



UNIVERSITA' DEGLI STUDI DI VERONA

DEPARTMENT OF

Biotechnology

GRADUATE SCHOOL OF

Natural Sciences and Engineering

DOCTORAL PROGRAM IN

Biotechnology

XXXII cycle

TITLE OF THE DOCTORAL THESIS

Bioinformatics approaches for hybrid de novo genome assembly

S.S.D. BIO/18




Coordinator: Prof. Matteo Ballottari

Tutor: Prof. Massimo Delledonne

Doctoral Student: Luca Marcolungo

Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione – non commerciale
Non opere derivate 3.0 Italia. Per leggere una copia della licenza visita il sito web:

<http://creativecommons.org/licenses/by-nc-nd/3.0/it/>

-  **Attribuzione** Devi riconoscere [una menzione di paternità adeguata](#), fornire un link alla licenza e [indicare se sono state effettuate delle modifiche](#). Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.
-  **NonCommerciale** Non puoi usare il materiale per [scopi commerciali](#).
-  **Non opere derivate** —Se [remixi, trasformi il materiale o ti basi su di esso](#), non puoi distribuire il materiale così modificato.

Bioinformatics approaches for hybrid de novo genome assembly
Luca Marcolungo

Tesi di Dottorato
Verona, 10 Dicembre 2019

Abstract

De novo genome assembly, the computational process to reconstruct the genomic sequence from scratch stitching together overlapping reads, plays a key role in computational biology and, to date, it cannot be considered a solved problem. Many bioinformatics approaches are available to deal with different type of data generated by diverse technologies. Assemblies relying on short read data resulted to be highly fragmented, reconstructing short contigs interrupted in repetitive region; on the other side long-read based approaches still suffer of high sequencing error rate, worsening the final consensus quality. This thesis aimed to assess the impact of different assembly approaches on the reconstruction of a highly repetitive genome, identifying the strengths and limiting the weaknesses of such approaches through the integration of orthogonal data types. Moreover, a benchmarking study has been undertaken to improve the contiguity of this genome, describing the improvements obtained thanks to the integration of additional data layers.

Assemblies performed using short reads confirmed the limitation in the reconstruction of long sequences for both the software adopted. The use of long reads allowed to improve the genome assembly contiguity, reconstructing also a greater number of gene models. Despite the enhancement of contiguity, base level accuracy of long reads-based assembly could still not reach higher levels. Therefore, short reads were integrated within the assembly process to limit the base level errors present in the reconstructed sequences up to 96%. To order and orient the assembled polished contigs into longer scaffolds, data derived from three different technologies (linked read, chromosome conformation capture and optical mapping) have been analysed. The best contiguity metrics were obtained using chromosome conformation data, which permit to obtain chromosome-scale scaffolds. To evaluate the obtained results, data derived from linked reads and optical mapping have been used to identify putative misassemblies in the scaffolds. Both the datasets allowed the identification of misassemblies, highlighting the importance of integrating data derived from orthogonal technologies in the de novo assembly process.

This work underlines the importance of adopting bioinformatics approaches able to deal with data type generated by different technologies. In this way, results could be more accurately validated for the reconstruction of assemblies that could be eventually considered reference genomes.

Summary

Abstract	3
Introduction.....	7
Algorithms for de novo Genome assembly.....	10
Limitation de novo genome assembly approaches	12
Long range information to enhance genome assembly.....	14
Limitation of long-range scaffolding approaches	16
Case study.....	17
Aim of the thesis.....	17
Materials and Methods	18
Public repository.....	18
Haematococcus pluvialis sample	18
Basecalling, reads processing and error correction.....	18
De novo genome assembly	20
PacBio reads.....	21
Assembly polishing.....	25
Identification of base-level errors.....	26
Merging long read-based assemblies.....	26
Assembly scaffolding and anchoring.....	27
Assessment of assembly quality and identification of misassembly.....	27
Results.....	28
De novo genome assembly of short reads data (Illumina)	28
De novo genome assembly of Long reads data (PacBio).....	31
De novo genome assembly of Long reads data (ONT)	35
Impact of reads length in de novo assembly	39
Genome assembly polishing	41
Merging long reads-based assemblies	43
Benchmarking of bioinformatics approaches for genome scaffolding and anchoring .	45
Assessing Chromosome Conformation capture technology result	49
Discussion	51
References	56

Introduction

Genome assembly is the process to retrieve the original genomic sequence given a bunch of sequenced DNA fragments (reads). Since the advent of sequencing technique *de novo* assembly represented a key step aiming to generate the genomic sequence of an organism which is fundamental for downstream analysis like comparative genomics, variant discovery or genome editing. *De novo* genome assembly is usually compared to jigsaw puzzle in which each piece is represented by a read that need to be placed in the original position of the picture identifying similarities with proximities pieces. From bioinformatic point of view, assembly procedure is performed by computational tools that find overlaps between reads reconstructing longer contiguous sequence (contigs). Then, using complementary long-range information, contigs can be ordered and oriented into longer, gapped sequences (scaffolds) (Figure 1). Genome assembly approaches changed during years adapting algorithms to the evolving sequencing technologies peculiarities. Despite the persistent improvement of sequencing technologies, genome assembly is still an unsolved problem and represent a unavoidable step due to the incapacity of reading entire chromosome sequence at once. Thus, to obtain the DNA sequence, it is necessary to sample many copies of the genome, fragment it randomly and sequence it. Generated reads need to be assembled using bioinformatics algorithms that have been developed exploiting different approaches.

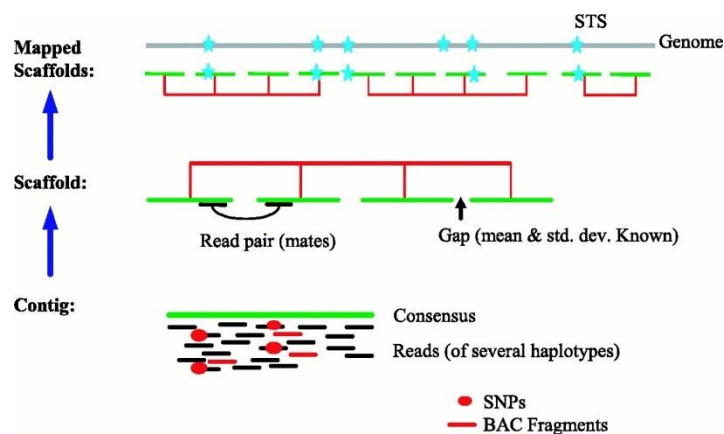


Figure 1. *Schematization of de novo assemble approach.* This image is from Venter et al. *The Sequence of the Human Genome.* Science. 2001

Initially DNA sequencing was performed in a single-end mode (SE) that is read a genomic DNA fragment from one side. Assembling reads generated using this approach, presented many limitation: the information is limited to the sequenced regions and read orientation is not known [1]. In DNA sequence assembly, longer distance information helps the correct positioning of the reads into longer contigs. To address this issue, in 1997 was introduced the pair-end sequencing mode by Weber and Myers [2] in which the sampled DNA fragment is sequenced from both the extremities. The first enhancement that this approach presented is the possibility to reads twice number of reads from the same number of DNA fragments, then, strictly related to de novo assembly, the paired information that can be used in the reconstruction process is extended to the whole fragment. Indeed, even if the middle portion is not sequenced, the information that two reads originated from the same fragment can be exploited during the reconstruction step to bypass complicated regions (Figure 2).

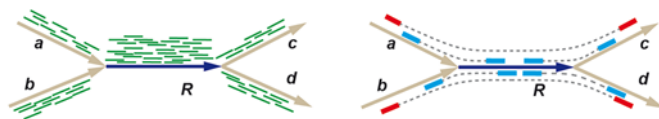


Figure 2. Single-end sequencing vs. Pair-end mode. Blue and red reads have been sequenced using a pair-end approach. This permit to identify unambiguously the thought the repetitive region R. This image is from Sohn and Nam. The present and future of de novo whole-genome assembly. Briefings in Bioinformatics. 2019

A boosting in range information was introduced by the so-called third generation sequencing technologies (Single Molecule Real Time and Nanopore sequencing) that have the ability to read continuous DNA sequences up to megabases in length [3]. The enhancement introduced in genome assembly community was remarkable. Long reads permit to read the complete sequence of repetitive regions reaching the non-repetitive boundaries. The non-repetitive sequence at the extremities resulted in a more reliable alignment during the assembly, incrementing the contigs size.

Comparing the statistics of the first human genome assembly with a recently published genome, is it possible to highlight how assembly process improved in

less than twenty years. Contigs N50 of the first human genome assembly was reported to be 82 Kbp [4], while, recently published human genomes, based on long reads, exceed easily megabases in length [5]. Interestingly, the reads N50 assembled in the latter work is greater than 100 Kb, highlighting how genome assembly approaches need to adapt to the evolving sequencing data.

Despite the improvement highlighted above, genome assembly suffer of the sequencing errors present in the reads worsening the final consensus accuracy. Moreover, low complexity regions and heterozygosity can lead to misassembled regions. Genome assembly is still an active research area aiming to find bioinformatic approaches to mitigate all the problematic arising from genome sequencing.

Algorithms for de novo Genome assembly

Broadly speaking, assembly methods rely on the generation of a graph representing the connection of shorter portion of DNA. Identification of unambiguous path through this graph led to the generation of contigs sequences. De novo assembly algorithms can be grouped in two major categories: Overlap layout consensus (OLC) and de Bruijn graphs (DBG).

Overlap Layout Consensus algorithm (OLC)

Overlap Layout consensus algorithm (OLC) has been extensively applied in computational biology and, as suggested by the name, it is composed by three steps (Figure 3).

In the **overlap** stage, whole dataset of reads is aligned in an all-vs-all manner identifying overlaps that exceed length threshold. Reads are placed as nodes and overlap creates links between nodes. During this procedure, the main challenge is represented by identifying the correct alignment: sequencing errors, repetitive genomic regions and heterozygosity can lead to false alignments that impact negatively in the next phases, generating misassembly or false connection in the graph. Then, on the bases of alignments, an overlap graph is constructed.

Once generated the overlap graph, the **layout** phase is performed to walk through the graph finding unambiguous paths and to convert them into contigs. Erroneous overlaps detected in the previous would create false connection that impact negatively in the layout procedure.

Finally, a **consensus** step is performed identifying a consensus between all aligned reads and generating contigs.

One of the first developed software implementing OLC algorithm is TIGR ASSEMBLER [6] employed in the firsts genome assemblies free-living organisms: *H.influenza* [7] and *M.genitalium* [8].

Subsequently, others OLC-based software have been developed, one of the most known is the WGA ASSEMBLER used for the assembly of the *D.melanogaster* genome and the first human genome [9][10].

OLC approach resulted to be successfully applied to Sanger sequencing which have a low throughput, generating reads of about 1000 bp in length. With the advent of second sequencing technologies the instrument throughput exponentially increased with a reduction of read length. These factors made OLC approach inadequate for short read data.

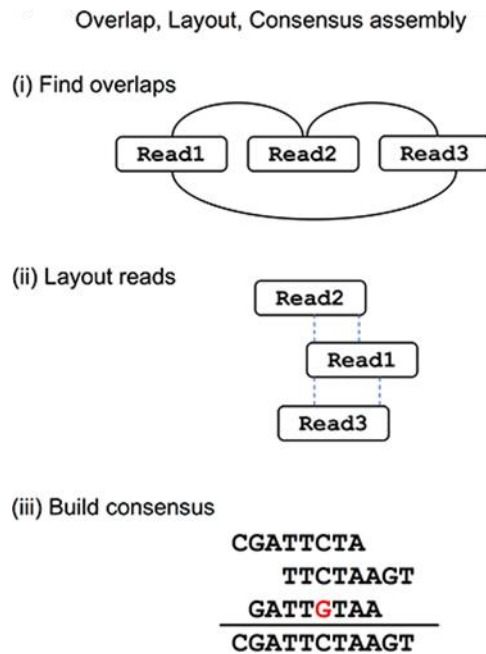


Figure 3. **Overlap Layout Consensus (OLC) algorithm.** This image is from Ayling et al. *New approaches for metagenome assembly with short reads*. *Briefings in Bioinformatics*, 2019.

De Bruijn Graph algorithm (DBG)

This model have been proposed for the first time applied to genome assembly in 2001 by Pevzner et al [11]. This approach was developed adapting to second generation data (NGS), a technique that were developing in those years. NGS produced short reads (e.g. 30-300 bp) with high accuracy. To use DBG in genome assembly was introduced a counterintuitive approach: short reads are further split into shorter pieces of length k, called kmers. Each kmer represent a node in a directed graph and two nodes are connected if share k-1 sequence (Figure 4). After generating a graph using all the reads in a dataset, DBG algorithms try to reconstruct contigs finding unambiguous path in the graph.

Limitation de novo genome assembly approaches

Since in the overlap step an all-vs-all alignment is performed, this algorithm is sensitive to the input reads number. The graph size increases linearly with the number of input sequences, and, consequently, the number of connections between nodes increase by a logarithmic scale. During the Layout stage, finding the best path through the graph is a Hamiltonian Path Problem which is NP-complete. Hence, OLC methods are designed to deal with sequencing technologies producing a low throughput. Moreover, decreased sequencing length did not permit to identify reliable overlap between reads. OLC algorithms have been, thus, highly exploited during the sanger sequencing era but were set aside when NGS hit the market. Subsequently, with the introduction of long reads technologies, OLC approaches return to be extensively used and many OLC-based software like Falcon [12], Canu [13] Flye[14], Wtdbg2 [15] and Shasta [16] have been developed. Third generation sequencing, indeed, generated long reads (up to hundreds of kilobases) permitting to take full advantages of overlap-based algorithms. On the other side, long reads contain a higher error rate (8 -15%), mostly due to small insertions or deletions (Indels) that may hamper a correct alignment.

To mitigate this issue, different approaches have been proposed:

- Introduction of a preliminary error correction (or preassembly) step in which input reads are corrected using a consensus approach before to proceed with the assembly. From one side, this permit to increase overlap identity between the reads improving overlap graph generation and the final genome assembly. This approach has been implemented in widely used software like Falcon and Canu and, despite its effectiveness, it is computationally slow.
- Introduction of a final consensus step to refine the assembled sequences. The improvement of long reads sequence quality permitted to avoid the preliminary, computationally expensive correction step in favour of a final consensus stage. A final consensus step can be implemented with the assembler, like Flye, or can be performed independently after the assembly.

Despite these measures, genome assemblies based on third generation sequencing still contains many errors that may hamper subsequent biotechnological experiment [17].

As mentioned for OLC approaches, during DBG generation sequencing errors can affect negatively the final assembly quality. Sequencing errors generate many false kmers that would produce erroneous nodes in the graph. To avoid this, as mentioned for OLC approaches, a preliminary error correction step is performed: sequenced reads are scanned in search of low frequency kmers, generating from sequencing errors. In an ideal scenario in which sequencing reads do not contain errors and all the kmers present in the genome are unique, the number on nodes in the assembly graph would be the same that in genome. Therefore, on the contrary to OLC approaches, sequencing depth do not impact on the graph size. Since the above assumptions could not be true, in practice, assembly graph generated by DBG approach is much higher than the genome graph and, thus, need to be properly cleaned. Reads errors can produce spurious branch or alternative paths in the graph and, thus, after generating the graph, DBG algorithms for de novo assembly perform the so-called tip pruning in which not connected, low supported branches are removed. Then, alternative path in the graph arising from sequencing errors commonly called “bubbles” are removed relying on the number of kmers supporting each path

Long range information to enhance genome assembly

Since the first genome assembly projects started, methods to get long range information have been pursued performing the so-called scaffolding procedure able to overcome problematic regions that generate breaks in the contigs. For instance, during the assembly of the first human genome mate-pair information was considered to join nearby contigs into scaffolds [10]. Indeed, sequencing a DNA fragments from both the extremities extend the information that can be converted into sequence favouring *de novo* assembly [18]. Even if the middle portion contains a repetitive region, unique sequenced extremities can be used to join the non-repetitive adjacent regions. Modern approaches to exploit long range information rely on newly developed sequencing and mapping technologies like linked reads, optical mapping and Chromosome conformation capture (Hi-C). Recently, different software has been implemented to exploit this information in favour of *de novo* assembly.

Linked reads

Proximity information carried by the barcode contained in the linked fragments can be exploited in a *de novo* genome assembly in three different way.

- 1) Linked reads can be independently assembled. Relying on an Illumina sequencing, short accurate reads can be assembled using a DBG approach. Subsequently, barcode information can be exploited to simplify the assembly graph and to join contigs through ambiguous assembly path.
- 2) Scaffolding. In this scenario, linked reads are aligned to the draft assembly and those contigs on which align reads labelled with the same barcode are identified and joined together.
- 3) Validation and correction of *de novo* genome assembly. Once aligned on a genome assembly, linked reads can be considered as a virtual long read spanning the first to the last fragment containing the same barcode. With this approach is it possible to calculate physical coverage of the assembly. Those assembled regions where physical decreases under a confidence

threshold (e.g. there aren't enough DNA fragment spanning that region) are considered as misassembled and contigs (or scaffold) can be broken.

Chromosomes conformation capture

Hi-C has been applied in genome scaffolding since it has the capability to catch all DNA interaction between genomic loci into the nucleolus without a limit in the linear distance which can vary between few kbp up to hundreds of megabases. Once aligned to the draft assembly, contigs are grouped considering shared pairs read. Hi-C does not measure distance between loci, but their frequency of interaction and thus, contigs sharing more interaction are placed nearby into scaffolds. Recently, many papers have been published highlighting the potentiality of this technology to obtain chromosome-scale scaffolds [19]–[22].

Genome mapping data

Genome mapping relying on the analysis of High Molecular Weight (HMW) DNA. Assembled genome maps can be used in a *de novo* assembly project to anchor the draft contigs to the maps. Since the analysed DNA fragments are in order of megabases in length it is easy to understand the advantages that this technology can guarantee in a *de novo* assembly project. Coming back to the jigsaw puzzle, this orthogonal technology represents a framework on which anchor the information deriving from NGS technologies. Being a sequencing free technology, bias arising from sequencing (PCR artefact, extreme GC content) are mitigated. Moreover, the long information range of the technology permit to span repetitive sequences improving the reconstruction of genomic regions that otherwise would be obscured.

Limitation of long-range scaffolding approaches

Linked reads data, as for Hi-C, rely on short reads sequencing and, thus, are affected by the limitation of Illumina sequencing. Moreover, these two approaches include PCR based library preparation. On the contrary of Hi-C, linked reads have a shorter information range deriving from library preparation. This limitation could limit the impact of the scaffolding procedure. Moreover, some limitation arises due to short read alignment and the difficulty to accurately estimate interaction frequency lead to contigs inversions and misassembly [23]. Indeed, when Hi-C fragments are mapped to the draft genome, the two pairs are aligned independently one each other and, thus, short reads alignment limitation are evident, especially in repetitive regions.

Mapping approach relying on the generation of a restriction site map of the genome. Due to a technological limitation, restriction site could not be too close and, on average, one site every 10 kb is recognised. This poses a limitation on the size of the contigs that can be properly anchored. Indeed, shorter contigs wouldn't have enough sites to be aligned against the map resulting not anchored. Therefore, to take full advantages of mapping assembly contiguity metrics of the input assembly need to be adequate

Case study

Haematococcus pluvialis is a unicellular green alga of relevant interest for industrial application, indeed it is currently the main organism cultivated for astaxanthin production, one of the most powerful natural antioxidants. However biotechnological approaches to improve this species are restricted by the absence of genomic information. Reference genome for this species will allow genome editing studies and potential targets for biotechnological manipulation to improve biomass and then astaxanthin production in *Haematococcus pluvialis*.

Aim of the thesis

Many bioinformatic algorithms for genome reconstruction, adapting to the evolving sequencing technologies, are in continuous development. This thesis aims to assess the impact of different assembly approaches on the reconstruction of *H.pluvialis* genome, identifying strengths and trying to limit the weaknesses through the integration of orthogonal technologies. Moreover, a benchmark study has been performed to evaluate the impact of different long-range scaffolding data for a de novo genome assembly

Materials and Methods

Public repository

To ensure readability and data analysis reproducibility, all scripts, commands and configuration files used in this work are publicly available at web repository <https://bitbucket.org/liukvr/luca-marcolungo-phd-thesis/src/master/>

Haematococcus pluvialis sample

Haematococcus pluvialis sample, strain K-0084, analysed in this thesis was kindly supplied by Sole Lab directed by Prof. Ballottari at the Department of Biotechnology, University of Verona.

Basecalling, reads processing and error correction

Pacific Bioscience reads processing

Raw sequencing files generated by PacBio RSII instrument (bas.h5) were converted to *fasta* files using `bash5tool.py` script from `pbh5tools` toolkit (<https://github.com/PacificBiosciences/pbh5tools>) using the following command line.

```
$ bash5tools.py --outFilePrefix output_name --readType subreads --outType fasta
```

Oxford Nanopore Technology basecalling and reads filtering

Raw signal files (*fast5* format) were converted into *fastq* file using Guppy v3.1.5 with accurate basecalling model. Subsequently, sequencing adapters were trimmed using Porechop v0.2.4 (<https://github.com/rrwick/Porechop>) and fastq file converted into fasta file using seqtk v1.3(<https://github.com/lh3/seqtk>).

```
$ guppy_basecaller --cpu_threads_per_caller threads_number -c guppy_config_file -i fast5_folder --hp_correct TRUE -s fastq --num_callers 1
```

```
$ porechop ont_reads.fastq -o ont-reads_trimmed.fastq --format fastq --verbosity 3 --threads threads_number
```

```
$ seqtk seq -A ont-reads_trimmed.fastq > ont-reads_trimmed.fasta
```

Error correction of Illumina reads

Illumina reads were searched for sequencing errors and corrected using BayesHammer module of SPAdes toolkit (<https://github.com/ablab/spades>) with the following command.

```
$ spades.py -o output_folder --tmp-dir temp_folder --only-error-correction -t 16 -m 256 --pe1-1 illumina_R1.fastq.gz --pe1-2 illumina_R2.fastq.gz
```

De novo genome assembly

Illumina reads

Filtered reads have been assembled using SPAdes v3.11.1 [24] and SOAPdenovo v2.04 [25] using different kmer length range from 41 to 107. The best result in terms of contiguity was obtained using kmer 77 for SOAPdenovo and exploiting a multi-kmer approach for SPAdes with kmer length 95, 99, 103, 107.

```
#SPAdes v3.11.1
$ spades.py --dataset spades_config.yaml -t 25 -m 400 --only-assembler -k
95,99,103,107 --tmp-dir temp_dir -o Hp_illumina

#SOAPdenovo v2.04
$ SOAPdenovo-127mer all -o K77 -p 20 -K 77 -F -V -R -s
config_file_soapdenovo2.config
```

PacBio reads

PacBio subreads were assembled using Canu v1.5 [13], Falcon v1.4.4 [12], Flye v2.5 [14] and Wtdbg2 v2.5 [15] using following command lines. Configuration file provided to Falcon pipeline is reported.

```
#Canu v1.5
$ canu -p H.pluvialis -d H.pluvialis_PacBio genomeSize=300m -pacbio-raw
subreads.fasta

#Falcon v1.4.4
$ fc_run file.cfg

#Flye v2.5
$ flye --pacbio-raw subreads.fasta -g 300m --out-dir output_dir -t threads_number -i
1

#Wtdbg2 v2.5
$ wtdbg2 -i subreads.fasta -o HP.wtdbg2 -t 30 -L5000
$ wtpoa-cns -t 22 -i HP.wtdbg2.ctg.lay -fo HP.wtdbg2.fasta
```

```

#Falcon pipeline configuration file

input_type = preads

length_cutoff = 4000

# The length cutoff used for overlapping the preassembled reads
length_cutoff_pr = 4000

### resource usage ###
# grid settings for...
# daligner step of raw reads
jobqueue = production
sge_option_da = -pe smp 8 -q %(jobqueue)s
# las-merging of raw reads
sge_option_la = -pe smp 2 -q %(jobqueue)s
# consensus calling for preads
sge_option_cns = -pe smp 12 -q %(jobqueue)s
# daligner on preads
sge_option_pda = -pe smp 8 -q %(jobqueue)s
# las-merging on preads
sge_option_pla = -pe smp 24 -q %(jobqueue)s
# final overlap/assembly
sge_option_fc = -pe smp 24 -q %(jobqueue)s

# job concurrency settings for...
# all jobs
default_concurrent_jobs = 7
# preassembly
pa_concurrent_jobs = 7
# consensus calling of preads
cns_concurrent_jobs = 4
# overlap detection
ovlp_concurrent_jobs = 7

# daligner parameter options for...
# initial overlap of raw reads
pa_HPCdaligner_option = -v -B4 -M50 -e.70 -l1000 -s100
# overlap of preads
ovlp_HPCdaligner_option = -v -B4 -M50 -h60 -e.96 -l500 -s100

# parameters for creation of dazzler database of...
# raw reads
pa_DBsplit_option = -x500 -s400 -a
# preads
ovlp_DBsplit_option = -x500 -s400

# settings for consensus calling for preads
falcon_sense_option = --output_multi --min_idt 0.70 --min_cov5 --max_n_read 200 --n_core 7

# setting for filtering of final overlap of preads
overlap_filtering_setting = --max_diff 100 --max_cov 100 --min_cov5 --bestn 10 --n_core 10

```

Oxford Nanopore technology reads

PacBio subreads were assembled using Canu v1.5, Shasta v0.1.0 [16], Flye v2.5 and Wtdbg2 v2.5 using following command lines.

```
#Canu v1.5
$ canu -p H.pluvialis -d H.pluvialis_ONT genomeSize=300m -nanopore-raw
nanopore_reads.fastq

#Flye v2.5
$ flye --nanopore-raw nanopore_reads.fasta -g 300m --out-dir out_dir -t
threads_number -i 1

#Shasta
$ shasta-Linux-0.1.0 --input nanopore_reads.fasta --Reads.minReadLength 0

#Wtdbg2 v2.5
$ wtdbg2 -i nanopore.fasta -o HP.wtdbg2 -t 30 -L 0
```

Optical map assembly

De novo genome map assembly was performed using Bionano Access v1.4.1 exploiting RefAligner v9232 (<https://bionanogenomics.com/support/software-downloads/>). Command line used for map assembly are reported below. Bionano configuration file was modified increasing the “MaxCoverage” parameter to 2500.

```
#BspQI genome map assembly

$ python pipelineCL.py -l output' -t RefAligner/1.0 -C \
clusterArgumentsBG_saphyr_phi.xml' \
-b Hp_Bionano_BssSI_RawMolecules.bnx \
-y -d -U -i 5 -F 1 -W 0.4 \
-a Bionano_configuration_file_map_assembly.xml \
-r reference.map \
-R 0.5 -f 0.2 -J 48 -j 60 -jp 240 -T 240 -N 6

#BssSI genome map assembly

$ python pipelineCL.py -l output -t RefAligner/1.0 -C \
clusterArgumentsBG_saphyr_phi.xml \
-b Hp_Bionano_BspQI_RawMolecules.bnx \
-y -d -U -i 5 -F 1 -W 0.4 \
-a Bionano_configuration_file_map_assembly.xml \
-r reference.map \
-R 0.5 -f 0.2 -J 48 -j 60 -jp 240 -T 240 -N 6
```


Assembly polishing

Long read based assemblies underwent to assembly polishing using both long reads and short reads correction. PacBio raw data were converted to bam file using bax2bam v0.0.8 (<https://github.com/PacificBiosciences/bax2bam>), aligned to Flye based assembly using pbmm2 v1.1.0 wrapper (<https://github.com/PacificBiosciences/pbmm2>). Then the based assembly was corrected using GenomicConsensus (<https://github.com/PacificBiosciences/GenomicConsensus>) using Quiver algorithm. ONT based assembly was correct using ONT reads. Firstly, ONT reads were aligned to the Flye based assembly using minimap2 v2.17 [26], then errors were corrected using two round of racon v1.4.3 (<https://github.com/isovic/racon>) and Medaka v0.8.1 (<https://nanoporetech.github.io/medaka/>). Raw assemblies along with long read polished assemblies were corrected using Illumina dataset: short reads were mapped to the assemblies using bwa v0.7.17, resulted alignment file was polished used by Pilon v1.23 [27] to correct base-level-errors.

```
#Polishing using PacBio dataset
$ bax2bam *.bax.h5
$ pbmm2 align Hp.pb.flye.fasta.mmi list_of_reads.fofn aligned.bam
$ quiver aligned.bam -r draft_genome.fasta -o quiver_polished.fasta -o

#Polishing using ONT dataset
$ minimap2 -a -L -t threads_number -x map-ont draft_genome.fasta ont_reads.fasta
> alignment.sam
$ racon -m 8 -x -6 -g -8 -w 500 -t 30 reads alignment.sam draft_genome.fasta 2>log1
> racon_polished.fasta
$ medaka_consensus -l ont_reads.fasta -d draft-assembly.fasta -o output_dir -t
threads_number

#Polishing using Illumina dataset
$ bwa mem -t threads_number reference.fasta read1.fastq.gz read2.fastq.gz |
samtools sort --threads 5 -o aligned.bam

$ java -Xmx160G -jar pilon.jar --genome draft_genome.fasta --frags alignment.bam --
output_pilon_polished --outdir Pilon_out --verbose --changes --vcf --vcfn --tracks --fix
```

Identification of base-level errors

To identify base-level errors still present in the assembly short reads data have been aligned to the sequence using BWA v0.7.17 [28]. The resulted BAM files were processed by local realignment around insertion–deletion sites, duplicate marking and recalibration using Genome Analysis Toolkit v4.0.2.1 [29]. Finally, variant calling was performed using GATK HaplotypeCaller v4.1.2.0.

```
$ bwa mem -t threads_number draf_assembly.fasta read1.fastq.gz read2.fastq.gz |  
samtools sort --threads 5 -o aligned.bam;
```

```
$ java -jar picard.jar MarkDuplicates VALIDATION_STRINGENCY=SILENT  
MAX_RECORDS_IN_RAM=4000000 INPUT=file.bam OUTPUT=alignment.rg.bam  
METRICS_FILE=duplicates.txt REMOVE_DUPLICATES=true CREATE_INDEX=true
```

```
$ java -jar gatk.jar -T IndelRealigner -R draf_assembly.fasta -l alignment.rg.bam -  
targetIntervals alignment.intervals -o alignment.realigned.bam
```

```
Java -jar gatk4.0.2.1.jar -R reference.fasta -T UnifiedGenotyper aligned.bam -o  
snps.raw.vcf -stand_call_conf 50.0 -dcov 200 -glm BOTH
```

Merging long read-based assemblies

Long reads-based assemblies were merged using QuickMerge v0.3 [30] . Briefly, the software aligns two genome assembly recovering those genomic regions reconstructed in only one. Moreover, contigs can be joined into scaffolds exploiting information of the other assembly.

```
$ nucmer -l 100 -prefix out self_assembly.fasta hybrid_assembly.fasta
```

```
$ delta-filter -r -q -l 10000 out.delta > out.rq.delta
```

```
$ Quickmerge -d alignment.delta -q ../pacbio.polished.fasta -r ../ont.polished.fasta  
-hco 5.0 -c 1.5 -l 0 -ml 5000 -p out
```

Assembly scaffolding and anchoring

Merged assembly has been scaffolded and anchored using 10X Genomics data, Bionano optical map and Hi-C data.

```
#Linked read data
$ scaff10x-nodes threads_number -longread 1 -gap 100 -matrix 2000 -reads 6 -link
4 -score 20 -edge 50000 -file 1 -plot output.png -block 50000 -data input.dat
draft_assembly.fasta 10x_scaffolded.fasta

#Hi-C
$ juicer.sh -g draft_genome -s Sau3AI -p draft_genome.chrSize -D juicer_script_dir

$ run-asm-pipeline.sh --editor-coarse-resolution 50000 --editor-coarse-region
100000 --editor-saturation-centile 40 --editor-repeat-coverage 10 -q 20 -r 5 --
editor-fine-resolution 1000 Hp.pbFlye.ontFlye.QM.fasta merged_nodups.txt

#Bionano optical map
$ Rscript runTGH.R -R RefAligner -b1 bionano_BssSI_assembly.cmap -b2
bionano_BspQI_assembly.cmap -N draft_assembly.fasta -e1 GCTCTTC -e2 CACGAG
-t cur_results.tar.gz -s status.txt -f
Bionano_configuration_file_hybridScaffold_two_enzymes.xml -O output
```

Assessment of assembly quality and identification of misassembly

Linked reads and optical mapping data have been exploited to identify misassembly in the Hi-C based assembly using break10X (<https://github.com/wtsi-hpag/Scaff10X>) v3.1 and RefAligner v9232 software respectively.

```
#Linked read data
$ break10x-nodes threads_number -gap 100 -reads 5 -score 20 -cover 50 -ratio 15 -
data input.dat hic_assembly.fasta corrected.fasta corrected

#Bionano optical map
$ Rscript runTGH.R -R RefAligner -b1 541_EXP_REFINEFINAL1.cmap -b2
543_EXP_REFINEFINAL1.cmap -N assembly.fasta -e1 CACGAG -e2 GCTCTTC -t
cur_results.tar.gz -s status.txt -f
Bionano_configuration_file_hybridScaffold_two_enzymes.xml - output
```

Results

De novo genome assembly of short reads data (Illumina)

Starting data

A genomic library has been sequenced on NovaSeq 6000 instrument using 150 Pair-End protocol generating 300 million sequencing fragments. Before to be assembled, reads underwent error-correction using SPAdes v3.11.1 BayesHammer module to limit the impact of false kmers (e.g. arising from sequencing errors) to assembly graph and the consequent genome assembly.

De novo genome assembly using de Bruijn Graph-based methods.

Corrected reads have been assembled using two different software implementing DBG algorithms: SPAdes v3.11.1 and SOAPdenovo v2.04. Different assemblies have been generated using a range of Kmer length and, for each software, the assembly showing the best contiguity metric is reported (Table 1). SOAPdenovo reconstructed 208 Mbp of sequence, 132 of which are contained in scaffolds with an N50 of 4 Kbp. On the other side, N50 value of SPAdes assembly couldn't be calculated because scaffolds length does not reach half of the total assembly size. This also reflect the ability of SPAdes to reconstruct contiguous sequence. Indeed, the number of gaps is 2,358 compared to the 34,474 of the SOAPdenovo assembly. Total contig number is 119,704 and 155,059 with an N50 value of 5,836 and 6,201 for SOAPdenovo and SPAdes assemblies respectively.

	SOAPdenovo Kmer 77		SPAdes Kmer 95, 99, 103, 107	
Total assembly length (Mbp)	208.0		219.7	
Total scaffolds length (Mbp)	132.1		17.5	
Number of scaffolds	28,927		2,168	
Scaffolds N50 (bp)	4,446			
Scaffolds average length (bp)	4,569		8,092	
Longest scaffold (bp)	79,691		70,878	
Number of Gaps	34,474		2,358	
Gaps size (bp)	623,511		39,975	
Contigs in scaffolds	56,114		4,430	
Remaining contigs	63,590		150,629	
Remaining contig total length (Mbp)	75.8		202.3	
Remaining contigs N50 (bp)	4,499		6,028	
Remaining contigs average length (bp)	1,192		1,342	
Longest remaining contig (bp)	79,453		79,543	
Total number of contigs	119,704		155,059	
Total contigs N50 (bp)	5,836		6,201	
Total contigs average length (bp)	1,723		1,417	
GC percentage	58.3%		59.7	
	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)
Sequences > 1 Kbp	32,150	184.2	34,111	180.2
Sequences > 5 Kbp	11,568	136.2	11,659	126.9
Sequences > 10 Kbp	5,215	91.0	4,757	78.2
Sequences > 30 Kbp	375	14.4	240	8.8

Table 1. Comparison of assembly generated using Illumina data. For each software, assembly statistics are reported. The rows show the total assembly length, the total scaffolds length, the scaffolds N50 (calculated considering the total assembly size), the scaffolds average length, the longest scaffold and the gaps metrics (Number of undefined bases and number of gaps). Subsequently is reported the number of ungapped sequences (remaining contigs), their total length, N50 value average length and the longest. Moreover, number of sequences (both scaffolds and contigs) and relative cumulative length are reported for sequences greater than 1, 5, 10, 30 Kbp.

Assembly completeness

BUSCO v4.0.6 has been used to assess assembly completeness using chlorophyte lineage specific single copy core gene set (chlorophyte_odb10). SOAPdenovo assembly contains 87.0% of the orthologue's genes present in the dataset, while SPAdes assembly contains 82.7% of those.

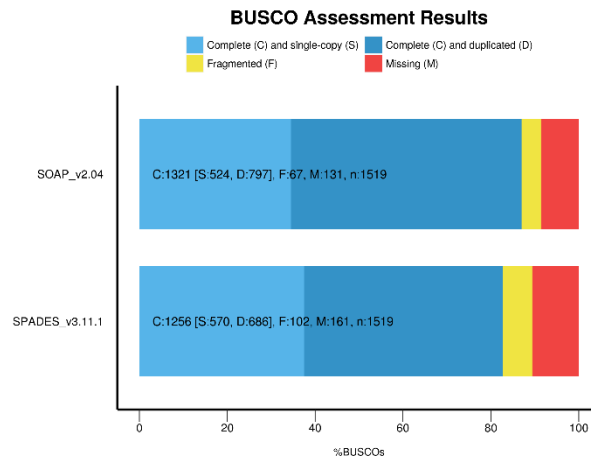


Figure 4: BUSCO completeness values on the Illumina-based assemblies

De novo genome assembly of Long reads data (PacBio)

Starting data

Long reads sequencing, generated with PacBio RSII instrument, comprises a total of 21 Gbp of data corresponding to about 70-fold coverage (Table 2). Reads N50 is 11.5 Kbp and the longest sequenced reads is 48 Kbp.

Instrument	PacBio RSII
Chemistry	P6-C4
Number of SMRTcells	21
Number of sequenced bases	21,816,661,710
Number of sequenced reads	2,828,818
Average reads length	7,712
Reads N50	11,507
Longest read	48,520
Expected fold coverage	70X

Table 2. *PacBio sequencing data statistics*. Table reports statistics of the PacBio subreads. Have been report the sequencer name, the sequencing chemistry used, the number of SMRTcells used. The total number of sequenced bases and sequenced reads, the average reads length, the N50 value of the reads, the longest sequenced read and the expected fold coverage considering an expected genome size of 300 Mbp.

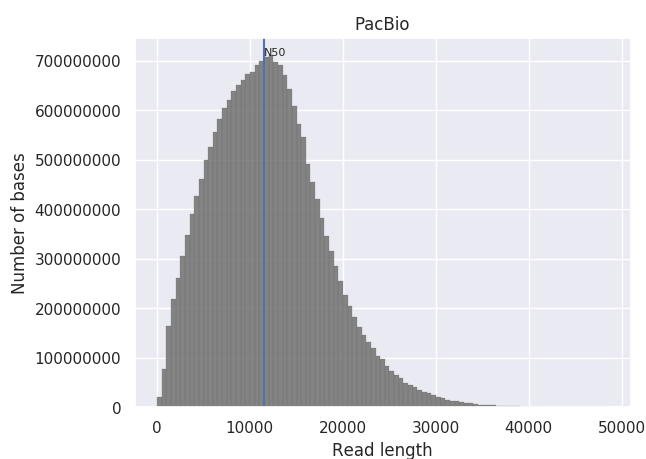


Figure 5: *PacBio Weighted reads length distribution*

Denovo assembly using OLC-based methods

Results obtained using Illumina data highlighted the limitation of short reads sequencing in a de novo assembly process. To overcome this limitation the sample have been sequenced using long reads (PacBio). DBG-based software assessed for short reads data cannot be successfully applied for long reads. Higher error rate profile would generate plenty of erroneous kmer hampering the graph building and the consequent genome assembly. On the contrary, OLC approach can handle errors presents in long reads allowing mismatch in the Overlap stage allowing to find overlap even though the identity is not 100%. Two well-known OLC-based software have been exploited to assemble long reads data: Falcon v1.4.4 and Canu v1.5. Moreover, two other newly developed assemblers, implementing a slightly different OLC approach, have been tested: Flye v2.5 using a repeat graph approach and Wtdbg2 v2.5 implementing Fuzzy De Bruijn Graph algorithm.

	Canu		Falcon		Flye		Wtdbg2	
Assembly length (Mbp)	266.2		240.3		272.5		241.6	
Number of contigs	5,192		6,676		6,176		5,493	
Average contigs length (Kbp)	51.2		36.0		44.1		43.9	
Longest contig (Kbp)	1,402.4		526.2		457.5		430.1	
Contigs N50 (Kbp)	67.0		81.5		86.5		81.1	
Contigs N90 (Kbp)	25.7		14.9		23.8		19.2	
GC percentage	59.6%		59.5%		60.1%		59.6%	
	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)
Contigs > 10 Kbp	5,065	265.3	4,454	227.9	4,416	264.9	4,383	233.9
Contigs > 50 Kbp	1,767	168.4	1,522	163.3	1,977	201.8	1,588	165.4
Contigs > 100 Kbp	542	82.8	621	99.4	753	114.7	624	96.5
Contigs > 300 Kbp	12	5.0	24	8.9	22	7.8	19	6.5

Table3. Comparison of assembly generated using PacBio data. For each software, assembly statistics are reported. The rows show the total assembly length, the number of reconstructed contigs, the average contigs length, the longest assembled contig, the N50 and N90 value of the assembly and the GC percentage. Moreover, number of contigs and relative contigs length are reported for contigs greater than 10, 50, 100, 300 Kbp.

Assembly completeness

To assess assembly completeness, we evaluated the four generated assemblies using BUSCO with the chlorophyte lineage specific single copy core gene set (chlorophyte_odb10). As for the contiguity, Flye assembly resulted the one with the higher completeness percentage (96.5%) of core genes, compared to the 88.6%, 93.9% and 80.5% of the assembly generate by Canu, Falcon and Wtdbg2 respectively.

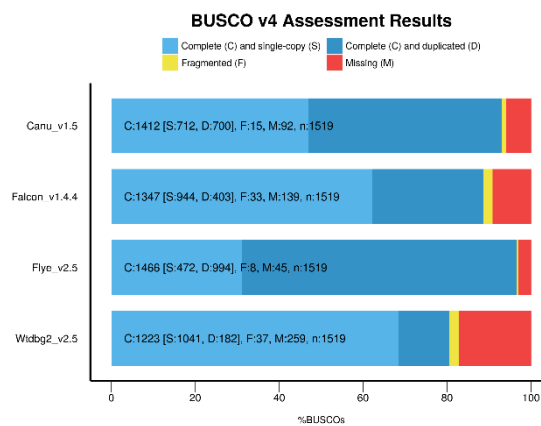


Figure 6. BUSCO completeness values on the PacBio-based assemblies

De novo genome assembly of Long reads data (ONT)

Starting data

Long reads sequencing, generated with Oxford Nanopore Technology MinION instrument, comprises a total of 12 Gbp of data corresponding to about 40-fold coverage (Table 4). Reads N50 is 15.8 Kbp and the longest sequenced reads is more than 400 Kbp.

Instrument	ONT MinION
Number of sequenced bases	11,940,386,371
Number of sequenced reads	2,061,059
Average reads length	7,712
Reads N50	15,852
Longer reads	430 Kbp
Expected fold coverage	41X

Table 4. *ONT sequencing data statistics*. Table reports statistics of the ONT reads. Have been report the sequencer name, the total number of sequenced bases and sequenced reads, the average reads length, the N50 value of the reads, the longest sequenced read and the expected fold coverage considering an expected genome size of 300 Mbp.

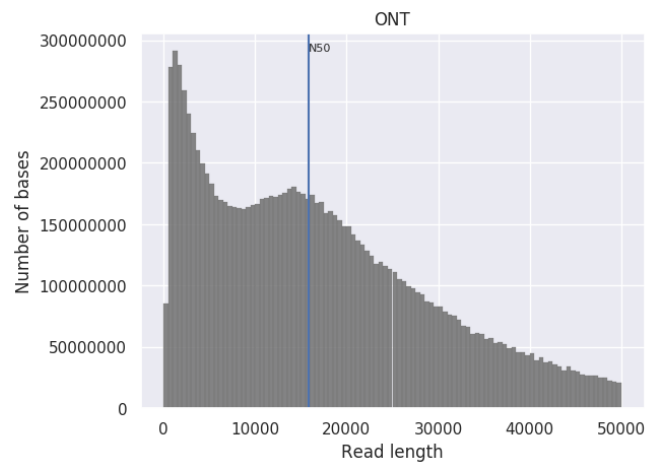


Figure 7: *ONT Weighted reads length distribution*

De novo genome assembly using OLC-based methods

Results obtained with PacBio data showed an improvement to the sort-read based assemblies. On the other hand, the contiguity metrics of the generated assembly still cannot be considered satisfactory. Considering the PacBio based assembly, short contigs couldn't be incorporated in a hybrid assembly using optical mapping data. Being a low-resolution technology, contig shorter than 60-90 Kbp cannot be properly anchored to the map. Thus, being only 50% of the Fly assembly sequence is contained in contigs longer than 86 Kbp, about 50% of the assembly would be anchored. To further increase contigs length, it was necessary to generate longer reads. Sequenced dataset is then assembled using four different OLC-based software. Falcon, which has been developed to work with PacBio reads has been replaced with Shasta v0.1.0 a brand-new assembler developed with the goal of assembling ONT reads.

	Canu		Flye		Wtdbg2		Shasta	
Assembly length (Mbp)	307.5		278.6		235.2		257.8	
Number of contigs	3,607		5,009		7,245		42,329	
Average contigs length (Kbp)	85.2		55.6		32.4		6.1	
Longest contig	1,533.3		1,189.2		911.9		848.3	
Contigs N50 (Kbp)	188.1		247.8		59.8		140.1	
Contigs N90 (Kbp)	39.5		56.1		12.8		5.6	
GC percentage	59.8%		60.3%		59.8%		59.2%	
	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)
Contigs > 10 Kbp	3,079	305.1	1,897	270.2	5,054	220.2	2,345	229.2
Contigs > 50 Kbp	1,604	266.1	1,270	253.7	1,034	129.1	1,411	204.7
Contigs > 100 Kbp	945	219.6	883	225.4	408	86.2	806	160.9
Contigs > 300 Kbp	194	89.7	238	111.5	68	29.8	106	43.0

Table 5. Comparison of assembly generated using ONT data. For each software, assembly statistics are reported. The rows show the total assembly length, the number of reconstructed contigs, the average contigs length, the longest assembled contig, the N50 and N90 value of the assembly and the GC percentage. Moreover, number of contigs and relative contigs length are reported for contigs greater than 10, 50, 100, 300 Kbp.

Generated assemblies have a genome size ranging from 235 Mbp (Wtdbg2) to 307 Mbp (Canu) (Table 5). Shasta reconstructed the greater number of contigs, most of them shorter than 10Kbp. Despite the assembly with the highest average contigs length is the Canu assembly (85 Kbp), if considering the N50 value Flye resulted to be the most contiguous assembly: 247 Kbp compared to the 188 Kbp, 140Kb and 59Kbp of Canu, Shasta and Wtdbg2 respectively. Moreover, considering N90 value, Flye assembly resulted the best assembly having reconstructing 90% of the sequence in contigs longer or equal to 56 Kbp.

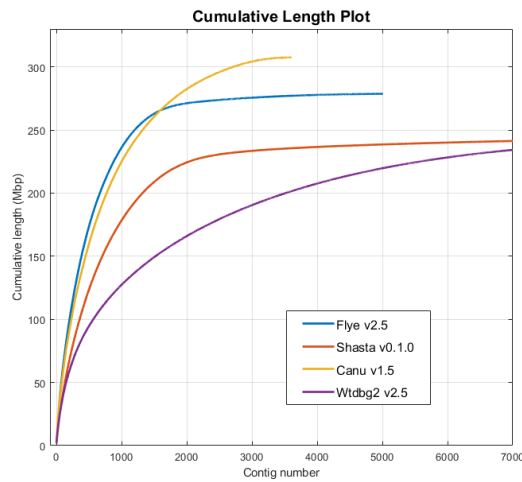


Figure 8. *ONT-based assemblies cumulative length plot*

Assembly completeness

As performed for PacBio-based assemblies the completeness has been assessed using BUSCO. Compared with the results obtained using PacBio data it is possible to notice that an overall decreasing of the reconstructed gene space. Flye assemblers confirm to be the most accurate reconstructing the 93.3% of the orthologous genes compared to the 88.2%, 86.7%, 62.9% of Shasta, Canu and Wtdbg2 respectively (Figure 10).

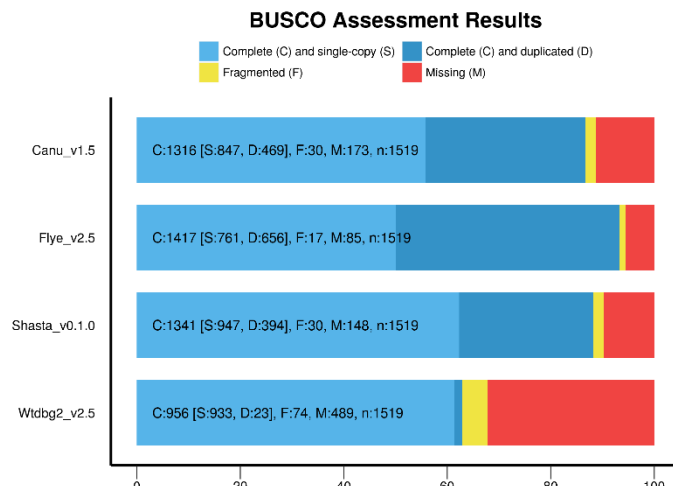


Figure 9. BUSCO completeness values on the ONT-based assemblies

Impact of reads length in de novo assembly

To assess the impact of reads length in de novo assembly process we downsampled the ONT dataset considering only those reads longer than: 5Kb (33-fold coverage), 10Kb (27-fold coverage) and 15Kb (21-fold coverage) (Figure 11).

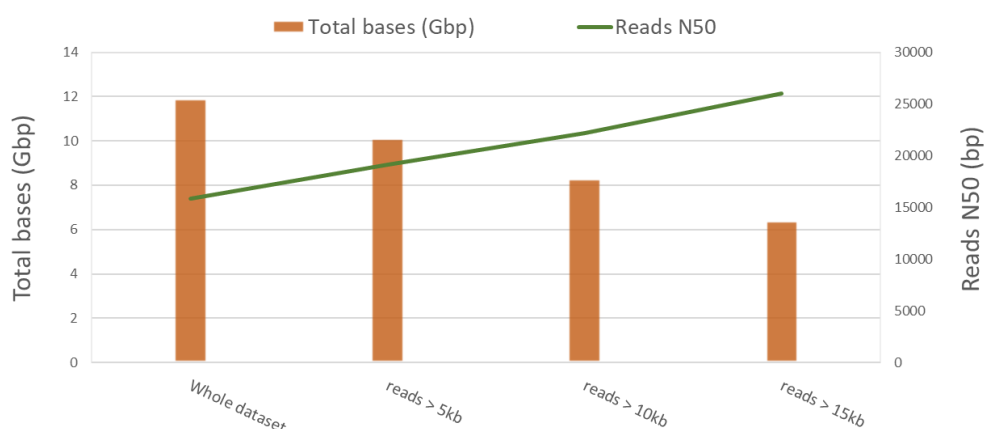


Figure 10. **ONT datasets.** Number of total bases along with reads N50 value is reported for the whole and filtered datasets

As expected, filtering out shorter reads, N50 value increase, reaching up 26 Kbp for the dataset comprising reads longer than 15 Kbp. Results show that assembly N50 increase when considering filtered datasets compared to the value obtained using the whole dataset. Despite the halved reads coverage, assembly generate using reads longer than 15 Kbp resulted having a greater N50 compared to the one generated with the whole dataset, value increased from 247 Kb to 272 Kb. Highest N50 value (278 Kb) has been obtained assembling dataset composed by reads longer than 10 Kb. The number of contigs reduced when considering filtered datasets: in all three cases the number of reconstructed contigs is about half of the number obtained using the whole dataset. This reduction is reflecting also in the average contigs reads length which is, in the filtered datasets, more than double of the original assembly (Table 6).

	Whole dataset	Reads > 5Kb	Reads > 10Kb	Reads > 15Kb				
Assembly length (Mbp)	278.6	311.9	316.8	316.9				
Number of contigs	5,009	2,578	2,634	2,469				
Average contigs length (Kbp)	55.6	121.0	12.3	128.3				
Longest contig (Kbp)	1,189.2	1,374.5	1,712.7	1,291.2				
Contigs N50 (Kbp)	247.8	271.9	278.3	272.6				
Contigs N90 (Kbp)	56.1	73.7	71.7	77.1				
GC percentage	60.3%	60.3%	60.3%	60.3%				
	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)
Contigs > 10 Kbp	1,897	270.2	1,905	309.3	1,964	314.2	1,877	314.7
Contigs > 50 Kbp	1,270	253.7	1,426	295.6	1,450	299.8	1,430	301.7
Contigs > 100 Kbp	883	225.4	975	263.1	986	266.5	989	269.2
Contigs > 300 Kbp	238	111.5	298	141.9	301	144.6	302	145.8

Table 6. Comparison of assembly generated using Flye assembler over different ONT filtered data. For each dataset, assembly statistics generated using Flye is reported. The rows show the total assembly length, the number of reconstructed contigs, the average contigs length, the longest assembled contig, the N50 and N90 value of the assembly and the GC percentage. Moreover, number of contigs and relative contigs length are reported for contigs greater than 10, 50, 100, 300 Kbp.

Genome assembly polishing

Long reads permit to span over repetitive regions allowing to resolve more low complex regions than short reads technologies. However, sequencing errors present in raw reads, have an impact on the base-level accuracy of the final assembly. Despite different approaches have been implemented trying to limit the propagation of sequencing errors to final assembly, (e.g. reads correction and consensus calling step) the quality of long-reads based assemblies is still not optimal and complementary bioinformatics approaches need to be exploited to improve overall quality. To tackle this limitation assembly polishing have been performed using long reads sequencing, short reads sequencing and a combination of both approaches. PacBio based assembly have been polished using PacBio reads with *GenomicConsensus v3.4.1* using *Quiver* algorithm; ONT based assembly have been polished using ONT reads using *Racon v1.4.3* and *Medaka v0.8.1*; polishing performed using short reads data have been performed using *Pilon v1.23*. Polishing improvement have been assessed using short reads data, identifying variants present in the genomes using *GATKToolkit v3.8.1*. Majority of errors present in the PacBio an ONT based assemblies generated are InDels, comprising 85% and 75% of total errors respectively (Figure 12). ONT-based assembly resulted to be the most error-prone, having 1.6 million position identified as erroneous compared to the 181,267

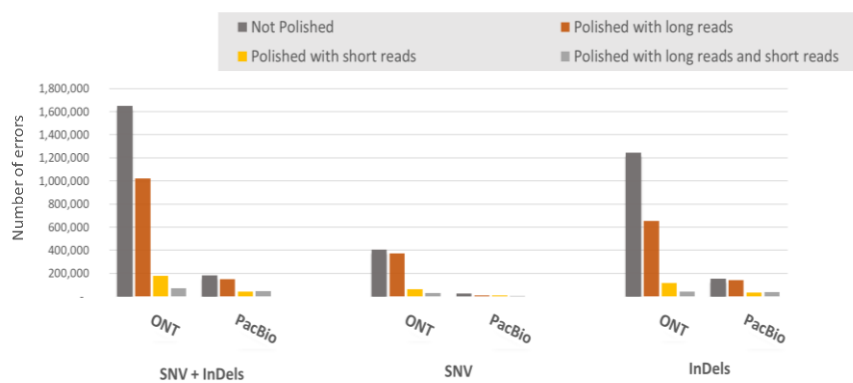


Figure 11. **Number of errors in de novo genome assemblies.** The number of Single Nucleotide variation (SNV) and short (<50 bp) insertions or deletion (InDels) is reported for the assembly generate using PacBio or ONT dataset. Number of variants identified in draft assembly as well as polished assembly is reported.

of the PacBio assembly. After polishing using long reads data number of errors present in ONT assembly reduced by 40% reaching 1,022,792 while PacBio decreased to 148,253. A higher reduction of errors was observed using short reads data; after error correction ONT and PacBio assemblies presented 177,799 and 43,501 errors respectively. Correcting using short read data subsequently to long reads data is possible to observe a reduced number of errors in the ONT data: 70,820 while PacBio assembly presented 44,443 errors. This increased number of errors derived from a higher number of SNV, while Indels resulted in a lower number.

	SNV + InDels		SNV		InDels	
	ONT	PacBio	ONT	PacBio	ONT	PacBio
Not Polished	1,650,034	181,267	406,670	27,009	1,243,364	154,258
Polished with long reads	1,022,792	148,253	370,875	8,480	651,917	139,773
Polished with short reads	177,799	43,501	62,305	10,464	115,494	33,037
Polished with long reads and short reads	70,820	44,443	29,614	5,539	41,206	38,904

Table 7 Number of errors in the assembly: Different polishing approaches have been tested on the PacBio and ONT assemblies. For each assembly (not polished, polished using only long reads, polished using only short reads and polished using long reads and short reads) number of SNV, InDels and total variants have been reported

Reduction in base level errors was reflected also to BUSCO completeness assessment. As for the analysis showed before, a greater improvement was observed for ONT-based assembly (Figure 13A), which BUSCO score increased from 93.3% to 97.1% compared to the PacBio-based assembly which BUSCO score was nearly unchanged.

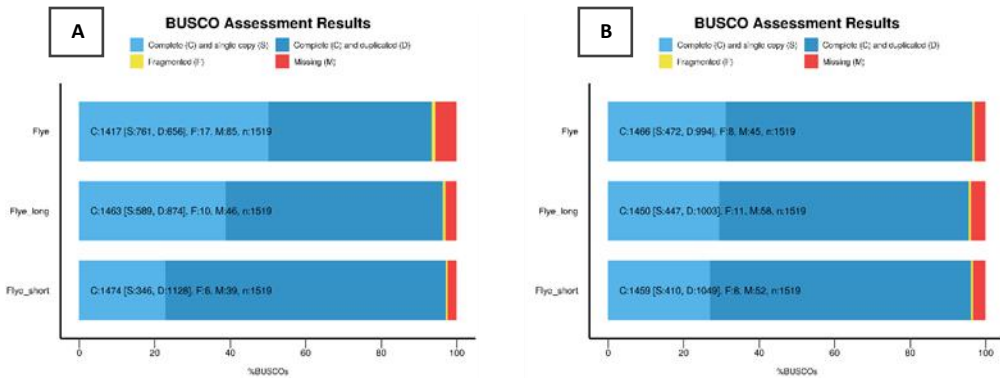


Figure 12: BUSCO completeness values on the ONT-based assemblies [A] and PacBio based assemblies [B]. From the top to the bottom, values represent BUSCO genes identified in the Flye not polished assembly, Flye assembly polished with long reads and Flye assembly polished with long and short reads.

Merging long reads-based assemblies

To further improve assembly contiguity two long reads-based assembly generated with Flye were merged using meta-assembler QuickMerge v0.3. This software performs whole genome alignment filling gaps or not reconstructed regions of one assembly exploiting information of the other. The resulted merged assembly has an increased assembly size (309 Mbp) with a contig N50 value of 284 kb (Table 8). This procedure permitted, moreover, to decrease the number of total contigs to 3,646 with an average contigs length of 84 Kbp. Merged assembly will be used as starting assembly to assess the impact of different scaffolding methods.

	Flye (PacBio)		Flye (ONT)		PacBio + ONT	
Assembly length	272.5		278.6		309.1	
Number of contigs	6,176		5,009		3,646	
Average contigs length	44.1		55.6		84.7	
Longest contig	457.5		1,189.2		1,861.1	
Contigs N50	86.5		247.8		284.0	
Contigs N90	23.8		56.1		49.9	
GC percentage	60.1%		60.3%		60.1%	
	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)
Contigs > 10 Kbp	4,416	264.9	1,897	270.2	2,330	303.6
Contigs > 50 Kbp	1,977	201.8	1,270	253.7	1,254	278.1
Contigs > 100 Kbp	753	114.7	883	225.4	877	250.9
Contigs > 300 Kbp	22	7.8	238	111.5	289	145.5

Table 8. Merging long-read based assemblies. Table reports statistics of the assembly generated using PacBio dataset (assembled using Flye), ONT dataset (assembled using Flye) and the merged assembly. For each assembly are shown: the total assembly length, the number of reconstructed contigs, the average contigs length, the longest assembled contig, the N50 and N90 value of the assembly and the GC percentage. Moreover, number of contigs and relative contigs length are reported for contigs greater than 10, 50, 100, 300 Kbp.

Benchmarking of bioinformatics approaches for genome scaffolding and anchoring

Except for short genomes like bacterial or viral, contig assembly most likely do not represent chromosomes or chromosome arms, interrupting in low complex regions or in those regions difficult to sequence. In the last years different methods for genome scaffolding have been developed exploiting orthogonal long-range technologies to scaffold contigs. We tested in parallel three different bioinformatics approaches that exploit linked reads data, optical mapping data and chromosome conformation capture information to improve assembly contiguity generating scaffolds. Scaff10X v4.2 have been applied to retrieve information contained in linked reads data (e.g. produced by 10X Genomics), Bionano Solve pipeline v3.4 is a toolkit developed by Bionano Genomics aimed to analyse optical mapping data and finally, 3D-DNA pipeline using juicer v1.5.7 has been used to retrieve proximity information from a Hi-C library preparation. Using the merged assembly described above as starting point we benchmarked the just mentioned methods.

Linked reads

Illumina sequencing of one 10X Genomics library has been performed on HiSeq X instrument using 150PE protocol generating 88 million fragments. Exploiting barcode information carried by sequenced reads, input contigs have been joined by Scaff10X into 180 scaffolds comprising 106 Mbp. Since this doesn't reach half of the assembled size (309 Mbp) is not possible to calculate N50 value. The 180 scaffolds contain 270 Gaps. When join two contigs is not possible, when using linked reads data, to accurate calculate the real gap size and, thus, scaffolding software insert a pre-selected gap length (e.g. 100 bp). More than three thousand contigs couldn't be placed in scaffold which cover 202 Mbp of the assembly (Table 10).

Optical mapping data

Sample was analysed using double-enzyme approach, increasing the label density, enhancing contigs anchoring. Analysis was performed on the Saphyr instrument generating a total of 1.1 million molecules covering 361 Gbp of data (more than 1,000-fold coverage). Molecules N50 is 196 Kbp and 210 Kbp for BspQI enzyme and BssSI enzyme respectively.

Input data was assembled using BioNano Solve pipeline generating two genome maps (Table 9), one for each selected enzyme.

	Enzyme BspQI	Enzyme BssSI
Genome map length (Mbp)	323.9	370
Number of optigs	699	741
Average optigs length (Kbp)	463.4	499
Longest optigs (Mbp)	3.0	3.2
Optigs N50 (Kbp)	520.0	630.0

Table 9. Genome map assembly statistics. For each enzyme, table reports genome maps statics including total genome map length, number of reconstructed optigs, average optigs length, longest reconstructed optigs and N50 value of the assembly.

Draft assembly was then aligned to both maps generating the hybrid assembly. Seventy-four percent of draft assembly, encompassing 228 Mbp, has been anchored to the map generating a hybrid assembly of 263 Mbp containing 604 gaps with a total of unknown bases of 33 Mbp. Scaffolds encompass 223 Mbp of the anchored sequences and have a N50 value of 849 Kbp. Contigs anchored, but not scaffolded are 153, with a total length of 40 Mbp. Not anchored contigs contain a total length of 79 Mbp with an average size of 24 Kbp highlighting that the majority of unanchored contigs are shorter than the optical map resolution.

Chromosome Conformation Capture technology

A Hi-C library have been sequenced using NovaSeq 600 instrument generating 300 million fragments. Hi-C library has been mapped to the draft assembly and the information regarding the spatial proximity of genomic regions have been exploited to join contigs in scaffolds. 3D-DNA pipeline successfully joined 1,819 contigs

generating 55 scaffolds containing 245 Mbp of sequence with an N50 of 5.1 Mbp. Scaffolds contains 1,765 gaps but, as motioned for linked reads, gaps size cannot be calculated confidentially and thus the pipeline introduce between adjacent contigs a user-selected length (e.g. 500 bp). Longest scaffold is 13.4 Mbp and the average scaffolds size is 4.5 Mbp.

	Bionano											
	10X genomics		Anchored		Not anchored		Anchored + Not Anchored		Hi-C			
Total assembly length (Mbp)	309.1		263.9		79.1		343.1		310.0			
Total scaffolds length (Mbp)	106.3		223.1				223.0		245.6			
Number of scaffolds	180		271				271		55			
Scaffolds N50 (Kbp)			849.5				656.2		5,095.1			
Scaffolds average length (Kbp)	590.6		823.2				823.2		4,466.4			
Longest scaffold (bp)	1.8		3.0				3.0		13.4			
Number of Gaps	270		604				604		1,765			
Gaps size (Kbp)	27.0		33,975.1				33.975.1		882.5			
Contigs in scaffolds	450		875				875		1,819			
Remaining contigs	3,196		153		3,174		3,327		3,428			
Remaining contig total length (Mbp)	202.8		40.8		79.1		120.0		64.3			
Remaining contigs N50 (Kbp)	269.1		303.2		54.8		93.0		42.8			
Remaining contigs average length (Kbp)	63.4		267.2		24.9		36.0		18.7			
Longest remaining contig (Kbp)	1,861.1		951.2		391.8		951.2		350.4			
		Number of sequences		Cumulative length (Mbp)		Number of sequences		Cumulative length (Mbp)		Number of sequences		Cumulative length (Mbp)
Sequences > 10 Kbp	2,060		303.6		424		263.9		1,745		73.1	
Sequences > 50 Kbp	984		278.2		424		263.9		489		42.7	
Sequences > 100 Kbp	696		258.1		418		263.4		127		17.9	
Sequences > 300 Kbp	320		188.1		295		238.0		2		0.696	

Table 10. Comparison of assembly generated using bioinformatics methods for scaffolding and anchoring. For each method, assembly statistics are reported. The rows show the total assembly length, the total scaffolds length, the scaffolds N50 (calculated considering the total assembly size), the scaffolds average length, the longest scaffold and the gaps metrics (Number of undefined bases and number of gaps). Subsequently is reported the number of ungapped sequences (remaining contigs), their total length, N50 value average length and the longest. Moreover, number of sequences (both scaffolds and contigs) and relative cumulative length are reported for sequences greater than 10, 50, 100, 300 Kbp.

Assessing Chromosome Conformation capture technology result

Results reported in the chapter above highlighted how Chromosome Conformation Capture technology can reconstruct chromosome-scale assembly. To assess the result obtained we decided to use two bioinformatics approaches to validate the structure of the assembly generate using Hi-C. The first one, rely on the fragment coverage, calculated exploiting linked reads data using break10X v3.1 software. Even though linked reads have been sequenced using short read technology, information carried by the barcode allow to retrieve the information of the original DNA fragment generating the barcoded-reads. Thus, it is possible to exploit fragment coverage instead of reads coverage to evaluate assembly accuracy. Those regions representing a misassembly, will not be supported by any fragment (e.g. any fragment will span over misassembled region) and will be labelled as erroneous. The second approach is based on optical maps. Genome maps have been aligned to the assembly using RefAligner v9232 and regions which highlight incongruences have been corrected breaking the contig or scaffolds in the misassembled region from BionanoSolve Pipeline v3.4. Hi-C assembly resulted fragmented using both technologies meaning that. Using fragment coverage have been identified 117 misassembly, corrected assembly resulted having a greater number of scaffolds and contigs. Consequently, scaffolds N50 decrease to 2.0 Mbp. Optical mapping pipeline permits to break sequences where misassembly is identify and then to join corrected sequences in right order and orientation base on map data. Fifty-four sequenced were corrected, identifying 524 misassembly. Resulted assembly contains 439 scaffolds with an N50 of 864 Kbp.

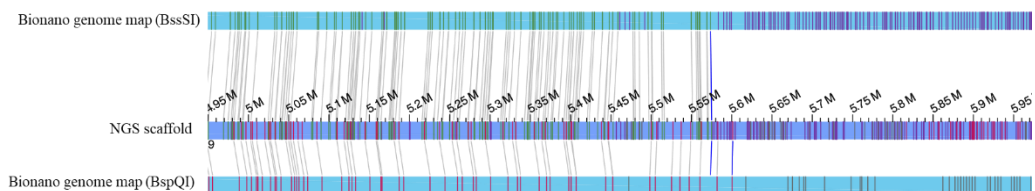


Figure 13: **Incongruence between NGS and Optical mapping data.** The figure shows a region where the two reconstructed genome maps align properly to the Hi-C scaffolded assembly until the misassemble region.

	Corrected with linked reads		Corrected with optical mapping data	
Total assembly length (Mbp)	310.0		326.5	
Total scaffolds length (Mbp)	245.1		265.1	
Number of scaffolds	157		439	
Scaffolds N50 (Kbp)	2,043.8		864.7	
Scaffolds average length (Kbp)	1,561,569		603.9	
Longest scaffold (bp)	5,717,429		3,942,643	
Number of Gaps	1,765		1,962	
Gaps size (Kbp)	882,500		17,459.3	
Contigs in scaffolds	1,921		2,376	
Remaining contigs	3,443		3,479	
Remaining contig total length (Mbp)	64.8		61.4	
Remaining contigs N50 (Kbp)	43.2		38.0	
Remaining contigs average length (Kbp)	18.8		17.6	
Longest remaining contig (Kbp)	350.4		576.8	
	Number of sequences	Cumulative length (Mbp)	Number of sequences	Cumulative length (Mbp)
Sequences > 10 Kbp	1781	302.2	2001	318.2
Sequences > 50 Kbp	500	274.2	658	288.2
Sequences > 100 Kbp	241	257.5	402	271.5
Sequences > 300 Kbp	135	241.0	256	245.9

Table 11. Comparison of Hi-C assembly corrected using linked reads and optical mapping data. For each correction, assembly statistics are reported. The rows show the total assembly length, the total scaffolds length, the scaffolds N50 (calculated considering the total assembly size), the scaffolds average length, the longest scaffold and the gaps metrics (Number of undefined bases and number of gaps). Subsequently is reported the number of ungapped sequences (remaining contigs), their total length, N50 value average length and the longest. Moreover, number of sequences (both scaffolds and contigs) and relative cumulative length are reported for sequences greater than 10, 50, 100, 300 Kbp.

Discussion

High quality genome assembly represents the fundament of further genomic studies. Continuously evolving sequencing technologies motivated the development of adequate bioinformatic tools to assemble sequenced reads. However, limitation of de novo assembly methods and sequencing technologies makes genome assembly an active research field [31][32][33]. Short reads sequencing, highly exploited during early 2000s, permitted to reconstruct genome assembly with high base-level accuracy. DBG algorithms was implemented to manage this kind of data, however, limited by sequencing read length, DBG methods do not permit to discriminate between repetitive regions hampering the reconstruction of high contiguous contigs. Indeed, in eukaryotic organism, transposable element generate ambiguous path in the assembly graph and represent the most relevant cause of contigs breaks during genome reconstruction [34][35]. *H.pluvialis* assemblies, generated using short reads data, resulted greatly fragmented using both tested software, with a contigs N50 about 5 Kbp, too low to annotate many gene models. Assembly generated by SPAdes resulted slightly larger than the one assembled by SOAPdenovo: 219 Mbp and 208 Mbp respectively. Interestingly, SPAdes is able to reconstruct longer contiguous sequences, highlighted also by the much smaller number of gaps and the lower total gaps length. More than 90% of assembly generated by SPAdes reside in contigs, underlining the effectiveness of Gap Closure module developed in the software. Genome size confirms limitation of kmer based approaches in reconstructing repetitive regions [36] which resulted to be collapsed if the kmer length cannot span the whole repetition, decreasing the size of final assembly. During the last years, development of long reads technologies permitted to sequence longer fragment of DNA enabling to depict a greater portion of the genome [37]. OLC based software using PacBio dataset permitted, indeed, to reconstruct a more contiguous assemblies, which are overall larger (60Mbp) deriving from an increased read length able to span over repetitive regions, anchor to the unique extremities and, thus, reconstruct repetitive regions. Due to an increased error rate compared to short reads data, DBG is not suitable for long reads. Indeed, this approach would detect

plenty of false kmer arising from sequencing errors generating spurious path in the assembly graph. Flye permitted to reconstruct a larger genome having the highest N50 value. Nevertheless, the four tested software showed comparable performance in terms of contiguity. The largest assembled sequence was reconstructed by Canu and it is 1.4 Mbp in length. While the increased assembly size obtained using long reads and mapping technologies, compared to Illumina based ones, was a consequence of the ability to reconstruct repetitive regions, the discordant assembly size obtained using Nanopore, PacBio and optical mapping data highlight the possibility that this species is highly heterozygous. Indeed, different implementation of assembly algorithms lead to a different amount of collapsing haplotypes affecting the final assembly size [38].

Concerning base level accuracy, Falcon and CANU perform a reads correction step before to proceed with the assembly, while Flye and Wtdbg2 compensate the absence of correction phase with final consensus step during which raw reads are used to polish the assembly. This procedure resulted more effective in Flye assembly highlighted by the higher number of reconstructed genes models compared to the others three approaches.

Long reads permitted to obtain assemblies containing greater portion of the gene space, namely 10% more of BUSCO genes was reconstructed using long reads approach. To confirm the fact that read length impact positively on the contiguity metrics and gene space reconstruction, a de novo assembly was performed using increased reads length dataset obtained using ONT technology. This, permitted to obtain longer and contiguous contigs, having an N50 of 246 Kbp, almost three times the contiguity obtained using PacBio dataset. Differently from the PacBio based assemblies, tested software showed more differences in terms of contigs and genome size but the best contiguity metrics along with best BUSCO value was identified using Flye based assembly as well. An overall worsening of BUSCO value have been observed using ONT reads, probably due by a higher sequencing error rate and a lower sequencing coverage. Despite the newly introduced software Shasta doesn't implement an error correctio step, the percentage of complete genes identified by BUSCO is comparable with Canu which it does. Wtdbg2 resulted the worst in terms of reconstructed gene models.

Depth of coverage is always considered one of the main parameters during the setup of a genome assembly project. We investigated the impact of this parameter, related to read length using the ONT dataset, highlighting how the lowering coverage, discarding shorter sequences, do not impact negatively on the resulted assembly. On the contrary, discarding shorter reads, assemblies resulted in a fewer number of contigs and with a greater N50 value, highlighting the importance of having longer reads even at the expense of reads coverage. From the other side, keep sequencing to increase depth of coverage would be unnecessary if sequenced reads length is not adequate. From bioinformatic point of view, longer sequences spanning over repetitive region, even with lower coverage, permit to identify unique overlaps and, thus, to walk through duplicated regions reconstructing those portions of genome that, otherwise, would be obscured.

The benefit of assemble with long reads technologies, however, have some drawbacks, deriving from the higher error rate of raw sequences. Indeed, long-reads based assembly suffer base-level errors, the majority of which are small insertion or deletions (InDels). These kinds of errors can cause frameshift in the final assembly hampering an accurate gene prediction [17]. Many approaches have been developed trying to limit the impact of this disadvantage during the assembly procedure [39][40]. Despite these efforts, assembly algorithms can't fully resolve this concern, resulting contigs with a relatively low base-level accuracy. As highlighted in this work, bioinformatics approaches to correct base-level errors are then necessary to refine genome assembly using complementary data (e.g. Illumina reads) that, despite a limited length (up to 300bp), enable, thanks to the high accuracy (> 99%), to polish the sequence. Refinement performed using a combination of long and short reads data permitted to increase *H.pluvialis* genome quality to QV38 in the case of PacBio assembly enhancing also further genes annotations. Despite the increased read length and the consequent improved contiguity, ONT based assemblies contain more errors than PacBio. Considering the assemblies refined using both long and short reads, ONT assembly contains almost double number of errors (70,820) compared to the PacBio assembly (44,443).

Results presented suggested that, when approaching genome assembly, a trade-off between contiguity and base-level accuracy should be considered. Despite huge improvements in the last few years, long reads technologies behave differently, and results need to be complemented with short reads data to enhance base-level quality.

Bioinformatic approaches to combine long and short reads permit to obtain high quality genomes, combining the ability to reconstruct low complex regions of long reads sequencing and the high base-level quality of short reads technologies. However, apart for some prokaryote or small viruses' genomes, contigs usually do not represent chromosomes [41], centromeres or long segmental duplication still represent complex regions hard to be confidentially reconstructed. Scaffolding approaches have been developed with the aim to compensate this limitation exploiting long range technologies that permit to bypass repetitive regions and to order and orient contigs into scaffolds. Tested methods, relying on linked reads, optical mapping data and chromosome conformation capture, disclosed different performance. The latter one, permitted to reconstruct chromosome scale scaffolds. Long-range information generates using Hi-C library derive from three-dimensional interaction of distant loci into the nucleus. Hi-C library is sequenced using Illumina instrument that still suffer of the mapping limitation of short sequencing technologies. Moreover, bioinformatics approaches tend to introduce inversions deriving from noise in the Hi-C data [23]. Thus, orthogonal technology are still necessary to validate the scaffolding result [42] [43].

Knowing the limitation of chromosome conformation capture technology, we verified the results using two bioinformatics approaches that permit to verify the assembly quality. Exploiting fragment coverage of linked reads is it possible to detect regions not supported by sequenced data, identifying misassembled regions. This approach allowed to break Hi-C scaffolds in 117 positions suggesting that the assembly contained misassembly. Using mapping data, the number of misassembled regions was even higher: scaffolds have been broken in 524 positions lowering the assembly N50 value to 864 Kbp.

Development of bioinformatic methods for genome sequencing and scaffolding permit to obtain assembly with unprecedented quality allowing starting of massive

sequencing projects initiative (e.g. Vertebrate Genome Project, Bat1K, Genome 10K). However, genome assembly cannot be considered a solved problem, especially for highly repetitive genomes, and still is not available a methodology or a sequencing technology without some drawbacks.

In this thesis different bioinformatic approaches and sequencing technologies have been employed to reconstruct the highly repetitive *H.pluvialis* genome. Sequential integration of additional data layers permitted to bypass single methodology limitation. The knowledge of bioinformatics approaches, how to integrate different methodologies and the importance of validate the results is necessary to reconstruct genome assembly that can also be considered a reference genome.

References

- [1] E. W. Myers Jr, “A history of DNA sequence assembly,” *it - Inf. Technol.*, vol. 58, no. 3, pp. 126–132, 2016.
- [2] J. L. Weber and E. W. Myers, “Human Whole-Genome Shotgun Sequencing,” no. 715, pp. 401–409, 1997.
- [3] A. Payne, N. Holmes, V. Rakyan, and M. Loose, “Bulkvis: A graphical viewer for Oxford nanopore bulk FAST5 files,” *Bioinformatics*, vol. 35, no. 13, pp. 2193–2198, 2019.
- [4] S. Lander *et al.*, “Initial sequencing and analysis of the human genome International Human Genome Sequencing Consortium* The Sanger Centre: Beijing Genomics Institute/Human Genome Center,” *Nature*, vol. 409, no. February, 2001.
- [5] M. Jain *et al.*, “Nanopore sequencing and assembly of a human genome with ultra-long reads,” *Nat. Biotechnol.*, vol. 36, no. 4, pp. 338–345, 2018.
- [6] G. G. SUTTON, O. WHITE, M. D. ADAMS, and A. R. KERLAVAGE, “TIGR Assembler: A New Tool for Assembling Large Shotgun Sequencing Projects,” *Genome Sci. Technol.*, vol. 1, no. 1, pp. 9–19, 1995.
- [7] R. D. Fleischmann *et al.*, “Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd,” *Science (80-.)*, vol. 269, no. 5223, pp. 496–512, 1995.
- [8] C. M. Fraser *et al.*, “The Minimal Gene Complement Myco plasma genitalium of,” *Science (80-.)*, vol. 270, no. 5235, pp. 397–403, 1995.
- [9] M. D. Adams *et al.*, “The genome sequence of *Drosophila melanogaster*,” *Science (80-.)*, vol. 287, no. 5461, pp. 2185–2195, 2000.
- [10] J. Craig Venter *et al.*, “The sequence of the human genome,” *Science (80-.)*, vol. 291, no. 5507, pp. 1304–1351, 2001.
- [11] P. A. Pevzner, H. Tang, and M. S. Waterman, “An Eulerian path approach

- to DNA fragment assembly,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 98, no. 17, pp. 9748–9753, 2001.
- [12] C. S. Chin *et al.*, “Phased diploid genome assembly with single-molecule real-time sequencing,” *Nat. Methods*, vol. 13, no. 12, pp. 1050–1054, 2016.
- [13] S. Koren, B. P. Walenz, K. Berlin, J. R. Miller, N. H. Bergman, and A. M. Phillippy, “Canu: Scalable and accurate long-read assembly via adaptive κ -mer weighting and repeat separation,” *Genome Res.*, vol. 27, no. 5, pp. 722–736, 2017.
- [14] M. Kolmogorov, J. Yuan, Y. Lin, and P. A. Pevzner, “Assembly of long, error-prone reads using repeat graphs,” *Nat. Biotechnol.*, vol. 37, no. 5, pp. 540–546, 2019.
- [15] J. Ruan and H. Li, “Fast and accurate long-read assembly with wtdbg2,” *bioRxiv*, p. 530972, 2019.
- [16] K. Shafin *et al.*, “Efficient de novo assembly of eleven human genomes using PromethION sequencing and a novel nanopore toolkit A Preprint,” 2019.
- [17] M. Watson and A. Warr, “Errors in long-read assemblies can critically affect protein prediction,” *Nature Biotechnology*. 2019.
- [18] J. Wetzel, C. Kingsford, and M. Pop, “Assessing the benefits of using mate-pairs to resolve repeats in de novo short-read prokaryotic assemblies,” *BMC Bioinformatics*, vol. 12, no. 1, p. 95, 2011.
- [19] Q. Li *et al.*, “A chromosome-scale genome assembly of cucumber (*Cucumis sativus* L.),” *Gigascience*, vol. 8, no. 6, pp. 1–10, 2019.
- [20] L. Zhang *et al.*, “A high-quality apple genome assembly reveals the association of a retrotransposon and red fruit colour,” *Nat. Commun.*, vol. 10, no. 1, pp. 1–13, 2019.
- [21] D. J. Bertioli *et al.*, “The genome sequence of segmental allotetraploid peanut *Arachis hypogaea*,” *Nat. Genet.*, vol. 51, no. 5, pp. 877–884, 2019.

- [22] W. Tang *et al.*, “Chromosome-scale genome assembly of kiwifruit *Actinidia eriantha* with single-molecule sequencing and chromatin interaction mapping,” *Gigascience*, vol. 8, no. 4, pp. 1–10, 2019.
- [23] J. Ghurye and M. Pop, “Modern technologies and algorithms for scaffolding assembled genomes,” *PLoS Comput. Biol.*, vol. 15, no. 6, pp. 1–20, 2019.
- [24] A. Bankevich *et al.*, “SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing,” *J. Comput. Biol.*, vol. 19, no. 5, pp. 455–77, May 2012.
- [25] R. Luo *et al.*, “SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler,” *Gigascience*, vol. 1:18, no. 1, pp. 1–6, 2012.
- [26] H. Li, “Minimap2: Pairwise alignment for nucleotide sequences,” *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018.
- [27] B. J. Walker *et al.*, “Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement,” *PLoS One*, vol. 9, no. 11, 2014.
- [28] H. Li and R. Durbin, “Fast and accurate short read alignment with Burrows-Wheeler transform,” *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.
- [29] G. A. Van der Auwera *et al.*, *From fastQ data to high-confidence variant calls: The genome analysis toolkit best practices pipeline*, no. SUPL.43. 2013.
- [30] M. Chakraborty, J. G. Baldwin-Brown, A. D. Long, and J. J. Emerson, “Contiguous and accurate de novo assembly of metazoan genomes with modest long read coverage,” *Nucleic Acids Res.*, vol. 44, no. 19, pp. 1–12, 2016.
- [31] F. J. Sedlazeck, H. Lee, C. A. Darby, and M. C. Schatz, “Piercing the dark matter: bioinformatics of long-range sequencing and mapping,” *Nat. Rev.*

Genet., 2018.

- [32] J. Il Sohn and J. W. Nam, “The present and future of de novo whole-genome assembly,” *Brief. Bioinform.*, vol. 19, no. 1, pp. 23–40, 2018.
- [33] A. M. Giani, G. R. Gallo, L. Gianfranceschi, and G. Formenti, “Long walk to genomics: History and current approaches to genome sequencing and assembly,” *Comput. Struct. Biotechnol. J.*, no. November, 2019.
- [34] P. Bongartz, “Resolving repeat families with long reads,” *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–11, 2019.
- [35] N. Ricker, H. Qian, and R. R. Fulthorpe, “The limitations of draft assemblies for understanding prokaryotic adaptation and evolution,” *Genomics*, vol. 100, no. 3, pp. 167–175, 2012.
- [36] O. K. Tørresen *et al.*, “Tandem repeats lead to sequence assembly errors and impose multi-level challenges for genome and protein databases,” *Nucleic Acids Res.*, vol. 47, no. 21, pp. 10994–11006, 2019.
- [37] A. V. Zimin *et al.*, “An improved assembly of the loblolly pine megagenome using long-read single-molecule sequencing,” *Gigascience*, vol. 6, no. 1, pp. 1–4, 2017.
- [38] N. Guiglielmoni, A. Derzelle, K. van Doninck, and J.-F. Flot, “Overcoming uncollapsed haplotypes in long-read assemblies of non-model organisms,” *bioRxiv*, p. 2020.03.16.993428, 2020.
- [39] A. S. Alic, D. Ruzafa, J. Dopazo, and I. Blanquer, “Objective review of de novo stand-alone error correction methods for NGS data,” *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, vol. 6, no. 2, pp. 111–146, 2016.
- [40] H. Zhang, C. Jain, and S. Aluru, “A comprehensive evaluation of long read error correction methods,” *bioRxiv*, p. 519330, 2019.
- [41] “A reference standard for genome biology,” *Nat. Biotechnol.*, vol. 36, no. 12, p. 1121, 2018.
- [42] D. M. Bickhart *et al.*, “Single-molecule sequencing and chromatin

conformation capture enable de novo reference assembly of the domestic goat genome,” *Nat. Genet.*, vol. 49, no. 4, pp. 643–650, 2017.

- [43] A. Wallberg *et al.*, “A hybrid de novo genome assembly of the honeybee, *Apis mellifera*, with chromosome-length scaffolds,” *BMC Genomics*, vol. 20, no. 1, pp. 1–19, 2019.