., the true positive set), we first selected all contigs from the three assemblies which (i) could be aligned to the optical map via REFALIGNER with a minimum confidence of at least 25 and (ii) had at least one BLAST alignment (v2.7.1, default parameters) to the reference genome with an e-value lower than 1e-50 and an alignment length higher than 8 kbp. A total of 73 contigs satisfied these two conditions. Among all the contigs that satisfied (i) and (ii), we defined a contig C to be a *true chimeric contig* if C had at least two alignments which satisfied any of the following three conditions: (1) C aligned to different chromosomes; (2) the orientation of C's alignments were different; or (3) the difference between the distance of alignments on the contig and the distance of alignments on the reference sequence was larger than 100 Kbp. A total of 6 contigs were identified as chimeric (out of 73). Precision and Sensitivity were defined as for cowpea (Section 2.2.1).

Experimental results are reported in Table 2.9 for CHIMERICOGNIZER, and Table 2.10 for BIONANO HYBRID SCAFFOLD. CHIMERICOGNIZER correctly identified five of them and did not report any false positives (see Table 2.9). BIONANO HYBRID SCAFFOLD detected five chimeric contigs, but none of them was correct (see Table 2.10).

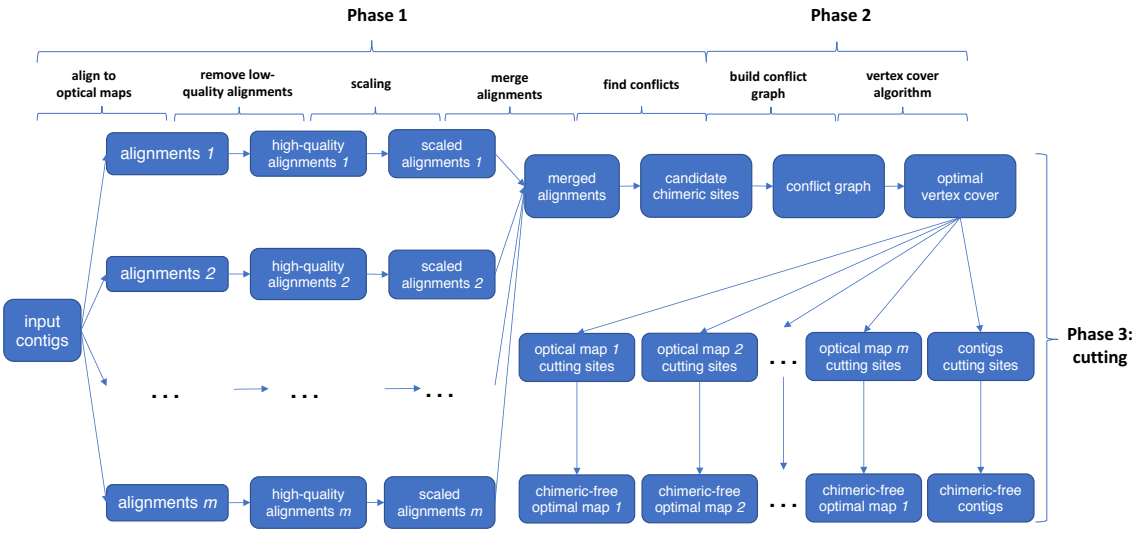
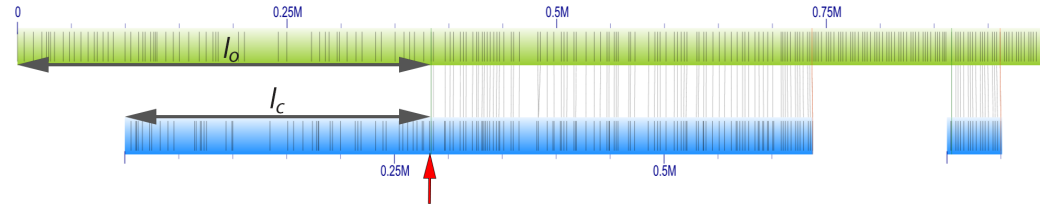


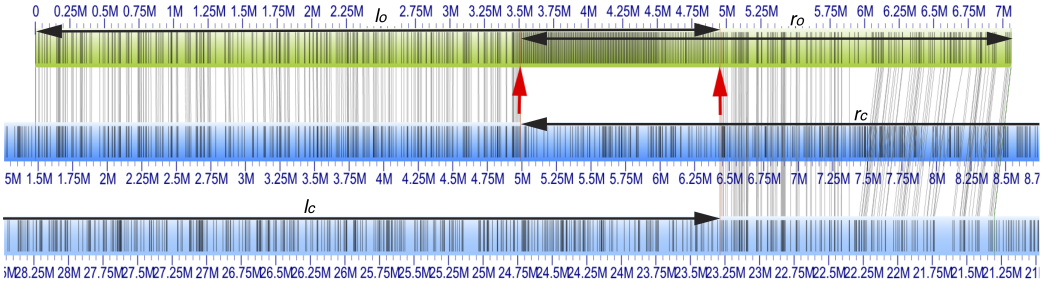
Figure 2.1: Algorithmic pipeline of ChimeriCognizer

CANU assembly	`corMhapSensitivity`	`corMaxEvidenceErate`	`corOutCoverage`	Quiver
1	high	default	default	
2	high	0.15	100	
3	normal	0.15	100	
4	high	default	100	✓
5	low	default	default	✓
6	low	default	100	✓

Table 2.1: Parameter choices for Canu v1.3: three assemblies were polished with Quiver



(A)



(B)

Figure 2.2: Examples of a conflicting alignment between an optical molecule (green) and an assembled contig (blue); vertical lines indicate the location of restriction enzyme sites; (A) a chimeric contig (blue) and its candidate location for a split indicated by the red arrow (l_o is the optical molecule left overhang, l_c is the contig left overhang; the left end of alignment is declared a *conflict site* if i) both l_o and l_c are longer than some minimum length (default 50 kbp) and ii) at least one restriction enzyme sites appear in both l_o and l_c; both conditions are satisfied in this case); (B) a chimeric optical molecule (green) and candidate locations for splits indicated by the red arrows (l_o is the optical molecule left overhang, l_c is the contig left overhang, r_o is the optical molecule right overhang, r_c is the contig right overhang)

2.3 Conclusions

In this chapter, we presented a tool called ChimericOgnizer that takes advantage of one or more Bionano Hybrid Scaffold optical maps to accurately detect and correct chimeric contigs. Experimental results show that ChimericOgnizer is very accurate, and significantly better than the chimeric detection method offered by the Bionano Hybrid Scaffold pipeline. ChimericOgnizer can also detect and correct chimeric optical molecules.

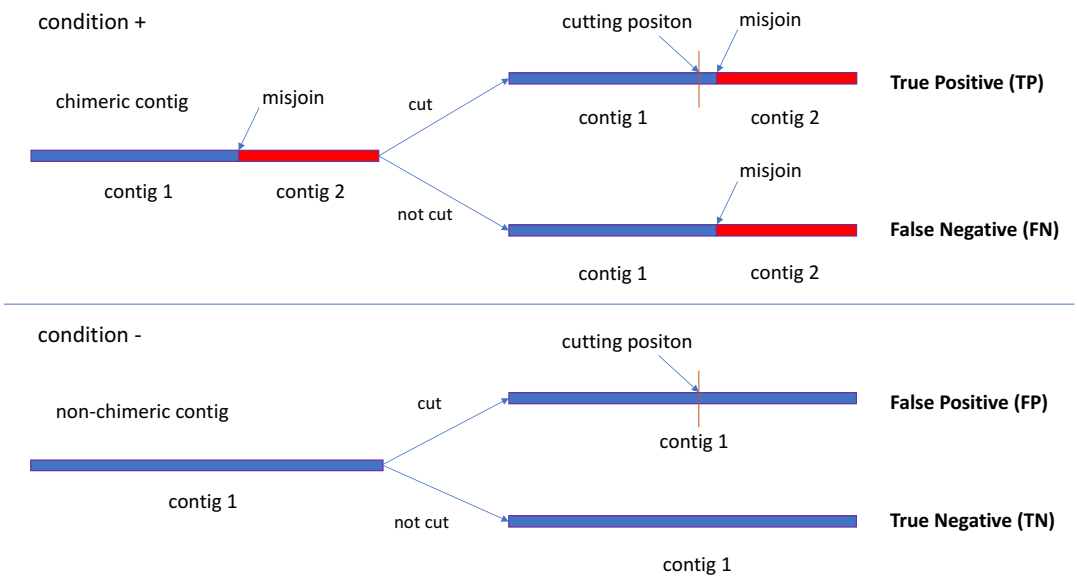
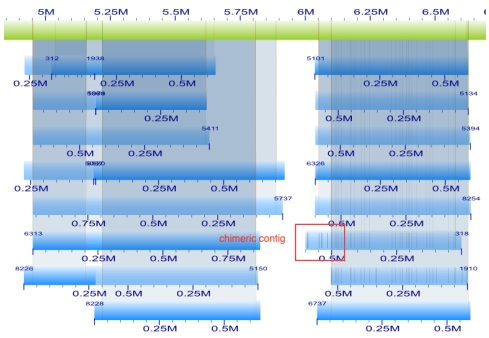
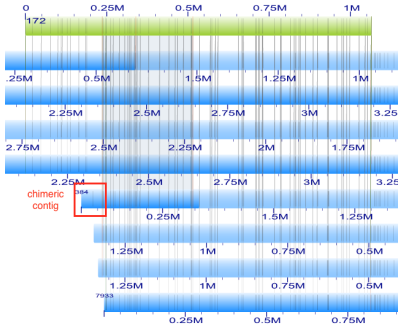


Figure 2.3: Illustrating how we computed true positives, false negatives, false positives and true negatives; when a contig contains a mis-join (TOP, condition positive), CHIMERICOGNIZER may decide to cut it (true positive) or not (false negative); when a contig does not contain a mis-join (BOTTOM, condition negative), CHIMERICOGNIZER may decide to cut it (false positive) or not (true negative); precision is TP/(TP+FP), sensitivity is TP/(TP+FN)

(A)



(B)



(C)

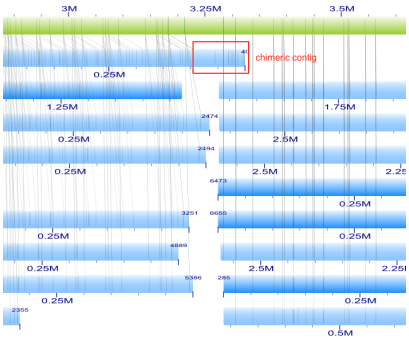


Figure 2.4: A few examples of chimeric contigs missed by the human expert, but correctly identified by ChimericOgnizer

	Chimericognizer with two optical maps							
	ABruijn	Falcon	Canu$_1$	Canu$_2$	Canu$_3$	Canu$_4$	Canu$_5$	Canu$_6$
contig N50 (bp)	2,084,664	2,918,725	3,427,506	3,175,625	2,798,135	5,633,882	5,312,333	4,757,094
contig L50	69	47	42	48	50	28	27	31
total assembled (bp)	478,230,679	511,933,729	504,711,938	516,558,510	515,964,327	511,101,122	506,285,539	517,496,317
# contigs	516	1,826	1,061	1,099	1,125	948	879	948
# contigs ≥100kbp	410	399	287	340	316	269	201	277
# contigs ≥1Mbp	149	115	125	135	141	94	98	103
# contigs ≥10Mbp	0	1	2	4	2	10	9	10
longest contig (bp)	9,801,038	10,554,495	14,090,735	14,331,160	12,496,821	17,211,165	18,473,372	18,498,533
Illumina reads, % mapped (202M)	99.72399%	99.58149%	99.97449%	99.97389%	99.97389%	99.97743%	99.97343%	99.97763%
Illumina reads, % properly paired (202M)	92.29997%	91.94896%	92.54645%	92.63437%	92.62722%	92.64222%	92.62153%	92.64414%
Illumina reads, % mapped, MapQ≥30 (202M)	64.20883%	59.48734%	64.65541%	63.00774%	63.47912%	64.80935%	64.85658%	64.59832%
total length with 100% consistent LG (bp)	425,557,449	344,074,378	421,565,015	418,588,863	409,262,310	425,812,490	423,058,141	420,659,561
	Chimericognizer with one optical map							
	ABruijn	Falcon	Canu$_1$	Canu$_2$	Canu$_3$	Canu$_4$	Canu$_5$	Canu$_6$
contig N50 (bp)	2,084,664	3,000,247	3,427,506	3,175,625	2,798,135	5,633,882	5,312,333	4,757,094
contig L50	69	46	42	48	50	28	27	31
total assembled (bp)	478,230,679	511,933,729	504,711,938	516,558,510	515,964,327	511,101,122	506,285,539	517,496,317
# contigs	510	1,814	1,059	1,098	1,125	947	879	947
# contigs ≥100kbp	407	391	286	340	316	268	201	277
# contigs ≥1Mbp	149	115	125	135	141	94	98	103
# contigs ≥10Mbp	0	1	2	4	2	10	9	10
longest contig (bp)	9,801,038	10,554,495	14,090,735	14,331,160	12,496,821	17,211,165	18,473,372	18,498,533
Illumina reads, % mapped (202M)	99.72400%	99.58149%	99.97449%	99.97389%	99.96996%	99.97743%	99.97343%	99.97763%
Illumina reads, % properly paired (202M)	92.29986%	91.94953%	92.54646%	92.63438%	92.62728%	92.64221%	92.62152%	92.64384%
Illumina reads, % mapped, MapQ≥30 (202M)	64.20894%	59.48738%	64.65538%	63.00775%	63.47915%	64.80937%	64.85659%	64.59879%
total length with 100% consistent LG (bp)	425,557,449	344,074,378	421,565,015	418,588,863	409,262,310	425,812,490	423,058,141	420,659,561
	Chimeric contigs detected/removed manually by an expert							
	ABruijn	Falcon	Canu$_1$	Canu$_2$	Canu$_3$	Canu$_4$	Canu$_5$	Canu$_6$
contig N50 (bp)	1,896,002	2,869,362	3,280,469	2,797,949	2,666,731	5,340,274	4,859,617	4,498,063
contig L50	74	49	42	51	55	29	30	32
contig NG50 (bp)	1,330,435	1,737,012	2,431,239	1,949,515	2,068,575	3,451,071	3,767,556	3,417,577
contig LG50	119	73	63	73	77	42	43	45
total assembled (bp)	478,230,679	511,933,729	503,187,311	516,537,734	515,949,175	507,773,747	506,154,442	516,817,613
# contigs	538	1,820	1,038	1,110	1,140	897	894	928
# contigs ≥100kbp	437	404	299	354	334	278	220	288
# contigs ≥1Mbp	151	118	128	142	145	103	104	107
# contigs ≥10Mbp	0	1	2	2	0	9	7	8
longest contig (bp)	8,846,014	10,554,495	14,090,735	14,331,160	9,775,097	17,211,165	18,473,372	18,498,533
Illumina reads, % mapped (202M)	99.72397%	99.58150%	99.94933%	99.97389%	99.94468%	99.97474%	99.96894%	99.97707%
Illumina reads, % properly paired (202M)	92.30106%	91.95107%	92.52969%	92.63057%	92.62330%	92.59763%	92.59433%	92.64181%
Illumina reads, % mapped, MapQ≥30 (202M)	64.21367%	59.49035%	64.38425%	63.00587%	63.22414%	62.84466%	64.35764%	63.50279%
total length with 100% consistent LG (bp)	379,029,914	312,593,019	356,505,616	349,534,672	347,586,448	425,812,490	331,956,528	338,556,993

Table 2.2: Assembly statistics of the eight cowpea assemblies after chimeric contigs were removed (top) by Chimericognizer using two optical map, (middle) by Chimericognizer using one optical map, and (bottom) by an expert; reads were mapped with BWA

	ABruijn	Falcon	Canu$_1$	Canu$_2$	Canu$_3$	Canu$_4$	Canu$_5$	Canu$_6$
# TP	10.8	35.5	21.0	20.8	22.4	17.8	17.2	18.0
# TP + FP	10.8	35.9	21.7	20.8	23.0	18.6	17.3	18.0
# P	11.0	37.0	22.0	22.0	23.0	19.0	18.0	19.0
precision	100.00%	98.92%	96.79%	100.00%	97.45%	95.70%	99.44%	100.00%
sensitivity	98.18%	95.95%	95.45%	94.55%	97.39%	93.68%	95.56%	94.74%
avg position error (bp)	16,704	26,380	32,054	18,426	19,415	38,338	17,753	18,809

Table 2.3: Performance statistics for Chimericognizer on the eight cowpea assemblies injected with synthetic chimeric contigs (i.e., 2% of the contigs longer than 500 Kbp selected at random where joined) and two optical maps; values in this table are the averages over ten experiments; TP, FP and P represent true positive, false positive and positive, respectively; avg position error is the average distance in base pairs between Chimericognizer's cutting position and the true mis-join position

	ABruijn	Falcon	Canu$_1$	Canu$_2$	Canu$_3$	Canu$_4$	Canu$_5$	Canu$_6$
# TP	9.5	30.8	17.5	18.8	18.9	15.3	14.6	14.5
# TP + FP	9.5	31.7	17.5	19.2	19.7	15.3	14.6	14.5
# P	11.0	37.0	22.0	22.0	23.0	19.0	18.0	19.0
precision	100.00%	97.17%	100.00%	98.04%	96.05%	100.00%	100.00%	100.00%
sensitivity	86.36%	83.24%	79.55%	85.45%	82.17%	80.53%	81.11%	76.32%
avg position error (bp)	17,560	27,969	18,506	21,778	73,255	19,853	16,693	22,266

Table 2.4: Performance statistics for Chimericognizer on the eight cowpea assemblies injected with synthetic chimeric contigs (i.e., 2% of the contigs longer than 500 Kbp selected at random where joined) and one optical map (BspQI); values in this table are the averages over ten experiments; TP, FP and P represent true positive, false positive and positive, respectively; avg position error is the average distance in base pairs between Chimericognizer's cutting position and the true mis-join position

	ABruijn	Falcon	Canu$_1$	Canu$_2$	Canu$_3$	Canu$_4$	Canu$_5$	Canu$_6$
# TP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
# TP + FP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
# P	11.0	37.0	22.0	22.0	23.0	19.0	18.0	19.0
precision	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
sensitivity	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg position error (bp)	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Table 2.5: Performance statistics for Bionano Hybrid Scaffold on the eight cowpea assemblies injected with synthetic chimeric contigs (i.e., 2% of the contigs longer than 500 Kbp selected at random where joined) and one optical map (BspQI); values in this table are the averages over ten experiments; TP, FP and P represent true positive, false positive and positive, respectively; avg position error is the average distance in base pairs between Bionano Hybrid Scaffold's cutting position and the true mis-join position

	one optical map		two optical maps	
	BspQI	BssSI	BspQI	BssSI
# TP	2.3	3.4	2.8	3.7
# TP + FP	2.3	3.4	2.8	3.7
# P	3.0	4.0	3.0	4.0
precision	100.00%	100.00%	100.00%	100.00%
sensitivity	76.67%	85.00%	93.33%	92.50%

Table 2.6: Performance statistics for Chimericognizer on cowpea datasets composed by one or two synthetic optical maps and eight real assemblies; for the "one optical map" column, we injected chimeric optical molecules in either BspQI or BssSI, ran Chimericognizer on that optical map, and measured precision/sensitivity on the molecules of that optical map; for the "two optical maps" column, we injected chimeric optical molecules in both optical maps, ran Chimericognizer with two optical maps, and measured precision/sensitivity on molecules of each optical map separately; values in this table are the averages over ten experiments; TP, FP and P represent true positive, false positive and positive, respectively

# assemblies	1	2	3	4	5	6	7	8
# TP	20.2	39.3	56.9	78.7	107.0	121.5	142.5	163.5
# TP + FP	22.4	40.1	57.6	80.5	108.5	123.6	144.4	166.1
# P	21.6	41.6	60.2	83.0	112.5	127.7	149.1	171.0
precision	89.35%	97.86%	98.75%	97.70%	98.59%	98.33%	98.69%	98.44%
sensitivity	93.34%	94.39%	94.55%	94.81%	95.05%	95.06%	95.59%	95.61%
average position error (bp)	121,396	17,935	20,852	18,905	29,384	25,395	33,402	24,274

Table 2.7: Performance statistics for CHIMERICOGNIZER on synthetic cowpea datasets composed of a variable number of assemblies and two optical maps; values in this table represent the total for all assemblies selected (averaged over ten experiments); TP, FP and P represent true positive, false positive and positive, respectively; avg position error is the average distance in base pairs between CHIMERICOGNIZER's cutting position and the true mis-join position

# assemblies	1	2	3	4	5	6	7	8
# TP	18.3	34.7	50.1	66.7	85.8	106.6	122.7	139.9
# TP + FP	25.8	38.2	51.9	68.1	87.1	108.1	124.1	142.0
# P	22.3	42.4	63.8	83.5	103.0	131.2	151.8	171.0
precision	68.43%	91.06%	96.64%	98.00%	98.56%	98.67%	98.87%	98.52%
sensitivity	81.49%	82.69%	78.76%	80.36%	83.22%	81.25%	80.85%	81.81%
average position error (bp)	270,414	102,461	19,633	41,662	21,143	25,795	25,468	29,249

Table 2.8: Performance statistics for CHIMERICOGNIZER on synthetic cowpea datasets composed of a variable number of assemblies and one optical map (BspQI); values in this table represent the total for all assemblies selected (averaged over ten experiments); TP, FP and P represent true positive, false positive and positive, respectively; avg position error is the average distance in base pairs between CHIMERICOGNIZER's cutting position and the true mis-join position

# TP	5
# TP + FP	6
# P	6
precision	83.33%
sensitivity	83.33%

Table 2.9: Performance statistics for CHIMERICOGNIZER on the *D. melanogaster* dataset (composed by one optical map and three assemblies); TP, FP and P represent true positive, false positive and positive, respectively

# TP	0
# TP + FP	5
# P	6
precision	0.00%
sensitivity	0.00%

Table 2.10: Performance statistics for BIONANO HYBRID SCAFFOLD on the *D. melanogaster* dataset (composed by one optical map and three assemblies); TP, FP and P represent true positive, false positive and positive, respectively

Chapter 3

Novo&Stitch: Accurate Reconciliation of Multiple *de novo* Genome Assemblies via Optical Maps

In this chapter, we focus on the assembly reconciliation problem, which is a part of the *de novo* genome assembly pipeline.

As said, despite significant algorithmic progress, the *de novo* genome assembly problem remains challenging due to the high repetitive content of eukaryotic genomes, short read length, uneven sequencing coverage, non-uniform sequencing errors and chimeric reads. Several *de novo* genome assembly tools are available, for both second and third generation sequencing data. Most assemblers for second generation sequencing data rely

on the *de Bruijn graph* (e.g., [64, 96, 79, 65, 34, 91, 44, 79, 11, 97, 49]) which allows one to avoid the pairwise overlap step on the massive number of short reads in input. Assemblers for third generation sequencing data mainly use the *overlap graph* to store prefix-suffix overlaps between the long (noisy) reads in input [30, 56]. Not only are these assembly tools fundamentally different at the algorithmic level, but their designers have made different choices in the tradeoff between maximizing assembly contiguity (e.g., N50) and minimizing the probability of misassemblies (e.g., misjoins). In addition, often these assembly tools have dozens of parameters that allow one to adjust these trade-offs, but these parameters can be difficult to optimize for a specific input dataset and target genome. As a result, it is common practice to generate as many assemblies as possible within the time frame of the sequencing project using different assemblers and/or parameter settings, and try to identify the highest quality assembly based on assembly statistics. However, it is surprisingly difficult to identify the "best" assembly. For instance, the assembly with the highest N50 is likely to be the one with most chimeric contigs.

The concept of *assembly reconciliation* has been proposed recently as a more appealing alternative. Instead of selecting the best assembly, assembly reconciliation algorithms try to take advantage of all the individual assemblies. They produce a higher quality consensus assembly by merging all of the candidate assemblies, so that the contiguity of the assembly increases without introducing misassembles. The problem of assembly reconciliation is also quite challenging. While several assembly reconciliation tools are available (see, e.g., [100], [45], [93], [89], [51], [82], [2], [88], [40], [81], [27], [24], [37], [13], [73], [31], [23]), our research group have recently demonstrated that none of these tools can consistently generate

a "reconcilated" assembly which has a significantly higher quality than the assemblies given in input [1].

Here, we introduce an assembly reconciliation algorithm called NOVO&STITCH that takes advantage of optical maps to accurately carry out assembly reconciliation. One or more optical maps are used to obtain coordinates for the contigs, which are then stitched based on their alignments. The presence of the optical map dramatically reduces the complexity of the problem and the possibility of a misjoin.

3.1 Problem definition

Optical mapping technology allows life scientists to produces genome-wide maps by fingerprinting long DNA molecules, typically via nicking restriction enzymes. Linear DNA fragments are stretched on a glass surface or in a nanochannel array, then the locations of restriction sites are identified with the help of dyes or fluorescent labels. An *optical map* is composed by a set of optical map *molecules*, each of which is represented by an ordered set of positions for the restriction enzyme sites.

In the following, we will use $S = \{s_1, s_2 \dots s_n\}$ to denote the set of contigs in the genome assembly, where each s_i is a string over the alphabet $\{A, C, G, T\}$. Given our interest in assembly reconciliation, S is going to be the union of multiple assemblies, obtained from multiple assemblers and/or parameters settings. In other words, S is expected to be highly redundant, i.e., regions of the genome are expected to be covered by multiple contigs. We will use $M = \{o_1, o_2, \dots o_m\}$ to denote the optical map, where each optical map molecule o_j is an ordered set of integers, corresponding to the distances in base pairs between two

adjacent restriction enzyme sites on molecule o_j. By digesting *in silico* the contig s_i using the same restriction enzyme used to produce the optical map and matching the sequence of adjacent distances between sites, one can align the contigs in S to optical map M. High quality alignments allow some of the contigs to be anchored at specific coordinates on the optical map. In addition, contigs can be oriented with respect to each other. When multiple contigs align to the same optical map molecule, an estimate of the distance between them can be obtained. If the distance is positive, a gap is introduced and a *scaffold* can be formed [76]. When the distance is negative (i.e., contigs are overlapping), it may be possible to stitch them.

Given our interest in merging multiple assemblies, here we focus on the case when contigs are overlapping. A series of practical factors make the problem of stitching overlapping contigs non-trivial. These factors include imprecisions in optical maps (e.g., mistakes in the optical map assembly), inaccurate alignment between contigs and optical molecules, and multiple anchoring positions for the same contigs that are not consistent with each other. As a consequence, it is appropriate to frame this problem as an optimization problem.

As said, we are given multiple assemblies represented by a set of contigs S, an optical map M and a set of alignments $A = \{a_{1,1}, a_{1,2}, \ldots a_{n,m}\}$ of S to M, where $a_{i,j}$ is the alignment of contig s_i to optical map molecule o_j. The problem is to stitch overlapping contigs based on A and obtain a new set of longer contigs $T = \{t_1, t_2, \ldots t_k\}$ such that (i) T covers the same portion of the genome covered by S, (ii) k is as small as possible and (iii) the conflicts of T with respect to A are minimized. This optimization problem is not

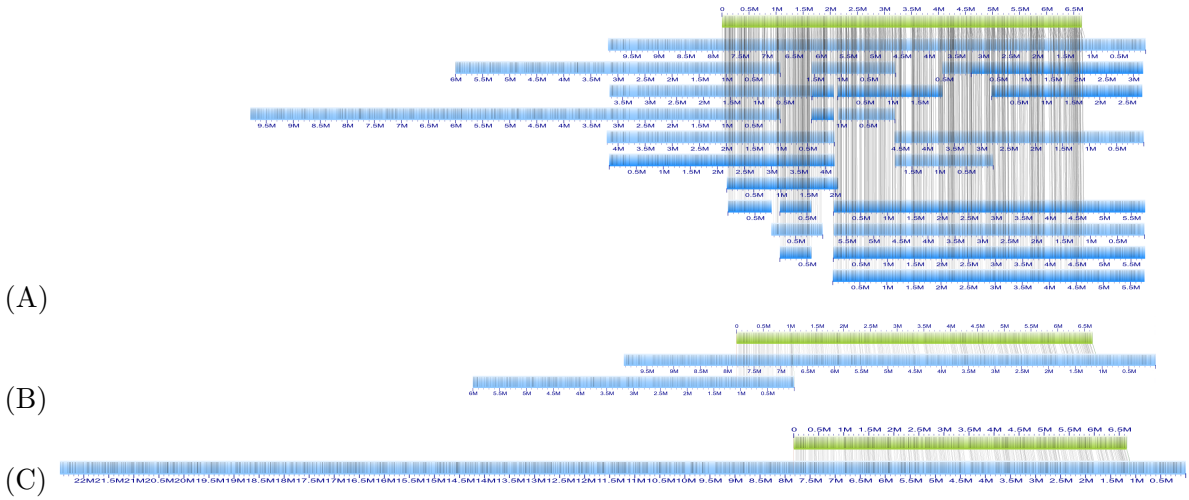


Figure 3.1: (A) Contigs of eight assemblies mapped to one optical molecule; (B) minimum tiling path of the contigs in A; (C) final stitched contig, at the end of the iterative stitching process

rigorously defined unless one defines precisely the concept of *conflict*, but this description captures the spirit of what we want to accomplish. Even if the notion of conflict could be made precise, this multi-objective optimization problem would be hard to solve. Instead of solving this problem, we propose an iterative method that accomplishes a similar objective.

3.2 Methods

The proposed stitching method is an iterative algorithm. Each iteration is composed of three phases: data reduction, stitching and post-processing. The real example in Figure 3.1 will help understanding the phases. In (A) eight assemblies of cowpea were concatenated and aligned the contigs (blue) on the optical map (green) (see "Experimental results" for details on these assemblies). Observe that among the eight assemblies, contigs produced by some assemblers can extend much further than others. In the first phase, the smallest

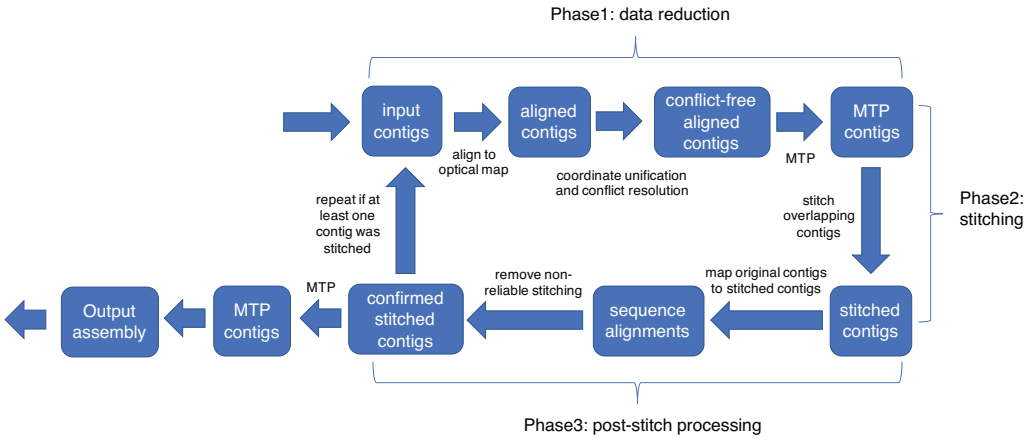


Figure 3.2: Pipeline of the proposed algorithm

subset of contigs that cover the same genomic region of the eight assemblies is selected (B in the figure). In the second phase, the two contigs in (B) are stitched to produce (C) in the figure. Observe that in this case the resulting 22Mb contig is much longer than the expected 12.5Mb due to additional stitching that occurred in later iterations. In the third phase, stitched contigs are checked for consistency, then the entire process is iterated. The pipeline of the algorithm is illustrated in Figure 3.2.

3.2.1 Phase 1: Coordinate unification, conflict resolution and MTP

At high level, phase one has three major steps. More details of each step are provided below. In step 1, we align *in silico*-digested chimeric-free contigs to the optical map (e.g., for a Bionano optical map, we use RefAligner), but not all alignments are used. We only consider alignments that (i) exceed a minimum confidence level (typically confidence 25 in the case of RefAligner) and (ii) do not create conflict with each other (see below for details). Since some contig can have high-quality conflict-free alignments to more than one

optical molecule (which could indicate alternative overlaps), a "unification" step is necessary (step 2, see below for details). Finally in step 3, we compute the *minimum tiling path* (MTP) of the contigs. Formally, let S be the initial set of contigs, and M be the optical map. Let $A = \{a_{1,1}, a_{1,2}, \ldots a_{n,m}\}$ be the set of high-quality conflict-free alignments of S to M, where $a_{i,j}$ is the alignment of contig s_i to optical map molecule o_j. Let $R = \{r_1, r_2, \ldots r_k\}$ be the set of intervals of M covered by the contigs S through the set of alignments A. A *minimum tiling path* of S is the smallest set $P \subseteq S$ such that P covers every intervals in R.

Our algorithm for reducing false alignments relies on a *conflict graph*. The *conflict graph* is an undirected hypergraph in which each vertex represents an alignment, and each hyperedge connects four vertices when the corresponding four alignments conflict with each other. Nodes of the hypergraph are weighted by the confidence of the alignment. Let us call $a_{i,p}$, $a_{i,q}$, $a_{j,p}$, $a_{j,q}$ the alignments of contig p and q on optical molecule i and j, respectively. We say that $a_{i,p}$, $a_{i,q}$, $a_{j,p}$, $a_{j,q}$ is a *conflict* if any of them have an *orientation conflict* or a *coordinate conflict*. An *orientation conflict* occurs when the orientations of $a_{i,p}$ and $a_{j,p}$, and the orientations of $a_{i,q}$ and $a_{j,q}$ are neither both 5' to 3' nor both 3' to 5' (depending on whether i and j are from the same strand of the genome or not). A *coordinate conflict* occurs when the distance between $a_{i,p}$ and $a_{j,p}$ is significantly different from the distance between $a_{i,q}$ and $a_{j,q}$. When four alignments have a conflict, at least one must be a false alignment. We model the problem of removing false alignments as a weighted vertex cover problem on the conflict hypergraph.

Since the weighted vertex cover problem on hypergraph is NP-hard, we use an approximation algorithm. We formulate weighted vertex cover as an integer program, as

follows

$$\begin{aligned} &\text{minimize} && \sum_{i \in V} w_i x_i \\ &\text{subject to} && x_i + x_j + x_k + x_l \geq 1 \quad \text{for every hyperedge } (i,j,k,l) \in E \\ & && x_i \in \{0,1\} \qquad\qquad \text{for every vertex } i \in V \end{aligned} \tag{3.1}$$

where V and E are the vertex set and hyperedge set of the conflict hypergraph, respectively.

In order to solve the integer program we relax it to a linear program, and solve the linear program by standard software packages (e.g., GLPK or CPLEX). The solution of the linear program is transformed into an integer solution as follows. We sort each variable $x_i > 1/4$ in decreasing order and we add the corresponding vertex to the solution if the new vertex covers at least one hyperedge that was not covered before. This greedy algorithm is a 4-approximation algorithm (see below), which is the best known approximation achievable in polynomial time for hypergraph with hyperedges of constant size [12].

Theorem 1 *The LP-based greedy algorithm for the* WEIGHTED VERTEX COVER *on hypergraph gives an approximation ratio of 4.*

Proof. Let C be a vertex cover. Consider a hyperedge $(i,j,k,l) \in E$. Since $x_i + x_j + x_k + x_l \geq 1$, either $x_i \geq 1$ or $x_j \geq 1$ or $x_k \geq 1$ or $x_l \geq 1$. Therefore, (i,j,k,l) is covered. If C^* is an optimum vertex cover, then $w(C) \leq 4w(C^*)$ because

$$w(C^*) \geq \sum_{i \in V} w_i x_i^* \geq \sum_{i \in S} w_i x_i^* \geq \frac{1}{4} \sum_{i \in S} w_i \geq \frac{1}{4} \sum_{i \in C} w_i = w(C)/4$$

First inequality: LP is a relaxation of ILP. Second inequality: $S \subseteq V$. Third inequality: $x_i^* \geq 1/4$ for all $i \in S$. Fourth inequality: $C \subseteq S$. ■

Details of the approximation algorithm are shown in Algorithm 1. In NOVO&STITCH, the *conflict graph* is first divided into connected components. The approximation algorithm is run on connected components with more than twenty vertices. For components with at most twenty nodes, we run the exhaustive (optimal) algorithm.

Algorithm 1 Greedy algorithm for weighted vertex cover problem on hypergraph

1: **procedure** LP_ROUND_GREEDY(V, E)
2: Compute the optimum solution x^* to LP relaxation (2).
3: $S = \{i \in V : x_i^* \geq 1/4\}$
4: $C = \emptyset$
5: $E^{'} = E$
6: **for** $i \in S$ **do**
7: $ifnew = False$
8: **for** $e \in E^{'}$ **do**
9: **if** $i \in e$ **then**
10: $ifnew = True$
11: remove e from $E^{'}$
12: **if** $ifnew = True$ **then** ▷ pick i if it appears in at least one new superedges
13: add i to C
14: **return** C

Our algorithms for computing the MTP and coordinate unification use the *association graph* between optical molecules and contigs. The *association graph* is an undirected graph in which each vertex represents an optical molecule and an edge indicates that the

two molecules share at least one contig aligned to both of them. The weight of edge (o_i, o_j) between molecule o_i and o_j is obtained from the confidences of the alignment of all common contigs, that is $1/\sum_{s \in S_i \cap S_j}(\text{conf}(s, o_i) + \text{conf}(s, o_j))$ where S_i and S_j are the sets of contigs aligned to o_i and o_j, respectively, and $\text{conf}(s, o)$ is the confidence score provided by RefAligner between contig s and molecule o. The confidence score represents the quality of the alignment (higher is better). For the MTP and unification step, we do not use association graph directly, rather the minimum spanning forest (MSF) of the association graph. By construction, MSF identifies the most reliable alignments (i.e., highest total confidence) between contigs and molecules.

We first unify the coordinates of all contigs with respect to the molecules they are aligned to using the MSF of A, as follows. We traverse each MST, starting from the vertex that represents the molecule that has the highest total alignment confidence score (for the contigs aligned to it). That node becomes the *root* of the MST and it defines the origin of the coordinate system. As we traverse the MST, we assign the coordinates of each contig on a molecule based on average position of all the common contigs. Specifically, the position of molecule x with respect to molecule r is $(1/|C|)\sum_{c \in C}(\text{mid}(x, c) - \text{mid}(r, c))$ where C is the set of contigs aligned to both r and c, and $\text{mid}(m, c)$ gives the middle points of contig c's alignment on molecule m.

Once the unification process is complete, we rebuild the association graph using the updated coordinates. At this stage we also remove contigs which are completely contained in other contigs, since they will not be used in the stitching. In order to compute the MTP we build another graph, called *overlap graph*. The *overlap graph* O is an unweighted directed

acyclic graph (DAG) in which vertices represent contigs and directed edges indicate overlaps between the corresponding contigs (oriented left to right along the coordinates induced by the alignment). Each optical molecule induces an overlap (sub)graph for the contigs aligned to it, but since a contig can align to multiple molecules, some of the overlap subgraphs can be connected. In order to efficiently connect the subgraphs in O, we use the MSF of A. Recall that by construction the nodes (which are optical molecules) in the same MST of A share common contigs. We process each minimum spanning tree in A, as we did above. As we traverse the MST we connect the corresponding subgraphs in O. Edges are added to O only if no cycles are introduced.

Once the overlap graph is finalized, we compute the MTP on each connected component of the graph. In the ideal case, each connected component O_i of O (which is a DAG by construction) is expected to have exactly one source and one sink because the genome is one-dimensional and the chain of overlaps is expected to have exactly one leftmost contig (source of the DAG) and exactly one rightmost contig (sink). When a connected component O_i has one source s and one sink t, the MTP problem reduces to finding the path from s to t with the smallest number of vertices. In practice however, O_i may have a set O_S of sources and a set O_T of sinks. A simple example explains why this could happen. Imagine three staggered contigs A, B, C of the same length. Assume these contigs overlap two disjoint optical molecules: the first overlaps A, B, C in their prefixes, while the second overlap only B, C on their suffixes. One should expect the overlap DAG to be $A \to B \to C$. But now assume that the quality of the alignment of B with the first molecule is poor, so in the first molecule we get $A \to C$, in the second molecule we get $B \to C$. When we merge

them, we end up with a DAG with two sources. When either $|O_S| > 1$ or $|O_T| > 1$, the MTP problem requires finding the smallest subgraph P_i of O_i such that for any source-sink pair $(s,t), s \in O_S, t \in O_T$ in which t is reachable from s in O_i, t is also reachable from s in P_i. We call this problem the smallest sub-DAG problem, defined as follow.

Definition 2 (Smallest SubDAG) INPUT: *A connected directed acyclic graph $G = (V, E)$, with source set S, and sink set T.* OUTPUT: *A subgraph $G' = (V', E')$ of G such that (i) G' is a connected directed acyclic graph, (ii) $S \subseteq V'$ (iii) $T \subseteq V'$, (iv) $|V'|$ is the smallest among all the subgraphs satisfying (i-iii).*

Theorem 3 SMALLEST SUBDAG *is NP-hard.*

Proof. We show that SET COVER $\leq_P$ SMALLEST SUBDAG. Let $\langle U, C \rangle$ be an instance of SET COVER, where U is the universe of sets and C represent the collection of sets. Given $\langle U, C \rangle$ we build an instance $\langle G = (V, E), S, T \rangle$ of SMALLEST SUBDAG as follows. For each element in U, build a vertex in V and in T. For each set in C, build a vertex in V. Let $S = \{s\}$ and add s to V. For each set c in C and each element e in U, if e belongs to c, build an edge in E from the vertex corresponding to c to the vertex corresponding to e. For each set c in C, build an edge in E from s to the vertex corresponding to c. The equivalence between these two problems is obvious.

■

Given the hardness of the SMALLEST SUBDAG problem, we propose a greedy heuristics. First, we find the shortest path from each source to each sink. The shortest path among all these paths is chosen as the initial path. Then, the source and sink vertices left

are added to the solution iteratively by calculating the shortest path between them to the current sub-DAG. Details of this greedy algorithm is shown as Algorithm 2.

3.2.2 Phase 2 and 3: Contig stitching and post-processing

In phase 2 we first compute the sequence alignments for MTP contigs that are overlapping according to coordinates obtained in phase 1. For each pair of overlapping contigs, we determine the best alignment between the corresponding sequence. If the best alignment is (i) above a certain length and (ii) of sufficient quality (e-value), and (iii) consistent with the optical map coordinates, the stitching is carried out. When stitching two aligned contigs c_1 and c_2, both c_1 and c_2 are composed of three parts, left overhang l_1, l_2, right overhang r_1, r_2 and common region (aligned region) m_1, m_2. The new stitched contig d is formed by the concatenation of (i) the longest between l_1 and l_2 (ii) either m_1, m_2 depending which one is closer to the 5' of its respective contig and (iii) the longest between r_1 and r_2. If c_2 is stitched with c_1, neither c_1 or c_2 will be used for stitching with other contig in this iteration. Contig d could be stitched to other contigs in later iterations.

In Phase 3, we check the correctness of the stitching by aligning the two original contigs to the new stitched contig. The difficulty of this process stems from the possible fragmentation of alignments. Sequence alignment tools (e.g., BLAST) can generate a large set of alignments, most of which are not informative. To determine the best overall alignment, we find the subset of mutually compatible alignments (from the set of all alignments) which has the longest total length. We say that two alignments are *compatible* if their overlap is smaller than a given fraction of the shorter alignment.

Definition 4 (Optimal set of mutually compatible alignments) INPUT: *A list A of n alignments and their lengths, a compatibility matrix C in which* $C[i,j] = True$ *if alignment i is compatible with alignment j.* OUTPUT: *A subset* A' *of A in which (i) for any pair of alignments* $a, b \in A'$, $C[a,b] = True$ *and (ii) the total length of the alignments in* A' *is the largest among all the subsets of A satisfying (i).*

To solve this problem, we use a dynamic programming algorithm. All the alignments are first sorted by starting positions. Let $S[i]$ be the total length of the alignments selected from from $A[1 \ldots i]$ that includes alignment $A[i]$. First we initialize $S[1]$ to the length of first alignment. The rest of the dynamic programming vector can be filled using the following recurrence relation

$$S[i] = L[i] + \max_{j=1 \ldots i-1} \{S[j] : C[i,j] = True\}$$

The pseudo code of this dynamic programming algorithm is described in Algorithm 3. The time complexity of this dynamic programming algorithm is $O(n^2)$. To speed it up, we remove alignments shorter than a given threshold to reduce the value of n.

After the optimal compatible alignment is computed, we compute the proportion of the two original overlapping contigs mapped to the stitched contig. If the proportion is below a predefined threshold, the stitching is cancelled.

Phase 1, 2 and 3 are repeated iteratively until no further stitching takes place. Recall that when we unify the coordinates, we compute the average position of all common contigs. When a contig appear in multiple fragments, it can affect the coordinates of other contigs, which in turn can change the detection of overlapping contigs. When we stitch contigs, the coordinates of several contigs can change, which can reveal overlaps that were

not detected before. That is why we use an iterative strategy: later iterations can "make up" for stitches that were missed in earlier iterations.

The final assembly produced in output is the MTP of the latest stitched assembly.

3.3 Experimental results

We tested NOVO&STITCH on multiple PacBio assemblies of (i) cowpea (*Vigna unguiculata*) and (ii) *Phytophthora infestans*. Both sequencing projects are currently underway at UC Riverside. Cowpea is a legume crop that is resilient to hot and drought-prone climates, and a primary source of protein in sub-Saharan Africa and other parts of the developing world. *P. infestans* is responsible for the late blight diseases of tomato and potato. It was the major culprit for the European potato famines of the 19th century. Worldwide the disease causes around $6 billion of damage to crops each year.

3.3.1 Experimental results on cowpea assemblies

Cowpea (*Vigna unguiculata*) is a diploid with a chromosome number 2n = 22 and an estimated genome size of 620 Mb. The genome has very low heterozygosity, so that in practice it can be considered as haploid. We sequenced an elite African variety (IT97K-499-35) using single-molecule real-time sequencing (Pacific Biosciences RSII). A total of 87 SMRT cells yielded about 6M reads for a total of 56.84 Gbp (91.7x genome equivalent). To test NOVO&STITCH we generated several assemblies with a mix of parameters, polishing qualities and assembly tools. We used CANU [7, 42], FALCON [14] and ABRUIJN [46] to generate eight assemblies. CANU was run with different parameters to generate six of the

eight assemblies (parameters shown in Table 3.1). CANU$_1$, CANU$_2$ and CANU$_6$ were polished with QUIVER.

CANU assembly	`corMhapSensitivity`	`corMaxEvidenceErate`	`corOutCoverage`	QUIVER
1	low	default	default	✓
2	low	default	100	✓
3	high	default	default	
4	high	0.15	100	
5	normal	0.15	100	
6	high	default	100	✓

Table 3.1: Parameter choices for CANU v1.3: three of these assemblies were polished with QUIVER

The basic statistics for the eight assemblies are provided in Table 3.2. In addition to standard contiguity statistics (N50[1], L50[2], NG50[3], and LG50[4]), total assembled size and contig length distributions, we evaluated the assemblies using several other independent metrics. We mapped (i) about 129K cowpea WGS contigs assembled from short reads ([54], assembly v.0.03), (ii) about 200M 2X100 paired-end Illumina reads generated at UCR in 2014, and (iii) transcripts assembled from RNA-Seq short reads. In Table 3.2 we report the percentage of DNA sequenced mapped with BWA with a minimum MapQ of 30. Finally, we compared the assemblies against the high-density genetic map available from [54]. To evaluate possible chimeric contigs, we BLASTed 121bp-long design sequence for the 51,128 genome-wide SNPs described in [54] against each assembly, then we identified which contigs had SNPs mapped to them, and what linkage group (chromosome) of the genetic map those mapped SNPs belonged to. Chimeric contigs are revealed when their mapped SNPs belong

[1]length for which the set of contigs of that length or longer accounts for at least half of the assembly size
[2]minimum number of contigs accounting for at least half of the assembly
[3]length for which the set of contigs of that length or longer accounts for at least half of the 620Mb genome
[4]minimum number of contigs accounting for at least half of the 620Mb genome

	CANU$_1$	CANU$_2$	ABRUIJN	FALCON	CANU$_3$	CANU$_4$	CANU$_5$	CANU$_6$
contig N50 (bp)	4,859,617	4,498,063	1,896,002	2,869,362	3,280,469	2,797,949	2,666,731	**5,340,274**
contig L50	30	32	74	49	42	51	55	**29**
contig NG50 (bp)	**3,767,556**	3,417,577	1,330,435	1,737,012	2,431,239	1,949,515	2,068,575	3,451,071
contig LG50	43	45	119	73	63	73	77	**42**
total assembled (bp)	506,154,442	**516,817,613**	478,230,679	511,933,729	503,187,311	516,537,734	515,949,175	507,773,747
# contigs	894	928	**538**	1,820	1,038	1,110	1,140	897
# contigs ≥100kbp	220	288	437	404	299	354	334	278
# contigs ≥1Mbp	104	107	151	118	128	142	145	103
# contigs ≥10Mbp	7	8	0	1	2	2	0	**9**
longest contig (bp)	18,473,372	**18,498,533**	8,846,014	10,554,495	14,090,735	14,331,160	9,775,097	17,211,165
WGS contigs ≥500bp, % mapped (129K)	98.27412%	**98.77014%**	88.30652%	97.84959%	98.30618%	98.25853%	98.23673%	98.73930%
UCR2014 reads, % properly paired (202M)	92.59433%	**92.64181%**	92.30106%	91.95107%	92.52969%	92.63057%	92.62330%	92.59763%
UCR2014 reads, % mapped (202M)	64.35764%	63.50279%	64.21367%	59.49035%	**64.38425%**	63.00587%	63.22414%	62.84466%
assembled transcripts, % mapped (157K)	92.60644%	94.83972%	**94.95582%**	94.16235%	92.65416%	92.52276%	92.46959%	94.85657%
total length with 100% consistent LG (bp)	331,956,528	338,556,993	379,029,914	312,593,019	356,505,616	349,534,672	347,586,448	**425,812,490**

Table 3.2: Assembly statistics of eight assemblies for cowpea; all reads/transcripts/BAC assemblies were mapped with BWA, MapQ≥30; number in boldface are the best statistics (min or max) across assemblies; for # contigs ≥100kbp and ≥1Mbp it is not obvious whether to report min or max

to more than one linkage group. The last line of Table 3.2 reports the total size of contigs in each assembly for which (i) they have at least one SNPs mapped to it and (ii) all SNPs belong to the same linkage group (i.e. likely to be non-chimeric). Observe in Table 3.2 that there is no single assembly that is the "best" in each row. CANU$_6$ has the highest N50 and the lowest L50, but CANU$_2$ has the longest contig. CANU$_1$ has the highest NG50. ABRUIJN has the smallest number of contigs.

NOVO&STITCH was run on the eight assemblies in Table 3.2 using two Bionano Genomics optical maps, the first obtained using the BspQI nicking enzyme (which recognizes "GCTCTTC"), and the second obtained with the BssSI nicking enzyme ("CACGAG"). For each optical map we used two sets of parameters, called "strict" (`-a 3000 -b 0.1 -c 10000 -d 0.5 -e 0.9 -h 25 -r 0.2`) and "loose" (`-a 0 -b 0.2 -c 5000 -d 0.5 -e 0.8 -h 25 -r 0.2`). Please refer to the README at `https://github.com/ucrbioinfo/Novo_Stitch` for details about these parameters. For convenience, in first column of Table 3.3, we copied the best statistics across the eight assemblies in Table 3.2. Note that no individual assembly,

	best of 8	BspQI (loose)	BspQI (strict)	BssSI (loose)	BssSI (strict)
contig N50 (bp)	4,859,617	9,944,851	9,944,851	9,584,779	9,584,779
contig L50	29	19	19	19	19
contig NG50 (bp)	3,767,556	9,944,851	8,187,172	7,956,155	7,826,863
contig LG50	42	19	24	24	24
total assembled (bp)	516,817,613	522,393,141	523,526,657	520,162,831	523,249,509
# contigs	538	791	798	791	798
# contigs ≥100kbp	N/A	211	218	211	218
# contigs ≥1Mbp	N/A	72	72	66	69
# contigs ≥10Mbp	9	18	18	17	17
longest contig (bp)	18,498,533	21,980,320	21,980,320	22,385,362	22,385,362
WGS contigs ≥500bp, % mapped (129K)	98.77014%	97.77496%	97.79009%	97.40359%	97.02018%
UCR2014 reads, % properly paired (202M)	92.64181%	92.57437%	92.58778%	92.47305%	92.50176%
UCR2014 reads, % mapped (202M)	64.38425%	62.20807%	62.11027%	61.82553%	61.63417%
assembled transcripts, % mapped (157K)	94.95582%	93.93669%	93.90570%	94.01125%	93.46803%
% contigs with 100% consistent LG	425,812,490	429,367,225	430,234,966	423,454,837	434,621,644

Table 3.3: Assembly statistics of NOVO&STITCH on the eight cowpea assemblies using either the BspQI or the BssSI optical map, "best of 8" is a copy the best statistics (boldface) among the eight assemblies in Table 3.2 – no individual assembly, however, has these statistics; see text about strict and loose parameters; all DNA sequences were mapped with BWA, MapQ≥30

however, has these statistics. Observe in Table 3.3 that NOVO&STITCH almost doubled the N50, reduced the L50 from 29 to 19, increased the number of contigs ≥10Mb from 9 to 17-18. Mapping statistics remained unaltered, as well as the agreement with the genetic map. Taken all together, these statistics indicate that NOVO&STITCH produced a much more contiguous assemblies, with no more chimeric contigs than the eight input assemblies.

3.3.2 Experimental results on *P. infestans* assemblies

We sequenced a strain of *P. infestans* from California called "1306". Strain 1306 is a diploid (other *P. infestans* strains are triploid or aneuploid), has 11-14 chromosomes and an estimated genome size of 220 Mb. *P. infestans* 1306 was sequenced using single-molecule real-time sequencing (Pacific Biosciences RSII). A total of 17 SMRT cells yielded about 3.1M reads for a total of 24.87 Gbp (113x genome equivalent). We tested NOVO&STITCH on six assemblies of *P. infestans*. We generated two assemblies with CANU v1.5, one on

	Falcon	Canu$_{10K}$	Canu$_{full}$	ABruijn$_{10K}$	ABruijn$_{corr}$	ABruijn$_{trim}$	N&S$_{loose}$	N&S$_{strict}$
contig N50 (bp)	481,068	131,313	135,263	356,459	293,280	302,893	769,322	730,890
contig L50	107	462	473	142	171	169	74	82
total assembled (bp)	215,910,203	305,686,040	292,352,599	195,768,168	177,232,870	175,149,119	240,150,657	250,416,680
# contigs	1,364	3,496	2,863	835	888	867	1,304	1,329
# contigs ≥100kbp	445	667	725	561	539	522	398	423
# contigs ≥1Mbp	36	12	7	19	9	10	54	55
longest contig (bp)	4,206,720	1,810,393	1,813,497	2,437,907	2,004,950	1,638,783	4,930,683	4797067
miSeq reads, % mapped (47M)	98.4995%	98.6503%	98.2370%	98.0305%	98.3051%	98.2923%	98.0928%	98.0958%
miSeq reads, % properly paired (47M)	96.3825%	97.6855%	95.5383%	93.5911%	94.8410%	94.8218%	95.6384%	95.7194%
1% miSeq reads, % mapped (0.47M)	97.6510%	97.9313%	96.7831%	96.4854%	97.3612%	97.3092%	97.1461%	97.1521%
Dovetail reads, % mapped (202M)	97.7712%	97.8934%	97.6519%	97.5093%	97.6161%	97.5835%	97.4953%	97.4989%
Dovetail reads, % properly paired (202M)	38.7274%	37.5416%	37.4140%	38.6057%	38.5723%	38.4578%	37.9324%	37.7392%
0.1% Dovetail reads, % mapped (0.2M)	91.6264%	92.0876%	91.3643%	90.8826%	91.3535%	91.2612%	91.1237%	91.1447%

Table 3.4: Statistics of six input assemblies for *P. infestans* and two stitched assemblies (N&S = Novo&Stitch) with strict and loose parameters; all reads were mapped with Bwa, except for 1% of miSeq and 0.1% of Dovetail which were mapped using Blast (e-value<1e-30)

the entire dataset (Canu$_{full}$, 113x coverage) and one on PacBio reads longer than 10Kb (Canu$_{10K}$, 80.9x coverage). We generated three assemblies with ABruijn v0.4 on three datasets, namely (i) PacBio reads longer than 10Kb (ABruijn$_{10K}$, $k = 17$, 80.9x coverage), (ii) all PacBio reads corrected by Canu (ABruijn$_{corr}$, $k = 17$, 75.4x coverage) and (iii) all PacBio reads corrected and trimmed by Canu (ABruijn$_{trim}$, $k = 17$, 73.6x coverage). One assembly was produced with Falcon on the whole dataset by the UC Davis core facility.

Novo&Stitch was run on the six assemblies in Table 3.4 using a Bionano Genomics optical map. To evaluate the quality of these assemblies, we mapped about 47M miSeq reads and 202M Dovetail read using BWA. We also mapped a fraction of those reads using Blast, which does not penalize the mapping quality in case of alignment of a read to multiple locations. The last two columns report the statistics of Novo&Stitch using strict and loose parameters. Observe again, how Novo&Stitch significantly improved the contiguity of the assembly (N50, L50, longest contig, etc.) while maintaining mapping statistics similar to the six input assemblies.

3.4 Conclusions

In this chapter, we presented a new assembly reconciliation tool called NOVO&STITCH for improving the contiguity of *de novo* genome assemblies using optical maps. NOVO&STITCH uses the alignment of contigs from multiple input assemblies to an optimal map to detect overlaps between contigs and drive the stitching process. Experimental results on *V. unguiculata* and *P. infestans* clearly demonstrates that the addition of the optical map can significantly improve the contiguity of genome assemblies. The optical map can be used again on the improved stitched assembly to create scaffolds.

Algorithm 2 Greedy algorithm for SMALLEST SUBDAG problem

```
1:  procedure GA(G = (V, E), S, T)
2:      current_set ← S ∪ T
3:      subgraph ← G
4:      for s in S and t in T do
5:          path ← BFS(G, s, t)                ▷ compute the shortest path in G from s to t
6:          if no_vertices(path) < no_vertices(subgraph) then
7:              subgraph, s*, t* ← path, s, t
8:      current_set ← current_set −{s*, t*}
9:      while current_set ≠ ∅ do
10:         path* ← G
11:         for x in current_set do
12:             for y in subgraph do
13:                 if x ∈ S then                          ▷ x is a source
14:                     path ← BFS(G, x, y)
15:                 else                                   ▷ x is a sink
16:                     path ← BFS(G, y, x)
17:                 if no_vertices(path) < no_vertices(path*) then
18:                     path*, x* ← path, x
19:         current_set, subgraph ← current_set −x*, subgraph ∪ path*
20:     return subgraph
```

Algorithm 3 Dynamic programming algorithm for optimal set of compatible alignments

```
1:  procedure COMPATIBLE(A, L, C)
2:      A', S[1], Last[1] ← ∅, L[1], NULL
3:      for i ← 2 to n do
4:          S[i], j* ← L[i], 0
5:          for j ← 1 to i − 1 do
6:              if C[i, j] = True then
7:                  if S[i] < S[j] + L[i] then
8:                      S[i], j* ← S[j] + L[i], j
9:          Last[i] ← j*
10:     S* ← 0, pos* ← 0
11:     for i ← 1 to n do
12:         if S[i] > S* then
13:             S*, pos* ← S[i], i
14:     i ← pos*
15:     while i ≠ NULL do
16:         A', i ← A' ∪ A[i], Last[i]
17:     return A'
```

Chapter 4

OMGS: Optical Map-based Genome Scaffolding

As metioned in the Introduction, genome scaffolding tools either use paired-end/mate-pair/linked/Hi-C reads or genome-wide maps. The first group includes scaffolding tools for second generation sequencing data, such as Bambus [66, 41], GRASS [29], MIP [70], Opera [26], SCARPA [22], SOPRA [19] and SSPACE [9] and the scaffolding modules from assemblers ABySS [79], SGA [78] and SOAPdenovo2 [49]. Since the relative orientation and approximate distance between paired-end/mate-pair/linked/Hi-C reads are known, the consistent alignment of a sufficient number of reads to two contigs can indicate their relative order, their orientation and the distance between them. An extensive comparison of scaffolding methods in this first group of tools can be found in [33].

The second group uses genome-wide maps such as genetic maps [85], physical maps, or optical maps. According to the markers provided by these maps, contigs can be anchored

to specific positions so that their order and orientations can be determined. The distance between contigs can also be estimated with varying degree of accuracy depending on the density of the map.

A few scaffolding algorithms that use optical maps are available. SOMA appears to be the first published tool that can take advantage of optical maps but it can only deal with a non-fragmented optical map [58]. The scaffolding tool proposed in [69] was used for two bacterial genomes *Yersinia pestis* and *Yersinia enterocolitica*, but the software is no longer publicly available. In the last few years, Bionano optical maps have become very popular, and have been used to improve the assembly contiguity in many large-scale *de novo* genome assembly projects (e.g., goat, apple, barley, maize, quinoa, sea bass [8, 63, 18, 50]). To the best of our knowledge, the main tools used to generate scaffolds using Bionano optical maps are SewingMachine from KSU [76] and HybridScaffold from Bionano Genomics (unpublished, 2016). SewingMachine seems to be favored by practitioners over HybridScaffold.

Both HybridScaffold and SewingMachine have, however, a serious limitation: they can only deal with one optical map at a time, forcing users to alternate or iterate over optical maps when multiple maps are available. In this chapter, we introduce a novel scaffolding algorithm called OMGS that for the first time can take advantage of any number of optical maps. OMGS solves several optimization problems to generate scaffolds with optimal contiguity and correctness.

4.1 Problem definition

The input to the problem is the genome assembly to be scaffolded (represented by a set of assembled contigs), and one or more optical maps (represented by a set of sets of genomic distances). We use $C = \{c_i | i = 1, \ldots, l\}$ to denote the set of contigs in the genome assembly, where each c_i is a string over the alphabet $\{A, C, G, T\}$. Henceforth, we assume that the contigs in C are chimera-free.

An optical map is composed by a set of optical molecules, each of which is represented by an ordered set of positions for the restriction enzyme sites. As said, optical molecules are obtained by an assembly process similar to sequence assembly, but we will reserve the term *contig* for sequenced contigs. We use $M = \{m_i | i = 1, \ldots, n\}$ to denote the optical map, where each optical molecule m_i is an ordered set of integers, corresponding to the distances in base pairs between two adjacent restriction enzyme sites on molecule m_i. By digesting *in silico* the contigs in C using the same restriction enzyme used to produce the optical map and matching the sequence of adjacent distances between sites, one can align the contigs in C to the optical map M. If one is given multiple optical maps obtained using different restriction enzymes, M will be the union of the molecules from all optical maps. In this case, each genomic location is expected to be covered by multiple molecules in M. As said, high quality alignments allows one to anchor and orient contigs to specific coordinates on the optical map. When multiple contigs are aligned to the same optical map molecule, one can order them and estimate the distance between them. By filling these gaps with a number of N's equal to the estimated distance, longer DNA sequences called *scaffolds* can be obtained.

A series of practical factors make the problem of scaffolding non-trivial. These factors include imprecisions in optical maps (e.g., mis-joins introduced during the assembly of the optical map [38]), unreliable alignments between contigs and optical molecules, and multiple inconsistent anchoring positions for the same contigs. As a consequence, it is appropriate to frame this scaffolding problem as an optimization problem.

We are now ready to define the problem. We are given an assembly represented by a set of contigs C, a set of optical map molecules M and a set of alignments $A = \{a_{1,1}, a_{1,2}, \ldots a_{l,n}\}$ of C to M, where $a_{i,j}$ is the alignment of contig c_i to optical map molecule o_j. The problem is to obtain a set of scaffolds $S = \{s_1, s_2, \ldots s_k\}$ where each s_i is a string over the alphabet $\{A, C, G, T, N\}$, such that (i) each contig c_i is contained/assigned to exactly one scaffold, (ii) the *contiguity* of S is maximized and (iii) the conflicts of S with respect to A are minimized. This optimization problem is not rigorously defined unless one defines precisely the concepts of *contiguity* and *conflict*, but this description captures the spirit of what we want to accomplish. In genome assembly, the assembly contiguity is usually captured by statistical measures like the N50/L50 or the NG50/LG50. The notion of conflict is not easily quantified, and even if it was made precise, this multi-objective optimization problem would be hard to solve. We decompose this problem into two separate steps, namely (a) scaffold detection and (b) gap estimation, as explained below.

4.2 Methods

As said, our proposed method is composed of two phases: scaffold detection and gap estimation. In the first phase, contigs are grouped into scaffolds and the order of contigs

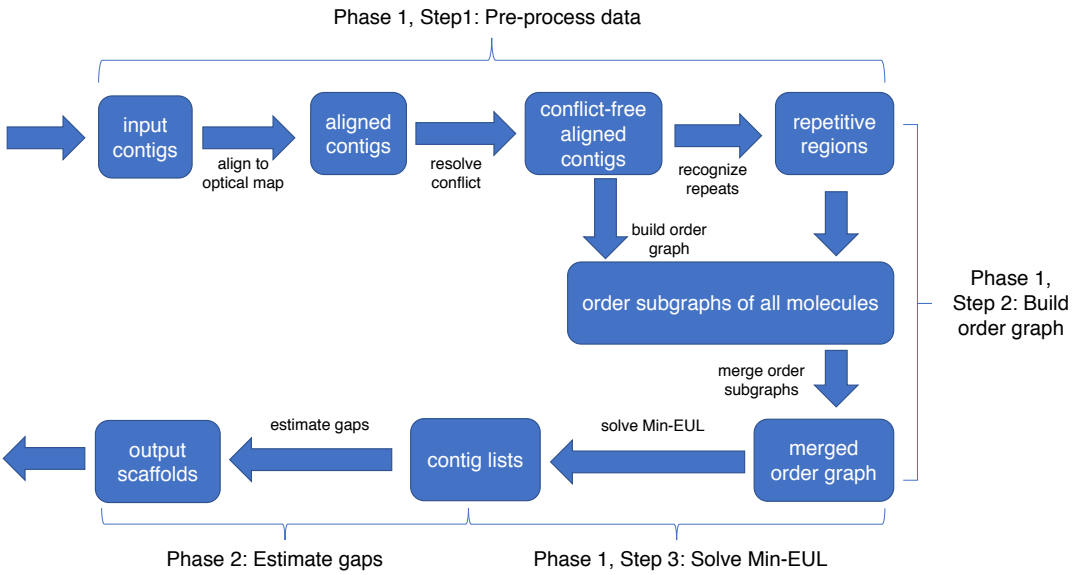


Figure 4.1: Pipeline of the proposed algorithm

in each scaffold is determined. In the second phase, distances between neighboring contigs assigned to scaffolds are estimated. The pipeline of the proposed algorithm is illustrated in Figure 4.1.

4.2.1 Phase 1: Detecting scaffolds

Phase 1 has three major steps. In Step 1, we align *in silico*-digested chimeric-free contigs to the optical maps (e.g., for a Bionano optical map, we use RefAligner), but not all alignments are used in Step 2. We only consider alignments that (i) exceed a minimum confidence level (e.g, confidence 15 in the case of RefAligner); (ii) do not overlap each other more than a given genomic distance (e.g, 20 kbp) and (iii) do not create conflict with each other. The method we use here to select conflict-free alignments was introduced in our

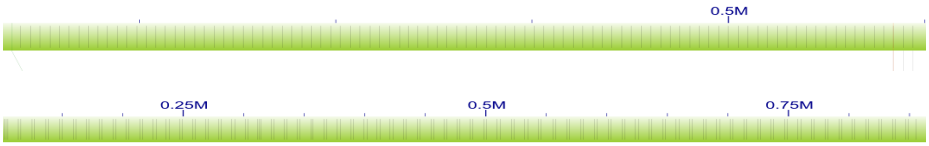
previous work [62]. In Step 2, we compute candidate scaffolds by building the *order graph* and formulating an optimization problem on it. In Step 3, either the exhaustive algorithm or a $\log n$-approximation algorithm is used to solve the optimization problem (depending on the size of the graph) and produce the final scaffolds.

Building the order graph

The order graph O is a directed weighted graph in which each vertex represents a contig. Given two contigs c_i and c_j aligned to an optical molecule o with alignments a_i and a_j, we create a directed edge (c_i, c_j) in O if (i) the starting coordinate of alignment a_i (that we call $a_i.start$ henceforth) is smaller than the starting coordinate of alignment a_j (that we call $a_j.start$ henceforth) and (ii) there is no other alignment a_k such that $a_k.start$ is between $a_i.start$ and $a_j.start$ and (iii) there are no conflict sites between $a_i.end$ and $a_j.start$ on the optical molecule, as defined below. For each alignment a between optical molecule o and contig c, we compute the left overhang l_o and right overhang r_o from o and the left overhang l_c and right overhang r_c from c. The left-end of alignment a is declared a *conflict site* if (i) both l_o and l_c are longer than some minimum length (e.g., 50 kbp) and (ii) at least one restriction enzyme sites appear in both l_o and l_c. A symmetric argument applies to the right-end of the alignment, which determines the values for r_o and r_c.

Directed edge (c_i, c_j) is assigned a weight equal to $\text{qual}(o, a_i.end, a_j.start)$ * ($\text{conf}(a_i)$+ $\text{conf}(a_j)$), where (i) $\text{qual}(o, a_i.end, a_j.start)$ is the *quality* of the region between $a_i.end$ and $a_j.start$ on molecule o (higher is better, defined next) and (ii) $\text{conf}(a)$ is the confidence score provided by RefAligner alignment a (higher is better). The quantity $\text{qual}(o, s, t)$ is defined based on the length of a repetitive region between coordinates (s, t). Based on our

(A)



(B)

Figure 4.2: Examples of single-site repetitive region (A) and two-site repetitive region (B) in optical maps. Observe the small variations in the repetitive patterns in (B)

experience, assembly mis-joins on optical molecule almost always happen in repetitive regions [38]. Given the length of repetitive region len_rep(o, s, t) in base pairs (defined below), we define the quality of o in the interval (s, t) as qual$(o, s, t) = e^{-\text{len_rep}(o,s,t)/100000}$. When a_i and a_j have a small overlap (e.g., shorter than 20 kbp), we set len_rep$(o, s, t) = 0$.

We recognize repetitive regions in optical molecules based on the distribution of restriction enzyme sites. For a molecule o with n sites, let m_i be the coordinate of the i-th site for $i = 1, \ldots, n$. As said, molecule o can be represented as a list of positions $\{m_i | i = 1, \ldots, n\}$. In order to determine the repetitive regions in o, we slide a window that covers k sites (e.g., $k = 10$ sites). At each position $j = 1, \ldots, n - k + 1$, we select window $w_j = \{m_j, \ldots, m_{j+k-1}\}$. While repetitive regions in genome can be highly complex (see, e.g., [99]), we observed only two types of repetitive regions in optical molecules, namely single-site repetitive region (see Figure 4.2A) and two-site repetitive region (see Figure 4.2B). It is entirely possible that more complex repetitive regions exist: if they do, they seem rare. Based on this observation, in order to decide whether window w_j is repetitive, we first compute two lists of pairwise distances between sites, namely $D_{j,1} = \{m_{j+l} - m_{j+l-1} | l = 1, \ldots, k - 1\}$ and $D_{j,2} = \{m_{j+l+1} - m_{j+l-1} | l = 1, \ldots, k - 2\}$ that we call *distance lists*, then we apply the statistical test described next.

In our statistical test we assume that the values in the distance lists that belong to repetitive regions are independent and identically distributed as a Gaussian. We further assume that each specific distance list ($D_{j,1}$ or $D_{j,2}$) is associated with a Gaussian with a specific mean $\mu_{j,q}$ ($q \in \{1, 2\}$). Finally, we assume that the variance σ^2 is globally shared by all molecules. An estimator of the mean is $\mu_{j,q}$ is $\hat{\mu}_{j,q} = \sum_{i=1}^{k-q} d_i/(k-q)$, where $d_i \in D_{j,q}$ and k is the window size. To estimate σ^2, we first get an initial (rough) estimate of the repetitive regions on all molecules. Given a particular $D_{j,q}$, let d_{max} and d_{min} be the maximum and minimum distance in $D_{j,q}$. We declare a distance list $D_{j,q}$ to be *estimated repetitive* if $d_{max} - d_{min}$ is smaller than a given distance (e.g., 1.5 kbp). We collect all estimated repetitive lists in set $R = \{D_p \text{ is estimated repetitive} | p = 1, \ldots, P\}$ and the estimated mean $\hat{\mu}_p$ for each distance list D_p in the set R, where P is the total number of estimated repetitive lists. According to the density function of Gaussian distribution, the log likelihood of one D_p is

$$-\frac{|D_p|}{2}\log(2\pi) - \frac{|D_p|}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{d_i \in D_p}(d_i - \hat{\mu}_p)^2.$$

The total log likelihood is the sum of the log likelihoods across all D_p's in R, which is

$$\log L(\sigma^2) = -\frac{\sum_{p=1}^{P}|D_p|}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{p=1}^{P}\sum_{d_i \in D_p}(d_i - \hat{\mu}_p)^2,$$

after ignoring all terms not related to σ^2. To maximize $\log L(\sigma^2)$, we require that the derivative of total log likelihood

$$\frac{\partial \log L(\sigma^2)}{\partial \sigma^2} = 0,$$

that is,

$$-\frac{\sum_{p=1}^{P}|D_p|}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{p=1}^{P}\sum_{d_i \in D_p}(d_i - \hat{\mu}_p)^2 = 0.$$

After some simplification, the estimator for variance becomes

$$\hat{\sigma}^2 = \frac{\sum_{p=1}^{P} \sum_{d_i \in D_p} (d_i - \hat{\mu}_p)^2}{\sum_{p=1}^{P} |D_p|}.$$

Then, we carry out the test on the statistic $d_{max} - d_{min}$ for each $D_{j,q}$. It is well-known that the joint density function of order statistics is

$$f_{X(i),X(j)}(u,v) = \frac{n!}{(i-1)!(j-1-i)!(n-j)!} f_x(u) f_x(v) [F_x(u)]^{i-1} [F_x(v) - F_x(u)]^{j-1-i} [1 - F_x(v)]^{n-j} \tag{4.1}$$

for $-\infty < u < v < +\infty$, where $X(i)$ and $X(j)$ are the i-th and j-th order statistics in $X_1, \ldots, X_n$ and F_x and f_x are the distribution function and density function of each X_i, respectively. Using (4.1), the joint density function of (d_{max},d_{min}) can be expressed as

$$f_{d_{max},d_{min}}(u,v) = n(n-1) f_{d_i}(u) f_{d_i}(v) [F_{d_i}(v) - F_{d_i}(u)]^{n-2}$$

for $-\infty < u < v < +\infty$, where F_{d_i} and f_{d_i} are the distribution function and density function of $d_i \sim N(\hat{\mu}_{j,q}, \hat{\sigma}^2)$, respectively.

Now, let $X = d_{max} - d_{min}$ and $Y = d_{min}$. Then $d_{max} = X + Y$ and $d_{min} = Y$, and the corresponding Jacobian determinant is

$$J = \begin{vmatrix} \partial d_{\max}/\partial X & \partial d_{\max}/\partial Y \\ \partial d_{\min}/\partial X & \partial d_{\min}/\partial Y \end{vmatrix} = \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} = 1.$$

Thus, the joint density function of (X, Y) is given by

$$f_{X,Y}(x,y) = f_{d_{max},d_{min}}(x+y, y)|J| = n(n-1) f_{d_i}(y) f_{d_i}(x+y) [F_{d_i}(x+y) - F_{d_i}(y)]^{n-2},$$

where $x \geq 0$ and $-\infty < y < +\infty$. By integrating over Y, the density function of $X = d_{max} - d_{min}$ becomes

$$f_{d_{max}-d_{min}}(x) = \int_{-\infty}^{+\infty} n(n-1) f_{d_i}(y) f_{d_i}(x+y) [F_{d_i}(x+y) - F_{d_i}(y)]^{n-2} dy, x \geq 0.$$

Let now X be a random variable associated with the distribution $f_{d_{max}-d_{min}}$. If the p-value $P(X > d_{max} - d_{min})$ is greater than a predefined threshold (e.g., 0.001), we accept the null hypothesis and declare that window w_j is repetitive. The repetitive regions for the entire molecule o is the union of all the windows w_j's recognized as repetitive according to the test above.

Once the order graph of each optical molecule is built, we connect all the order graphs which share the same contigs using the association graph introduced in [62]. The association graph is an undirected graph in which each vertex represents an optical molecule and an edge indicates that the two molecules share at least one contig aligned to both of them. We use depth first search (DFS) to first build a spanning forest of the association graph. Then, we traverse each spanning tree and connect the corresponding order subgraph to the final order graph. Every time we add a new graph, new vertices and new edges might be added. If an edge already exist, the weights of the new edges are added to the weights of existing edges.

Generating scaffolds

Once the order graph O is finalized, we generate the ordered sequence of contigs in each scaffold. In the ideal case, each connected component O_i of O is a directed acyclic graph (DAG) because the genome is one-dimensional and the order of any pair of contigs is unique. In practice however, O_i may contain cycles caused by the inaccuracy of the alignments and mis-joins in optical molecules. To convert each cyclic component O_i into a DAG, we solve the Minimum Feedback Arc Set problem on O_i. In this problem, the objective is to find the minimum subset of edges (called *feedback arc set*) containing at least one edge of every

cycle in the input graph. Since the minimum feedback edge set problem is APX-hard, we use the greedy local heuristics introduced in [4] to solve it.

We then break each DAG G_i of connected component O_i into subgraphs as follows. In each subgraph, we require that the order of every pair of vertices to be uniquely determined by the directed edges. This allows us to uniquely determine the order of the contigs for each scaffold. The formal definition of this optimization problem is as follows.

Definition 5 (Minimum Edge Unique Linearization problem) INPUT: *A weighted directed acyclic graph $G = (V, E)$.* OUTPUT: *A subset of edges $E' \subseteq E$ such that (i) in each connected component G'_i of the graph $G' = (V, E - E')$ obtained after removing E', the order of all vertices can be uniquely determined, and (ii) the total weights of the edges in E' is the minimum among all the subset of edges satisfying (i).*

In Theorem 6 below, we show that the MINIMUM EDGE UNIQUE LINEARIZATION problem (MIN-EUL) is NP-hard by proving that it is equivalent to the MINIMUM EDGE CLIQUE PARTITION problem (MIN-ECP), which is know to be NP-hard [21]. In MIN-ECP, we are given a general undirected graph, and we need to partition its vertices into disjoint clusters such that each cluster forms a clique and the total weight of the edges between clusters is minimized.

Theorem 6 MIN-EUL *is equivalent to* MIN-ECP.

Proof. First, we show that MIN-EUL polynomially reduces to MIN-ECP. Given an instance $G = (V, E)$ of MIN-EUL, we build an instance $G' = (V', E')$ of MIN-ECP as follows. Let $V' = V$. For each pair of vertices $u, v \in V'$ where v is reachable from u, define

an undirected edge between u and v in E'. For each directed edge $(u, v) \in E$, set the weight of the corresponding undirected edge $(u, v) \in E'$ as 1. Set the weights of the other edges in E' as 0. Then it is easy to see that a MIN-EUL solution to G' is equivalent to a MIN-ECP solution to G and vice versa.

Now we show that MIN-ECP polynomially reduces to MIN-EUL. Given an instance $G' = (V', E')$ (assuming G' is connected) of MIN-ECP, we build an instance $G = (V, E)$ of MIN-EUL as follows. Let $V = V'$. Pick any total linear order O of all vertices in V'. For each undirected edge $(u, v) \in E'$ where $\text{rank}(u) < \text{rank}(v)$ in O, define a directed edge from u to v in E and set its weight to be the same as its corresponding undirected edge in E'. For any two vertices $u, v \in V$, where $\text{rank}(u) < \text{rank}(v)$ and $(u, v) \notin E'$, add a new vertex $x_{uv} \in V$ with $\text{rank}(x_{uv}) = \text{rank}(v)$ and a directed edge u to x_{uv} of weight 1 in E. Now for each pair of vertices $u, v \in V$ where $\text{rank}(u) < \text{rank}(v)$ and $(u, v) \notin E$, add a directed edge u to v with weight zero in E. Then it is easy to see that a MIN-EUL solution to G corresponds to a MIN-ECP solution to G' and vice versa. ■

Given the complexity of MIN-EUL, we propose an exponential time exact algorithm and a polynomial time $\log n$-approximation algorithm for solving it. To describe the exact algorithm, we need to introduce some notations. A *conjunction* vertex in a DAG is a vertex which has more than one incoming edge or outgoing edge. A *candidate* edge is an edge which connects at least one conjunction vertex. In Theorem 7 below, we prove that the optimal solution E' of MIN-EUL must only contain candidate edges. Let E_c be the set of all candidate edges in the DAG G, for each subset E'_j of E_c, we check whether the graph $G' = (V, E - E'_j)$ satisfies requirement (i) in Definition 5 after removing E'_j from G. Among

all the feasible E'_j, we produce the set of edges with minimum total weights. To check whether E'_j is feasible, we use a variant of topological sorting which requires one to produce a unique topological ordering. To do so, we require that in every iteration of topological sorting, the candidate node to be added to sorted graph is always unique. Details of this algorithm are shown as Algorithm 4.

Algorithm 4 Sketch of the algorithm for checking whether a DAG provides an unique ordering

```
1:  procedure ORDER_UNIQUENESS_CHECK(G = (V, E))
2:      S = nodes with no incoming edges
3:      while S ≠ ∅ do
4:          if |S| > 1 then
5:              return False
6:          remove a node n from S
7:          for each node m with an edge e = (n, m) do
8:              remove edge e from the E
9:              if m has no other incoming edges then
10:                 insert m into S
11:     return True
```

Theorem 7 *The optimal solution E' of* MIN-EUL *only contains candidate edges.*

Proof. For sake of contradiction, we assume that E' contains a non-candidate edges (u, v). Since E' is optimal, $G' = (V, E - E')$ satisfies condition (i) in Definition 5. Since both u and v are conjunction vertices, u has only one incoming edge and v has only one outgoing edge.

Therefore, by adding (u, v) to $G' = (V, E - E')$, we still satisfy condition (i) in Definition 5. Since the weight of (u, v) is positive, the total weight of $E - E' + \{(u, v)\}$ is larger than $E - E'$. Therefore $E' - \{(u, v)\}$ is optimal, contradicting the optimality of E'. ■

As said, MIN-EUL is equivalent to MIN-ECP (Theorem 6). In addition, the authors of [21] showed that for any instance of MIN-ECP one can find an equivalent instance of the MINIMUM DISAGREEMENT CORRELATION CLUSTERING problem. As a consequence, any algorithm for the MINIMUM DISAGREEMENT CORRELATION CLUSTERING problem could be used to solve MIN-EUL. In our tool OMGS, we implemented a $O(\log n)$-approximation algorithm based on linear programming, originally proposed in [20]. Standard linear programming packages (e.g., GLPK or CPLEX) are used to solve the linear program. We use the exact algorithm for DAGs with no more than twenty candidate edges, and the approximation algorithm for larger DAGs.

4.2.2 Phase 2: Estimating gaps

Let $s = \{c_i | i = 1, \ldots, h\}$ be one of the scaffold generated in Phase 1 where each c_i is a contig. In Phase 2, we estimate the length l_i of the gap between each pair c_i and c_{i+1} of adjacent contigs. We estimate all gap lengths $L = \{l_i | i = 1, \ldots, h - 1\}$ at the same time using the distances between the contigs provided by the alignments and the corresponding order subgraphs. We assume that each l_i is chi-square distributed with α_i degrees of freedom. The choice of chi-square distribution is due to its additive properties, namely the sum of independent chi-squared variables is also chi-squared distributed. Recall that each order subgraph O_k provides an unique ordering $x_k = \{c_j | j = 1, \ldots, r\}$ of the contigs aligned to molecule o_k, while the coordinates of the alignment provide the distances between all pairs

of adjacent contigs c_j and c_{j+1} as $y_k = \{d_j | j = 1, \ldots, r-1\}$. We use the distances d_j as samples to estimate gap lengths l_i. If edge (c_j, c_{j+1}) in O_k is removed in the order graph O when solving MIN-EUL in Phase 1, d_j will be considered not reliable and removed from y_k.

In the ideal case, d_j should be a sample of a single l_i (i.e., $c_j c_{j+1}$ in x_k corresponds to $c_p c_{p+1}$ in s). In practice however, $c_j c_{j+1}$ in x_k will corresponds to a different pair $c_p c_q$ in s where $q > p+1$ (i.e., $c_{p+1} \ldots c_{q-1}$ are missing from the order subgraph because some alignments with low confidence were removed in Step 1 of Phase 1). In this situation, after subtracting the length of missing contigs from d_j, $d_j - \sum_{c=c_{p+1}}^{c_{q-1}} |c|$ is a sample of $\sum_{i=p}^{q-1} l_i$ where $|c|$ represents the length of contig c. Since $l_p, \ldots, l_{q-1}$ are independent chi-square random variables, $\sum_{i=p}^{q-1} l_i$ is chi-square distributed with degree of freedom $\sum_{i=p}^{q-1} \alpha_i$. Since the density function of a chi-square random variable X with degree of freedom k is

$$f_X(x) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2}$$

where Γ is the gamma function, the likelihood of $\sum_{i=p}^{q-1} l_i$ with observation

$$\gamma = d_j - \sum_{c=c_{p+1}}^{c_{q-1}} |c|$$

is

$$\frac{1}{2^{\beta}\Gamma(\beta)} \gamma^{\beta-1} e^{-\gamma/2},$$

where $\beta = \sum_{i=p}^{q-1} \frac{\alpha_i}{2}$. Therefore, the log likelihood function for one sample is

$$\log l = (\beta - 1) \log \gamma - \frac{\gamma}{2} - \beta \log 2 - \log \Gamma(\beta).$$

The total log likelihood is the sum of the log likelihoods across all samples.

To find the α_i maximizing the total log likelihood, we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [3]. Since the mean of a chi-square distribution equals

its degree of freedom, we obtain the estimated gaps $\hat{l}_i = \hat{\alpha}_i$. For the case in which the l_i are pre-estimated as negative in the first step, the second and third steps are ignored and the pre-estimated distances are used as final estimates.

Finally, we add $\lceil \hat{l}_i \rceil$ nucleotides (represented by `Ns`) between each pair of contigs c_i and c_{i+1}. When $\hat{l}_i < 0$, we add exactly 100 `Ns` between c_i and c_{i+1}, which is the convention for a gap of unknown length.

4.3 Experimental results

We compared OMGS against KSU SewingMachine (version 1.0.6, released in 2015) and Bionano HybridScaffold (version 4741, released in 2016) which, to the best of our knowledge, are the only available scaffolding tools for Bionano Genomics optical maps. All tools were run with default parameters, unless otherwise specified. We collected experimental results on scaffolds of (i) cowpea (*Vigna unguiculata*) and (ii) fruit fly (*Drosophila melanogaster*).

4.3.1 Experimental results on cowpea assemblies

Cowpea is a diploid with a chromosome number $2n = 22$ and an estimated genome size of 620 Mb. We sequenced the cowpea genome using single-molecule real-time sequencing (Pacific Biosciences RSII). A total of 87 SMRT cells yielded about 6M reads for a total of 56.84 Gbp (91.7x genome equivalent). We tested the three scaffolding tool on a high-quality assembly produced by Canu [7, 42] with parameters `corMhapSensitivity=high` and `corOutCoverage=100`, then polished it with Quiver. We used Chimericognizer

to detect and break chimeric contigs, using seven other assemblies generated by Canu, Falcon [14] and ABruijn [46] as explained in [61].

In addition to standard contiguity statistics (N50[1], L50[2]), total assembled size and scaffold length distribution, we determined incorrect/chimeric scaffolds by comparing them against the high-density genetic map available from [54]. We BLASTed 121bp-long design sequence for the 51,128 genome-wide SNPs described in [54] against each assembly, then we identified which contigs had SNPs mapped to them, and what linkage group (chromosome) of the genetic map those mapped SNPs belonged to. Chimeric contigs were revealed when their mapped SNPs belonged to more than one linkage group. The last line of Table 4.1 and Table 4.2 report the total size of contigs in each assembly for which (i) they have at least one SNPs mapped to it and (ii) all SNPs belong to the same linkage group (i.e., likely to be non-chimeric).

As said, the three scaffolding tools were run on a chimera-free assembly of cowpea described above using two available Bionano Genomics optical maps (the first obtained using the BspQI nicking enzyme, and the second obtained with the BssSI nicking enzyme). Since SewingMachine can only use a single optical map, we alternated the optical maps in input (BspQI map first, then BssSI and vice versa). SewingMachine provides two outputs depending on the minimum allowed alignment confidence, namely 'default' and 'relax'. Mode 'relax' considers more alignments than 'default', but it has a higher chance of introducing mis-joins. HybridScaffold failed on the BssSI map, so we could not test it on alternating maps.

[1] length for which the set of contigs/scaffolds of that length or longer accounts for at least half of the assembly size

[2] minimum number of contigs/scaffolds accounting for at least half of the assembly

Table 4.1 shows that when using a single optical map, OMGS can generate comparable or better scaffolds than SewingMachine and HybridScaffold. With two optical maps, OMGS' correctness ("contigs/scaffolds with 100% consistent LG") and contiguity (N50) are significantly better than other two tools. Observe that OMGS' correctness ("contigs/scaffolds with 100% consistent LG") is even better than the input assembly. This can happen when contigs with SNPs belonging to same linkage group are scaffolded with contigs that have no SNP.

We also compared the performance of OMGS, SewingMachine and HybridScaffold when using optical maps corrected by Chimericognizer (on the same cowpea assembly). Observe in Table 4.2 that OMGS, SewingMachine and HybridScaffold increased the correctness but decreased the contiguity when the corrected BspQI optical map was used. The results on the corrected BssSI optical map or both corrected optical maps did not change significantly. But again, OMGS produced better scaffolds than SewingMachine and HybridScaffold.

4.3.2 Experimental results on *D. melanogaster* assemblies

D. melanogaster has four pairs of chromosomes: three autosomes, and one pair of sex chromosomes. The fruit fly's genome is about 139.5 Mb. We downloaded three *D. melanogaster* assemblies generated in [80] (`https://github.com/danrdanny/Nanopore_ISO1`). The first assembly (295 contigs, total size 141 Mb, N50 = 3 Mb) was generated using Canu [7, 42] on Oxford Nanopore (ONT) reads longer than 1kb. The second assembly (208 contigs, total size 132 Mb, N50 = 3.9 Mb) was generated using MiniMap and

		One optical map							
		BspQI				BssSI			
	Input	SM (default)	SM (relax)	HS	OMGS	SM (default)	SM (relax)	HS	OMGS
contig/scaffold N50 (bp)	5,633,882	13,154,336	13,154,336	12,211,658	**14,339,314**	10,620,326	10,886,079	N/A	**11,536,649**
contig/scaffold L50	28	15	15	17	14	18	17	N/A	15
total assembled (bp)	511,101,122	521,209,608	521,210,640	516,455,893	518,265,608	518,987,660	518,945,404	N/A	518,252,638
# contigs/scaffolds	948	863	863	877	847	849	846	N/A	832
# contigs/scaffolds ≥100kbp	269	185	185	198	170	177	174	N/A	165
# contigs/scaffolds ≥1Mbp	94	59	59	63	56	63	62	N/A	59
# contigs/scaffolds ≥10Mbp	10	20	20	21	20	18	18	N/A	17
contigs/scaffolds with consistent LG (bp)	425,812,490	404,408,642	404,409,674	381,974,417	**410,552,582**	**425,572,265**	425,530,009	N/A	424,143,108

		Two optical maps				
		BspQI+BssSI	BspQI+BssSI	BssSI+BspQI	BssSI+BspQI	BspQI&BssSI
	Input	SM (default)	SM (relax)	SM (default)	SM (relax)	OMGS
contig/scaffold N50 (bp)	5,633,882	14,892,230	14,892,230	13,527,997	14,892,235	**16,364,046**
contig/scaffold L50	28	13	13	14	13	12
total assembled (bp)	511,101,122	525,577,823	525,198,231	525,827,900	525,105,345	521,324,385
# contigs/scaffolds	948	822	823	816	814	802
# contigs/scaffolds ≥100kbp	269	149	150	145	143	137
# contigs/scaffolds ≥1Mbp	94	46	46	48	46	44
# contigs/scaffolds ≥10Mbp	10	21	21	22	22	21
contigs/scaffolds with consistent LG (bp)	425,812,490	385,449,577	385,069,985	425,678,421	403,637,207	**432,639,234**

Table 4.1: Comparing OMGS, SewingMachine (SM) and HybridScaffold (HS) on a cowpea assembly using one or two optical maps. Numbers in boldface highlight the best N50 and scaffold consistency with the genetic map for one map (BspQI and BssSI) or two maps ('A+B' refers to the use of map A followed by map B, 'A&B' refers to the use of both maps at the same time).

MiniAsm [43] using only ONT reads. The third assembly (339 contigs, total size 134 Mb, N50 = 10 Mb) was generated by Platanus [39] and Dbg2Olc [94] using 67.4x of Illumina paired-end reads and the longest 30x ONT reads. The first and third assemblies were polished using nanopolish [48] and Pilon [90]. The Bionano optical for *D. melanogaster* map was provided by the authors of [80]. This BspQI optical map (363 molecules, total size = 246 Mb, N50 = 841 kb) was created using IrysSolve 2.1 from 78,397 raw Bionano molecules (19.9 Gb of data with a mean read length 253 kb).

As said, all tools were run with default parameters, with the exception of OMGS' minimum confidence, which was set at 20 (default is 15). To evaluate the performance of OMGS, HybridScaffold and SewingMachine, we compared their output scaffolds to the high-quality reference genome of *D. melanogaster* (release 6.21, downloaded from FlyBase). We reported the total length of correct/non-chimeric scaffolds as a measure of the

		One optical map							
		BspQI				BssSI			
	Input	SM (default)	SM (relax)	HS	OMGS	SM (default)	SM (relax)	HS	OMGS
contig/scaffold N50 (bp)	5,633,882	12,487,373	12,487,373	12,495,655	**13,505,314**	9,420,899	10,886,079	N/A	**11,256,770**
contig/scaffold L50	28	16	16	15	14	19	17	N/A	16
total assembled (bp)	511,101,122	519,785,777	519,785,777	515,519,585	518,405,022	517,678,278	517,636,022	N/A	517,318,151
# contigs/scaffolds	948	863	863	871	849	854	851	N/A	837
# contigs/scaffolds ≥100kbp	269	185	185	192	172	182	179	N/A	169
# contigs/scaffolds ≥1Mbp	94	60	60	60	58	66	65	N/A	62
# contigs/scaffolds ≥10Mbp	10	19	19	19	19	17	17	N/A	17
contigs/scaffolds with consistent LG (bp)	425,812,490	413,819,557	413,819,557	402,840,302	**421,466,164**	**424,262,883**	424,220,627	N/A	423,117,331

		Two optical maps				
		BspQI+BssSI	BspQI+BssSI	BssSI+BspQI	BssSI+BspQI	BspQI&BssSI
	Input	SM (default)	SM (relax)	SM (default)	SM (relax)	OMGS
contig/scaffold N50 (bp)	5,633,882	14,354,752	14,354,752	13,527,997	14,892,235	**16,364,046**
contig/scaffold L50	28	14	14	14	13	12
total assembled (bp)	511,101,122	523,520,329	523,139,705	521,540,185	525,105,345	520,697,623
# contigs/scaffolds	948	823	824	817	814	805
# contigs/scaffolds ≥100kbp	269	150	151	146	143	139
# contigs/scaffolds ≥1Mbp	94	48	48	48	46	46
# contigs/scaffolds ≥10Mbp	10	21	21	21	22	21
contigs/scaffolds with consistent LG (bp)	425,812,490	402,344,751	401,964,127	420,269,616	403,637,207	**431,921,182**

Table 4.2: Comparing OMGS, SewingMachine (SM) and HybridScaffold (HS) on a cowpea assembly using optical maps corrected by Chimericognizer. Numbers in boldface highlight the best N50 and scaffold consistency with the genetic map for one map (BspQI and BssSI) or two maps ('A+B' refers to the use of map A followed by map B, 'A&B' refers to the use of both maps at the same time).

overall correctness. To determine which scaffolds were incorrect/chimeric we first selected BLAST alignments of the scaffolds against the reference genome which had an e-value lower than 1e-50 and an alignment length higher than 30 kbp. We defined a scaffold S to be *chimeric* if S had at least two high-quality alignments which satisfied one or more of the following conditions: (i) S aligned to different chromosomes; (ii) the orientation of S's alignments were different; or (iii) the difference between the distance of alignments on the scaffold and the distance of alignments on the reference sequence was larger than 100 Kbp.

Table 4.3 reports the main statistics for the three *D. melanogaster* scaffolded assemblies. Even with one map, OMGS' scaffolds are better than SewingMachine and HybridScaffold.

	MiniAsm assembly				
	Input	SM (default)	SM (relax)	HS	OMGS
contig/scaffold N50 (bp)	3,866,686	4,494,241	**4,906,224**	3,866,686	**4,906,224**
contig/scaffold L50	9	8	8	9	8
total assembled (bp)	131,856,353	132,480,826	133,233,999	132,138,056	132,838,677
# contigs/scaffolds	208	205	203	206	206
# contigs/scaffolds ≥100kbp	85	82	80	83	83
# contigs/scaffolds ≥1Mbp	26	26	25	26	25
# contigs/scaffolds ≥10Mbp	2	2	2	2	2
non-chimeric contigs/scaffolds (bp)	131,317,873	125,305,638	**132,695,519**	131,174,201	132,300,197

	Canu assembly				
	Input	SM (default)	SM (relax)	HS	OMGS
contig/scaffold N50 (bp)	3,004,953	3,004,953	3,004,953	3,918,649	**5,336,340**
contig/scaffold L50	11	11	11	10	7
total assembled (bp)	140,720,404	140,923,974	140,923,974	140,867,226	140,960,395
# contigs/scaffolds	295	291	291	286	280
# contigs/scaffolds ≥100kbp	111	107	107	102	96
# contigs/scaffolds ≥1Mbp	31	31	31	29	27
# contigs/scaffolds ≥10Mbp	1	1	1	1	5
non-chimeric contigs/scaffolds (bp)	140,720,404	140,923,974	140,923,974	140,867,226	**140,960,395**

	Dbg2Olc assembly				
	Input	SM (default)	SM (relax)	HS	OMGS
contig/scaffold N50 (bp)	10,113,899	11,223,142	11,223,142	12,785,467	**12,928,771**
contig/scaffold L50	6	5	5	5	4
total assembled (bp)	134,109,164	134,164,629	134,164,629	134,162,857	134,208,377
# contigs/scaffolds	339	337	337	331	327
# contigs/scaffolds ≥100kbp	78	76	76	70	66
# contigs/scaffolds ≥1Mbp	22	22	22	17	16
# contigs/scaffolds ≥10Mbp	6	6	6	5	7
non-chimeric contigs/scaffolds (bp)	134,109,164	134,164,629	134,164,629	134,162,857	**134,208,377**

Table 4.3: Comparing OMGS, SewingMachine (SM) and HybridScaffold (HS) on three *D. melanogaster* assemblies (produced by MiniAsm, Canu, and Dbg2Olc) using the BspQI optical map. Numbers in boldface highlight the best N50 and the best scaffold consistency with the reference genome

4.4 Conclusions

In this chapter, we presented a scaffolding tool called OMGS for improving the contiguity of *de novo* genome assembly using one or multiple optical maps. OMGS solves several optimization problems to generate scaffolds with optimal contiguity and correctness. Experimental results on *V. unguiculata* and *D. melanogaster* clearly demonstrate that OMGS outperforms SewingMachine and HybridScaffold both in contiguity and correctness using multiple optical maps.

Chapter 5

Conclusions

In this dissertation, we addressed some of the computational issues associated with genome scaffolding, with the help of optical maps. We proposed an algorithm for scaffolding (OMGS), and two pre-processing algorithms aimed at either breaking chimeric contigs (Chimericognizer), or stitching overlapping contigs for multiple assemblies (Novo&Stitch).

OMGS is the first tool that can take advantages of multiple optical maps at same time to carry out scaffolding. Novo&Stitch is the first tool that can take advantage of optical maps to accurately carry out assembly reconciliation. Chimericognizer is significantly more accurate than the chimeric detection method offered by the Bionano Hybrid Scaffold pipeline, and is the first tool which can correct optical maps.

Here, we list some research problems we plan to study in the future. First, we will improve Chimericognizer by considering the conflicts between optical maps and between assemblies. In the current version of Chimericognizer, we only solve the conflicts between

optical map and assembly. Generally speaking, after *in silico*-digesting, the assemblies can been seen as same as optical maps. So it's reasonable to ignore the difference between optical maps and assemblies and compare all pairs of them when detecting mis-joins. We believe more comparisons will make detection more accurate, but more time consuming. There more efficient algorithm will be needed.

Second, we will try to improve genome scaffolding by taking advantage of different types of maps such as optical maps, genetic maps, Hi-C data and long reads at same time. In current scaffolding pipeline, people apply each of them to assembly DNA sequences or correct mis-joins alternately or iteratively. We believe that applying all the information at same time in an optical way has a change to improve both the contiguity and correctness of assembly.

Last but not least, we will study gap filling which is a post-processing step of scaffolding. We plan to create a novel algorithm for gap filling scaffolds by long reads with the help of optical maps.

5.1 Publications

This dissertation includes three peer-reviewed publications. The findings on NOVO&STITCH was presented at Conference on Intelligent Systems for Molecular Biology (ISMB) 2018, Chicago, IL and published in *Bioinformatics*. The work on OMGS was presented at Conference on Research in Computational Molecular Biology (RECOMB) 2019, Washington DC and will be published in the *Journal of Computational Biology*. CHIMERICOGNIZER was published in *Bioinformatics*.

Full list of publications by W. Pan:

1. **W. Pan**, T. Jiang, S. Lonardi. "OMGS: Optical Map-based Genome Scaffolding." *Proceedings of Conference on Research in Computational Molecular Biology (RECOMB)*, pp. 190-207, Washington, DC, 2019.

2. **W. Pan**, S. Lonardi. "Accurate detection of chimeric contigs via Bionano optical maps." *Bioinformatics*, vol. 35, no. 10, pp. 1760-1762, 2018.

3. **W. Pan**, S. Wanamaker, A. Ah-Fong, H. Judelson, S. Lonardi. "Novo&Stitch: Accurate Reconciliation of Genome Assemblies via Optical Maps." *Proceedings of Conference on Intelligent Systems for Molecular Biology (ISMB)*, Chicago, IL, 2018. *Bioinformatics*, vol. 34, no. 13, pp. i43-i51, 2018.

4. C. Schwartz, J.F. Cheng, R. Evans, C.A. Schwartz, J.M. Wagner, S. Anglin, A. Beitz, **W. Pan**, S. Lonardi, M. Blenner, H.S. Alper. "Validating genome-wide CRISPR-Cas9 function improves screening in the oleaginous yeast Yarrowia lipolytica." *Metabolic Engineering*, vol. 55, pp. 102-110, 2018.

5. A. Polishko, M. A. Hasan, **W. Pan**, E. Bunnik, K. L. Roch, S. Lonardi. "ThIEF: Finding Genome-wide Trajectories of Epigenetics Marks." *Proceedings of the Workshop on Algorithms in Bioinformatics (WABI)*, 19:1-19:16, Boston, MA, 2017.

Bibliography

[1] Hind Alhakami, Hamid Mirebrahim, and Stefano Lonardi. A comparative evaluation of genome assembly reconciliation tools. *Genome biology*, 18(1):93, 2017.

[2] Juan Lucas Argueso, Marcelo F Carazzolle, Piotr A Mieczkowski, Fabiana M Duarte, Osmar VC Netto, Silvia K Missawa, Felipe Galzerani, Gustavo GL Costa, Ramon O Vidal, Melline F Noronha, et al. Genome structure of a saccharomyces cerevisiae strain widely used in bioethanol production. *Genome research*, 19(12):2258–2270, 2009.

[3] Mordecai Avriel. *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

[4] Ali Baharev, Hermann Schichl, Arnold Neumaier, and TOBIAS Achterberg. An exact method for the minimum feedback arc set problem. *University of Vienna*, 10:35–60, 2015.

[5] S Batzoglou. Algorithmic challenges in mammalian genome sequence assembly. *Encyclopedia of genomics, proteomics and bioinformatics*, 2005.

[6] Serafim Batzoglou, David B Jaffe, Ken Stanley, Jonathan Butler, Sante Gnerre, Evan Mauceli, Bonnie Berger, Jill P Mesirov, and Eric S Lander. Arachne: a whole-genome shotgun assembler. *Genome research*, 12(1):177–189, 2002.

[7] 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.

[8] Derek M Bickhart, Benjamin D Rosen, Sergey Koren, Brian L Sayre, Alex R Hastie, Saki Chan, Joyce Lee, Ernest T Lam, Ivan Liachko, Shawn T Sullivan, et al. Single-molecule sequencing and chromatin conformation capture enable de novo reference assembly of the domestic goat genome. *Nature genetics*, 49(4):643, 2017.

[9] Marten Boetzer, Christiaan V Henkel, Hans J Jansen, Derek Butler, and Walter Pirovano. Scaffolding pre-assembled contigs using sspace. *Bioinformatics*, 27(4):578–579, 2010.

[10] Joshua N Burton, Andrew Adey, Rupali P Patwardhan, Ruolan Qiu, Jacob O Kitzman, and Jay Shendure. Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nature biotechnology*, 31(12):1119, 2013.

[11] Matt J Cahill, Claudio U Köser, Nicholas E Ross, and John AC Archer. Read length and repeat resolution: exploring prokaryote genomes using next-generation sequencing technologies. *PloS one*, 5(7):e11518, 2010.

[12] Jean Cardinal, Marek Karpinski, Richard Schmied, and Claus Viehmann. Approximating vertex cover in dense hypergraphs. *Journal of discrete algorithms*, 13:67–77, 2012.

[13] Todd A Castoe, AP Jason de Koning, Kathryn T Hall, Daren C Card, Drew R Schield, Matthew K Fujita, Robert P Ruggiero, Jack F Degner, Juan M Daza, Wanjun Gu, et al. The burmese python genome reveals the molecular basis for extreme adaptation in snakes. *Proceedings of the National Academy of Sciences*, 110(51):20645–20650, 2013.

[14] Chen-Shan Chin, Paul Peluso, Fritz J Sedlazeck, Maria Nattestad, Gregory T Concepcion, Alicia Clum, Christopher Dunn, Ronan O'Malley, Rosa Figueroa-Balderas, Abraham Morales-Cruz, et al. Phased diploid genome assembly with single-molecule real-time sequencing. *Nature methods*, 13(12):1050–1054, 2016.

[15] James Clarke, Hai-Chen Wu, Lakmal Jayasinghe, Alpesh Patel, Stuart Reid, and Hagan Bayley. Continuous base identification for single-molecule nanopore DNA sequencing. *Nature nanotechnology*, 4(4):265–270, 2009.

[16] Kenneth L Clarkson. A modification of the greedy algorithm for vertex cover. *Inf. Process. Lett.*, 16(1):23–25, January 1983.

[17] International Human Genome Sequencing Consortium et al. Initial sequencing and analysis of the human genome. *nature*, 409(6822):860, 2001.

[18] Nicolas Daccord, Jean-Marc Celton, Gareth Linsmith, Claude Becker, Nathalie Choisne, Elio Schijlen, Henri van de Geest, Luca Bianco, Diego Micheletti, Riccardo Velasco, et al. High-quality de novo assembly of the apple genome and methylome dynamics of early fruit development. *Nature genetics*, 49(7):1099, 2017.

[19] Adel Dayarian, Todd P Michael, and Anirvan M Sengupta. Sopra: Scaffolding algorithm for paired reads via statistical optimization. *BMC bioinformatics*, 11(1):345, 2010.

[20] 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.

[21] Anders Dessmark, Jesper Jansson, Andrzej Lingas, Eva-Marta Lundell, and Mia Persson. On the approximability of maximum and minimum edge clique partition problems. *International Journal of Foundations of Computer Science*, 18(02):217–226, 2007.

[22] Nilgun Donmez and Michael Brudno. Scarpa: scaffolding reads with practical algorithms. *Bioinformatics*, 29(4):428–434, 2012.

[23] Emilie Dordet-Frisoni, Eveline Sagné, Eric Baranowski, Marc Breton, Laurent Xavier Nouvel, Alain Blanchard, Marc Serge Marenda, Florence Tardy, Pascal Sirand-Pugnet, and Christine Citti. Chromosomal transfers in mycoplasmas: when minimal genomes go mobile. *MBio*, 5(6):e01958–14, 2014.

[24] Alexander W Eastman, Brian Weselowski, Naeem Nathoo, and Ze-Chun Yuan. Complete genome sequence of paenibacillus polymyxa cr1, a plant growth-promoting bacterium isolated from the corn rhizosphere exhibiting potential for biocontrol, biomass degradation, and biofuel production. *Genome Announc.*, 2(1):e01218–13, 2014.

[25] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, et al. Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 2009.

[26] Song Gao, Niranjan Nagarajan, and Wing-Kin Sung. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. In *International Conference on Research in Computational Molecular Biology*, pages 437–451. Springer, 2011.

[27] Annalisa Giampetruzzi, Michela Chiumenti, Maria Saponari, Giacinto Donvito, Alessandro Italiano, Giuliana Loconsole, Donato Boscia, Corrado Cariddi, Giovanni Paolo Martelli, and Pasquale Saldarelli. Draft genome sequence of the xylella fastidiosa codiro strain. *Genome Announc.*, 3(1):e01538–14, 2015.

[28] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, 17(6):333–351, May 2016.

[29] Alexey A Gritsenko, Jurgen F Nijkamp, Marcel JT Reinders, and Dick de Ridder. Grass: a generic algorithm for scaffolding next-generation sequencing assemblies. *Bioinformatics*, 28(11):1429–1437, 2012.

[30] David Hernandez, Patrice François, Laurent Farinelli, Magne Østerås, and Jacques Schrenzel. De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. *Genome research*, 18(5):802–809, 2008.

[31] Xiaoqiu Huang, Jianmin Wang, Srinivas Aluru, Shiaw-Pyng Yang, and LaDeana Hillier. Pcap: a whole-genome assembly program. *Genome research*, 13(9):2164–2170, 2003.

[32] Xiaoqiu Huang and Shiaw-Pyng Yang. Generating a genome assembly with pcap. *Current protocols in bioinformatics*, 11(1):11–3, 2005.

[33] Martin Hunt, Chris Newbold, Matthew Berriman, and Thomas D. Otto. A comprehensive evaluation of assembly scaffolding tools. *Genome Biology*, 15(3):R42, Mar 2014.

[34] Ramana M Idury and Michael S Waterman. A new algorithm for DNA sequence assembly. *Journal of computational biology*, 2(2):291–306, 1995.

[35] David B Jaffe, Jonathan Butler, Sante Gnerre, Evan Mauceli, Kerstin Lindblad-Toh, Jill P Mesirov, Michael C Zody, and Eric S Lander. Whole-genome sequence assembly for mammalian genomes: Arachne 2. *Genome research*, 13(1):91–96, 2003.

[36] Miten Jain, Hugh E Olsen, Benedict Paten, and Mark Akeson. The oxford nanopore minion: delivery of nanopore sequencing to the genomics community. *Genome biology*, 17(1):239, 2016.

[37] Young-Min Jeong, Won-Hyung Chung, Jeong-Hwan Mun, Namshin Kim, and Hee-Ju Yu. De novo assembly and characterization of the complete chloroplast genome of radish (raphanus sativus l.). *Gene*, 551(1):39–48, 2014.

[38] Wen-Biao Jiao, Gonzalo Garcia Accinelli, Benjamin Hartwig, Christiane Kiefer, David Baker, Edouard Severing, Eva-Maria Willing, Mathieu Piednoel, Stefan Woetzel, Eva Madrid-Herrero, Bruno Huettel, Ulrike Hümann, Richard Reinhard, Marcus A Koch, Daniel Swan, Bernardo Clavijo, George Coupland, and Korbinian Schneeberger. Improving and correcting the contiguity of long-read genome assemblies of three plant species using optical mapping and chromosome conformation capture data. *Genome Res.*, 27(5):778–786, May 2017.

[39] Rei Kajitani, Kouta Toshimoto, Hideki Noguchi, Atsushi Toyoda, Yoshitoshi Ogura, Miki Okuno, Mitsuru Yabana, Masayuki Harada, Eiji Nagayasu, Haruhiko Maruyama, et al. Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads. *Genome research*, 24(8):1384–1395, 2014.

[40] Mikhail Kolmogorov, Brian Raney, Benedict Paten, and Son Pham. Ragouta reference-assisted assembly tool for bacterial genomes. *Bioinformatics*, 30(12):i302–i309, 2014.

[41] Sergey Koren, Todd J Treangen, and Mihai Pop. Bambus 2: scaffolding metagenomes. *Bioinformatics*, 27(21):2964–2971, 2011.

[42] Sergey Koren, Brian P Walenz, Konstantin Berlin, Jason R Miller, Nicholas H Bergman, and Adam M Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome research*, 27(5):722–736, 2017.

[43] Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 2016.

[44] Ruiqiang Li, Hongmei Zhu, Jue Ruan, Wubin Qian, Xiaodong Fang, Zhongbin Shi, Yingrui Li, Shengting Li, Gao Shan, Karsten Kristiansen, et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome research*, 20(2):265–272, 2010.

[45] Shin-Hung Lin and Yu-Chieh Liao. Cisa: contig integrator for sequence assembly of bacterial genomes. *PloS one*, 8(3):e60843, 2013.

[46] Yu Lin, Jeffrey Yuan, Mikhail Kolmogorov, Max W Shen, Mark Chaisson, and Pavel A Pevzner. Assembly of long error-prone reads using de bruijn graphs. *Proceedings of the National Academy of Sciences*, 113(52):E8396–E8405, 2016.

[47] Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. Comparison of next-generation sequencing systems. *BioMed Research International*, 2012, 2012.

[48] Nicholas J Loman, Joshua Quick, and Jared T Simpson. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature methods*, 12(8):733, 2015.

[49] Ruibang Luo, Binghang Liu, Yinlong Xie, Zhenyu Li, Weihua Huang, Jianying Yuan, Guangzhu He, Yanxiang Chen, Qi Pan, Yunjie Liu, et al. Soapdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience*, 1(1):18, 2012.

[50] Martin Mascher, Heidrun Gundlach, Axel Himmelbach, Sebastian Beier, Sven O Twardziok, Thomas Wicker, Volodymyr Radchuk, Christoph Dockter, Pete E Hedley, Joanne Russell, et al. A chromosome conformation capture ordered sequence of the barley genome. *Nature*, 544(7651):427, 2017.

[51] Luz Mayela Soto-Jimenez, Karel Estrada, and Alejandro Sanchez-Flores. Garm: genome assembly, reconciliation and merging pipeline. *Current topics in medicinal chemistry*, 14(3):418–424, 2014.

[52] Camilo Mora, Derek P Tittensor, Sina Adl, Alastair G B Simpson, and Boris Worm. How Many Species Are There on Earth and in the Ocean? *PLoS Biology*, 2011.

[53] Martin D. Muggli et al. Misassembly detection using paired-end sequence reads and optical mapping data. *Bioinformatics*, 31(12):i80–i88, 2015.

[54] María Muñoz-Amatriaín, Hamid Mirebrahim, Pei Xu, Steve I Wanamaker, MingCheng Luo, Hind Alhakami, Matthew Alpert, Ibrahim Atokple, Benoit J Batieno, Ousmane Boukar, et al. Genome resources for climate-resilient cowpea, an essential crop for food security. *The Plant Journal*, 89(5):1042–1054, 2017.

[55] David J Munroe and Timothy JR Harris. Third-generation sequencing fireworks at marco island. *Nature biotechnology*, 28(5):426, 2010.

[56] Eugene W Myers. The fragment assembly string graph. *Bioinformatics*, 21(suppl_2):ii79–ii85, 2005.

[57] Eugene W Myers, Granger G Sutton, Art L Delcher, Ian M Dew, Dan P Fasulo, Michael J Flanigan, Saul A Kravitz, Clark M Mobarry, Knut HJ Reinert, Karin A Remington, et al. A whole-genome assembly of drosophila. *Science*, 287(5461):2196–2204, 2000.

[58] Niranjan Nagarajan, Timothy D Read, and Mihai Pop. Scaffolding and validation of bacterial genome assemblies using optical restriction maps. *Bioinformatics*, 24(10):1229–1235, 2008.

[59] Giuseppe Narzisi and Bud Mishra. Comparing de novo genome assembly: the long and short of it. *PloS one*, 6(4):e19175, 2011.

[60] Daniel Nathans and Hamilton O Smith. Restriction endonucleases in the analysis and restructuring of dna molecules. *Annual review of biochemistry*, 44(1):273–293, 1975.

[61] Weihua Pan and Stefano Lonardi. Accurate detection of chimeric contigs via bionano optical maps. *Bioinformatics*, 2018.

[62] Weihua Pan, Steve I Wanamaker, Audrey MV Ah-Fong, Howard S Judelson, and Stefano Lonardi. Novo&stitch: accurate reconciliation of genome assemblies via optical maps. *Bioinformatics*, 34(13):i43–i51, 2018.

[63] Matthew Pendleton, Robert Sebra, Andy Wing Chun Pang, Ajay Ummat, Oscar Franzen, Tobias Rausch, Adrian M Stütz, William Stedman, Thomas Anantharaman, Alex Hastie, et al. Assembly and diploid architecture of an individual human genome via single-molecule technologies. *Nature methods*, 12(8):780, 2015.

[64] Yu Peng, Henry CM Leung, Siu-Ming Yiu, and Francis YL Chin. Idba–a practical iterative de bruijn graph de novo assembler. In *Annual international conference on research in computational molecular biology*, pages 426–440. Springer, 2010.

[65] 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.

[66] Mihai Pop, Daniel S Kosack, and Steven L Salzberg. Hierarchical scaffolding with bambus. *Genome research*, 14(1):149–159, 2004.

[67] Richard J Roberts, Mauricio O Carneiro, and Michael C Schatz. The advantages of smrt sequencing. *Genome biology*, 14(6):405, 2013.

[68] Nicole Rusk. Cheap third-generation sequencing. *Nature Methods*, 6(4):244, 2009.

[69] Subrata Saha and Sanguthevar Rajasekaran. Efficient and scalable scaffolding using optical restriction maps. *BMC genomics*, 15(5):S5, 2014.

[70] Leena Salmela, Veli Mäkinen, Niko Välimäki, Johannes Ylinen, and Esko Ukkonen. Fast scaffolding with small independent mixed integer programs. *Bioinformatics*, 27(23):3259–3265, 2011.

[71] Akhtar Samad, EF Huff, Weiwen Cai, and David C Schwartz. Optical mapping: a novel, single-molecule approach to genomic analysis. *Genome research*, 5(1):1–4, 1995.

[72] Eric E Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human molecular genetics*, 19(R2):R227–R240, 2010.

[73] Manfred Schartl, Ronald B Walter, Yingjia Shen, Tzintzuni Garcia, Julian Catchen, Angel Amores, Ingo Braasch, Domitille Chalopin, Jean-Nicolas Volff, Klaus-Peter Lesch, et al. The genome of the platyfish, xiphophorus maculatus, provides insights into evolutionary adaptation and several complex traits. *Nature genetics*, 45(5):567, 2013.

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

[75] Fritz J Sedlazeck, Hayan Lee, Charlotte A Darby, and Michael C Schatz. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics*, 19(6):329, 2018.

[76] Jennifer M Shelton, Michelle C Coleman, Nic Herndon, Nanyan Lu, Ernest T Lam, Thomas Anantharaman, Palak Sheth, and Susan J Brown. Tools and pipelines for bionano data: molecule assembly pipeline and fasta super scaffolding tool. *BMC genomics*, 16(1):734, 2015.

[77] Jennifer M Shelton, Michelle C Coleman, Nic Herndon, Nanyan Lu, Ernest T Lam, Thomas Anantharaman, Palak Sheth, and Susan J Brown. Tools and pipelines for BioNano data: molecule assembly pipeline and FASTA super scaffolding tool. *BMC Genomics*, 16:734, September 2015.

[78] Jared T Simpson and Richard Durbin. Efficient de novo assembly of large genomes using compressed data structures. *Genome research*, 22(3):549–556, 2012.

[79] Jared T Simpson, Kim Wong, Shaun D Jackman, Jacqueline E Schein, Steven JM Jones, and Inanç Birol. Abyss: a parallel assembler for short read sequence data. *Genome research*, 19(6):1117–1123, 2009.

[80] Edwin A. Solares et al. Rapid low-cost assembly of the drosophila melanogaster reference genome using low-coverage, long-read sequencing. *G3: Genes, Genomes, Genetics*, 2018.

[81] Daniel D Sommer, Arthur L Delcher, Steven L Salzberg, and Mihai Pop. Minimus: a fast, lightweight genome assembler. *BMC bioinformatics*, 8(1):64, 2007.

[82] Hayssam Soueidan, Florence Maurier, Alexis Groppi, Pascal Sirand-Pugnet, Florence Tardy, Christine Citti, Virginie Dupuy, and Macha Nikolski. Finishing bacterial genome assemblies with mix. *BMC bioinformatics*, 14(15):S16, 2013.

[83] Granger Sutton and Ian Dew. Shotgun fragment assembly. *Systems Biology: Volume I: Genomics: Volume I: Genomics*, page 79, 2006.

[84] Haibao Tang, Eric Lyons, and Christopher D Town. Optical mapping in plant comparative genomics. *Gigascience*, 4(1):3, February 2015.

[85] Haibao Tang, Xingtan Zhang, Chenyong Miao, Jisen Zhang, Ray Ming, James C Schnable, Patrick S Schnable, Eric Lyons, and Jianguo Lu. Allmaps: robust scaffold ordering based on multiple maps. *Genome biology*, 16(1):3, 2015.

[86] Erwin L Van Dijk, Hélène Auger, Yan Jaszczyszyn, and Claude Thermes. Ten years of next-generation sequencing technology. *Trends in genetics*, 30(9):418–426, 2014.

[87] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

[88] Francesco Vezzi, Federica Cattonaro, and Alberto Policriti. e-rga: enhanced reference guided assembly of complex genomes. *EMBnet. journal*, 17(1):46–54, 2011.

[89] Riccardo Vicedomini, Francesco Vezzi, Simone Scalabrin, Lars Arvestad, and Alberto Policriti. Gam-ngs: genomic assemblies merger for next generation sequencing. *BMC bioinformatics*, 14(7):S6, 2013.

[90] Bruce J Walker, Thomas Abeel, Terrance Shea, Margaret Priest, Amr Abouelliel, Sharadha Sakthikumar, Christina A Cuomo, Qiandong Zeng, Jennifer Wortman, Sarah K Young, et al. Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PloS one*, 9(11):e112963, 2014.

[91] René L Warren, Granger G Sutton, Steven JM Jones, and Robert A Holt. Assembling millions of short dna sequences using ssake. *Bioinformatics*, 23(4):500–501, 2006.

[92] Alejandro Hernandez Wences and Michael C Schatz. Metassembler: merging and optimizing de novo genome assemblies. *Genome biology*, 16(1):207, 2015.

[93] Guohui Yao, Liang Ye, Hongyu Gao, Patrick Minx, Wesley C Warren, and George M Weinstock. Graph accordance of next-generation sequence assemblies. *Bioinformatics*, 28(1):13–16, 2011.

[94] Chengxi Ye, Christopher M Hill, Shigang Wu, Jue Ruan, and Zhanshan Sam Ma. Dbg2olc: efficient assembly of large genomes using long erroneous reads of the third generation sequencing technologies. *Scientific reports*, 6:31900, 2016.

[95] Yuxuan Yuan et al. BioNanoAnalyst: a visualisation tool to assess genome assembly quality using BioNano data. *BMC Bioinformatics*, 18(1):323, June 2017.

[96] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821–829, 2008.

[97] Daniel R Zerbino, Gayle K McEwen, Elliott H Margulies, and Ewan Birney. Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. *PloS one*, 4(12):e8407, 2009.

[98] Wenyu Zhang, Jiajia Chen, Yang Yang, Yifei Tang, Jing Shang, and Bairong Shen. A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies. *PloS one*, 6(3):e17915, 2011.

[99] Jie Zheng and S. Lonardi. Discovery of repetitive patterns in dna with accurate boundaries. In *Fifth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'05)*, pages 105–112, Oct 2005.

[100] Aleksey V Zimin, Douglas R Smith, Granger Sutton, and James A Yorke. Assembly reconciliation. *Bioinformatics*, 24(1):42–45, 2007.