**Ministry of Science and Higher Education of the Russian Federation**

**ITMO University**

**GRADUATION THESIS**

# Chromosome-Scale Genome Assembly from Long Noisy Reads using Hi-C Data.

Author: Anton Andreevich Zamyatin _____
(full name) (signature)

Subject area 01.04.02 Applied Mathematics and Informatics

Degree level Master

Thesis supervisor: Alexeev N.V., PhD, Lead Researcher

_____
(signature)

St. Petersburg, 2020

Student Anton Andreevich Zamyatin
(full name)
Group M42352 Faculty of Information Technologies and Programming

Subject area, program Bioinformatics and Systems Biology

Consultant(s):
Avdeyev P.V., George Washington University _____
(surname, initials, academic title, degree) (signature)

Thesis received "____"_____2020__

Originality of thesis: _____%

Thesis completed with the grade: _____

Date of defense "____"_____20 ____

Secretary of State Exam Commission_____ _____
(full name) (signature)

Number of pages _____

Number of supplementary materials/Blueprints _____

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“Национальный исследовательский университет ИТМО”**

# ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

# СБОРКА ГЕНОМОВ УРОВНЯ ХРОМОСОМНЫХ СКАФФОЛДОВ ИЗ ДЛИННЫХ РИДОВ С БОЛЬШОЙ ЧАСТОТОЙ ОШИБОК С ИСПОЛЬЗОВАНИЕМ ДАННЫХ Hi-C

Автор_____Замятин Антон Андреевич___ _____
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки _____01.04.02 Прикладная математика и информатика

Квалификация _____Магистр_____

Руководитель ВКР_____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

Санкт-Петербург, 2020 г.

Обучающийся__Замятин Антон Андреевич_____
(ФИО полностью)

Группа__М42352_ Факультет _Информационных технологий и программирования

Направленность, специализация _ Биоинформатика и системная биология ____

Консультант (ы):

а) __Авдеев Павел Владимирович_____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

б) _____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

ВКР принята “____”_____20 ____г.

Оригинальность ВКР _____%

ВКР выполнена с оценкой _____

Дата защиты “____”_____20 ____г.

Секретарь ГЭК _____ _____
(ФИО) (подпись)

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

**Ministry of Science and Higher Education of the Russian Federation**

**ITMO University**

**APPROVED**

Head of educational program

_____ _____

(Surname, initials) (signature)

«____» «_____» 20____

# OBJECTIVES
## FOR A GRADUATION THESIS

**Student** Anton Andreevich Zamyatin

(full name)

**Group** M42352 **Faculty** of Information Technologies and Programming

**Degree level** Master's

**Subject area** 01.04.02 Applied Mathematics and Informatics

**Major** Bioinformatics and Systems Biology

**Specialization** _____

**1 Thesis topic** Chromosome-scale genome assembly from long noisy reads using Hi-C data.

**Thesis supervisor** Alexeev Nikita Vladimirovich, PhD, Lead Researcher, ITMO University

(full name, place of employment, position, academic degree, academic title)

**2 Deadline for submission of complete thesis «**_____» «_____» 20_____

**3 Requirements and premise for the thesis**

The theoretical analysis of the literature on the topic. Performing the best strategy of genome assembly from long nanopore reads and draft assembly polishing using short Illumina reads for two mosquito species. Performing assembly chromosome-level scaffolding using Hi-C data. Genomes assembly assessment and validation. Performing genome assembly for two barnacle species from long pacbio reads. Polishing of assemblies using short Illumina reads. Genomes assembly assessment and validation.

**4 Content of the thesis (list of key issues)**

a) The terminology used in the thesis and description of main concepts and technologies. b) Mosquitos project. Project introduction, materials, and methods, project results c) Barnacles project. Project introduction, materials, and methods. Project results. d) Conclusion

**5 List of graphic materials (with a list of required material)**

Graphic materials representing obtained results are provided along within the thesis text. Additional materials for mosquitos project and barnacles project are in appendix A and B respectively.

**6 Source materials and publications** *reference materials must not be older than 10 years*

"A chromosome-scale assembly of the major African malaria vector Anopheles funestus" J Ghurye, s. Koren, S Small et.al. GigaScience 2019 DOI: 10.1093/gigascience/giz063

"De novo assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds" O. Dudchenko, S.Batra, A.Omer et. al. Science 2017 DOI: 10.1126/science.aal3327

**7 Objectives issued on «**____» «_____» 20____

Thesis supervisor _____
(signature)

Objectives assumed by _____ «____» «_____» 20____
(signature)

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**УТВЕРЖДАЮ**

Руководитель ОП

_____ _____

(Фамилия, И.О.) (подпись)

«____» «_____» 20____ г.

# З А Д А Н И Е
## НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

**Обучающийся**__Замятин Антон Андреевич__

(ФИО полностью)

**Группа**__M42352_ **Факультет** _Информационных технологий и программирования

**Квалификация** _____Магистр_____

**Направление подготовки** ___01.04.02 Прикладная математика и информатика _____

**Направленность образовательной программы** _ Биоинформатика и системная биология _

**Специализация** _____

**1 Тема ВКР**_Сборка геномов уровня хромосомных скаффолдов из длинных ридов_____

С большой частотой ошибок с использованием данных Hi-C_____

**Руководитель**_Алексеев Никита Владимирович, к.м.н, ведущий научный сотрудник, ИТМО

(ФИО полностью, место работы, должность, ученая степень, ученое звание)

---

**2 Срок сдачи студентом законченной работы до** «_____» «_____» 2020 г.

**3 Техническое задание и исходные данные к работе**

Анализ литературы по теме. Определить и применить лучшие стратегии сборки геномов из длинных ридов Oxford Nanopore и полировки промежуточных сборок короткими ридами Illumina для двух видов москитов. Осуществить сборку геномов до уровня хромосомных скаффолдов используя данные Hi-C. Оценить результаты промежуточные и конечные результаты сборки, валидировать финальную сборку. Осуществить сборку двух геномов полипов из длинных ридов Pacbio и полировку промежуточных сборок ридами Illumina. Оценить результаты, валидировать конечные сборки.

**4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)**

_А) Терминология, используемая в дипломной работе описание основных концепций и технологий. Б)_Проект по сборке геномов москитов. Вступление, материалы и методы, результаты. В) Проект по сборке геномов полипов Вступление, материалы и методы, результаты. Г) Заключение Д) Список использованной литературы

**5 Перечень графического материала (с указанием обязательного материала)**

Графические и табличные материалы по полученным результатам приведены в тексте. Дополнительные материалы по проекту сборки геномов москитов приведены в приложении А, по проекту сборки геномов полипов в приложении Б.

**6 Исходные материалы и пособия** *указанная литература должна быть не старше 10 лет*

_"A chromosome-scale assembly of the major African malaria vector Anopheles funestus" J Ghurye, s. Koren, S Small et.al. GigaScience 2019 DOI: 10.1093/gigascience/giz063

_"De novo assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds" O. Dudchenko, S.Batra, A.Omer et. al. Science 2017 DOI: 10.1126/science.aal3327

**7 Дата выдачи задания «**____» «_____» 20____г.

Руководитель ВКР_____
(подпись)

Задание принял к исполнению_____ «____» «_____» 20____г.
(подпись)

**Ministry of Science and Higher Education of the Russian Federation**

**ITMO University**

# SUMMARY
## OF A GRADUATION THESIS

**Student** Anton Andreevich Zamyatin
(full name)
**Title of the thesis** Chromosome-scale genome assembly from long noisy reads using Hi-C data.
**Name of organization** ITMO University

## DESCRIPTION OF THE GRADUATION THESIS

1 Research objective Produce and validate two chromosome-scale assemblies for mosquito species. Produce and validate two chromosome-scale assemblies for barnacle species.

2 Research tasks Performing the best strategy of genome assembly from long nanopore reads and draft assembly polishing using short Illumina reads for two mosquito species. Performing assembly chromosome-level scaffolding using Hi-C data. Genomes assembly assessment and validation. Performing genome assembly for two barnacle species from long pacbio reads. Polishing of assemblies using short Illumina reads. Genomes assembly assessment and validation.

3 Number of sources listed in the review section: 10
4 Total number of sources used in the thesis: 61
5 Sources by years:

| **Russian** | | | **Foreign** | | |
|---|---|---|---|---|---|
| In the last 5 years | 5 to 10 years | More than 10 years | In the last 5 years | 5 to 10 years | More than 10 years |
| - | - | - | 25 | 29 | 7 |

6 Use of online (internet) resources Yes
7 Use of modern computer software suites and technologies (List which ones were used and for which section of the thesis)

| **Software suites and technologies** | **Thesis section** |
|---|---|
| Nanopack tools package, Kraken2, minimap2, samtools, bedtools, BWA mem, FastQC, fastp, wtdbg2, miniasm, Flye, Canu, Quast, BUSCO, Pilon, Racon, Medaka, Nanopolish, HiCExplorer, SALSA2, 3D-DNA, Juicebox assembly tool, Juicer, Purge Haplotigs, D-genies | 2 |
| Falcon, Falcon-Unzip | 3 |
| SLURM, NCBI Blast+, jupyter-lab, ipython, R, miniconda. | 2,3 |

8 Short summary of results/conclusions
Different strategies for genome assembly and polishing were performed, results are assessed. Draft mosquito genomes were scaffolded into chromosome-level assemblies. Assemblies were validated. Two barnacle species were assembled with different assemblers, results are assessed. Genomes were polished and validated.

9 Grants received while working on the thesis No

10 Have you produced any publications or conference reports on the topic of the thesis No

Student Anton Andreevich Zamyatin_____
(Full name) (signature)

Thesis supervisor ____Alexeev N.V _____
(Full name) (signature)

“_____”_____20___

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

# АННОТАЦИЯ

## ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

**Обучающийся**_Замятин Антон Андреевич_____
(ФИО)

**Наименование темы ВКР:**_ Сборка геномов уровня хромосомных скаффолдов из длинных ридов_с большой частотой ошибок с использованием данных Hi-C_____

**Наименование организации, где выполнена ВКР**___Университет ИТМО _____

## ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования_Сборка геномов эукариот уровня хромосомных скаффолдов__
2 Задачи, решаемые в ВКР _Сборка и полировка геномов, скаффолдинг геномов, оценка и валидация сборок_____
3 Число источников, использованных при составлении обзора__10_____
4 Полное число источников, использованных в работе ____61_____
5 В том числе источников по годам

| **Отечественных** | | | **Иностранных** | | |
|---|---|---|---|---|---|
| Последние 5 лет | От 5 до 10 лет | Более 10 лет | Последние 5 лет | От 5 до 10 лет | Более 10 лет |
| - | - | - | 25 | 29 | 7 |

6 Использование информационных ресурсов Internet_____Да_____
(Да, нет, число ссылок в списке литературы)

7 Использование современных пакетов компьютерных программ и технологий (Указать, какие именно, и в каком разделе работы)

| **Пакеты компьютерных программ и технологий** | **Раздел работы** |
|---|---|
| Nanopack tools package, Kraken2, minimap2, samtools, bedtools, BWA mem, FastQC, fastp, wtdbg2, miniasm, Flye, Canu, Quast, BUSCO, Pilon, Racon, Medaka, Nanopolish, HiCExplorer, SALSA2, 3D-DNA, Juicebox assembly tool, Juicer, Purge Haplotigs, D-genies | 2 |
| Falcon, Falcon-Unzip | 3 |
| SLURM, NCBI Blast+, jupyter-lab, ipython, R, miniconda. | 2,3 |
| | |
| | |

8 Краткая характеристика полученных результатов
Для двух видов москитов различные стратегии геномной сборки и полировки геномов были применены, результаты оценены. Промежуточные геномы были в скаффолды хромосомного уровня. Сборки были валидированы._Два генома полипов были собраны различными сборщиками, результаты оценены. Геномы были отполированы и валидированы.

9 Полученные гранты, при выполнении работы ___Нет_____
(Название гранта)
_____

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы_Нет___
(Да, нет)

а) 1 _____
(Библиографическое описание публикаций)
2_____
3_____
_____

б) 1_____
(Библиографическое описание выступлений на конференциях)
2_____
3_____
_____
_____

Обучающийся__Замятин Антон Андреевич_ _____
(ФИО) (подпись)

Руководитель ВКР_Алексеев Никита Владимирович _____
(ФИО) (подпись)

“_____”_____20___г.

# CONTENTS

INTRODUCTION ..... 3
1. CHROMOSOME-SCALE ASSEMBLY ..... 5
1.1. Terminology. ..... 5
1.2. Technologies ..... 6
1.2.1. Illumina sequencing-by-synthesis. ..... 6
1.2.2. Oxford nanopore sequencing. ..... 8
1.2.3. PacBio sequencing. ..... 8
1.2.4. Hi-C. ..... 9
1.2.5. FISH experiments ..... 10
2. MOSQUITOS PROJECT. ..... 12
2.1. Project introduction ..... 12
2.2. Materials and methods ..... 13
2.2.1. Main pipeline. ..... 13
2.2.2. Data description ..... 14
2.2.3. Genome size estimation ..... 16
2.2.4. Nanopore reads contamination search ..... 18
2.2.5. Genome assemblers ..... 19
2.2.6. Wtdbg2 assembler ..... 20
2.2.7. Miniasm assembler ..... 20
2.2.8. Flye assembler ..... 20
2.2.9. CANU assembler ..... 21
2.2.10. Assemblies assessment Quast-lg and BUSCO genes ..... 21
2.2.11. Assemblies assessment auNg metric. ..... 23
2.2.12. Genome polishing. ..... 25
2.2.13. Scaffolding. ..... 26
2.2.14. SALSA2. ..... 28
2.2.15. 3D-DNA. ..... 29
2.2.16. By-hand scaffolding. ..... 30
2.2.17. Purge haplotigs. ..... 32
2.2.18. Validation. ..... 33
2.2.19. Validation. Rearrangements and dot-plots. ..... 33
2.2.20. Validation genes from the reference assembly ..... 36
2.2.21. Validation with marker sequences. ..... 37
2.2.22. Validation CQ analysis. ..... 40
2.3. Project results. ..... 41
3. BARNCLES PROJECT ..... 42

3.1. Project introduction. ..... 42
3.2. Materials and methods..... 42
3.2.1. Data description. ..... 42
3.2.2. Main pipeline..... 43
3.2.3. Falcon assembly..... 43
3.2.4. Contamination search..... 44
3.3. Project results. ..... 46
4. CONCLUSION. ..... 47
5. REFERENCES..... 48
APPENDIX A ..... 52
APPENDIX B. ..... 64

## INTRODUCTION

Sequencing technologies have revolutionized biological and medical research have led to the current genomic revolution. Knowledge about genomic sequences allows us to provide a tremendous number of different studies that expand understanding of our life and its connection to all living things in our world. Given a sequenced and assembled human genome we can use this information to find genetic causes of common and rare diseases, find the ways to treat them, we can build phylogenetic trees with genotyped variants and follow the history of human resettlement on our planet. Having genomes for model organisms allows us to experiment with these animals in cases which not possible conduct such experiments with humans, and save human lives with the results of these experiments. Knowledge of genome sequences for many more species provides us a better understanding of evolutionary processes. Here, in 2020, mankind tries not only to learn the sequence of each species genome including extinct but we have instruments to edit these sequences to achieve our goals in a predictive way. It becomes technology. But you cannot study or modify any species genome that is not sequenced and assembled in a proper way.

According to the U.S. National Center for Biotechnology Information (NCBI)[1] in 2020 we have sequenced and assembled 11 thousand eukaryotic genomes, almost 250 thousand prokaryotic genomes and 38 thousands of viral genomes. These numbers seem large but they are far from the number of known species. For example only Arthropoda phylum has 1,061,003 species according to the Catalogue of Life[2].

The quality of genome assembly plays an important role in modern research. New studies of genome rearrangements cannot be provided without chromosome-level assemblies. The contiguity of genome scaffolds allows us to better understand the organization of chromatin inside the cell nucleus. Possibility to sequence long repeat regions provides us insights into the organization of heterochromatin, large centromere, and telomere regions. But how to achieve this level of genome contiguity. Only long reads sequencing is not a cure. It can be that sequencer cant read particular regions at all. In that case we need good scaffolding. If we have a reference genome there are no problems with this. But how to be if we have no reference. We must use an additional source of information. In the past the best choice was to use mate-pairs reads. Now we have an incredible source of information about proximities in genome Hi-C. Hi-C method is very good for scaffolding but has some issues with low signal regions and ambiguity in haplotype regions.

After the finish of assembly and scaffolding genome assemblies must be validated to avoid misassembles and misjoints that ruin all work for creating chromosome-level reference genome.

My thesis is about all of these stages of chromosome-scale genome assembly. During my work with two genome assembly projects I followed through them and want to describe it here.

# 1. CHROMOSOME-SCALE ASSEMBLY

In this chapter, I describe the main concepts of chromosome-scale assembly and define terminology that is used in my thesis.

## 1.1. Terminology.

In my thesis I used common bioinformatics terminology that is widely used but must be determined for better understanding.

*DNA sequencing* - the process of determining the nucleic acid sequence.

*Sequencing read* - is an inferred sequence of base pairs corresponding to part of a sequenced molecule of DNA or RNA represented as a text string.

*Phred quality score* – is a measure of the reliability of base pair identification. Score Q is defined as a property which is logarithmically related to the base-calling error probabilities P. $Q = -10log_{10}P$. For each read we usually have second text string with Phred quality for each base pair encoded in symbols.

*Sequence alignment* is a computational process of sequences arrangement to find similarities between them. Also, the result of this process usually called the same. Levenshtein distance is usually used as a similarity metric. Alignment can be pairwise: global between whole sequences or local between a whole part of the reference sequence and whole query sequence; and multiple between more than two sequences.

*Consensus*. Given a collection of overlapping reads, that do not precisely match along with their overlaps, a consensus sequence for the collection is one for which the sum of the differences between the consensus sequence and each one of the reads is minimal.

*Contig* is a maximal set of reads in a layout which in aggregate covers a contiguous interval, the contig is a substring of the real genome sequence accurate to sequencing and assembly errors.

*Unitig* is a uniquely assemblable subset of overlapping fragments. A unitig is an assembly of fragments for which there are no competing choices in terms of internal overlaps. This means that a unitig is either a correctly assembled portion of a contig or it is an overcompressed assembly of several high-fidelity copies of a repeat.

*Scaffold* is a sequence of base pairs that is a set of contigs combined, arranged, and ordered according to additional information separated with poly N gaps. N is a symbol for unknown base pair.

*Genome assembly* is the computational process of deciphering the sequence composition of the genetic material (DNA) within the cell of an organism, using

sequencing data or a result of this process that is represented as a set of contigs or scaffolds.

*Haploid assembly* is a genome assembly in which any locus may be represented 0, 1, or >1 time, but entire chromosomes are only represented 0 or 1 times.

*Diploid assembly* is a genome assembly for which a chromosome assembly is available for both sets of an individual's chromosomes. It is anticipated that a diploid genome assembly is representing the genome of an individual. Therefore, it is not anticipated that alternate loci will be defined for this assembly, although it is possible that unlocalized or unplaced sequences could be part of the assembly.

*Haplotig* is a contig contained duplicated genetic sequences relative to the main chromosome sequences. The presence of haplotigs in assembly can be caused by haplotypes in diploid chromosomes for haploid assembly or using multiple organisms for sequencing library preparation.

*K-mers* – all possible substrings of length k that are contained in the target string, where the target string is a nucleotide sequence, usually whole genome assembly sequence.

## 1.2. Technologies

### 1.2.1. Illumina sequencing-by-synthesis

Illumina sequencing technology is one of the New Generation Sequencing (NGS) methods which also can be called second-generation sequencing. Nowadays Illumina sequencing is the most common sequencing method[3].

The Illumina sequencing workflow is composed of 3 basic steps: sample preparation, cluster generation, and sequencing.

There are several different ways to prepare samples. All preparation methods add adaptors to the ends of the DNA fragments. Through reduced cycle amplification additional motifs are introduced, such as the sequencing binding site, indices, and regions complementary to the flow cell oligos. Hybridization is enabled by the first of the two types of oligos on the surface.

Clustering is a process where each fragment molecule is isothermal amplified. The flow cell is a glass slide with lanes. Each lane is a channel coated with a lawn composed of two types of oligos. This oligo is complementary to the adapter region on one of the fragment strands. A polymerase creates a complement of the hybridized fragment. The double-stranded molecule is denatured and the original template is washed away. The strands are clonally amplified through bridge amplification. In this process the strand folds over and the adapter region hybridizes

to the second type of oligo on the flow cell. Polymerases generate the complementary strand forming a double-stranded bridge. This bridge is denatured, resulting in two single-stranded copies of the molecule that are tethered to the flow cell. The process is then repeated and occurs simultaneously for millions of clusters resulting in clonal amplification of all the fragments. After bridge amplification the reverse strands are cleaved and washed off. Leaving only the forward strands. The 3` ends are blocked to prevent unwanted priming.

Sequencing begins with the extension of the first sequencing primer to produce the first read. With each cycle, fluorescently tagged nucleotides compete for addition to the growing chain. Only one is incorporated based on the sequence of the template. After the addition of each nucleotide the clusters are excited by a light source and a characteristic fluorescent signal is emitted. This proprietary process is called Sequencing-by-Synthesis. The number of cycles determines the length of the read. The emission wavelength, along with the signal intensity, determines the base call. For a given cluster, all identical strands are read simultaneously. All clusters are sequenced in a massively parallel process. After completion of the first read, the read`s product is washed off, and the 3` ends of the template are deprotected. The template now folds over and binds the second oligo on the flow cell. Polymerases extend the second flow cell oligo forming a double-stranded bridge. This double-stranded DNA is then linearized and 3` ends are blocked. The original forward strand is cleaved off and washed away leaving only the reverse strand. Then sequencing of the second read starts in the same manner.

This entire process generates millions of reads representing all the fragments. The length of reads is between 90 and 300 bp according to protocol and sequencing machine that was used. Each DNA fragment is represented by forward and reverse reads[4].

For single-end sequencing the sequencer reads a fragment from only one end to the other generating the sequence of base pairs. In paired-end sequencing, it starts at one read end, finishes forward direction at the specified read length, and then starts another round of reading from the opposite end of the fragment. Sequencing both ends of each read is a more efficient use of the library. Having pairs of reads improves read alignment by improving the ability to resolve chromosomal rearrangements such as insertions, deletions, and inversions. During configuration, the paired-end sequence files must be added to the analysis as a pair of files so that the sample name is assigned to the set and so that they proceed through the subsequent steps of the analysis together.

### 1.2.2. Oxford nanopore sequencing.

ONT is the third-generation sequencing technology that has the capability to produce substantially longer reads than second-generation sequencing. Such an advantage has critical implications for both genome science and the study of biology in general. However, third-generation sequencing data have much higher error rates than previous technologies, which can complicate downstream genome assembly and analysis of the resulting data.

The core of the Nanopore sequencer is a flow cell bearing up to 2048 individually addressable nanopores. Before sequencing, adapters are ligated to both ends of genomic DNA or cDNA fragments. These adapters facilitate strand capture and loading of a processive enzyme at the 5′-end of one strand. The enzyme is required to ensure unidirectional single-nucleotide displacement along the strand at a millisecond time scale. The adapters also concentrate DNA substrates at the membrane surface proximal to the nanopore, boosting the DNA capture rate by several thousand-fold. In addition, the hairpin adapter permits contiguous sequencing of both strands of a duplex molecule by covalently attaching one strand to the other. Upon the capture of a DNA molecule in the nanopore, the enzyme processes along one strand (the 'template read'). After the enzyme passes through the hairpin, this process repeats for the complementary strand (the 'complement read') (citation). As the DNA passes through the pore, the sensor detects changes in ionic current caused by differences in the shifting nucleotide sequences occupying the pore. These ionic current changes are segmented as discrete events that have an associated duration, mean amplitude, and variance. This sequence of events is then interpreted computationally as a sequence of 3–6 nucleotide long k-mers ('words') using graphical models. To convert signals into nucleotide base pairs special software called basecaller is used. Different basecallers use hidden Markov Models (HMM) with a hierarchical Dirichlet process (HDP) or neural networks to classify signals as base pairs. Next-generation sequencing technologies do not directly detect base modifications in native DNA. By contrast, single-molecule sequencing of native DNA and RNA with nanopore technology can detect modifications on individual nucleotides. And the main advantage of this technology is the length of reads up to 100kbp. But also nanopore reads have a high error rate of about 5%, it depends on the protocol that was used for sequencing[5].

### 1.2.3. PacBio sequencing.

Pacific Bioscience sequencing is another third-generation sequencing method. It performs sequencing by synthesis.

The main technology is called Single Molecule Real-Time (SMRT) sequencing and it exploits the natural process of DNA replication[6].

At the first step DNA is isolated from cells and the SMRTbell library is crated by ligating adaptors to double-stranded DNA creating a circular template. Then primer and polymerase are added into the library that is placed in a sequencing machine.

The core of SMRT sequencing is an SMRT cell that contains millions of tiny wells (100nm in height) that called Zero-Mode Waveguides (ZMW). A single molecule of DNA is immobilized in ZMW and DNA-polymerases are on the bottom of ZMW. As polymerase incorporates with labeled nucleotide light is emitted. In this approach nucleotide incorporation is measured in real-time. The camera is installed at the bottom of ZMW taking a video monitoring polymerization reaction in real-time.

SMRT sequencing uses building blocks similar to traditional Sanger or Illumina sequencing that are slightly different. Nucleotides have no blocks at 3` end. This means that once it gets incorporated another base can get incorporated right after. The fluorescent group is attached to the phosphate group of the nucleotide. These phosphate groups with fluorescent groups are removed once a base is incorporated. This means that there is no need in separate chemistry to enable the reaction to proceed.

Pacbio sequencing produces long reads from 1-200 kbp in length. Reads have a smaller error rate than nanopore reads but still larger than the error rate of Illumina reads.

### 1.2.4. Hi-C.

Hi-C is a method that probes the three-dimensional architecture of whole genomes by coupling proximity-based ligation with massively parallel sequencing. Hi-C is a sequencing-based assay originally designed to interrogate the 3D structure of the genome inside a cell nucleus by measuring the contact frequency between all pairs of loci in the genome[7].

The contact frequency between a pair of loci strongly correlates with the one-dimensional distance between them. Hi-C data can provide linkage information across a variety of length scales, spanning tens of megabases. As a result, Hi-C data can be used for genome scaffolding. Shortly after its introduction, Hi-C was used to generate chromosome-scale scaffolds.

Chromatin inside the cell nucleus is packaged into three-dimensional structures that retain a relationship between genomic and physical distance sequences that are closer on the same chromosome are also closer in physical space. Hi-C method exploits this relationship between linkage and proximity to enable whole chromosomes scaffolding and phasing of genomes.

The DNA in the sample is cross-linked in vivo to fix DNA sequences present inside the same cell. Cross-linking traps sequence interactions across the entire genome and between different chromosomes. Cross-linked DNA is fragmented with endonucleases. Fragmented loci are then biotin elated and ligated creating chimeric junctions between adjacent sequences. This process is called proximity ligation. The more often two sequences are joined together the closer these two sequences are in genomic space. Biotinylated junctions are purified and subjected to paired-end sequencing. The proximity ligation reads are then mapped onto a draft assembly. Proximity information is used to assign contigs to chromosomes and order and orient them along chromosome-scale scaffolds. This results in fully scaffolded chromosomes of virtually any size. This process also detects structural variation and corrects assembly misjoins as well as maps the three-dimensional conformation of

chromatin within a population of cells[8].

### 1.2.5. FISH experiments

Fluorescent in situ hybridization (FISH) is a molecular cytogenetic technique that uses fluorescent probes that only bind to parts of the chromosome with a high degree of sequence complementarity. It is used to detect and localize the presence or absence of specific DNA sequences on chromosomes and to assess the localization of these sequences[9].

FISH works by exploiting the ability of one DNA strand to hybridize specifically to another DNA strand and uses small DNA fragments called probes that have a fluorescent label attached to them. The probes are complementary to the specific parts of a chromosome. The chromosomal DNA double-strand is denatured using heating and probes are able to hybridize to their complementary sequence. If a small deletion is present in the region complementary to the probe, the probe will not hybridize. If duplication is present, more of the identical probes can hybridize.

There are different kinds of probes that can be used in the FISH experiment. Probes specific to centromeres are from alpha and satellite III sequences, repetitive regions found in centromeres. Probes specific for telomeres from 300 kb locus at the end of a chromosome. Whole chromosome probes with different color labels that are collections of smaller probes, each of which binds to a different sequence along

the length of a given chromosome. Finally there are locus-specific probes that can be used for gene detection and localization. Also there are probes for RNA sequences that can be used for gene expression or amplification assessment.

Probes have two main types of labels: radioactive isotope labeling and nonradioactive isotope labelings such as biotin and fluorescent dyeing.

There are four steps of FISH: preparation of the fluorescent probes, denaturation of the probe and the target, hybridization of the probe, and the target and detection.

Usually prepared probes are provided by biotech companies. Custom probes can be synthesized based on needs. Denaturation process runs inside special ovens with temperature 46 degrees Celsius and followed by immersion ethanol solution. Hybridization runs using specific protocols and lasts for a few hours. After that hybridized samples are ready for the detection step using fluorescent microscopy.

## 2. MOSQUITOS PROJECT.

### 2.1. Project introduction

Malaria has a devastating global impact on public health and welfare, with the majority of the world's malaria cases occurring in sub-Saharan Africa. Anopheles mosquitoes are exclusive vectors of malaria, with species from the *An. gambiae* complex being the most important African vectors. *An. coluzzii* and *An. arabiensis*, along with *An. gambiae*, are the malaria vectors of the most widespread importance in Sub-Saharan Africa. *An. gambiae* and *An. coluzzii* have been classified as different species[10] because they are genetically distinct[11]. Although they undergo assortative mating[12], reproductive isolation is incomplete: hybrids are viable and fertile, and evidence exists for hybridization in nature, varying over space and time[13], [14]. *An. gambiae* and *An. coluzzii* are often sympatric but differ in the geographical range[15], larval ecology[16], behavior[17], and strategies for surviving the dry season[18]. They are both highly anthropophilic and endophilic, while *An. arabiensis* has a more opportunistic feeding behavior[19], [20]. Moreover, *An. arabiensis* feeds and rests predominantly outdoors replacing *An. gambiae* in some localities with high use of long-lasting insecticide-treated nets (LLINs) and indoor residual spraying (IRS)[20]. The discovery of pervasive genomic introgression between *An. arabiensis* and *An. gambiae* or *An. coluzzii*[21] opened an opportunity to investigate how traits enhancing vectorial capacity can be acquired through an interspecific genetic exchange.

The main goal of this study was to apply the long-read Oxford nanopore sequencing technology and the Hi-C approach, a groundbreaking technology that exploits in vivo chromatin proximity information, to produce de novo chromosome-scale genome assemblies for *An. arabiensis* and *An. coluzzii* species.

This project is affiliated with the Max Alekseyev`s lab from the computational biology institute, George Washington University (GWU) in collaboration with Igor Sharakov`s lab from the Department of Entomology, Virginia Polytechnic Institute.

My task in this project was to provide almost all computational processes of data quality control and statistics gathering, genome assembly, scaffolding, assessing of assemblies. Also I designed some and performed all validation analyses of final assemblies. All computational processes were performed on GWU High-Performance Computing (HPC) clusters.

## 2.2. Materials and methods

### 2.2.1. Main pipeline

I summarize project stages into the pipeline graphical representation of which is depicted in fig 1.

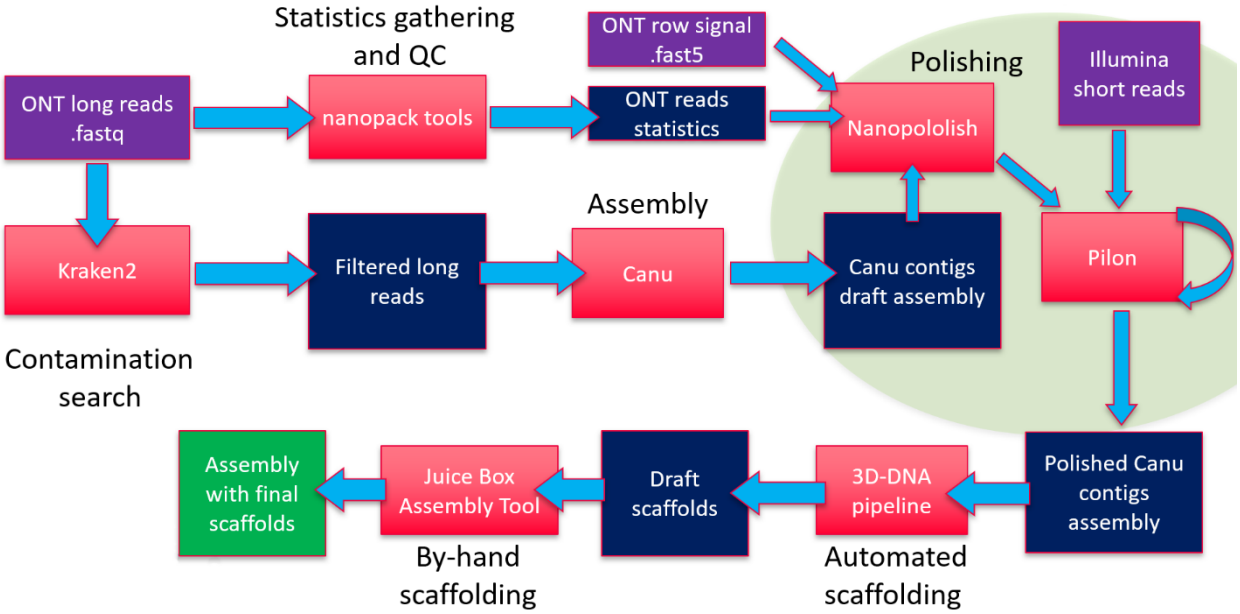


Fig.1. Main pipeline.

Violet bars represent input data, red bars – computational tools used to create the final version of assemblies, blue bars – intermediate data obtained in the assembly process.

This pipeline represents only steps of creating mosquito's chromosome-scale assemblies without a comprehensive comparison of assemblers, polishing strategies, automated scaffolding tools, and additional validation stage for final genome assemblies. All of these things will be described separately.

A closer look at the polishing cycle that was used is depicted in fig. 2.

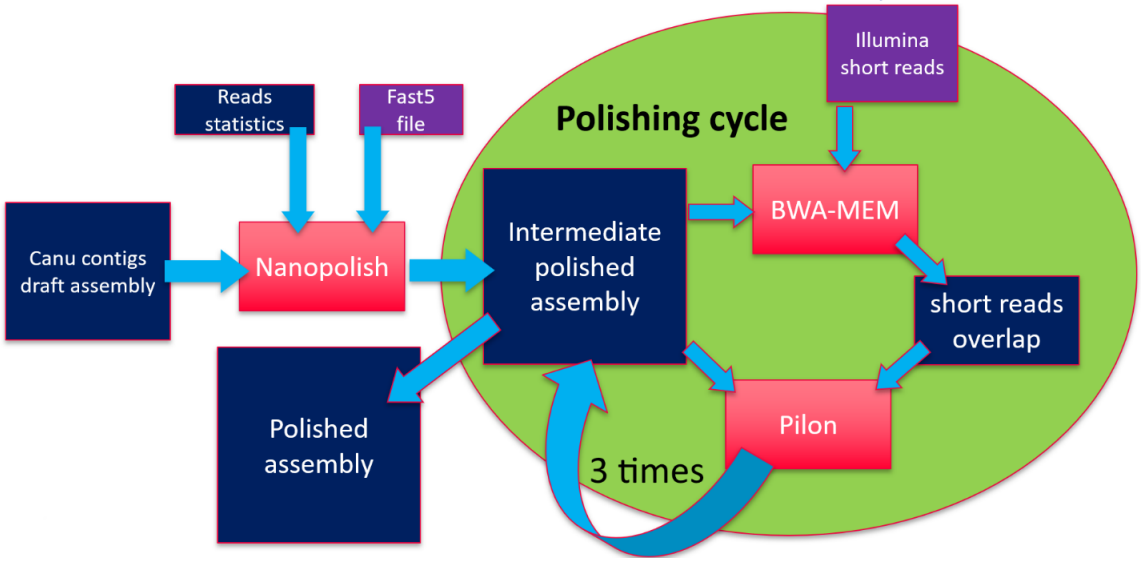


Fig.2. Draft genome assembly polishing cycle.

### 2.2.2. Data description

Three types of data were used:

- Long reads from Oxford Nanopore sequencing
- Short reads from Illumina sequencing
- Hi-C Illumina reads

The main source of genomic data for both anopheline assemblies was Oxford Nanopore long reads. Nanopore sequencing data was obtained from our collaborators. For both species, it consisted of nanopore long reads in .fastq file format, raw nanopore signal data in separated files in .fast5 format, and sequencing summary information in text format.

Quality control, basic analysis, and visualization of the long nanopore reads were done with Nanostat and Nanoplot from Nanopack software package[21]. For *An. coluzzii* genome, these tools reported 3.3M reads of the total length 28Gbp. The read length N50 is 19Kbp, and the read median length and quality are around 4 kbp and 10.3 kbp, respectively. For *An. arabiensis* genome, we have 5.0M reads of the total length 35Gbp. The read length N50 is 21Kbp and the read median length and quality are 2.2Kbp and 10.0, respectively. The detailed statistics reported by these tools can be found in supplementary table 1. Reads length and quality distribution plots are shown for *An.coluzzii* in supplementary fig. 1 for *An.arabiensis* in supplementary fig. 2.

From quality control we can see that *an.coluzzii* reads were initially trimmed by 7 quality but *An.arabiensis* reads were not. We decided not to trim *An.arabiensis* reads because assemblers that were used except miniasm[22] have their own algorithms to work with read quality. We could not establish any effect of quality trimming the main cause of this is different read coverage.

I performed reference alignment to the *An.gembiae* genome to estimate the average sequencing coverage of chromosomes. Alignments were done using minimap2 tool[23]. For *An.coluzzii* genome, the total numbers of aligned and unaligned reads equal 3.3M(99%) and 0.03M(1%), respectively. For *An.arabiensis* genome, the 4.5M(89%) and 0.56M(11%). Minimap2 outputs .sam files were converted to bam sorted and then for each base in reference genome coverage was computed and stored in the table file. The table file then was parsed with my python script and for each chromosome coverage statistics were computed. Also, I visualized chromosome coverage to understand what regions of reference chromosomes are not covered and built a coverage distribution histogram figure 3 in supplementary.

The alignment statistics confirmed the 100x coverage for *An.coluzzii* genome and 114x coverage for *An.arabiensis* genome. Mitochondrial DNA was not excluded from sequencing libraries and as we can see in fig. 3 that it has much more read coverage as chromosomal DNA about 4000x for both species.

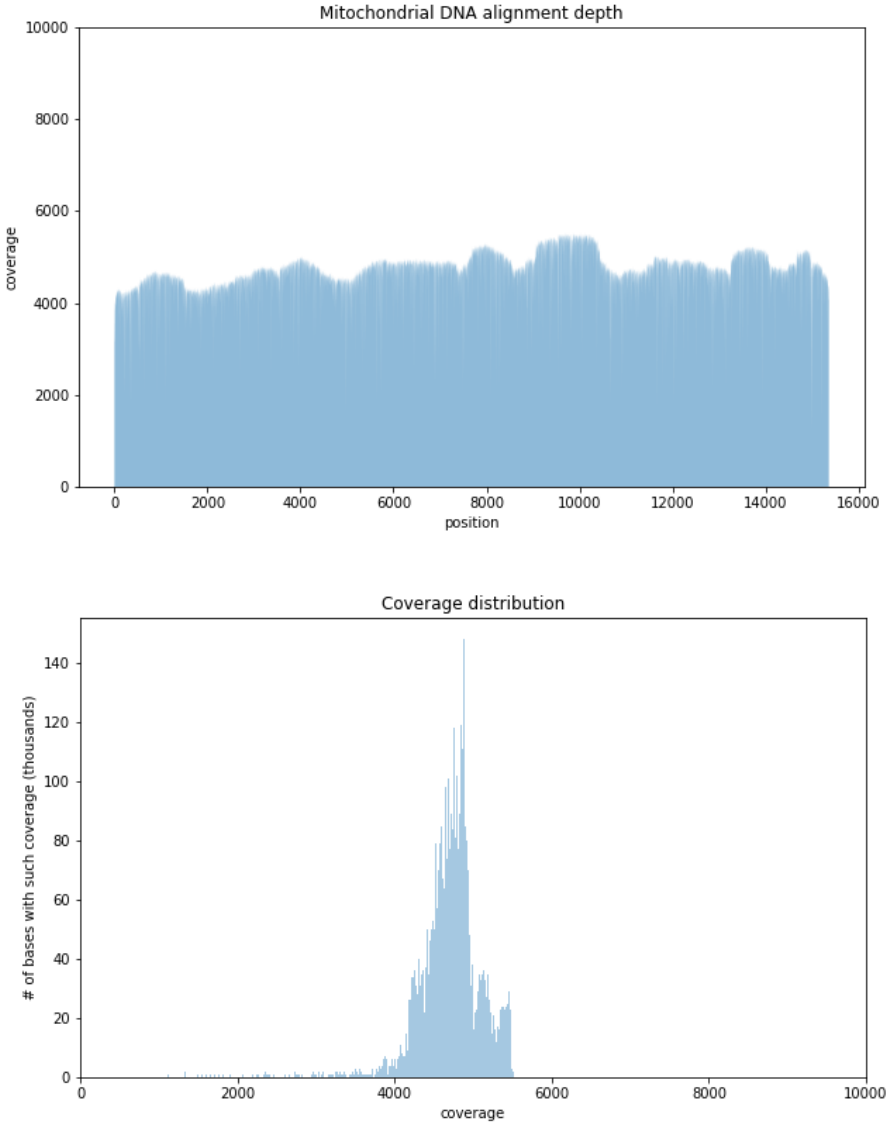


Fig.3. Alignment depth and coverage distribution of mitochondrial DNA of *An.arabienses*

Nanopore reads were filtered out from contamination using Kraken2 tool[24], [25] before the assembly process. I tried to estimate genome sizes using k-mer analysis of nanopore reads but it cannot be estimated due to a large error rate of nanopore sequencing. Contamination filtering and genome size estimation process will be described in the following sections.

For polishing assemblies obtained from nanopore reads, we used the Illumina short paired-end data with NCBI SRX accession number SRX3832577 for

*An.coluzzii* genome, and NCBI SRX accession numbers SRX084275, SRX084275, SRX084275, SRX111457, SRX111457, SRX111457, SRX111457, SRX200218 for *An.arabiensis* genome. The sequence quality control of the short pair-end reads was performed with FastQC (FastQC, RRID:SCR_014583)[26]. FastQC showed that *An.coluzzii* reads have high per-base sequence quality (exceeding 32 on Phred scale) and no adapter contamination. FastQC reported 122.3M reads of length in the range 36--200bp and the total length 22.8Gbp. For *An.arabiensis* genome, FastQC reported 260.6M reads of total length 53.2Gbp and average length 90bp. Based on the FastQC analysis, I filtered reads by the 31 quality and minimum read length, and further trimmed TruSeq adapters from reads using fastp v0.20.0[27]. This resulted in filtering 14% of reads and leaving just 224.8M reads.

Hi-C data for genome scaffolding was obtained from our collaborators. Hi-C libraries preparation protocol for *An.coluzzii* used MboI restriction enzymes, Arima protocol was used for *An.arabiensis*. Hi-C Illumina short paired-end reads quality control was inspected with FastQC. For both mosquito genomes, FastQC showed high per base sequence quality (exceeding 30 on Phred scale) and detected contamination with Illumina TrueSeq adapters in 0.17% reads of *An.coluzzii* and in 1.5% reads of *An.arabiensis*. All such contaminated reads were filtered out in both read sets, resulting in 231.9M and 141.9M reads for *An.coluzzii* and *An.arabiensis*, respectively.

### 2.2.3. Genome size estimation

Approximate genome size is one of the input parameters for assemblers. Thus it must be estimated before the assembly process.

Due to the inability to use nanopore sequencing data for genome size estimation I used Illumina reads which have less error rate.

The approximate genome size can be calculated by counting k-mer frequency of the sequencing data.(citation) The k should be sufficiently large that most of the genome can be distinguished. For most eukaryotic genomes at least 17 are usually used.

To understand the difficulties of such a way of estimation we must look at k-mer distribution of a typical real-world genome. The main issue that is faced in real-world genome sequencing projects is a non-uniform coverage of the genome. This can be attributed to technical and biological variables, for example biased amplification of certain genomic regions during polymerase chain reaction (PCR) and the presence of repetitive sequences in the genome.

The size of k-mers should be large enough allowing the k-mers to map uniquely to the genome (a concept used in designing primer/oligo length for PCR).

In the first step, k-mer frequency is calculated to determine the coverage of the genome achieved during sequencing. There are software tools like Jellyfish[27] that helps in finding the k-mer frequency in sequencing projects. The k-mer frequency follows a Poisson distribution, it can be treated like pseudo-normal around the mean coverage in distribution of k-mer counts.

Then the k-mer frequencies are calculated and plotted to visualize the distribution and to calculate mean coverage, see fig. 4.

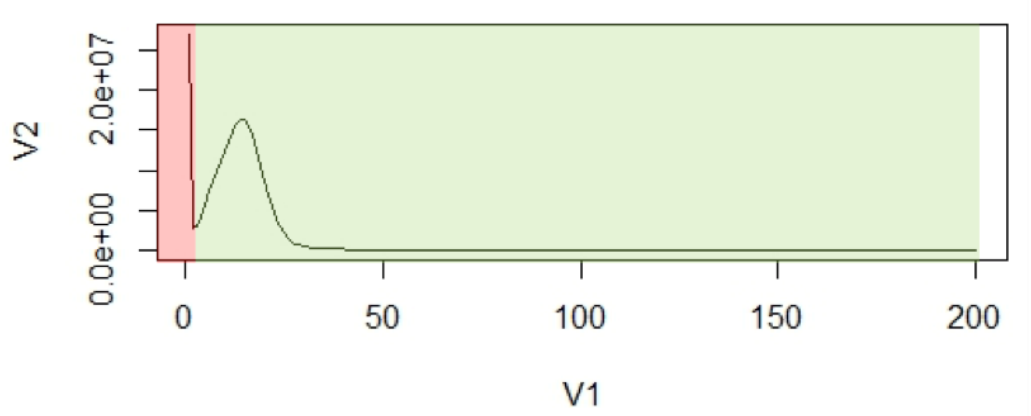


Fig.4. Distribution of k-mer frequencies example. The X-axis (V1), is the frequency or the number of times a given k-mer is observed. The Y-axis (V2), is the total number of k-mers with a given frequency.

The first peak in the red region is a result of rare and random sequencing errors in reads. These values can be trimmed to remove reads with sequencing errors from the estimation process. With the assumption that k-mers are uniquely mapped to the genome, they should be present only once in a genome sequence. Thus, their frequency will reflect the coverage of the genome. Mean coverage is used for calculating. The area under the curve will represent the total number of k-mers.

The genome size estimation will be:

$$N = \frac{total\ number\ of\ kmers}{coverage} = \frac{area\ under\ the\ curve}{mean\ coverage}$$

To estimate single copy region size we must count not a total number of k-mers but only the number of k-mers from this region or in other words we must calculate the area under the bell shape only removing all k-mers with higher frequency.

The whole-genome size of *An.coluzzii* and *An.arabiensis* was estimated by the k-mer analysis for k=19 based on Illumina short pair-end reads. I computed the frequency distribution of 19-mers in all high-quality short reads using jellyfish. For

*An.coluzzii* genome, the peak of the 19-mer distribution was at a depth of 54, and the whole-genome size was estimated as 301.3Mbp. The length of single genome regions was estimated as 204.1Mbp. For Illumina reads of *An.arabiensis* genome, the peak of the 19-mer distribution was at a depth of 88, and the genome size was estimated as 315.6 Mbp. The size of single genome regions was estimated to be 249.4Mbp. K-mere distribution in *An.arabiensis* is shown in fig. 5.

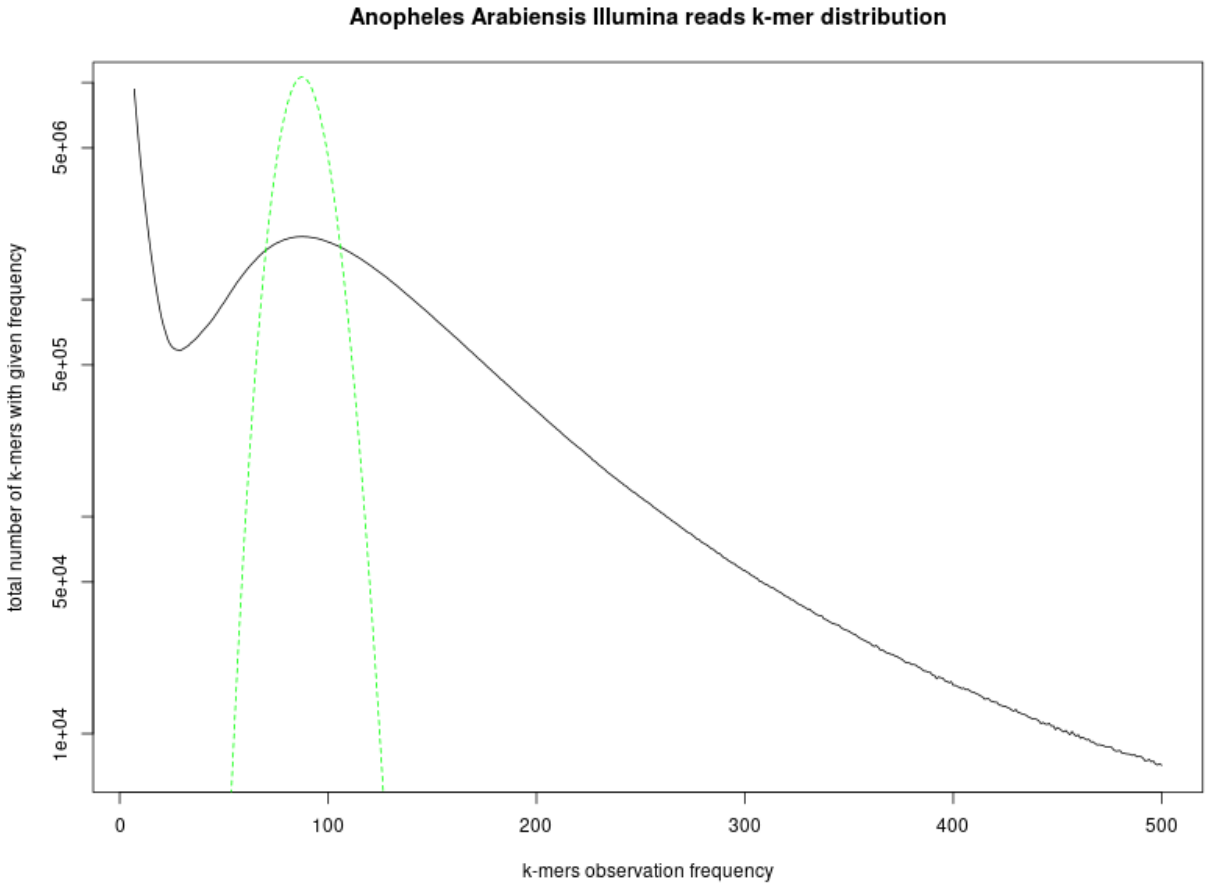


Fig.5. Distribution of k-mers (k=19) in *An.arabiensis* genome. Greenline is theoretical Poisson distribution. The Y-axis is log scaled.

### 2.2.4. Nanopore reads contamination search

Before assembly all contaminated nanopore reads were filtered out using taxonomic sequence classification system Kraken2.

There is another approach to filter contaminants as a step of genome validation after assembly[28]. But when I obtained the first draft assembly of *An.coluzzii* genome that was assembled with Canu[29] I have done a contamination search with Kraken2 and after that checked contaminated contigs with NCBI Blast+[30] over NCBI nucleotide database base. The results showed that there are chimeric contigs that have regions of contaminated DNA inside while other regions belong to mosquito species. To prevent information loss because of filtering contigs with

mosquito DNA we decided to search contaminants on the reads stage. Nanopore reads are quite long for this process and the results of the search are satisfying enough.

Kraken2 is a taxonomic sequence classifier that assigns taxonomic labels to DNA sequences. Kraken2 examines the k-mers within a query sequence and uses the information within those k-mers to query a database. That database maps k-mers to the lowest common ancestor (LCA) of all genomes known to contain a given k-mer.

I performed a contamination analysis with Kraken2 using a custom database, which includes all bacterial, archaeal, protozoan, viral RefSeq genomes, human genome, artificial contaminants, and manually added mosquito genomes from Vectorbase. For *An.coluzzii* genome, Kraken2 identified the origin of most reads (98.40%) as a mosquito. For *An.arabiensis* genome, the mosquito origin was detected for 89.55% of reads, but for 4.8% of reads the origin remained unknown. We considered the remaining reads (which were primarily attributed to the bacterial origin) as contaminated and filtered them out. At the same time, we retained the reads of unknown origin for a downstream analysis because they may represent novel mosquito sequences.

Technically contamination search was divided into three stages. The first stage is Kraken2 database building. There was an issue - to add some genome from .fasta file into database special taxonomy sign must be added into the name of each contig. That was done using a python script. The second stage is assigning a taxonomy and the third is reads filtering according to Kraken2 output. The third stage is an output processing. The output of Kraken2 is a table file with read names and taxonomy ids. I used another python script to parse the output and filter contaminant reads.

**2.2.5. Genome assemblers**

Genome assembly from nanopore sequencing data is an actively developing area. Recently, several tools for assembling nanopore reads were released. We decided to perform a comprehensive comparison of these software on the quality of assembly. To assess assembler performance, we choose several assemblers that were available at the time on nanopore reads for *An.coluzzii* genome, including wtdbg2 v1.1 (WTDBG, RRID:SCR_017225)[31], FLYE v2.4.1 (Flye, RRID:SCR_017016)[32], Miniasm v0.3-r179 (Miniasm, RRID:SCR_015880), and Canu v1.8 (Canu, RRID:SCR_015880). For wtdbg2, an optional polishing step using minimap2 with the same nanopore data was performed per the developer's

recommendation. After Canu assembling we have two assembly variants: contigs and unitigs assemblies.

### 2.2.6. Wtdbg2 assembler

Wtdbg2 is a de novo sequence assembler for long noisy reads produced by PacBio or Oxford Nanopore Technologies. It assembles raw reads without error correction and then builds the consensus from intermediate assembly output.

During assembly, wtdbg2 chops reads into 1024bp segments, merges similar segments into a vertex, and connects vertices based on the segment adjacency on reads. The resulting graph is called the fuzzy Bruijn graph (FBG). It is akin to the De Bruijn graph but permits mismatches/gaps and keeps read paths when collapsing k-mers. The use of FBG distinguishes wtdbg2 from the majority of long-read assemblers.

Wtdbg2 has two key components: an assembler wtdbg2 and a consenser wtpoa-cns. Executable wtdbg2 assembles raw reads and generates the contig layout and edge sequences in a file "prefix.ctg.lay.gz". Executable wtpoa-cns takes this file as input and produces the final consensus in .fasta.

I used an additional polishing step which is based on consensus with the same nanopore data minimap2 alignment. This step is recommended by developers if you don’t use other sources of sequencing data for further polishing.

### 2.2.7. Miniasm assembler

Miniasm is a very fast OLC-based (overlap layout consensus) de novo assembler for noisy long reads. It takes all-vs-all read alignments (typically by minimap2) as input and outputs an assembly graph in the GFA format. Different from mainstream assemblers, miniasm does not have a consensus step. It simply concatenates pieces of read sequences to generate the final unitig sequences. Thus the per-base error rate is similar to the raw input reads.

So far miniasm is in early development stage.

### 2.2.8. Flye assembler

Flye is a de novo assembler for single molecule sequencing reads, such as those produced by PacBio and Oxford Nanopore Technologies. It is designed for a wide range of datasets, from small bacterial projects to large mammalian-scale assemblies.

Flye is using repeat graph as a core data structure. In contrast to de Bruijn graphs (which requires exact k-mer matches), repeat graph is built using approximate sequence matches and can tolerate higher noise of single-molecule sequencing (SMS) reads.

The edges of repeat graph represent genomic sequence, and nodes define the junctions. Each edges is classified into unique or repetitive. The genome traverses the graph (in an unknown way), so as each unique edge appears exactly once in this traversal. Repeat graphs reveal the repeat structure of the genome, which helps to reconstruct an optimal assembly.

### 2.2.9. CANU assembler

CANU is a new single-molecule sequence assembler that improves upon and supersedes the now unsupported Celera Assembler.

CANU is a fork of the Celera Assembler, designed for high-noise single-molecule sequencing (such as the PacBio RS II/Sequel or Oxford Nanopore MinION/GridION).

CANU is a hierarchical assembly pipeline which runs in four steps:

- Detect overlaps in high-noise sequences using MinHash alignment process (MHAP)
- Generate corrected sequence consensus
- Trim corrected sequences
- Assemble trimmed corrected sequences

It can be easily installed from sources but only in single-node mode. To run CANU in grid mode on the HPC cluster it must be configured with SLURM system requirements. That was done by cluster administrator and CANU was installed as SLURM module. Also, it requires a specific JVM version that may be an issue.

### 2.2.10. Assemblies assessment Quast-lg and BUSCO genes.

Assembly assessment and ranking by best quality of assembly was done with two state-of-art tools QUAST-LG[33] and BUSCO genes[34]. For QUAST-LG, we used *An.gambiae* as a reference genome[35].

QUAST-LG is an extension of QUAST intended for evaluating large-scale genome assemblies (up to mammalian-size). It is included in the QUAST package starting from version 5.0.0.

QUAST default pipeline utilizes Minimap2. Also, it uses bedtools[36] for calculating raw and physical read coverage, which is shown in Icarus contig alignment viewer[37].

QUAST -lg output Metrics based only on contigs:

- Number of large contigs (i.e., longer than 500 bp) and total length of them;
- Length of the largest contig;

- N50 (length of a contig, such that all the contigs of at least the same length together cover at least 50% of the assembly) and other similar metrics;
- Numbers of misassembles of different kinds (inversions, relocations, translocations;

QUAST -lg output Metrics based on reference alignment:

- The number and the total length of unaligned contigs.
- Numbers of mismatches and indels, over the assembly and per 100 kb.
- Genome fraction %, assembled part of the reference.
- Duplication ratio, the total number of aligned bases in the assembly divided by the total number of those in the reference. If the assembly contains many contigs that cover the same regions, its duplication ratio will significantly exceed. This occurs due to multiple reasons, including overestimating repeat multiplicities and overlaps between contigs.
- NGA50, a reference-aware version of N50 metric. It is calculated using aligned blocks instead of contigs. Such blocks are obtained after removing unaligned regions and then splitting contigs at misassembly breakpoints. Thus, NGA50 is the length of a block, such that all the blocks of at least the same length together cover at least 50% of the reference.

BUSCO completeness assessments employ sets of Benchmarking Universal Single-Copy Orthologs from OrthoDB[38] to provide quantitative measures of the completeness of genome assemblies, annotated gene sets, and transcriptomes in terms of expected gene content. Genes that make up the BUSCO sets for each major lineage are selected from orthologous groups with genes present as single-copy orthologs in at least 90% of the species. While allowing for rare gene duplications or losses, this establishes an evolutionarily-informed expectation that these genes should be found as single-copy orthologs in any newly-sequenced genome. The evolutionary expectation means that if the BUSCOs cannot be identified in a genome assembly or annotated gene set, it is possible that the sequencing and/or assembly and/or annotation approaches have failed to capture the complete expected gene content.

The assessment tool implements a computational pipeline to identify and classify BUSCO group matches from genome assemblies, annotated gene sets, or transcriptomes, using HMMER hidden Markov models and de novo gene prediction with Augustus. Running the assessment tool requires working installations of

Python, HMMER[39], Blast+, and Augustus[40] (genome assessment only). Genome assembly assessment first identifies candidate regions to be assessed with tBLASTn searches using BUSCO consensus sequences. Gene structures are then predicted using Augustus with BUSCO block profiles. These predicted genes, or all genes from an annotated gene set or transcriptome, are then assessed using HMMER and lineage specific BUSCO profiles to classify matches. The recovered matches are classified as 'complete' if their lengths are within the expectation of the BUSCO profile match lengths. If these are found more than once they are classified as 'duplicated'. The matches that are only partially recovered are classified as 'fragmented', and BUSCO groups for which there are no matches that pass the tests of orthology are classified as 'missing'.

At practice BUSCO is a python script that consequently runs two cycles of tBLASTn -> Augustus -> HMMER pipeline. There are some issues in BUSCO usage: you must use the correct orthologs database and configure Augustus respectively, for this project the diptera database was used and Augustus was configured to use *aedes aegipty* model.

QUAST-LG and BUSCO results for all draft assemblies are presented in supplementary tables 2 and 3. CANU produced the best result, followed by Flye, then wtdbg2, and then miniasm.

**2.2.11. Assemblies assessment auNg metric.**

We used a new metric for assessing assembly contiguity the was proposed by Heng Li[41].

Given a de novo assembly, we often measure the "average" contig length by N50. A longer N50 indicates better contiguity. We can similarly define Nx such that contigs no shorter than Nx cover x% of the assembly. The Nx curve plots Nx as a function of x, where x is ranged from 0 to 100.

In the author's opinion there are two problems with N50. First, N50 is not contiguous. For a good human assembly, contigs of lengths around N50 can differ by several megabases in length. Discarding tiny contigs may lead to a big jump in N50. Relatedly, between two assemblies, a more contiguous assembly might happen to have a smaller N50 just by chance. Second, N50 may not reflect some improvements to the assembly. If we connect two contigs longer than N50 or connect two contigs shorter than N50, N50 is not changed; N50 is only improved if we connect a contig shorter than N50 and a contig longer than N50. If we assembler developers solely target N50, we may be misled by it.

This is an idea about how to overcome the two issues. N50 is a single point on the Nx curve. The entire Nx curve in fact gives us a better sense of contiguity. We can take the area under the curve, abbreviated as "auN", as a measurement of contiguity. The formula to calculate the area is:

$$auN = \sum_i L_i \cdot \frac{L_i}{\sum_j L_j} = \sum_i {L_i}^2 \Big/ \sum_j L_j$$

where $Li_i$ is the length of contig i. Although auN is inspired by the Nx curve, its calculation actually doesn't require to sort contigs by their lengths. It is easier to calculate in practice. The auN metric doesn't have the two aforementioned problems. It is more stable and less affected by big jumps in contig lengths. It considers the entire Nx curve. Connecting two contigs of any lengths will always lead to a longer auN. If we want to summarizes contig contiguity with a single number, auN is a better choice than N50. Similarly we can define auNG and auNGA.

In our assessment of draft assemblies we used NGx metric. I built NGx curves, see fig 6, and calculated auNGs for each assembly, see table 1.

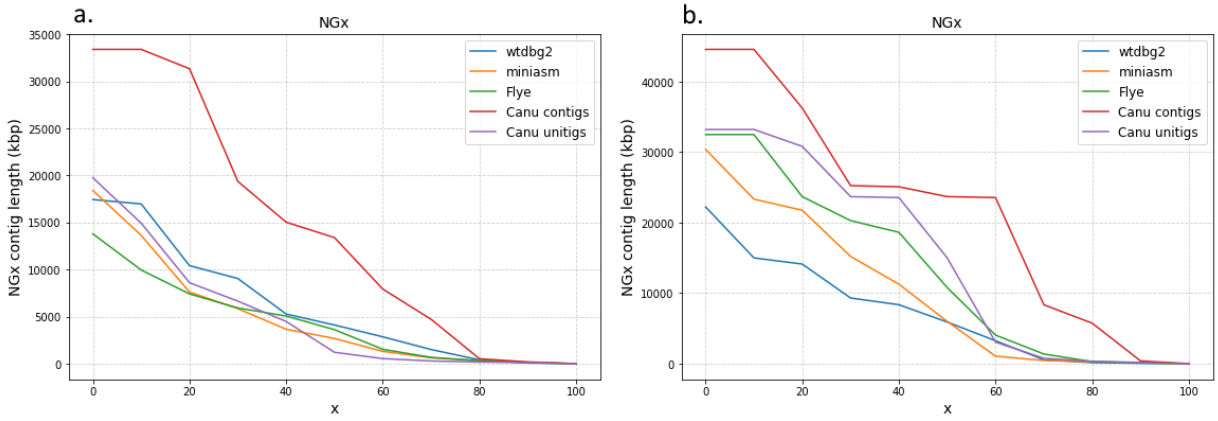


Fig.6. NGx curves for assemblies (a) *An.coluzzii* (b) *An.arabiensis*.

We can see that canu contigs assemblies are the best in case of contiguity for both anopheline species. This fact proves our choice to use canu contigs drafts for further polishing and scaffolding.

| assemblers | *An.coluzzii* | *An.arabiensis* |
|---|---|---|
| | auNG (Mbp) | |
| wtdbg2 | 5.87 | 7.03 |
| miniasm | 4.45 | 8.84 |
| flye | 4.21 | 12.5 |
| canu unitigs | 4.69 | 14.43 |
| canu contigs | 13.57 | 21.99 |

Table.1 AuNG values for assemblies.

### 2.2.12. Genome polishing.

The assembly of long reads from Oxford Nanopore Technologies typically requires resource-intensive polishing to obtain high-quality assemblies. There are two commonly recommended polishing strategies for fixing frequent insertion and deletion errors in assemblies obtained from nanopore reads. The first pipeline is to run Racon (Racon, RRID:SCR_017642)[42] several times using raw nanopore reads and then run Medaka (Medaka, RRID:SCR_005857). The second strategy is to run Nanopolish (Nanopolish, RRID:SCR_016157)[43] using signal-level data measured by the nanopore sequencer. In both strategies, for obtaining better quality assemblies, it is recommended to run Pilon (Pilon, RRID:SCR_014731)[44] several times using short high-quality reads. We tried different strategies on *An.coluzzii* contig assembly obtained by Canu. After each run of a polishing program, we queried the resulting genome for a set of diptera and metazoa conserved single-copy genes (see supplementary table 4). It should be noted that BUSCO single-copy genes usually cover a short portion of a genome and it remains unclear how these polishing tools perform on regions that contain repeats or represent non-coding sequences.

I ran Nanopolish on Canu contig assembly of An. coluzzii genome. Nanopolish corrected 283,935 substitutions, 1.6M insertions, and 51,104 deletions. My collaborator ran 4 rounds of Racon and then Medaka on the *An.arabiensis* assembly. If not stated otherwise, we report BUSCO score for the diptera gene set. The BUSCO score jumped from 77.6% to 93.6% of complete genes after Nanopolish and to 95.1% of complete genes after Racon+Medaka. Despite better BUSCO scores of Racon+Medaka polishing pipeline, we decided to proceed with assembly polished by Nanopolish because there exists an opinion that Nanopolish corrects errors in low-complexity regions better than Racon+Medaka. We also tried to run four rounds of Racon using nanopore raw reads after Nanopolish but that dropped the percentage of complete genes from 93.6% to 88.6%.

Using Canu contig assembly polished by Nanopolish, I ran Pilon several times by utilizing Illumina reads. After the first run of the Pilon, I had 97.9% of complete genes for diptera gene set. After three runs of Pilon, I reached 98.5% of complete genes. We did not run Pilon for the fourth time because the changes were insignificant during the third run. I also tried to run the second time Nanopolish after the first round of Pilon but this dropped the BUSCO's score to 95.9%.

I ran Nanopolish and three rounds of Pilon on Canu contig assembly of *An. arabiensis* genome. After Nanopolish, BUSCO score became equal 94.5% (for Canu contig assembly, it was 83%). Nanopolish corrected 143.5k substitutions, 1.1M

insertions, and 40.7k deletions. After three rounds of Pilon, I obtained the assembly with 98.6% complete genes. For the next scaffolding step, I also polished *An. coluzzii* and *An.arabiensis* unitig assembly obtained by Canu using Nanopolish and three rounds of Pilon.

The final BUSCO scores for *An.coluzzii* and *An.arabiensis* assemblies are similar with BUSCO score for *An. gambiae* PEST genome (i.e., 98.5%, 98.6%, and 98.3% of complete genes, respectively).

**2.2.13. Scaffolding.**

After draft assembly and polishing we have fasta files two for each species. One file for Canu contigs assembly and other for Canu unitigs. Our task was using information from Hi-C experiment reconstruct chromosome scaffolds.

I experimented with different tools to perform scaffolding. This process is not fully automatized now and the last steps must be done by hand. I used three tools HiCExplorer[45], SALSA2[46] and 3D-DNA[47] pipeline. Only last one have an option for by hand editing of results in Juicebox Assembly Tools (JBAT) [48] Hi-C map visualization and editing tool.

HiCExplorer pipeline did not produce any meaningful results. I describe here only SALSA2 and 3D-DNA.

I performed automated scaffolding with SALSA2 on *An.coluzzii* Canu contigs assembly and *An.arabiensis* contigs and unitigs Canu assembly. My scientific advisor performed automated scaffolding using 3D-DNA pipeline for the same assemblies. Then I assessed all scaffolded assemblies with Quast-lg. The results are shown in supplementary table 5.

We also visually inspected Hi-C heat maps of contact information produced by all four runs (see Fig. 7). Hi-C heat map of SALSA's assembly from contigs represents the best Hi-C heat map among considered ones. For example, SALSA 2 completely assembled chromosome X (upper left corner in Fig. 7a) in the right order, and only to inversion is required to fix orientation issues. While 3D-DNA heat maps are smoother (see Fig. 7c, d), it is clear that a lot of repeat sequences resolved during the assembly stage were cut from contigs or unitigs by this tool.

Overall, we conclude that SALSA2 is better suited for assemblies obtained from long reads than 3D-DNA. However, results from both of the tools can be further improved by manual correction of scaffolds performed by inspection of Hi-C heat maps. JBAT is the only tool available for manual correction of genome assemblies. While 3D-DNA are designed to be loadable into JBAT for manual correction, we were unable to convert SALSA2 output to a format that can be loaded

and corrected into JBAT. Therefore, despite better results produced by SALSA2, we decided to proceed with 3D-DNA scaffolds obtained from Canu contig assemblies for *An. coluzzii* and *An. arabiensis* genomes.

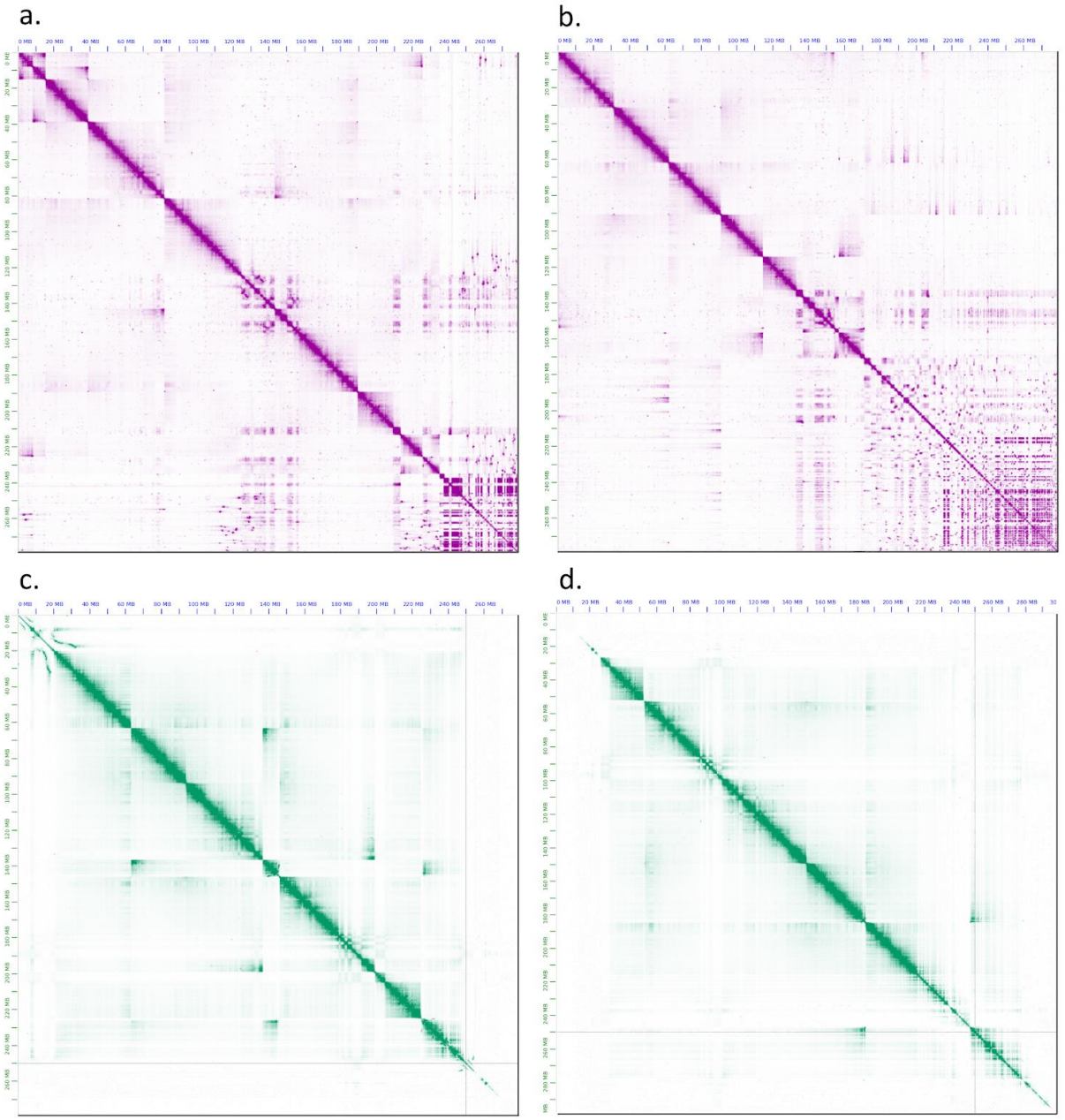


Fig.7. The heat maps of Hi-C contact information for assemblies of *An. arabiensis* genome obtained by (a) SALSA 2 from Canu's contig assembly, (b) SALSA 2 from Canu's unitig assembly, (c) 3D-DNA from from Canu's contig assembly, and (d) 3D-DNA from Canu's unitig assembly.

Special .hic file format is a container for Hi-C maps with different levels of resolution. Hi-C map is a matrix each element of which is a count of aligned Hi-C reads at positions on the genome that corresponds to the number of row and column. In that version of visualization using JBAT each point corresponds to the number of

reads that were aligned to particular coordinates in the genome. Count of reads is presented by the color of a dot. More color intensity means more reads were aligned. White dot means no aligned reads at these coordinates.

The resolution of Hi-C map determines how many base pairs of the genome are represented as one. In .hic file more than one map can be stored with different levels of resolution.

For visualizing .hic files can be used different online and offline tools. JBAT from Aiden Lab was chosen because .hic file was initially designed and standardized in Aiden Lab and it is the only tool available for manual correction of the genome.

**2.2.14. SALSA2.**

SALSA2 is a novel open-source Hi-C scaffolder that does not require an a priori estimate of chromosome number and minimizes errors by scaffolding with the assistance of an assembly graph.

To start the scaffolding, the first step is to map reads to the assembly. BWA mem tool[49] was used for reads mapping. The read mapping generates a .bam file. SALSA requires .bed file as the input. This file was processed using the bamToBed command from the Bedtools package. Also, SALSA requires .bed file to be sorted by the read name, rather than the alignment coordinates. This was done with sorting options in samtools sort command.

SALSA requires contig lengths as an input. File with contig lengths was created using samtools[50] faidx command on contig sequence file.

Hi-C experiments can use different restriction enzymes. SALSA2 uses the restriction sites frequency in contigs to normalize the Hi-C interaction frequency. Restriction site for the enzyme which was used for Hi-C experiment needs to be specified while running SALSA2.

I had contig sequences in polished genome assembly and the alignment .bam file but also wanted to use Hi-C data to correct input assembly errors. A method that allows us to correct some of the errors in the assembly with Hi-C data is already a part SALSA2.

And the last input file for salsa is .gfa assembly graph representing the ambiguous reconstructions. Canu outputs assembly graph files for each assembly and we had two graphs for each species for contig and unnitig assembly.

SALSA2 begins with a draft assembly. Hi-C reads are aligned to the contig sequences, and contigs are optionally split into regions lacking Hi-C coverage. A hybrid scaffold graph is constructed using both ambiguous edges from the assembly graph and edges from the Hi-C reads, scoring edges according to a “best buddy”

scheme. The best buddy weight BB(u, v) is the weight W(u, v) divided by the maximal weight of any edge incident upon nodes u or v, excluding the (u, v) edge itself. Scaffolds are iteratively constructed from this hybrid scaffold graph.

After the scaffolding process I created a Hi-C map files (.hic) using Juicer tool[51]. These maps were visualized using JBAT.

As we can see automatic output of SALSA was not a complete chromosome level genome. It must be corrected by hand. But there were no clear ways to do it by hand correction because of issues with the initial division of assembly into "chromosomal" regions that cannot be treated with JBAT without changing this separation in .hic file and creating special .assembly file needed to JBAT for by-hand scaffolding.

**2.2.15. 3D-DNA.**

3D-DNA is a custom computational pipeline to correct misassembles, anchor, order and orient fragments of DNA based on Hi-C data. Information about usage of 3D-DNA pipeline was obtained from "Genome Assembly Cookbook"[52] written by authors of this pipeline. An overview of the workflow is schematically given in figure 8.

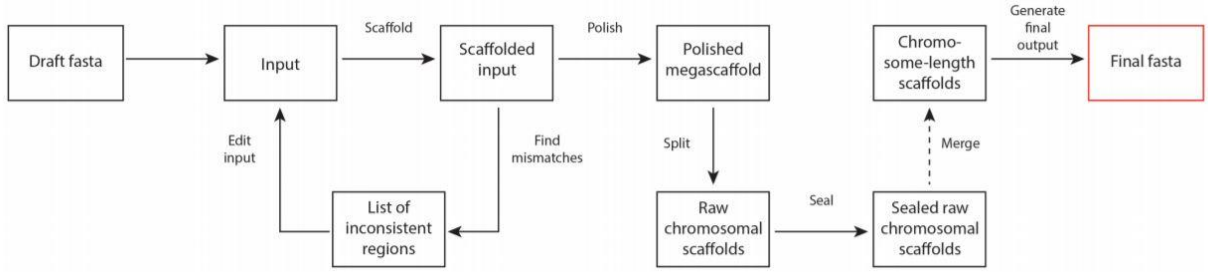


Fig.8. 3D-DNA pipeline from "Genome assembly cookbook"

Next is the short description of the automated scaffolding pipeline of 3D-DNA from the cookbook.

The pipeline starts with setting aside very small scaffolds (threshold side defined by the --input option). The remaining scaffolds are ordered and oriented, and the output is used to detect and correct misjoints in the input scaffolds. The corrected scaffolds again become subjects to ordering and orienting procedure (from scratch). This can be repeated several times (controlled by the --rounds option). Once the iterative scaffolding and misjoin detection are finished, the results are polished by running a coarse-grained misassembly detection and rescaffolding the resulting large pieces. The resulting megascaffold is then split into chromosomes, sealed to examine and restore false-positive edits introduced during misjoin detection.

Resulting Hi-C maps of 3D-DNA pipeline for Canu contigs assemblies are presented in fig. 9.

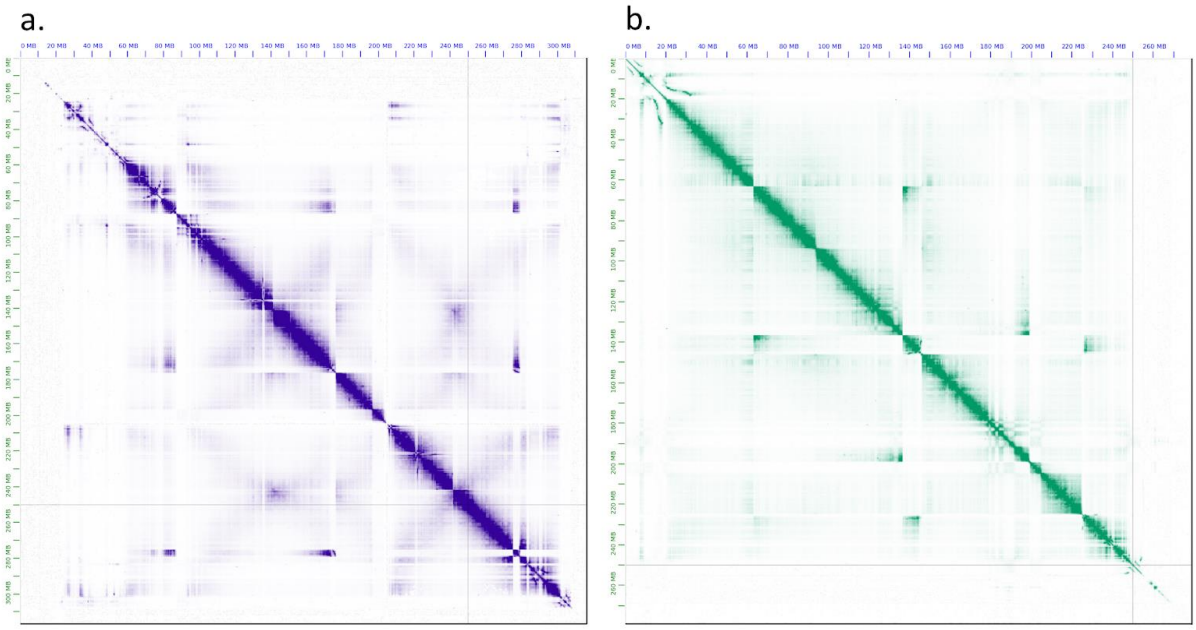


Fig.9. The heat maps of Hi-C contact information for assemblies of (a) An. coluzzii and (b) An. arabiensis genomes obtained by 3D DNA. The pictures were produced by Juicebox tool.

**2.2.16. By-hand scaffolding.**

The heat maps of Hi-C contact information (Fig. 9) indicated a need for manual correction for both assemblies by reordering, changing orientation, splitting contig sequences, and scaffold border allocation. The main goal of manual correction is to obtain chromosome-level scaffolds without assembly errors, haplotype sequences, and assembly artifacts. I also tried to minimize the operations of splitting contigs during the manual correction process. In order to improve our manual correction, I obtained additional information about contigs and scaffolds in the assemblies.

All contigs were classified with PurgeHaplotigs software[53] into primary contigs, haplotigs, and assembly artifacts based on the read-depth analysis.

I aligned contigs from each draft assembly to *An. gambiae* PEST assembly for obtaining information about the distribution of the contigs across the chromosomes. I used results of CQ analysis using Illumina reads from female and male mosquito genomes to detect the presence of contigs from chromosome Y.

Since Hi-C signals must be stronger for adjacent sequence regions, I reordered and changed the orientation of contigs in each assembly to keep Hi-C signal bright along the diagonal. I used the PurgeHaplotigs classification and the fact that haplotig sequences lead to parallel diagonal signals, for moving contigs into debris. I also

added to debris the contigs with low Hi-C signal and contigs that were classified as assembly artifacts. The rest contigs were reordered according to the Hi-C signal. Contigs in debris were further partitioned on several scaffolds. First, I grouped contigs with CQ values more than 0.1 into a separate scaffold that we called Y_unplaced. I was unable to arrange contigs inside Y_unplaced since the Hi-C signal is low for these contigs. Thanks to known tandem repeats, I was able to extract from debris contigs that belong to pericentromeric regions of chromosome X and autosomal pericentromeric regions of chromosome 2 and 3. Since the Hi-C signal is low for these contigs due to the low complexity of these regions, I was unable to place it correctly in scaffolds corresponding to chromosomes. Therefore, we decided to grouped these contigs into X_pericentromeric_DNA scaffold, and Autosomal_pericentromeric_DNA scaffold. Fig. 10 shows heat maps of Hi-C contact information after the manual correction for *An. coluzzii* and *An. arabiensis*.

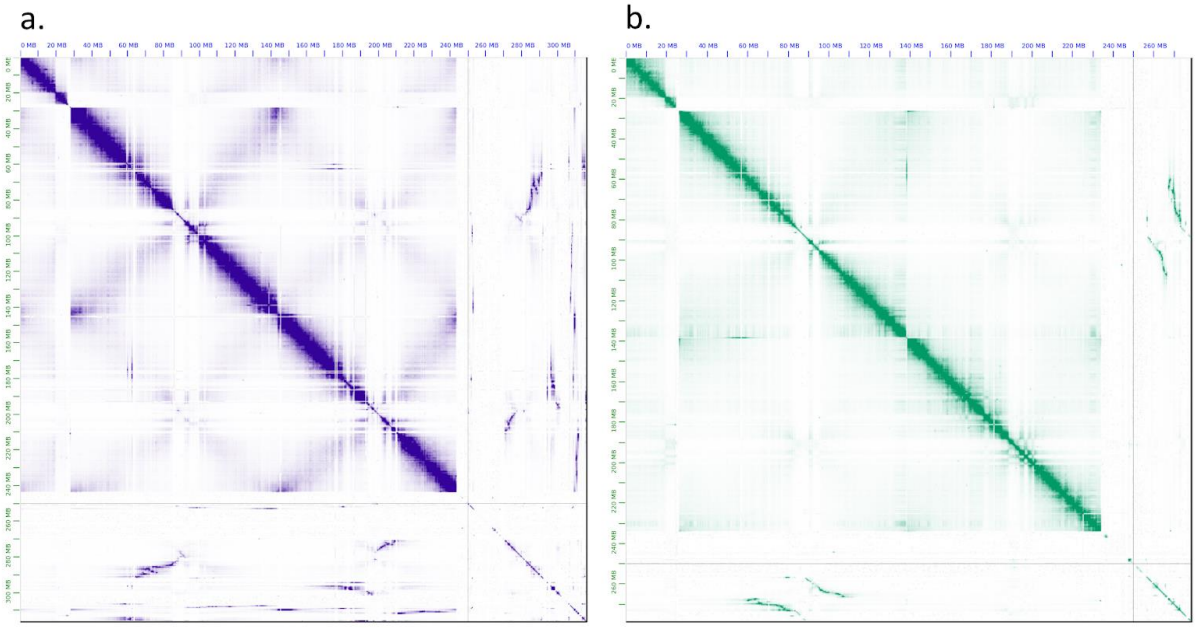


Fig.10. The heat maps of Hi-C contact information for assemblies of (a) *An. coluzzii* and (b) *An. arabiensis* genomes obtained after manual correction of assemblies depicted in Fig. 2. From left to right in each heat map, one can see chromosome X, chromosome 2, chromosome 3, and other scaffolds including contigs in debris. The pictures were produced by Juicebox tool.

We disregarded the remaining contigs in debris and obtained the final assemblies for An. coluzzii and An. arabiensis consisting of 7 scaffolds each:

- scaffold with name X corresponds to chromosome X;
- scaffold with name 2 corresponds to chromosome 2;

- scaffold with name 3 corresponds to chromosome 3;
- scaffold with name Mt corresponds to mitochondrial chromosome;
- scaffold with name X_pericentromeric_DNA correspond to pericentromeric regions of chromosome X;
- scaffold with name Autosomal_pericentromeric_DNA correspond to autosomal pericentromeric regions of chromosome 2 and 3;
- scaffold with name Y_unplaced contains some sequences from Y chromosome.

As a result, we obtained An. coluzzii assembly with BUSCO score equals 98.1% of complete genes for diplera gene set with total length equals 273.57Mbp. An. arabiensis assembly has BUSCO score equals 98.1% of complete genes for diplera gene set and the total length equals 257.22Mbp.

**2.2.17. Purge haplotigs.**

Purge Haplotigs is a computational pipeline to deal with diploid assemblies e.g. FALCON. But in case of presence in haploid assembly large number of haplotigs it can help to mark them using read coverage. The pipeline includes three steps.

Purge Haplotigs uses the alignment of long reads to the curated assembly. It was done with minimap2. The first step is creating a read-depth distribution. Distribution plots for both species are shown in figure 11.

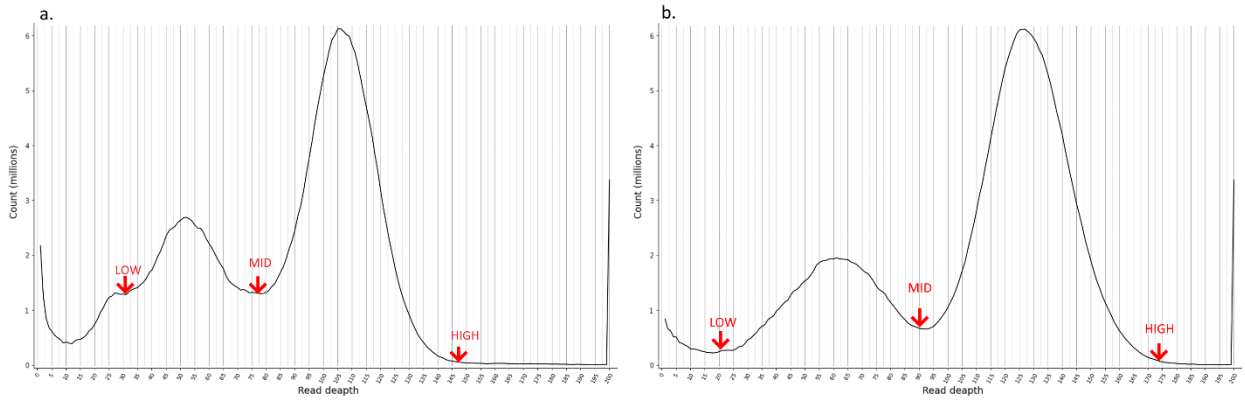


Fig.11. Purge Haplotigs read-depth distributions. (a) *An.coluzzii (*b) *An.arabiensis* draft canu contigs assemblies.

The X-axis is read-deapth or coverage Y-axis is nucleotide count. We can see that distribution is bimodal. This is because the real genome is diploid and there are haplotigs in assembly (not only from diploidy but from different individuals in sequencing library too). The first read-depth peak results from the duplicated regions that correspond to the 'haploid' level of coverage (0.5x). Contigs with this coverage are our suspects to be haplotigs. The second read-depth peak results from regions

that are haplotype-fused that corresponds to 'diploid' level of coverage (1x). Contigs with this level we will mark as primary. Contigs with an inadequate mix of coverage levels we will mark as assembly artifacts. At this step we must choose low, mid, and high cutoffs for coverage. Cutoffs for *an.coluzzii* are: 30, 78, and 132, for *an.arabiensis*: 25, 93, and 160.

The second step is producing a contig coverage stats .csv file with suspect contigs flagged for further analysis or removal.

The third step is the iterative purging pipeline. The script will automatically run a windowed coverage analysis and assess which contigs to reassign and which to keep.

At the end of the Purge Haplotigs process we had three .fasta files: primary contigs, haplotigs, and assembly artifacts. I used this information in the by-hand scaffolding process.

**2.2.18. Validation.**

To validate and compare resulting assemblies to existing mosquito assemblies, I generated whole-genome pairwise alignments between the available assemblies, analyzed all assemblies in the presence of known tandem repeats and transposable elements, and mapped genes from *An. gambiae* PEST to our assemblies.

**2.2.19. Validation. Rearrangements and dot-plots.**

To validate assembly completeness and contigs ordering in chromosomal scaffolds *An.coluzzi* and *An.arabiensis* assemblies were pairwise alligned to *an.gambiae* PEST strain assembly, which is the most complete chromosome level anopheline genome assembly, and to each other. I performed pairwise alignment dot-plots using the D-GENIES tool[54]. Alignments were done with the minimap2 algorithm. To see all chromosome-to chromosome alignments see fig. 4 in the supplement.

The first alignment of *An.coluzzii* to *An.gambiae* is shown in fig. 12.

Distribution of aligned genome regions by identity quartiles and not matched is (not matched, <25%, 25-50%, 50-75%, >75%) is 5.6%, 0.04%, 1.29%, 43.28%, 49.79%.

There are no interchromosomal rearrangements but in the X chromosome and in the 2L chromosomal arm intrachromosomal rearrangements are present. Gaps in dot-plot show that a.coluzzii assembly has more genomic information in pericentromeric regions than a.gambiae assembly.

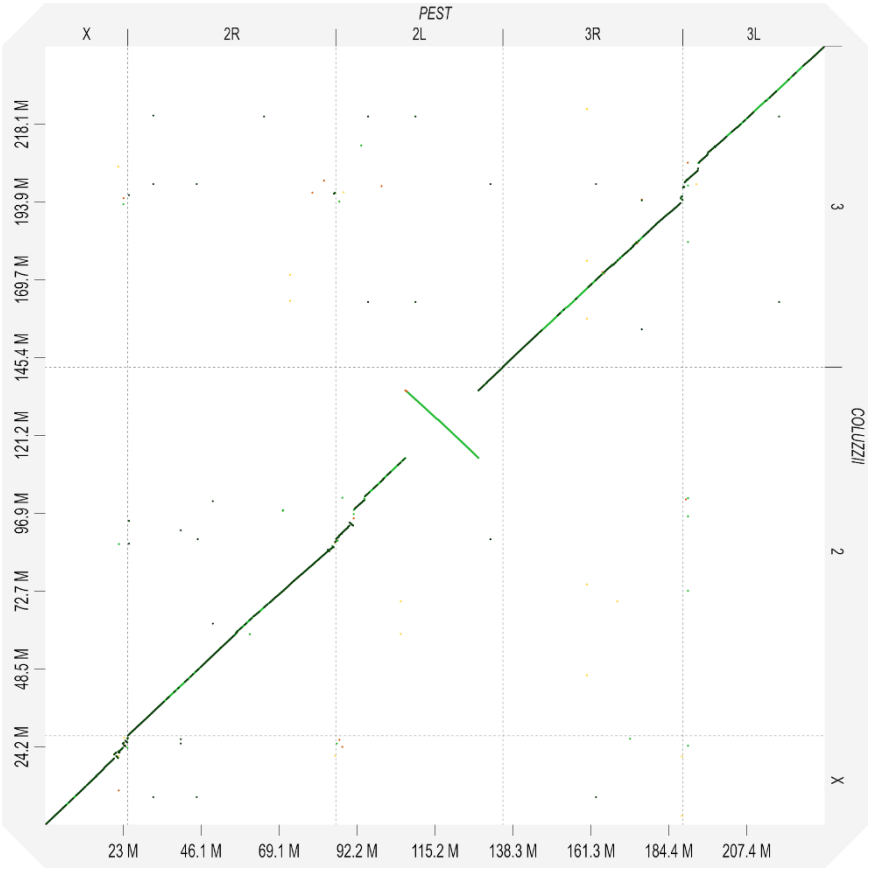


Fig.12. Pairwise alignment dot-plot for an.coluzzii to an.gambiae.

There are multiple rearranged regions at the centromeric end of the X chromosome. Pericentromeric region of a.coluzzii is more complete. Approximate intervals of rearrangement breakpoints for the X chromosome according to an.gambiae PEST X scaffold coordinates: 20.347Mbp - 20.357Mbp; 20.954Mbp - 20.967Mbp; 21.555Mbp - 21.567Mbp; 21.977Mbp - 21.987Mbp; 22.906Mbp - 22.936Mbp; 23.200Mbp - 23.210Mbp; 23.647Mbp - 23.684Mbp; 24.272Mbp - 24.283Mbp.

Two small rearrangements are in the pericentromeric region of 2R arm, one in the pericentromeric region of 2L arm, and one big rearrangement in 2L arm.

Pericentromeric region of the 2nd chromosome is more complete in an.coluzzii assembly.

Approximate intervals of rearrangement breakpoints for the 2R arm according to an.gambiae PEST 2R scaffold coordinates: 59.084Mbp - 59.115Mbp; 59.611Mbp - 59.679Mbp; 60.462Mbp - 60.472Mbp; 60.912Mbp - 61.285Mbp

Approximate intervals of rearrangement breakpoints for the 2L arm according to an.gambiae PEST 2L arm coordinates: 3.983Mbp - 4.041Mbp; 5.069Mbp - 5.214Mbp; 20.524Mbp - 20.528Mbp; 42.165Mbp - 42.166Mbp

Multiple small rearrangements are in the pericentromeric region of the 3rd chromosome. Pericentromeric region of the 3rd chromosome is more complete in an.coluzzii assembly.

The second alignment is *An.arabiensis* to *An.gambiae* (fig. 13).

Distribution of aligned genome regions by identity quartiles and not matched is (not matched, <25%, 25-50%, 50-75%, >75%) is 8.88%, 0.38%, 9.22%, 62.31%, 19.20%. We know that *An.arabiensis* is more distant from *An.gambiae* as *An.coluzzii*. And pairwise alignment proves this.

There are no interchromosomal rearrangements.

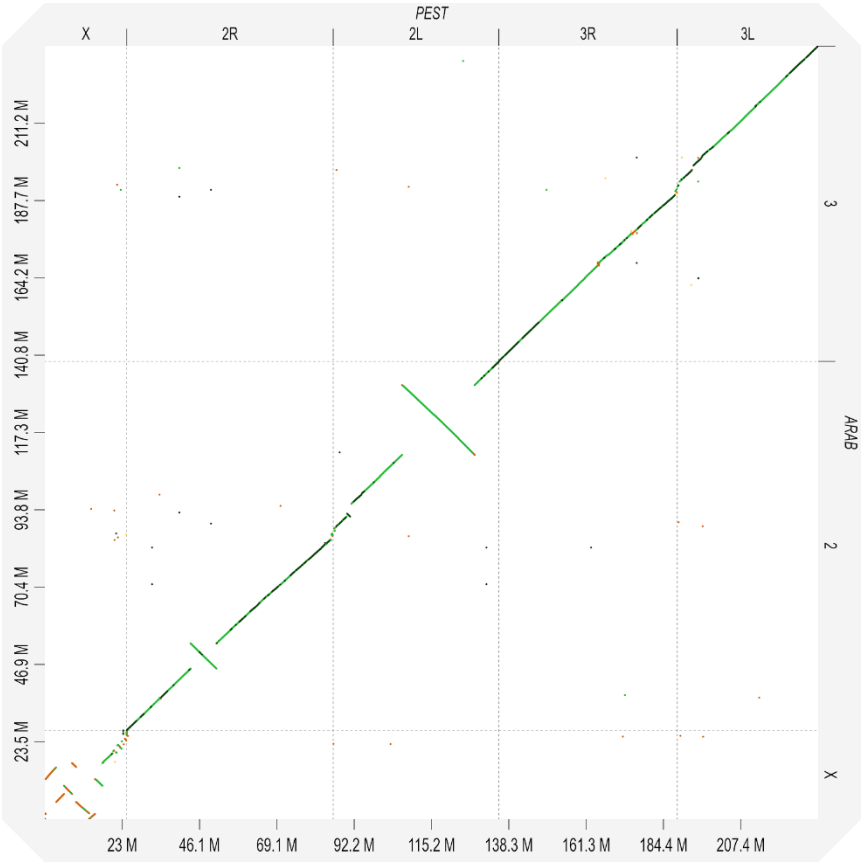


Fig.13. Pairwise alignment dot-plot for *An.arabiensis* to *An.gambiae*.

Multiple rearrangements are present in the X chromosome. Two big rearrangements are in 2R and 2L arms of the 2nd chromosome. Gaps in dot-plot show that a.arabiensis assembly has more genomic information in pericentromeric regions than a.gambiae assembly.

Almost the whole X chromosome consists of rearranged regions.

Approximate intervals of rearrangement breakpoints for the X chromosome according to an.gambiae PEST X scaffold coordinates: 0.022Mbp - 0.037Mbp; 0.238Mbp - 0.241Mbp; 3.281Mbp - 3.356Mbp; 5.677Mbp - 5.678Mbp; 8.104Mbp - 8.118Mbp; 9.282Mbp - 9.289Mbp; 13.160Mbp - 13.162Mbp; 14.903Mbp - 14.912Mbp; 17.157Mbp - 17.165Mbp.

In both arms of the 2nd chromosome, there are one big rearrangement and small rearrangements in the pericentromeric region. Coordinates for 2R:18.995Mbp - 19.032Mbp; 26.748Mbp - 26.751Mbp; 59.083Mbp - 59.119Mbp; 59.561Mbp - 59.682Mbp. Coordinates for 2L: 3.997Mbp - 4.090Mbp; 5.065Mbp - 5.455Mbp; 20.524Mbp - 20.528Mbp; 42.071Mbp - 42.166Mbp.

There are no rearrangements in 3d chromosome. Gaps show that a.arabiensis assembly has a more complete pericentromeric region.

To validate our observations pairwise alignment of *An.coluzzii* and *An.arabiensis* was built. We can see that all rearrangement are proven, see fig 14.

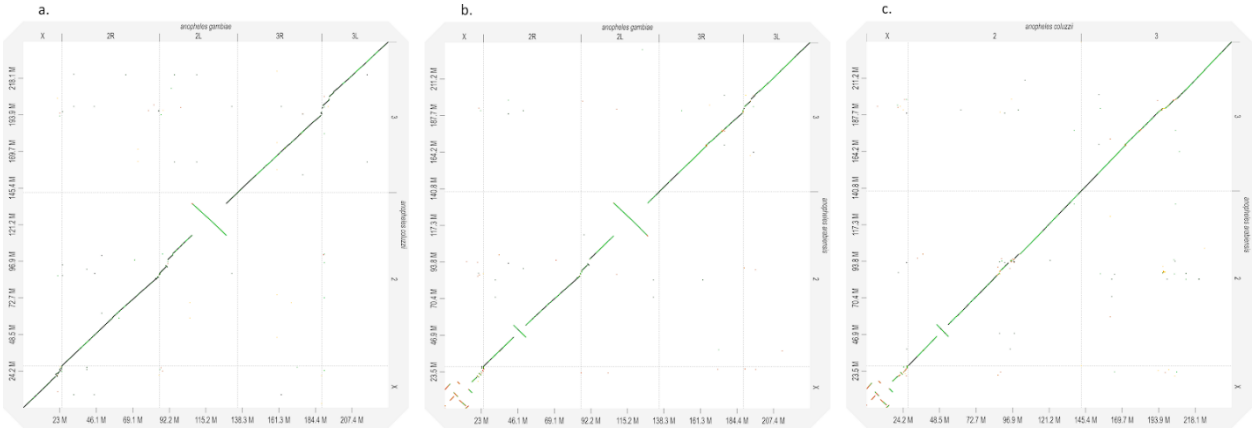


Fig.14. Three whole-genome pairwise alignments dot-plots a. *An.coluzzii* to *An.gambiae*, (b) *An.arabiensis* to *An.gambiae*, (c) *An.arabiensis* to *An.coluzzii*.

Some rearrangements that were founded are well known from biological studies especially big rearrangements in the 2nd chromosome and X chromosome for *An.arabiensis*[55].

### 2.2.20. Validation genes from the reference assembly

I mapped genes from An Gambiae using NCBI Blast+. 1D annotation tracks for Hi-C map in JBAT were created for genes from each PEST chromosome separately.

Overall, Blast mapped 13036 (99.84%) and 13031 (99.8%) genes from 13,057 *An. Gambiae's* genes for *An.coluzzii* and *An.arabiensis* assemblies, respectively. Moreover, we have 9442 (72.31%) and 8971 (68.71%) genes for *An.coluzzii* and *An.arabiensis* assemblies, respectively, mapped with zero e-value and alignment length bigger than 90% of gene length. Table 2 shows numbers of mapped genes among scaffolds with zero e-value and alignment length bigger than 90% and less than 110% of gene length. The lower number of aligned genes to scaffold corresponding to *An.arabiensis* chromosome X may potentially indicate a higher

divergence from *An. gambiae'*. Gene alignments further support evidence that obtained assemblies are high-quality ones.

| | | | An.Gambiae | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | aligned from | X | 2 | 3 | Mt | Y_unplaced | UNKN |
| | aligned to | total | 1063 | 6603 | 4897 | 13 | 2 | 479 |
| An.coluzzii | X | 584 | 564 | 5 | 4 | | | 11 |
| | 2 | 5055 | 9 | 4862 | 4 | | | 180 |
| | 3 | 3785 | 6 | 10 | 3566 | | | 203 |
| | Mt | 13 | | | | 13 | | |
| | Y_u | 4 | | 1 | 1 | | 2 | |
| | Autosomal_pc | 0 | | | | | | |
| | X_pc | 1 | | | 1 | | | |

| | | | An.Gambiae | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | aligned from | X | 2 | 3 | Mt | Y_unplaced | UNKN |
| | aligned to | total | 1063 | 6603 | 4897 | 13 | 2 | 479 |
| An.arabiensis | X | 397 | 378 | 3 | 3 | | | 13 |
| | 2 | 4930 | 11 | 4734 | 5 | | | 180 |
| | 3 | 3610 | 2 | 8 | 3419 | | | 181 |
| | Mt | 13 | | | | 13 | | |
| | Y_u | 10 | | 3 | | | 1 | 6 |
| | Autosomal_pc | 10 | | 3 | 6 | | | 1 |
| | X_pc | 1 | | | | | 1 | |

Table.2 The statistic of aligned genes from An. gambiae assembly to *An. coluzzii* and *An. arabiensis* assemblies.

### 2.2.21. Validation with marker sequences.

To validate our assemblies I performed a search of marker sequences in assemblies. I created blast databases for each genome and used NCBI Blast+ to map sequences. After processing blast outputs were transformed into bed files that can be visualized in genome browsers. For visualization purposes, I used IGV v.2.8.0[56]. Visualizing in the form of 1D annotation tracks for JBAT maps is not correct because our view resolution is limited by 1000bp in one dot but our sequences of interest have a length less than 1000bp. But tracks for JBAT were created to assess the correctness of arranging and ordering of repeats.

Ag93 repeat is a known marker[57] for the beginning of pericentromeric regions in autosomal chromosomes. We use it to assess the completeness of chromosomal arms. Information about Ag93 location was derived from the FISH experiment see fig. 15.

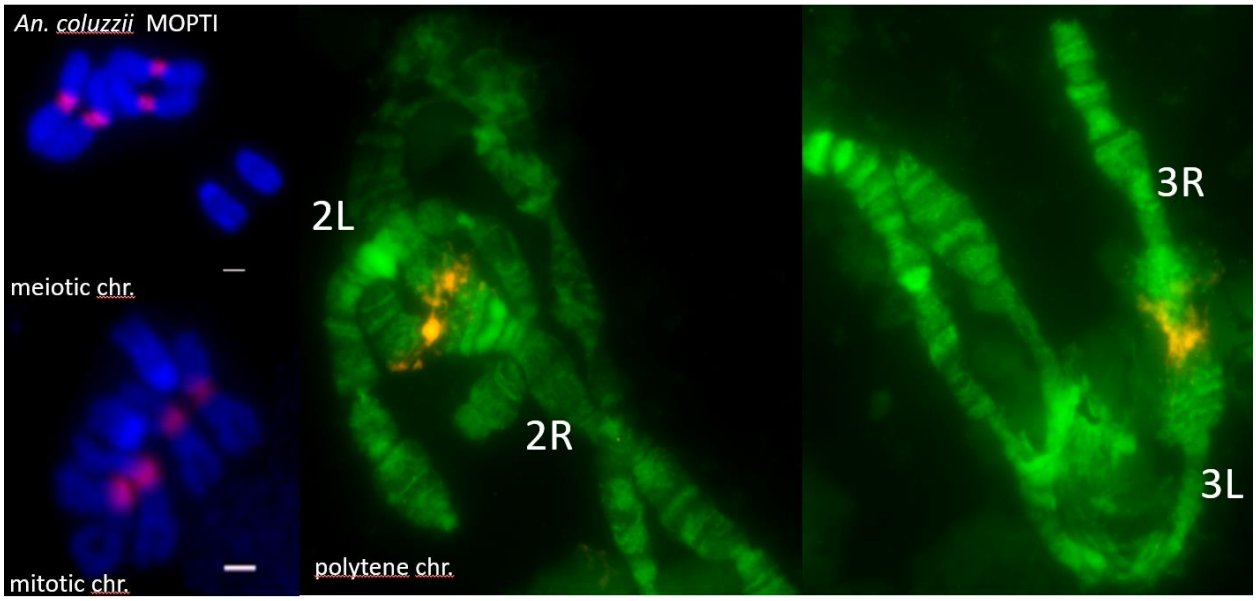


Fig.15. AG93 highlighted with color in meiotic, mitotic and polytene autosomal chromosomes in *An.coluzzii* MOPTY strain.

We can see that Ag93 regions surround centromeres of 2nd and 3rd chromosomes. After mapping Ag93 sequence to our assemblies I find that each autosomal chromosome arm is ended with Ag93 clusters. And our project supervisor decided to merge chromosomal arms into full chromosome with gap between because that result says that pericentromeric regions are almost complete. Contigs from remaining centromeric regions were determined using another marker sequences and combined into not ordered autosomal pericentromeric scaffold.

Another marker of autosomal pericentromeric regions is Ag53C[58] and its junction with Tsessebe III transposal element. According to another FISH experiment these repeat clusters and junctions are located in-between Ag93 repeat regions see fig 16.

A lot of Ag53C repeat clusters were found in both species in contigs that were defined as debris in scaffolding process. We cannot properly arrange them because of lack of Hi-C signal. Contigs from debris contained Ag53C and Ag93 repeats were combined into unordered autosomal_pericentromeric_DNA scaffold.

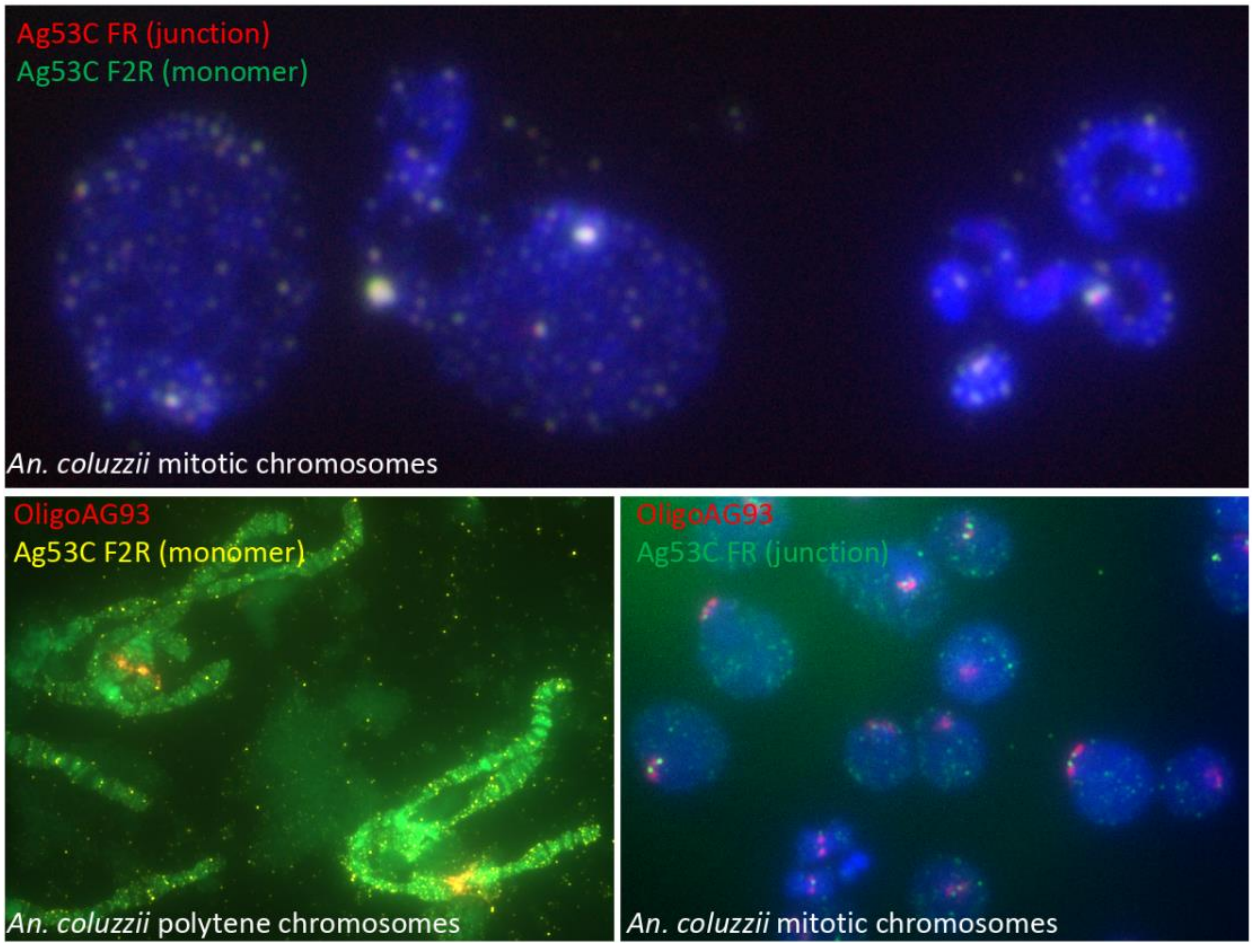


Fig.16. Ag53C regions highlighted in pericentromeric regions of an.coluzzii autosomal chromosomes.

AgX367 and AgY477 repeats are markers of pericentromeric regions for X and Y chromosomes respectively. We know that AgX367 and AgY477 are very similar to each other, see fig 17.

```
Y477 TTTGAGCATGTGTTTAAAGGGTAATATGACCCATAAAGGTTAAGCTCAGAGCTTAGGAACATATAGTAAATTGCCTCTAAAGTTGAAGGTTTTGTGGAAAGTCTT
X367 .....----------

Y477 CAAATGTGCTTCGGGGGACTATGACCCAGTATGAAACTTTTTCATCGCCAAGATCCTTGTTATTGTGTCCCAGGGCTTTGATTTGCTTATTCATGAAGCCCCAAT
X367 ----------T.....G.....G.....

Y477 GACAAAAGAACGATAATGAATGACCTTGCATTTTCGTCAAACATTCAAGCATGGCCATGGGGACGGATGAGAAAGCTCAAGTGATGTAGTTGGATGTTCCTTCAA
X367 ....T..A.....G.....AT.....C.....A.....G...A.....

Y477 ATGGCCATAACTTCGTAACCATGTGTCCTAGCGTGATGATTGAGACAGTTTTGGGAAGGTATTGAAGTGGTCTACAAGATCTGCGCATAGGTTTAAAGATCAGAA
X367 .....G.....T.....A.....A.....T.....TA..A.....A.....A.T

Y477 TCACTGGTAGCCTAGTAAATGGCCTCTGAATGCATTGTACTCGGGAAAACCTGTCAA
X367 ...T.TT...T.....C.....
```

Fig.17. Alignment of AgY477 and AgX367 consensus monomer sequences.

AgX367 is almost a subsequence of AgY477. Also from this paper we can assume the position of these repeats in the pericentromeric region of sex chromosomes according to FISH experiment results, see fig. 18.

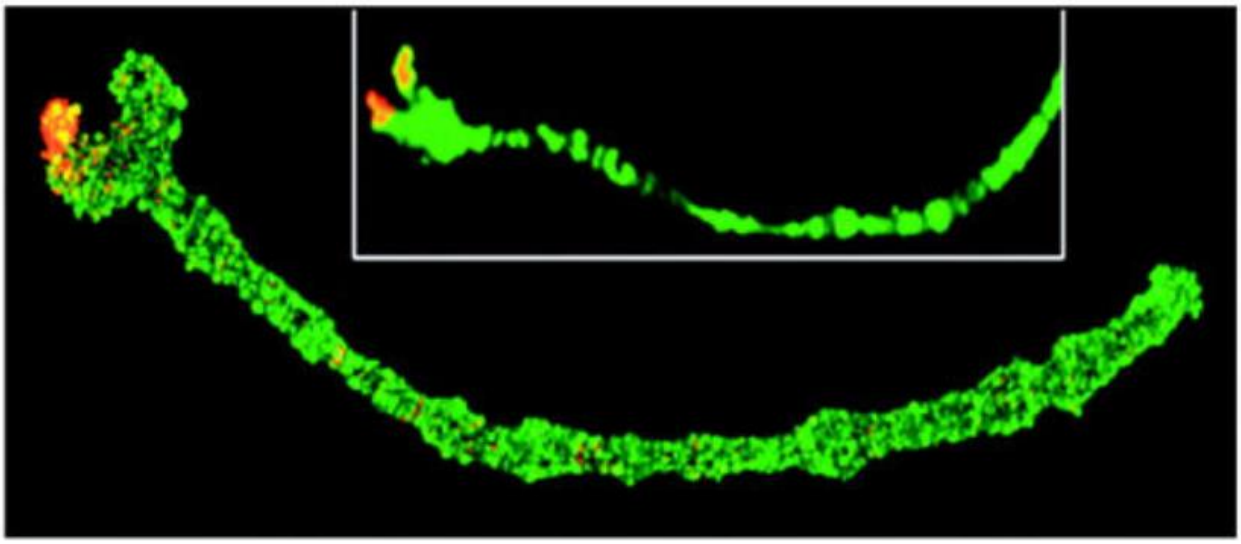

Fig.18. FISH of the AgY477 probes to A. gambiae ovarian nurse cell polytene chromosomes.

In *An.coluzzii* assembly all spots of AgX367 were found in unscaffolded contigs. These contigs were combined in unordered X_pericentromeric_DNA scaffold. In *An.arabiensis* there are spots of AgX367 in the X chromosome with another marker gene 18SrDNA. Thus X chromosome is more complete in *An.arabiensis* assembly. Other AgX367 contained contigs were combined in separate unordered scaffold like in *An.coluzzii* assembly.

We assumed that both assemblies must contain DNA sequences derived from Y chromosomes because of the presence of mails in the sequencing library. In *An.coluzzii* assembly these contigs were found. Hi-C signal for such contigs is too small to arrange and order them into the chromosomal scaffold. To determine what contig belongs to Y chromosome I used not only AgY477 sequence alignment but other sequences such as unique for Y chromosome Zanzibar gene. But the main source of information was CQ analysis data obtained from collaborators.

For both species contigs that were determined as Y chromosomal were combined into unordered Y_unplaced scaffold.

**2.2.22. Validation CQ analysis.**

Chromosome quotient (CQ) is a novel approach to systematically discover Y chromosome genes[59]. In the CQ method, genomic DNA from males and females is sequenced independently and aligned to candidate reference sequences. The female to male ratio of the number of alignments to a reference sequence, a parameter called the chromosome quotient (CQ), is used to determine whether the sequence is Y-linked.

I obtained tables with CQ values from the CQ analysis experiment for each species from our collaborators. CQ values were calculated for each 1kbp region of

genomes. I defined cutoffs for CQ values 0.3 and 0.1 as prescribed in original paper and with this information marked contigs that belong to the Y chromosome according to experimental results. During the scaffolding process these contigs were combined into unordered Y_unplaced scaffolds for each assembly.

**2.3. Project results.**

We ended with two assembled and scaffolded genomes for two mosquitos species *An.coluzzii* and *An.arabiensis*. Genomes were given to the NCBI to run annotation pipeline and publish. After annotation genomes will be in public access in NCBI open genomes database and in main mosquito vectors study project Vectorebase.

We performed a comprehensive comparison of modern assemblers for haploid assembly of eukaryotic species using long noisy nanopore reads. We assessed the quality of assemblies with different metrics and showed the results.

We performed a comparison of modern automated scaffolders. We performed by-hand scaffolding using different sources of information to prove decisions about contigs ordering, rearranging, and combining into chromosomal scaffolds.

We validate our final assemblies using biological data from the FISH experiment, CQ analysis, and bioinformatic data derived from previous assembled species.

Finally, we obtained 232.78 Mbp and 244.22 Mbp haploid male assemblies for *An.arabiensis* and *An.coluzzii*, respectively. Our assemblies contain pericentromeric heterochromatin sequences and sequences of the Y chromosomes. The comparison of these new assemblies with the existing assemblies for these species demonstrated that we obtained reference-quality genomes for these mosquito species.

## 3. BARNCLES PROJECT.

During my research work in the master program I was involved in another project with chromosome-scale genome assemblies. The task of the project was to assemble and assess two barnacle species de novo from long pacbio reads using Illumina data for polishing. In this chapter I want to describe some methods that differ or absent in mosquitos project.

### 3.1. Project introduction.

This project is about the de-novo assembling of two barnacles species: *Amphibalanus Amphitrite* and *Pollicipes pollicipes* also known as gooseneck barnacle. Both species have no close reference genome that makes an assessment of assemblies much harder. For gooseneck barnacle genome assembly was already done by the Dovetail inc. and our task was to polish and validate it. For *a.amphitrite* we had only sequencing data, and the task was to perform full pipeline of genome assembly, polishing, and validation.

This project is also affiliated with the computational biology institute, George Washington University (GWU)

### 3.2. Materials and methods.

#### 3.2.1. Data description.

The main data source for both barnacle species was pacbio sequencing output. Files were provided from collaborators in separate files in .bam format. I merged files for each species and gathered read statistics using nanopack tools package. Main reads statistics are represented in table 3.

| | A.Amphitrite | P.Policipes |
|---|---|---|
| **General summary:** | | |
| Mean read length: | 8,527.7 | 10,903.4 |
| Median read length: | 6,537 | 9,836.0 |
| Number of reads: | 13,337,349 | 1,443,598 |
| Read length N50: | 15,160 | 11,908 |
| Total bases: | 113,737,246,583 | 15,740,168,493 |

Table.3 Pacbio reads statistics.

Since these were raw pacbio reads in .bam format they did not have quality. Quality is calculated during their transformation into primary reads in the assembly process.

Illumina reads were also obtained from collaborators. I perform quality control with FastQC and filtering from low-quality reads and adapter sequences using fastp.

### 3.2.2. Main pipeline.

The main pipeline in the barnacles project is almost similar to the mosquitos project. At first I assemble long pacbio reads using two assemblers. Canu and Falcon. For gooseneck we prefer to use Dovetail assembly because it was scaffolded using their proprietary Hi-C Chicago method and Hi-C data was unavailable. We assess our assemblies and Dovetatil assembly using Busco and Quast-lg. Falcon assembly for *a.amphitrite* and Dovetail assembly for gooseneck were polished with Illumina data using 3 rounds of Pilon. Contigs from assemblies were classified with Purge Haplotigs into primary contigs and haplotigs. After that I performed contamination searches in both assemblies. As validation step I use alignment to the nearest species Hallilea Azteca and mapping genes from that species.

In this chapter I want to describe methods that are different from the mosquitos project: Falcon assembly, and contamination search using blast.

### 3.2.3. Falcon assembly.

FALCON and FALCON-Unzip[60] are de novo genome assemblers for PacBio long reads. FALCON is a diploid-aware assembler which follows the hierarchical genome assembly process (HGAP) and is optimized for large genome assembly. FALCON produces a set of primary contigs (p-contigs) as the primary assembly and a set of associate contigs (a-contigs) which represent divergent allelic variants. Each a-contig is associated with a homologous genomic region on a p-contig.

FALCON-Unzip is a true diploid assembler. It takes the contigs from FALCON and phases the reads based on heterozygous SNPs identified in the initial assembly. It then produces a set of partially-phased primary contigs and fully-phased haplotigs that represent divergent haplotypes.

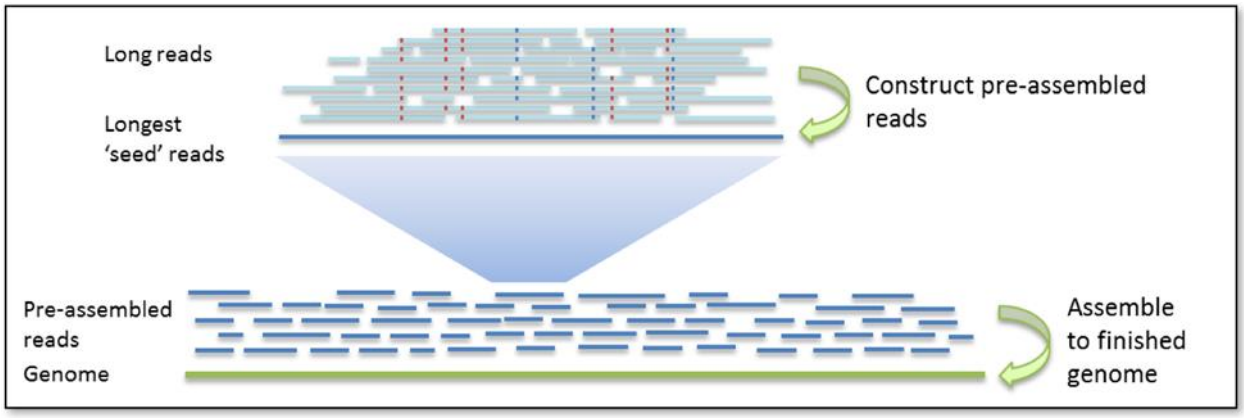


Fig.19. Falcon p-reads creation.

Assembly with PacBio data uses the hierarchical genome assembly process (HGAP). The first round is pre-assembly or error correction of the long reads. This involves the selection of seed reads or the longest reads in the dataset (user-defined length cutoff). All shorter reads are aligned to the seed reads, in order to generate consensus sequences with high accuracy see fig 19.

In the next round of HGAP, the p-reads are aligned to each other and assembled into genomic contigs.

Assembly is typically followed by a round of polishing where all raw PacBio subreads are aligned to the draft contigs and the genomic consensus is performed. Polishing greatly increases the base quality of the assembly.

I assemble both species genomes with Falcon -> Falcon-UNZIP pipeline. Busco results are presented in supplementary table 6.

Falcon and Falcon unzip can be run in grid mode on computational clusters, but this task is still computationally hard and each genome took 2 weeks. Also configure files for starting Falcon and Falcon unzip are very complicate and have a lot of parameters that must be set according to known information about data and genome such as genome ploidy, expected genome length, reads coverage, cutoffs for p-reads creation and so on.

At the same time Canu assembler works very fast on this genomes especially on a.amphitrite with very high pacbio coverage (more than 40). Canu worked on grid-mode only one day for both species.

**3.2.4. Contamination search.**

At first I tried to use the same method with Kraken2 as in mosquitos project but failed. Barnacles species are not well-sequenced before and there was no reference DNA in Kraken2 databases. When I performed Kraken2 on assembled contigs the majority was not classified at all. I tried to search the contamination on the stages of raw reads and p-reads. Raw reads cant be used because of pacbio error rates. But p-reads did not lead to the result, reads were classified almost to unrelated taxons like vertebrates, mushrooms or plants. I think that can be because of the absence of close reference and each read was classified with minor identity.

After these trials by the advice of my advisor in the article about human contamination in bacterial genomes[61] I found an interesting method of contamination search using blast. The core of this method is the division of genome contigs or scaffolds onto overlapping subreads. Each read then maps to the databases. Authors use NCBI refseq, we preferred to use NCBI nucleotide base

because we want to see every accession to the nearest sequenced organisms not only assembled ones.

I divide genomes into 10 kbp overlapping by 5 kbp subreads. Download nt database to the cluster and ran blast in megablast mode using custom output parameters.

I was interested in information that output in the next fields:

- ‘q_id': (query id) name of contig+position of 10 000bp region,
- 'q_start': the position of starting nucleotide in alignment,
- 'q_end': the position of the last nucleotide in alignment,
- 'ss_id': (subject sequence) id of matched sequence in NCBI nt database,
- 'eval': e-value (number of expected hits of similar quality (score) that could be found just by chance given the same size of a random database),
- 'bitscore': bitscore (the log2 scaled required size of a sequence database in which the current match could be found just by chance),
- 'len': length of the alignment,
- 'pident': % of matches,
- 'taxid': taxonomic id of matched sequence.

Blast results were filtered by top accession for each contig. Next step I obtain full taxonomy for each taxonomic id and created merged data frames for all this information.

I performed two levels of analysis. The first analysis of hits with an arbitrary length of alignment and e-value < 0.01. This method shows more homologous hits. The second analysis of hits with length of alignment more or equal 500 bp and e-value < 1e-50. This method shows more contaminant hits.

Then I built plots with the color representation of each contig subreads taxonomy classification. And contaminated contigs were easily founded. For example subset of gooseneck assembly scaffolds are presented in fig. 20.

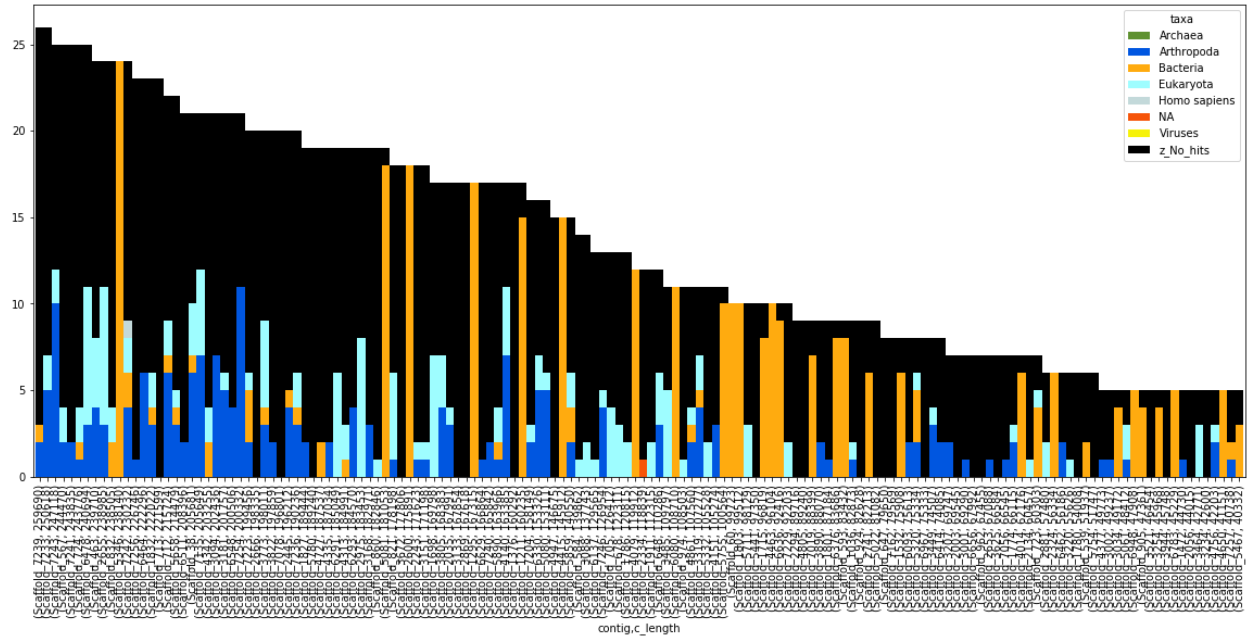


Fig.20. Goose neck contamination search plot scaffolds 51-200.

The black color is for unclassified subreads, orange for bacterial contamination, and blue for subreads classified as Arthropoda species (supposed as barnacles).

All contigs that were classified as contaminated were additionally aligned using blast to nucleotide database and bacterial contamination was proved.

This method is highly useful in case of contigs that consist of homogenous contaminants but if we have chimeric contamination inside contig in small proportion we still do not see it.

Using this method 8 from 2644 contigs of *a.amphitrite* assembly and 62 from 576 scaffolds of gooseneck assembly were filtered as bacterial contamination.

**3.3. Project results.**

At this project I assemble two draft genomes for each species with Canu and Falcon. Polished assemblies. Assess them and performed contamination search and validation. The project is in the process now, we are waiting for Hi-C data for *a.amphitrite* to start the scaffolding. After the finish of this project these genomes will be one of the first genomes for barnacle species and will be published in NCBI database.

## 4. CONCLUSION.

My thesis is a summary of my research work during the master's program. I followed a full cycle of de-novo reference-level genome assembly. All stages of this process were covered: producing genome drafts, polishing, scaffolding, validation. This work allowed me to work on the science frontier. Methods of scaffolding with Hi-C data are brand new in the field and our assemblies from long noisy reads with such scaffolding are one of the first.

I learned how to work with computational clusters and apply full specter of bioinformatics tools related to genome assembly as well as how to write scripts for data analysis. I designed computational workflows for genome validation using data from biological experiments (FISH, cq analysis) and for contamination search using blast over NCBI nucleotide database.

Thanks to these projects I found the sphere of my scientific interests. I am really amazed by the Hi-C method and it`s applications. I want to work with Hi-C data not only in analysis way but creating software for data visualization and contact map editing.

I think that chromosome-level genome assembly for non-model organisms is a very important and fast-developing area and I want to sharp my skills and apply my experience for further researches

## 5. REFERENCES.


[1] "U.S. National Library of Medicine Genome List." [Online]. Available: https://www.ncbi.nlm.nih.gov/genome/browse/#!/overview/.

[2] "Catalogue of Life: 2020-04-16 Beta. Annual Checklist Interface developed by Naturalis Biodiversity Center." [Online]. Available: https://www.catalogueoflife.org/col/browse/tree/id/89ac18bfcf1654a9662a600ba06bb494.

[3] H. P. J. Buermans and J. T. den Dunnen, "Next generation sequencing technology: Advances and applications," *Biochim. Biophys. Acta - Mol. Basis Dis.*, vol. 1842, no. 10, pp. 1932–1941, 2014.

[4] Illumina Inc., "Illumina sequencing introduction," *Illumina Seq. Introd.*, no. October, pp. 1–8, 2017.

[5] M. Jain, H. E. Olsen, B. Paten, and M. Akeson, "The Oxford Nanopore MinION : delivery of nanopore sequencing to the genomics community," *Genome Biol.*, pp. 1–11, 2016.

[6] S. Ardui, A. Ameur, J. R. Vermeesch, and M. S. Hestand, "Single molecule real-time (SMRT) sequencing comes of age: Applications and utilities for medical diagnostics," *Nucleic Acids Res.*, vol. 46, no. 5, pp. 2159–2168, 2018.

[7] E. Lieberman-Aiden *et al.*, "Comprehensive mapping of long-range interactions reveals folding principles of the human genome," *Science (80-. ).*, vol. 326, no. 5950, pp. 289–293, 2009.

[8] V. Lukyanchikova, V. Fishman, M. Nuriddinov, N. Battulin, O. L. Serov, and I. V Sharakhov, "The Hi-C approach improved genome assemblies of Anopheles species and revealed principles of 3D genome organization in dipteran insects," vol. 7, no. 2014, p. 2299, 2018.

[9] C. Cui, W. Shu, and P. Li, "Fluorescence in situ hybridization: Cell-based genetic diagnostic and research applications," *Front. Cell Dev. Biol.*, vol. 4, no. SEP, pp. 1–11, 2016.

[10] M. Coetzee, R. H. Hunt, R. Wilkerson, A. Della Torre, M. B. Coulibaly, and N. J. Besansky, "Anopheles coluzzii and anopheles amharicus, new members of the anopheles gambiae complex," *Zootaxa*, vol. 3619, no. 3, pp. 246–274, 2013.

[11] D. E. Neafsey *et al.*, "SNP genotyping defines complex gene-flow boundaries among african malaria vector mosquitoes," *Science (80-. ).*, vol. 330, no. 6003, pp. 514–517, 2010.

[12] F. Aboagye-Antwi *et al.*, "Experimental Swap of Anopheles gambiae's Assortative Mating Preferences Demonstrates Key Role of X-Chromosome Divergence Island in Incipient Sympatric Speciation," *PLoS Genet.*, vol. 11, no. 4, pp. 1–19, 2015.

[13] Y. Lee *et al.*, "Spatiotemporal dynamics of gene flow and hybrid fitness between the M and S forms of the malaria mosquito, Anopheles gambiae," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 49, pp. 19854–19859, 2013.

[14] D. Weetman, C. S. Wilding, K. Steen, J. Pinto, and M. J. Donnelly, "Gene flow-dependent genomic divergence between anopheles gambiae M and S

forms," *Mol. Biol. Evol.*, vol. 29, no. 1, pp. 279–291, 2012.
[15] B. Tene Fossog *et al.*, "Habitat segregation and ecological character displacement in cryptic African malaria mosquitoes," *Evol. Appl.*, vol. 8, no. 4, pp. 326–345, 2015.
[16] A. Diabaté *et al.*, "Larval Development of the Molecular Forms of <I>Anopheles gambiae</I> (Diptera: Culicidae) in Different Habitats: A Transplantation Experiment," *J. Med. Entomol.*, vol. 42, no. 4, pp. 548–553, 2006.
[17] G. Gimonneau, J. Bouyer, S. Morand, N. J. Besansky, A. Diabate, and F. Simard, "A behavioral mechanism underlying ecological divergence in the malaria mosquito Anopheles gambiae," *Behav. Ecol.*, vol. 21, no. 5, pp. 1087–1092, 2010.
[18] A. Dao *et al.*, "Signatures of aestivation and migration in Sahelian malaria mosquito populations," *Nature*, vol. 516, no. 7531, pp. 387–390, 2014.
[19] V. Petrarca and J. C. Beier, "Intraspecific chromosomal polymorphism in the Anopheles gambiae complex as a factor affecting malaria transmission in the Kisumu area of Kenya," *Am. J. Trop. Med. Hyg.*, vol. 46, no. 2, pp. 229–237, 1992.
[20] B. J. Main *et al.*, "The Genetic Basis of Host Preference and Resting Behavior in the Major African Malaria Vector, Anopheles arabiensis," *PLoS Genet.*, vol. 12, no. 9, pp. 1–17, 2016.
[21] D. E. Neafsey *et al.*, "Highly evolvable malaria vectors: The genomes of 16 Anopheles mosquitoes," *Science (80-. ).*, vol. 347, no. 6217, 2015.
[22] H. Li, "Minimap and miniasm: Fast mapping and de novo assembly for noisy long sequences," *Bioinformatics*, vol. 32, no. 14, pp. 2103–2110, 2016.
[23] H. Li, "Minimap2: Pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018.
[24] D. E. Wood and S. L. Salzberg, "Kraken : ultrafast metagenomic sequence classification using exact alignments," 2014.
[25] D. E. Wood, J. Lu, and B. Langmead, "Improved metagenomic analysis with Kraken 2," *Genome Biol.*, vol. 20, no. 1, pp. 1–13, 2019.
[26] S. Andrews, "FastQC: a quality control tool for high throughput sequence data.," 2010. [Online]. Available: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
[27] S. Chen, Y. Zhou, Y. Chen, and J. Gu, "Fastp: An ultra-fast all-in-one FASTQ preprocessor," *Bioinformatics*, vol. 34, no. 17, pp. i884–i890, 2018.
[28] J. Ghurye *et al.*, "A chromosome-scale assembly of the major African malaria vector Anopheles funestus," *Gigascience*, vol. 8, no. 6, pp. 1–8, 2019.
[29] 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 k - mer weighting and repeat separation," pp. 722–736, 2017.
[30] C. Camacho *et al.*, "BLAST+: Architecture and applications," *BMC Bioinformatics*, vol. 10, pp. 1–9, 2009.
[31] J. Ruan and H. Li, "Fast and accurate long-read assembly with wtdbg2,"

2019.
[32] 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.
[33] A. Gurevich, V. Saveliev, N. Vyahhi, and G. Tesler, “QUAST: Quality assessment tool for genome assemblies,” *Bioinformatics*, vol. 29, no. 8, pp. 1072–1075, 2013.
[34] F. A. Simão, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov, “BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs,” *Bioinformatics*, vol. 31, no. 19, pp. 3210–3212, 2015.
[35] “Anopheles Gambiae genome assembly (AgamP4). Vectorbase.,” *2019*. [Online]. Available: https://www.vectorbase.org/organisms/anopheles-gambiae/pest/agamp4.
[36] A. R. Quinlan and I. M. Hall, “BEDTools: A flexible suite of utilities for comparing genomic features,” *Bioinformatics*, vol. 26, no. 6, pp. 841–842, 2010.
[37] A. Mikheenko, G. Valin, A. Prjibelski, V. Saveliev, and A. Gurevich, “Icarus: Visualizer for de novo assembly evaluation,” *Bioinformatics*, vol. 32, no. 21, pp. 3321–3323, 2016.
[38] “The hierarchical catalog of orthologs.” [Online]. Available: https://www.orthodb.org/.
[39] L. S. Johnson, S. R. Eddy, and E. Portugaly, “Hidden Markov model speed heuristic and iterative HMM search procedure,” *BMC Bioinformatics*, vol. 11, 2010.
[40] O. Keller, M. Kollmar, M. Stanke, and S. Waack, “A novel hybrid gene prediction method employing protein multiple sequence alignments,” *Bioinformatics*, vol. 27, no. 6, pp. 757–763, 2011.
[41] Heng Li, “auN: a new metric to measure assembly contiguity,” 2020.
[42] R. Vaser, I. Sović, N. Nagarajan, and M. Šikić, “Fast and accurate de novo genome assembly from long uncorrected reads,” *Genome Res.*, vol. 27, no. 5, pp. 737–746, 2017.
[43] N. J. Loman, J. Quick, and J. T. Simpson, “A complete bacterial genome assembled de novo using only nanopore sequencing data,” *Nat. Methods*, vol. 12, no. 8, pp. 733–735, 2015.
[44] B. J. Walker *et al.*, “Pilon : An Integrated Tool for Comprehensive Microbial Variant Detection and Genome Assembly Improvement,” vol. 9, no. 11, 2014.
[45] F. Ramírez *et al.*, “High-resolution TADs reveal DNA sequences underlying genome organization in flies,” *Nat. Commun.*, vol. 9, no. 1, 2018.
[46] J. Ghurye *et al.*, “Integrating Hi-C links with assembly graphs for chromosome-scale assembly,” *PLoS Comput. Biol.*, vol. 15, no. 8, pp. 1–19, 2019.
[47] O. Dudchenko *et al.*, “De novo assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds,” *Science (80-. ).*, vol. 356, no.

6333, pp. 92–95, 2017.
[48] N. C. Durand *et al.*, “Juicebox Provides a Visualization System for Hi-C Contact Maps with Unlimited Zoom,” *Cell Syst.*, vol. 3, no. 1, pp. 99–101, 2016.
[49] H. Li, “Aligning sequence reads , clone sequences and assembly contigs with BWA-MEM,” vol. 00, no. 00, pp. 1–3, 2013.
[50] H. Li *et al.*, “The Sequence Alignment/Map format and SAMtools,” *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.
[51] N. C. Durand *et al.*, “Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments,” *Cell Syst.*, vol. 3, no. 1, pp. 95–98, 2016.
[52] “Genome Assembly Cookbook,” 2018.
[53] M. J. Roach, S. A. Schmidt, and A. R. Borneman, “Purge Haplotigs: Allelic contig reassignment for third-gen diploid genome assemblies,” *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–10, 2018.
[54] F. Cabanettes and C. Klopp, “D-GENIES: Dot plot large genomes in an interactive, efficient and simple way,” *PeerJ*, vol. 2018, no. 6, 2018.
[55] I. V. Sharakhov *et al.*, “Breakpoint structure reveals the unique origin of an interspecific chromosomal inversion (2La) in the Anopheles gambiae complex,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 16, pp. 6258–6262, 2006.
[56] J. T. Robinson *et al.*, “Integrative Genome Viewer,” *Nat. Biotechnol.*, vol. 29, no. 1, pp. 24–6, 2011.
[57] D. P. Melters *et al.*, “Comparative analysis of tandem repeats from hundreds of species reveals unique insights into centromere evolution,” *Genome Biol.*, vol. 14, no. 1, pp. 1–20, 2013.
[58] J. Krzywinski, D. Sangaré, and N. J. Besansky, “Satellite DNA from the Y chromosome of the malaria vector Anopheles gambiae,” *Genetics*, vol. 169, no. 1, pp. 185–196, 2005.
[59] A. B. Hall, Y. Qi, V. Timoshevskiy, M. V. Sharakhova, I. V. Sharakhov, and Z. Tu, “Six novel Y chromosome genes in Anopheles mosquitoes discovered by independently sequencing males and females,” *BMC Genomics*, vol. 14, no. 1, 2013.
[60] 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.
[61] F. P. Breitwieser, M. Pertea, A. V. Zimin, and S. L. Salzberg, “Human contamination in bacterial genomes has created thousands of spurious proteins,” *Genome Res.*, vol. 29, no. 6, pp. 954–960, 2019.

## APPENDIX A

Supplementary figures and tables for Mosquitos project.

| | An.Coluzzii | An.Arabiensis |
|---|---|---|
| **General summary:** | | |
| Mean read length: | 8509.6 | 6788.7 |
| Mean read quality: | 10.1 | 9.3 |
| Median read length: | 3833 | 2 256 |
| Median read quality: | 10.3 | 10 |
| Number of reads: | 3 299 012 | 5 094 106 |
| Read length N50: | 19 315 | 21 969 |
| Total bases: | 28 073 279 865 | 34 582 156 501 |
| **Number, percentage and megabases of reads above quality cutoffs** | | |
| >Q5: | 3299012 (100.0%) 28073.3Mb | 4718400 (92.6%) 33503.9M |
| >Q7: | 3297112 (99.9%) 28059.2Mb | 4457124 (87.5%) 32336.6Mb |
| >Q10: | 1994146 (60.4%) 17513.7Mb | 2520015 (49.5%) 20173.2Mb |
| >Q12: | 34417 (1.0%) 202.7Mb | 62557 (1.2%) 232.6Mb |
| >Q15: | 0 (0.0%) 0.0Mb | 0 (0.0%) 0.0Mb |
| **Top 5 highest mean basecall quality scores and their read lengths** | | |
| 1: | 13.8 (19847) | 14.5 (33760) |
| 2: | 13.7 (924) | 14.5 (651) |
| 3: | 13.7 (576) | 14.4 (484) |
| 4: | 13.7 (1109) | 14.3 (19429) |
| 5: | 13.6 (1416) | 14.3 (177) |
| **Top 5 longest reads and their mean basecall quality score** | | |
| 1: | 298483 (9.9) | 276317 (9.0) |
| 2: | 292246 (7.4) | 276021 (10.7) |
| 3: | 291487 (9.9) | 265300 (10.6) |
| 4: | 273986 (7.4) | 252875 (8.5) |
| 5: | 262854 (7.6) | 245928 (9.3) |

Table.1 Statistics for nanopore long reads


Fig.1. An.Coluzzii nanopore reads quality and lengths distribution plots

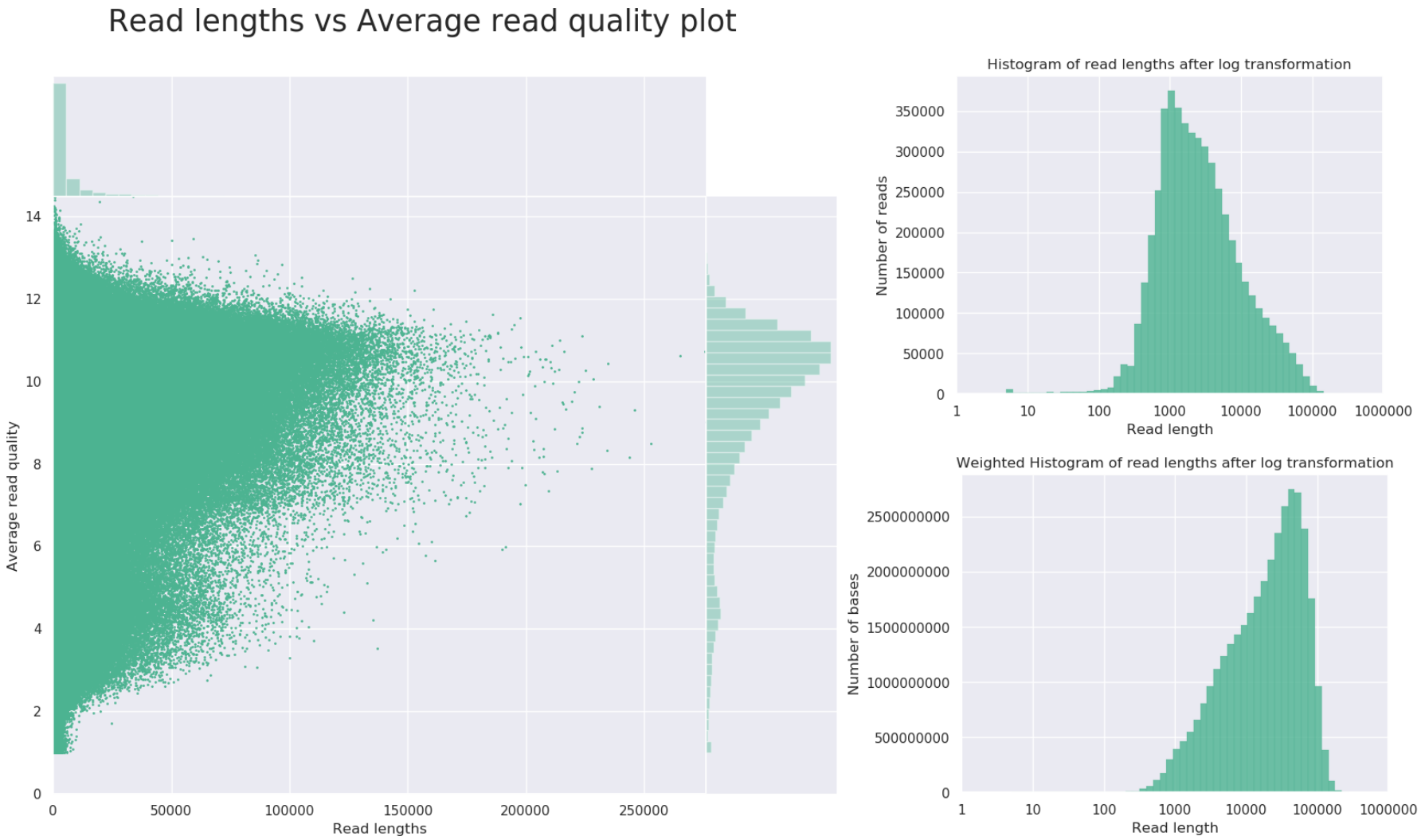


Fig.2. An.Arabiensis nanopore reads quality and lengths distribution plots

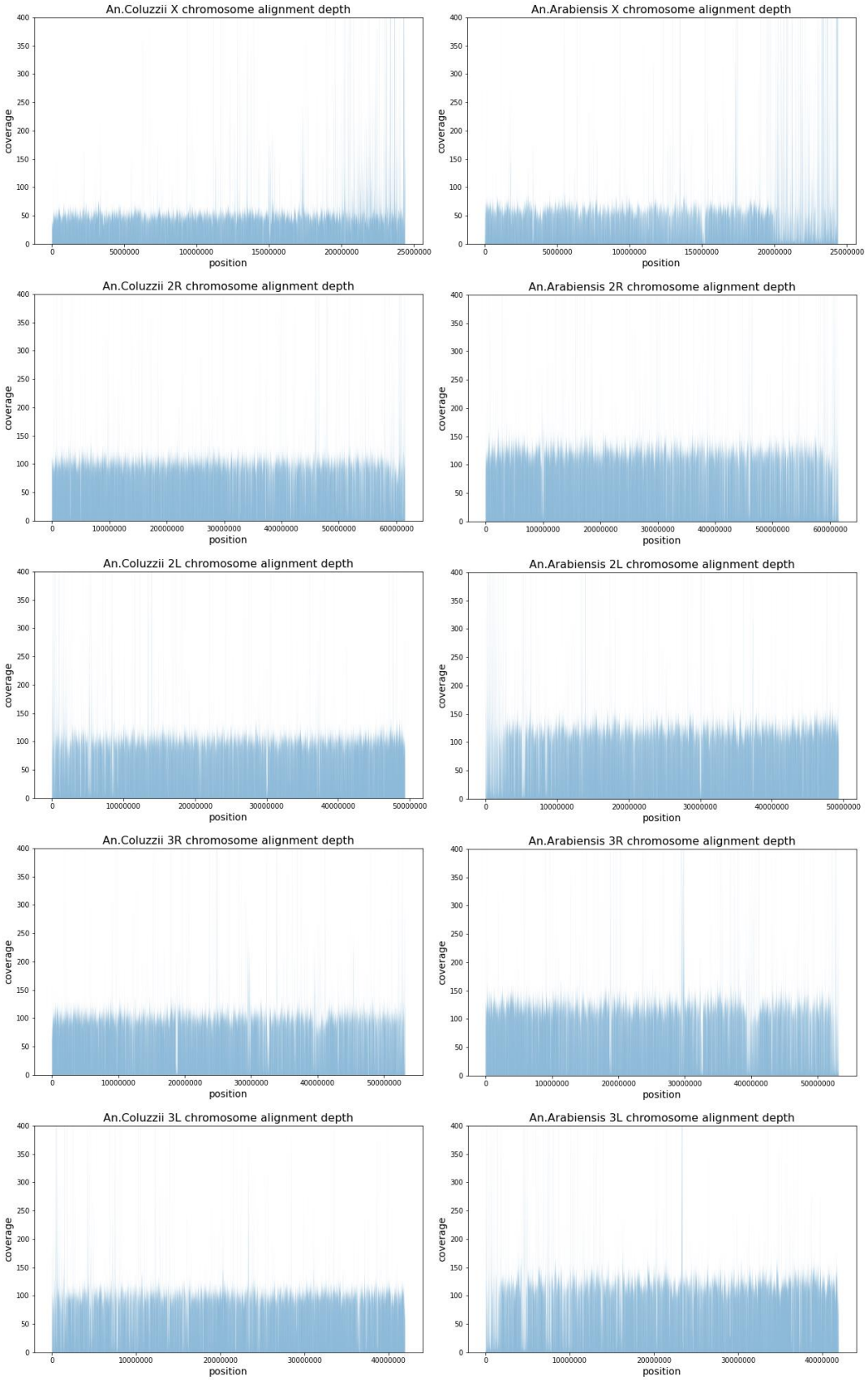


Fig.3. The alignment depth of nanopore reads from An. coluzzii (left column) and An. arabiensis (right column) genomes to the closely-related assembly of An. gambiae (AgamP4) genome.

| | Anopheles Coluzzii | | | | | Anopheles Arabiensis | | | |
|---|---|---|---|---|---|---|---|---|---|
| | wtdbg2 | miniasm | Flye | CANU - unitigs | CANU - contigs | wtdbg2 | Flye | CANU - contigs | CANU-unitigs |
| **Genome statistics** | **Genome statistics (A.Gambiae PEST assembly as reference)** | | | | | | | | |
| Genome fraction (%) | 54,32 | 0,16 | 52,58 | 59,18 | 58,48 | 52,869 | 43,074 | 56,24 | 59,89 |
| Duplication ratio | 1,07 | 1,02 | 1,14 | 1,36 | 1,25 | 1,13 | 1,139 | 1,23 | 1,31 |
| Largest alignment | 1 440 034 | 3 064 | 1 154 080 | 1 284 741 | 1 469 500 | 2 082 067 | 2 182 677 | 2 090 800 | 2 126 425 |
| Total aligned length | 145 808 258 | 420 914 | 150 581 571 | 202 100 538 | 184 201 098 | 150 768 812 | 123 807 719 | 173 732 578 | 196 854 393 |
| NG50 | 3 518 408 | 3 351 593 | 3 617 662 | 4 485 652 | 13 842 187 | 7 492 164 | 10 796 042 | 23 700 761 | 23 710 802 |
| NG75 | 712 008 | 956 538 | 565 263 | 550 578 | 6 304 690 | 737 639 | 1 341 337 | 6 985 567 | 1 144 137 |
| NA50 | 2 952 | NA | 2 166 | 4 228 | 3 874 | 629 | NA | 10 884 | 27 527 |
| NGA50 | 2 196 | NA | 2 658 | 44 375 | 17 350 | 1 625 | NA | 15 731 | 54 779 |
| LG50 | 17 | 20 | 21 | 16 | 7 | 12 | 7 | 5 | 5 |
| LG75 | 54 | 57 | 78 | 71 | 14 | 38 | 25 | 10 | 25 |
| LA50 | 3 508 | NA | 5 884 | 4 036 | 4 247 | 20 751 | NA | 1 135 | 994 |
| LGA50 | 4 675 | NA | 4 717 | 837 | 1 155 | 8 457 | NA | 980 | 677 |
| **Misassemblies** | **Misassemblies** | | | | | | | | |
| # misassemblies | 4 316 | 0 | 6 196 | 11 834 | 10 729 | 4 061 | 3 958 | 12 308 | 12 860 |
| # relocations | 1 706 | 0 | 2 467 | 4 461 | 3 880 | 1 320 | 1 287 | 4 396 | 4 276 |
| # translocations | 2 554 | 0 | 3 634 | 7 222 | 6 723 | 2 706 | 2 631 | 7 826 | 8 477 |
| # inversions | 56 | 0 | 95 | 151 | 126 | 35 | 40 | 86 | 107 |
| # misassembled contigs | 300 | 0 | 360 | 518 | 210 | 334 | 151 | 134 | 348 |
| Misassembled contigs length | 197 415 431 | 0 | 213 113 463 | 258 775 961 | 240 726 872 | 166 552 571 | 168 135 090 | 238 615 865 | 247 176 514 |
| # local misassemblies | 18 019 | 0 | 20 122 | 26 014 | 22 903 | 22 376 | 23 629 | 30 814 | 31 980 |
| # scaffold gap ext. mis. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| # scaffold gap loc. mis. | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| # possible TEs | 2 228 | 0 | 2 320 | 3 704 | 3 478 | 1 878 | 1 422 | 2 824 | 3 138 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| # unaligned mis. contigs | 510 | 86 | 321 | 358 | 191 | 831 | 266 | 45 | 93 |
| **Unaligned** | **Unaligned** | | | | | | | | |
| # fully unaligned contigs | 310 | 462 | 174 | 24 | 12 | 428 | 716 | 11 | 21 |
| Fully unaligned length | 6 082 863 | 88 230 278 | 3 238 628 | 1 858 773 | 985 300 | 15 125 468 | 26 219 848 | 1 255 286 | 2 375 670 |
| # partially unaligned contigs | 1 051 | 172 | 834 | 1 024 | 450 | 1 456 | 529 | 200 | 497 |
| Partially unaligned length | 115 076 975 | 230 334 583 | 124 479 154 | 139 181 121 | 128 372 967 | 132 515 795 | 139 638 537 | 101 637 092 | 98 726 070 |
| **Mismatches** | **Mismatches** | | | | | | | | |
| # mismatches | 3 560 597 | 5 311 | 3 801 380 | 4 738 319 | 4 239 427 | 5 677 775 | 4 537 311 | 6 933 636 | 7 814 852 |
| # indels | 1 160 243 | 6 779 | 1 724 034 | 1 269 221 | 1 167 976 | 1 040 792 | 1 294 719 | 994 669 | 784 501 |
| Indels length | 2 744 847 | 10 804 | 3 289 354 | 3 511 488 | 3 215 434 | 2 579 692 | 2 568 697 | 2 851 751 | 2 863 834 |
| # mismatches per 100 kbp | 2 596,36 | 1 289,52 | 2 863,66 | 3 171,32 | 2 871,56 | 4 254 | 4 173 | 4 884 | 5 169 |
| # indels per 100 kbp | 846,04 | 1 645,96 | 1 298,75 | 849,48 | 791,13 | 780 | 1 191 | 701 | 519 |
| # indels (<= 5 bp) | 1 081 711 | 6 686 | 1 654 237 | 1 163 569 | 1 070 969 | 959 948 | 1 235 354 | 897 417 | 672 054 |
| # indels (> 5 bp) | 78 532 | 93 | 69 797 | 105 652 | 97 007 | 80 844 | 59 365 | 97 252 | 112 447 |
| # N's | 0 | 0 | 500 | 0 | 0 | 0 | 500 | 0 | 0 |
| # N's per 100 kbp | 0,00 | 0,00 | 0,18 | 0,00 | 0,00 | 0 | 0,17 | 0,00 | 0,00 |
| **Statistics without reference** | **Statistics without reference** | | | | | | | | |
| # contigs | 1 391 | 634 | 1 048 | 1 055 | 465 | 1 920 | 1 280 | 211 | 521 |
| # contigs (>= 0 bp) | 1 392 | 638 | 1 618 | 1 073 | 474 | 1 928 | 2 048 | 220 | 541 |
| # contigs (>= 1000 bp) | 1 392 | 638 | 1 388 | 1 073 | 474 | 1 928 | 1 763 | 220 | 541 |
| # contigs (>= 5000 bp) | 1 348 | 630 | 861 | 1 051 | 463 | 1 883 | 1 060 | 208 | 517 |
| # contigs (>= 10000 bp) | 1 067 | 622 | 731 | 1 045 | 460 | 1 428 | 836 | 207 | 512 |
| # contigs (>= 25000 bp) | 653 | 616 | 624 | 1 029 | 455 | 929 | 644 | 207 | 507 |
| # contigs (>= 50000 bp) | 364 | 606 | 502 | 935 | 432 | 564 | 455 | 205 | 494 |
| Largest contig | 17 446 215 | 18 411 938 | 13 804 856 | 19 763 313 | 33 413 712 | 22 238 065 | 32 507 593 | 44 591 211 | 33 440 724 |
| Total length | 267 203 205 | 318 985 775 | 278 686 226 | 343 961 469 | 314 190 168 | 298 413 078 | 289 704 167 | 277 203 818 | 298 533 470 |
| Total length (>= 0 bp) | 267 206 174 | 318 993 565 | 279 473 599 | 343 996 597 | 314 208 573 | 298 431 996 | 290 752 171 | 277 218 797 | 298 564 895 |
| Total length (>= 1000 bp) | 267 206 174 | 318 993 565 | 279 331 846 | 343 996 597 | 314 208 573 | 298 431 996 | 290 567 540 | 277 218 797 | 298 564 895 |
| Total length (>= 5000 bp) | 267 016 028 | 318 970 674 | 277 964 236 | 343 945 192 | 314 182 073 | 298 254 454 | 288 856 343 | 277 192 427 | 298 517 822 |
| Total length (>= 10000 bp) | 264 994 751 | 318 910 721 | 277 080 562 | 343 900 991 | 314 159 967 | 294 987 018 | 287 284 715 | 277 183 822 | 298 485 334 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Total length (>= 25000 bp) | 257 974 837 | 318 820 092 | 275 269 049 | 343 630 110 | 314 083 595 | 286 915 173 | 284 122 552 | 277 183 822 | 298 387 283 |
| Total length (>= 50000 bp) | 247 916 099 | 318 460 456 | 270 692 452 | 340 005 869 | 313 137 556 | 273 706 097 | 277 219 122 | 277 094 612 | 297 857 656 |
| N50 | 4 111 930 | 2 679 573 | 3 617 326 | 1 211 583 | 13 401 784 | 5 903 928 | 10 796 042 | 23 700 761 | 15 085 722 |
| N75 | 897 490 | 448 329 | 507 563 | 222 639 | 2 423 489 | 268 064 | 603 035 | 6 985 567 | 475 129 |
| L50 | 16 | 28 | 22 | 30 | 8 | 14 | 7 | 5 | 6 |
| L75 | 48 | 111 | 86 | 229 | 21 | 85 | 39 | 10 | 54 |
| GC (%) | 44 | 44 | 43 | 44 | 44 | 43 | 43 | 44 | 44 |
| **K-mer-based statistics** | **K-mer-based statistics** | | | | | | | | |
| K-mer-based compl. (%) | 14,53 | 0,24 | 10,43 | 15,65 | 15,95 | 8,35 | 5,70 | 9,18 | 11,57 |
| K-mer-based cor. length (%) | 23,43 | 34,90 | 48,30 | 34,34 | 10,88 | 11,68 | 20,45 | 6,28 | 19,47 |
| K-mer-based mis. length (%) | 64,97 | 22,41 | 43,84 | 52,74 | 78,74 | 58,02 | 59,66 | 84,51 | 68,44 |
| K-mer-based undef. length (%) | 11,60 | 42,68 | 7,87 | 12,92 | 10,38 | 30,30 | 19,89 | 9,21 | 12,10 |
| # k-mer-based misjoins | 266 | 14 | 215 | 512 | 513 | 108 | 54 | 168 | 216 |
| # k-mer-based translocations | 171 | 4 | 113 | 267 | 254 | 82 | 29 | 117 | 174 |
| # k-mer-based 100kbp relocations | 95 | 10 | 102 | 245 | 259 | 26 | 25 | 51 | 42 |

Table.2 Quast-lg reports for draft assemblies

| | | BUSCO diptera | | | | | | BUSCO metazoa | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | assemblies | complete | single | duplicated | fragmented | missing | number | complete | single | duplicated | fragmented | missing | number |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A.Coluzzii | | **A.Coluzzii MOPTI Draft Assemblies** | | | | | | | | | | | |
| | CANU unitigs | 79.3% | 73.0% | 6.3% | 12.9% | 7.8% | 2799 | 94.4% | 84.4% | 10.0% | 2.7% | 2.9% | 978 |
| | CANU contigs | 77.6% | 72.7% | 4.9% | 13.7% | 8.7% | | 93.3% | 85.3% | 8.0% | 3.5% | 3.2% | |
| | wtdbg2 | 65.9% | 65.6% | 0.3% | 18.7% | 15.4% | | 87.4% | 86.9% | 0.5% | 6.9% | 5.7% | |
| | Flye | 62.7% | 61.7% | 1.0% | 19.0% | 18.3% | | 81.2% | 80.0% | 1.2% | 9.7% | 9.1% | |
| | miniasm | 1.4% | 1.4% | 0.0% | 3.1% | 95.5% | | 4.6% | 4.6% | 0.0% | 21.1% | 83.3% | |
| A.Arabiensis | | **A.Arabiensis DONGOLA draft assemblies** | | | | | | | | | | | |
| | CANU unitigs | 83.9% | 78.0% | 5.9% | 10.8% | 5.3% | 2799 | 94.4% | 87.5% | 6.9% | 3.3% | 2.3% | 978 |
| | CANU contigs | 83.0% | 78.% | 4.1% | 11.4% | 5.6% | | 94.1% | 88.8% | 5.3% | 3.5% | 2.4% | |
| | Flye | 63.0% | 62.6% | 0.4% | 18.5% | 18.5% | | 81.9% | 80.7% | 1.2% | 9.3% | 8.8% | |
| | wtdbg2 | 70.4% | 70.2% | 0.2% | 16.0% | 13.6% | | 85.7% | 85.1% | 0.6% | 5.1% | 9.2% | |
| | miniasm | 1.5% | 1.5% | 0.0% | 4.4% | 94.1% | | 8.2% | 8.2% | 0.0% | 18.8% | 73.0% | |

Table.3 BUSCO scores for draft assemblies

| | | BUSCO diptera | | | | | | BUSCO metazoa | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | assemblies | complete | single | duplicated | fragmented | missing | number | complete | single | duplicated | fragmented | missing | number |
| A.Coluzzii | | **CANU_contigs assembly polishing 1st strategy (np - Nanopolish, rx - x Racoon rounds, pilx - x rounds of Pilon)** | | | | | | | | | | | |
| | canu>np | 93.6% | 87.4% | 6.2% | 4.3% | 2.1% | 2799 | 98.3% | 89.2% | 9.1% | 0.5% | 1.2% | 978 |
| | canu>np>r4 | 88.6% | 82.7% | 5.9% | 7.7% | 3.7% | | 96.9% | 88.5% | 8.4% | 1.2% | 1.9% | |
| | canu>np>pil1 | 97.9% | 90.4% | 7.5% | 1.2% | 0.9% | | 98.8% | 89.1% | 9.7% | 0.2% | 1.0% | |
| | canu>np>pil1>np | 95.9% | 88.5% | 7.4% | 2.8% | 1.3% | | 98.5% | 89.2% | 9.3% | 0.5% | 1.0% | |
| | canu>np>pil2 | 98.2% | 90.1% | 8.1% | 0.9% | 0.9% | | 98.9% | 89.1% | 9.8% | 0.2% | 0.9% | |
| | canu>np>pil3 | 98.5% | 90.2% | 8.3% | 0.7% | 0.8% | | 98.9% | 89.1% | 9.8% | 0.2% | 0.9% | |
| | | **CANU_contigs assembly polishing 2nd strategy (rx - x rounds of Racon, med - Medaka)** | | | | | | | | | | | |
| | canu>r1 | 86.9% | 81.3% | 5.6% | 8.9% | 4.2% | 2799 | 96.4% | 88.4% | 8.0% | 1.8% | 1.8% | 978 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | canu>r2 | 87.9% | 82.5% | 5.4% | 8.3% | 3.8% | | 96.8% | 88.2% | 8.6% | 1.5% | 1.7% | |
| | canu>r3 | 87.7% | 82.1% | 5.6% | 8.6% | 3.7% | | 96.4% | 87.7% | 8.7% | 1.8% | 1.8% | |
| | canu>r4 | 87.7% | 81.9% | 5.8% | 8.6% | 3.7% | | 96.4% | 88.2% | 8.2% | 2.0% | 1.6% | |
| | canu>r4>med | 95.1% | 88.0% | 7.1% | 3.3% | 1.6% | | 98.4% | 89.5% | 8.9% | 0.6% | 1.0% | |
| A.Arabiensis | | **CANU_contigs assembly polishing (np - Nanopolish, pilx - x rounds of Pilon)** | | | | | | | | | | | |
| | canu>np | 94.4% | 90.0% | 4.4% | 3.7% | 1.9% | 2799 | 98.0% | 91.7% | 6.3% | 0.9% | 1.1% | 978 |
| | canu>np>pil1 | 98.3% | 93.3% | 5.0% | 0.9% | 0.8% | | 98.9% | 92.0% | 6.9% | 0.2% | 0.9% | |
| | canu>np>pil2 | 98.5% | 93.6% | 4.9% | 0.7% | 0.8% | | 98.9% | 91.9% | 7.0% | 0.2% | 0.9% | |
| | canu>np>pil3 | 98.5% | 93.5% | 5.0% | 0.7% | 0.8% | | 98.9% | 91.8% | 7.1% | 0.2% | 0.9% | |
| | | **CANU_unitigs assembly polishing (np - Nanopolish, pilx - x rounds of Pilon)** | | | | | | | | | | | |
| | canu_unitigs>np | 94.5% | 88.6% | 5.9% | 3.6% | 1.9% | 2799 | 98.0% | 90.1% | 7.9% | 0.7% | 1.3% | 978 |
| | canu_unitigs>np>pil1 | 98.2% | 91.6% | 6.6% | 0.9% | 0.9% | | 98.8% | 90.5% | 8.3% | 0.2% | 1.0% | |
| | canu_unitigs>np>pil2 | 98.6% | 91.8% | 6.8% | 0.6% | 0.8% | | 98.8% | 90.4% | 8.4% | 0.2% | 1.0% | |
| | canu_unitigs>np>pil3 | 98.6% | 91.7% | 6.9% | 0.6% | 0.8% | | 98.8% | 90.2% | 8.6% | 0.2% | 1.0% | |

Table.4 Canu assemblies polishing BUSCO scores

| | An.Coluzzii | | Arabiensis | | | |
|---|---|---|---|---|---|---|
| | contigs 3D-DNA | contigs SALSA2 | contigs 3D-DNA | unitigs 3D-DNA | contigs SALSA2 | unitigs SALSA2 |
| **Genome statistics** | **Genome statistics** | | | | | |
| Genome fraction (%) | 76.755 | 76.788 | 60.335 | 60.622 | 60.442 | 60.607 |
| Duplication ratio | 1.298 | 1.299 | 1.191 | 1.308 | 1.225 | 1.307 |
| Largest alignment | 1 887 112 | 1 602 245 | 2 132 901 | 2 132 915 | 2 132 901 | 2 132 915 |
| Total aligned length | 250 071 004 | 250 575 668 | 180 751 241 | 199 307 339 | 186 285 183 | 199 303 482 |
| NG50 | 55 177 780 | 31 651 344 | 5 162 957 | 5 958 256 | 71 432 794 | 23 943 874 |
| NG75 | 35 859 916 | 25 845 703 | 2 275 000 | 2 725 000 | 23 935 993 | 1 708 138 |
| NA50 | 114 112 | 115 512 | 40 879 | 19 514 | 39 573 | 33 207 |
| NGA50 | 176 362 | 178 481 | 31 046 | 39 605 | 46 965 | 58 625 |
| LG50 | 2 | 4 | 16 | 14 | 2 | 5 |
| LG75 | 4 | 6 | 36 | 31 | 4 | 16 |
| LA50 | 461 | 457 | 728 | 1 273 | 717 | 924 |
| LGA50 | 6 667 | 305 | 838 | 824 | 653 | 643 |
| **Misassemblies** | **Misassemblies** | | | | | |
| # misassemblies | 16 041 | 14 900 | 11 650 | 15 415 | 13 155 | 13 271 |
| # relocations | 6 274 | 5 381 | 4 033 | 5 080 | 4 819 | 4 610 |
| # translocations | 9 578 | 9 353 | 7 470 | 10 125 | 8 220 | 8 539 |
| # inversions | 189 | 166 | 147 | 210 | 116 | 122 |
| # misassembled contigs | 28 | 238 | 152 | 401 | 101 | 224 |
| Misassembled contigs length | 313 812 358 | 295 525 606 | 222 577 479 | 246 103 302 | 242 010 143 | 252 795 057 |
| # local misassemblies | 37 012 | 34 385 | 28 412 | 30 846 | 30 282 | 31 880 |
| # scaffold gap ext. mis. | 6 | 3 | 0 | 0 | 1 | 0 |
| # scaffold gap loc. mis. | 20 | 7 | 1 | 8 | 1 | 1 |
| # possible TEs | 4 706 | 4 400 | 2 818 | 3 248 | 2 908 | 3 144 |
| # unaligned mis. contigs | 9 | 85 | 52 | 169 | 34 | 81 |
| **Unaligned** | **Unaligned** | | | | | |
| # fully unaligned contigs | 1 | 7 | 4 | 39 | 10 | 19 |
| Fully unaligned length | 10 907 | 672 048 | 324 911 | 2 347 438 | 1 038 458 | 2 156 888 |
| # partially unaligned contigs | 45 | 363 | 236 | 716 | 142 | 324 |
| Partially unaligned length | 64 723 074 | 63 688 694 | 83 332 303 | 95 916 827 | 90 699 703 | 96 248 562 |
| **Mismatches** | **Mismatches** | | | | | |
| # mismatches | 7 178 903 | 7 186 647 | 7 200 413 | 7 782 616 | 7 459 690 | 7 897 370 |
| # indels | 709 703 | 712 521 | 578 950 | 663 900 | 601 187 | 664 809 |
| Indels length | 3 411 160 | 3 401 230 | 2 533 911 | 2 792 119 | 2 594 046 | 2 788 514 |
| # mismatches per 100 kbp | 3704.84 | 3707.23 | 4727.18 | 5085.22 | 4888.75 | 5161.54 |
| # indels per 100 kbp | 366.26 | 367.55 | 380.09 | 433.8 | 393.99 | 434.5 |
| # indels (<= 5 bp) | 586 126 | 589 684 | 472 547 | 549 170 | 492 487 | 549 678 |
| # indels (> 5 bp) | 123 577 | 122 837 | 106 403 | 114 730 | 108 700 | 115 131 |
| # N's | 222 000 | 47 000 | 160 500 | 273 500 | 30 500 | 88 500 |
| # N's per 100 kbp | 70.18 | 14.87 | 60.54 | 91.63 | 10.95 | 29.66 |
| **Statistics without reference** | **Statistics without reference** | | | | | |
| # contigs | 57 | 375 | 242 | 788 | 152 | 346 |
| # contigs (>= 0 bp) | 66 | 384 | 246 | 870 | 161 | 366 |
| # contigs (>= 1000 bp) | 66 | 384 | 246 | 870 | 161 | 366 |

| | | | | | | |
|---|---|---|---|---|---|---|
| # contigs (>= 5000 bp) | 52 | 373 | 242 | 763 | 149 | 342 |
| # contigs (>= 10000 bp) | 41 | 370 | 240 | 713 | 149 | 336 |
| # contigs (>= 25000 bp) | 36 | 365 | 229 | 590 | 149 | 334 |
| # contigs (>= 50000 bp) | 31 | 351 | 194 | 418 | 147 | 322 |
| Largest contig | 87 919 362 | 53 085 522 | 16 125 000 | 18 472 100 | 81 389 553 | 33 042 432 |
| Total length | 316 325 129 | 316 150 129 | 265 110 564 | 298 480 923 | 278 628 555 | 298 396 733 |
| Total length (>= 0 bp) | 316 343 571 | 316 168 571 | 265 117 083 | 298 613 175 | 278 643 557 | 298 428 175 |
| Total length (>= 1000 bp) | 316 343 571 | 316 168 571 | 265 117 083 | 298 613 175 | 278 643 557 | 298 428 175 |
| Total length (>= 5000 bp) | 316 308 034 | 316 142 034 | 265 110 564 | 298 385 985 | 278 617 151 | 298 381 068 |
| Total length (>= 10000 bp) | 316 245 926 | 316 119 926 | 265 098 861 | 298 027 017 | 278 617 151 | 298 338 640 |
| Total length (>= 25000 bp) | 316 166 532 | 316 043 547 | 264 892 037 | 295 855 943 | 278 617 151 | 298 303 942 |
| Total length (>= 50000 bp) | 316 019 783 | 315 501 298 | 263 899 871 | 289 998 342 | 278 527 354 | 297 812 403 |
| N50 | 53 705 741 | 27 116 470 | 5 536 397 | 5 542 566 | 71 432 794 | 18 845 809 |
| N75 | 35 677 594 | 2 433 575 | 2 400 000 | 1 467 468 | 23 935 993 | 1 368 796 |
| L50 | 3 | 5 | 15 | 17 | 2 | 6 |
| L75 | 5 | 12 | 34 | 39 | 4 | 28 |
| GC (%) | 43.77 | 43.77 | 44.41 | 44.2 | 44.32 | 44.19 |
| **K-mer-based statistics** | **K-mer-based statistics** | | | | | |
| K-mer-based compl. (%) | 26.21 | 26.21 | 12.51 | 12.28 | 43 963,00 | 12.28 |
| K-mer-based cor. length (%) | 0.52 | 7.24 | 19.16 | 44 154,00 | 5.43 | 43 962,00 |
| K-mer-based mis. length (%) | 98.58 | 83.42 | 76.37 | 68.07 | 86.92 | 77.92 |
| K-mer-based undef. length (%) | 0.9 | 9.33 | 4.47 | 12.82 | 7.65 | 43 872,00 |
| # k-mer-based misjoins | 1 194 | 985 | 331 | 291 | 282 | 266 |
| # k-mer-based translocations | 593 | 547 | 218 | 187 | 212 | 202 |
| # k-mer-based 100kbp relocations | 601 | 438 | 113 | 104 | 70 | 64 |

Table.5 Quast-lg reports for automated scaffolding results.

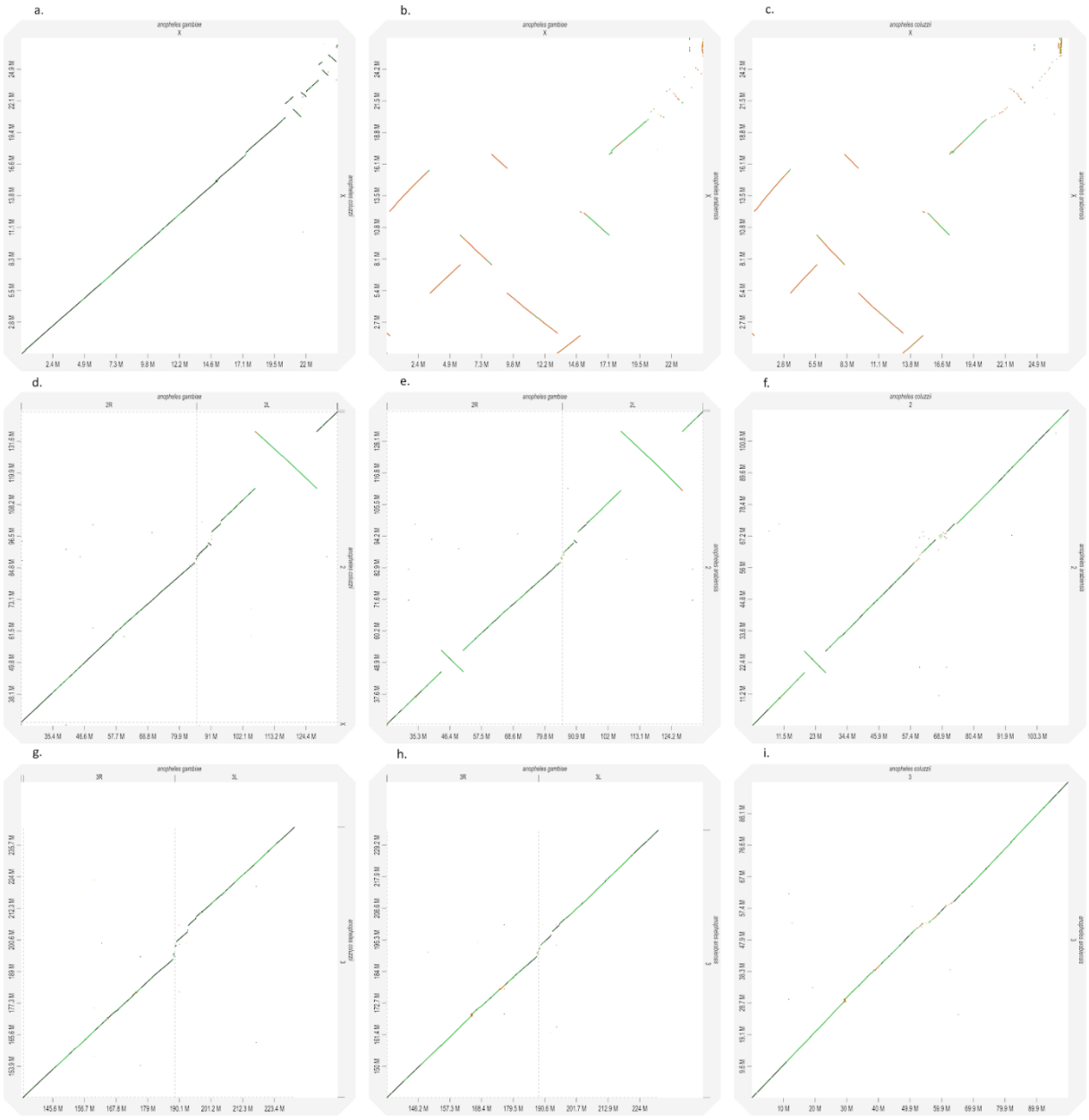


Fig.4. D-genies dot-plots of pairwise alignment separated by chromosomes

**APPENDIX B.**

| | | BUSCO arthropoda | | | | | | BUSCO eukaryota | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | assemblies | complete | single | duplicated | fragmented | missing | number | complete | single | duplicated | fragmented | missing | number |
| | | | | | | | | | | | | | |
| A.amphitrite | mil_CANU | 95.8% | 26.3% | 69.5% | 1.7% | 2.5% | 1066 | 95.3% | 22.4% | 72.9% | 0.7% | 4.0% | 303 |
| | FALCON draft | 92.7% | 61.0% | 31.7% | 3.4% | 3.9% | | 92.1% | 63.4% | 28.7% | 3.6% | 4.3% | |
| | FALCON_unzip_all | 95.3% | 23.5% | 71.8% | 1.7% | 3.0% | | 95.7% | 19.8% | 75.9% | 0.3% | 4.0% | |
| | FALCON_unzip_prim | 95.4% | 36.9% | 58.5% | 1.6% | 3.0% | | 95.7% | 36.0% | 59.7% | 0.7% | 3.6% | |
| | FALCON_unzip_pil1 | 95.5% | 36.7% | 58.8% | 1.5% | 3.0% | | 95.4% | 35.0% | 60.4% | 0.7% | 3.9% | |
| | FALCON_unzip_pil2 | 95.5% | 36.6% | 58.9% | 1.5% | 3.0% | | 95.4% | 34.0% | 61.4% | 1.0% | 3.6% | |
| | FALCON_unzip_pil3 | 95.5% | 36.7% | 58.8% | 1.5% | 3.0% | | 95.4% | 34.3% | 61.1% | 1.0% | 3.6% | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| goose_neck | dovetail_FALCON_ARROW | 92.0% | 58.3% | 33.7% | 1.8% | 6.2% | 1066 | 91.7% | 58.4% | 33.3% | 3.0% | 5.3% | 303 |
| | FALCON | 67.5% | 63.7% | 3.8% | 8.1% | 24.4% | | 66.3% | 64.0% | 2.3% | 9.2% | 24.5% | |
| | CANU_ctg | 95.2% | 56.5% | 38.7% | 1.7% | 3.1% | | 93.7% | 57.1% | 36.6% | 2.0% | 4.3% | |
| | | | | | | | | | | | | | |
| goose_neck | dovetail_Chicago | 92.2% | 69.6% | 22.6% | 1.8% | 6.0% | 1066 | 92.1% | 72.6% | 19.5% | 2.6% | 5.3% | 303 |
| | dovetail_after_HIC | 92.3% | 71.1% | 21.2% | 1.8% | 5.9% | | 92.5% | 74.3% | 18.2% | 2.0% | 5.5% | |
| | dovetail_purge_haplo | 92.5% | 79.7% | 12.8% | 1.9% | 5.6% | | 90.8% | 80.2% | 10.6% | 2.6% | 6.6% | |
| | dovetail_gapclosed | 92.3% | 71.1% | 21.2% | 1.8% | 5.9% | | 92.5% | 74.3% | 18.2% | 2.0% | 5.5% | |

Table.6 BUSCO results for barnacles assemblies.