

Auckland ICT Graduate School

## Internship Final Report

February 2023

Automating and Optimising Genome Assembly  
Process using Nextflow

Author: Qiong Zhou

Student ID: 365217677

Project supervisor: Joerg Simon Wicker

Plant & Food Research & Chen Wu

## **Declaration of Originality**

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Name: Qiong Zhou

**Abstract**—The quest for high-quality reference genomes for newly discovered species remains an ongoing challenge, with assembling chromosome-length sequences remaining difficult. However, recent advances in chromatin conformation capture (Hi-C) have introduced a new approach to scaffold genome assemblies. Despite their widespread use, there is currently no comprehensive evaluation method to determine which Hi-C scaffolder performs best for a particular species under specific conditions. In this article, a Hi-C scaffolding benchmarking pipeline using Nextflow is introduced, an automation tool in bioinformatics, to automate and optimize the bioinformatics workflow and experiments. The pipeline is designed to evaluate and compare the performance of different tools and strategies for Hi-C scaffolding, which can help researchers identify the most effective methods for their specific species under certain conditions. The pipeline has significant potential to contribute to the field of de novo genome assembly and offers several advantages, including its scalability, flexibility, and the ability to explore 3D genome structure. Overall, the development of this Hi-C scaffolding benchmarking pipeline has the potential to improve the accuracy and completeness of de novo genome assembly, facilitating more detailed and comprehensive analysis of the genome. I also reflects on the soft and technical skills learned during the industry project.

**Keywords**—Hi-C, Nextflow, Automated pipeline, Genome assembly, Scaffolding

## I. INTRODUCTION

THIS document is a report on a ten-week internship project which aims to automate and optimise the genome construction process in the company of Plant and Food Research.

### *About Plant and Food Research*

The company I participated in the industry project experience with was the New Zealand Plant and Food Research Institute (PFR), which is one of seven Royal Institutes in New Zealand and is jointly funded by the government and investors.

PFR is a New Zealand-based company that conducts research and development in the areas of plant and food science. The company has a strong focus on horticulture, viticulture, and aquaculture, and its work aims to improve the productivity, sustainability, and competitiveness of these industries. PFR has 15 research institutes, including its headquarters in Auckland Mt Albert, and four focal points in Australia and the USA. The company works closely with growers, producers, and other industry partners, as well as with government agencies and academic institutions, to support the growth and success of the country's plant and food sectors. For example, the company's bioinformatics analysis department studies the genetic information of different fruits or plants to try to make some genetic changes or improvements to some fruit to achieve a better taste, and then tries to increase their yield until it is stable and put on the market, such as the red heart kiwi. Genetic improvements are also worked on to improve indigenous plants to make them non-toxic and tasty, thus creating new varieties of indigenous fruit.

At PFR, it is believed that science can create a better future. PFR uses world-leading science to improve the way growers, fishers, harvesters, creators, and food preparers grow, fish, harvest, create, prepare, and share food. Every day, the

company has 1000 people working across Aotearoa New Zealand, and the world to help deliver healthy foods from the world's most sustainable systems. By finding smarter, greener options, PFR is helping secure the world it wants to live in tomorrow. The company works with partners to achieve these goals.

### *About the project I did*

The quest for high-quality reference genomes for newly discovered species is an ongoing challenge, and assembling chromosome-length sequences remains a difficult task due to sequencing machines cannot sequence a whole chromosome-length genome. However, recent advances in chromatin conformation capture (Hi-C) have introduced a new approach to scaffold genome assemblies. In the last decade, numerous Hi-C scaffolding methods have been developed and applied to different species. Despite their widespread use, there is currently no comprehensive evaluation method to determine which Hi-C scaffolder performs best for a particular species under specific conditions. A more serious problem is that data pre-processing, mapping, and aligning read data to sequences of pre-struts, constructing scaffolds and benchmarking them using different scaffold tools, and evaluating the output results of different scaffolds requires a significant investment of bioinformatician manual and time costs, with a high potential for human error.

For this project, I used Nextflow, an automation tool in bioinformatics, to create an automated pipeline to automate and optimise this bioinformatics workflow and experiments. A literature review was also conducted to identify the most popular Hi-C scaffolds, including Ychs and SALSA2, and implemented them into the benchmarking of this automated pipeline scaffold. An algorithmic calculation and objective evaluation of which Hi-C scaffolder performs best under specific conditions for a particular species were implemented at the end.

## II. LITERATURE REVIEW

Given that I do not have any biological knowledge or bioinformatics background, I started learning from the basics and therefore did a lot of literature survey and review to be able to understand the deep bioinformatics background and logic of the whole project and thus to be able to follow the plan in an organized manner to move the project forward. As in the following literature review chapters, starting from the bioinformatics foundation, I need to understand what is a genome, how a reference genome is constructed, what is ab initio gene assembly, why do gene assembly instead of reading the whole genome directly with a sequencer, what is a scaffold, how does a scaffold perform gene assembly and construct a high quality reference genome, what is Hi-C technology How Hi-C revolutionized genome assembly and scaffolding and why we automated this process of genome assembly scaffolding and used Nextflow as an automation tool to build the automation pipeline.

High-quality and complete reference genome assembly is essential for the application of genomics to biology, disease, and biodiversity conservation. Genome assembly is the process

of putting nucleotide sequences into the correct order and orientation. Although current third-generation sequencing technologies, such as Oxford Nanopore and PacBio HiFi [2], [34], produce much longer nucleotide fragments than previous technologies, significantly improving the continuity of assembly. However, long-range linked data is still required to accurately bridge gaps in assembly to enable the accurate assembly of large and complex plant and animal genomes at the chromosomal level.

Chromosome conformation capture techniques that have been developed, such as Hi-C, using proximity ligation and massively parallel sequencing to map chromatin interactions and infer the three-dimensional structure of chromosomes within the nucleus. Hi-C data can be used to determine the orientation and order of allelic genomes and assemble them into chromosomes by generating interaction frequency data for large genomic distances [18], [41], [50], [51].

### 1) *What is a Genome?*

A genome can be defined as the complete genetic information that an organism carries in its DNA, it provides all the information needed for an organism to function [35]. Genomes of different species vary in size, complexity and organization [56]. With the advent of high-throughput sequencing technologies, it has become possible to sequence genomes at unprecedented speed and low cost [37]. The study of genomics has been a topic of interest for decades, and the complete sequencing of the human genome was a major milestone in the field, which has since been expanded to include a wide range of organisms, from bacteria to plants, and even to more complex organisms like mammals [3]. The knowledge gained from genome sequencing has led to major advances in areas such as personalized medicine, agriculture and biotechnology, for example, genomics has made it possible to identify genetic mutations in diseases, paving the way for personalized treatments for individual patients [8]. Similarly, the sequencing of crop genomes has led to the development of improved varieties with greater resistance to pests and diseases, resulting in higher crop yields [29]. Despite these achievements, much remains to be discovered in the field of genomics. The vast amount of genomic data that has been generated presents a challenge to researchers, and new analytical tools and methods need to be developed to make sense of this data. In addition, ethical considerations surrounding the use of genomic data, such as privacy and consent, must also be taken into account.

### 2) *How is a reference genome constructed?*

The construction of a reference genome is a key step in genomics research that involves the generation, assembly and refinement of high-quality DNA sequence data [47]. The goal of this process is to create a comprehensive and accurate representation of a species-specific genome that can be used as a reference for subsequent genetic analyses [49].

The first step in this process is to obtain a high-quality DNA sample from the organism of interest. The DNA is then segmented, and short segments of DNA are sequenced using

high throughput sequencing technology, resulting in millions to billions of short sequence reads [54]. The next step is to assemble these short reads into longer contiguous sequences, known as contigs [61]. However, the assembly of large and complex genomes is challenging, especially when many plant genomes are large, complex and at the high ploidy level [46].

To meet this challenge, researchers use various techniques, such as long-line sequencing, optical mapping or scaffolding, to sequence and position alleles to create a more complete and contiguous assembly [9], [30]. Scaffolding involves the use of additional data, such as Hi-C or chromatin conformation capture data, which can help determine the relative spatial organization of different alleles in the genome. Automation of the scaffolding process is important to streamline the genome assembly process, reducing the time, computational resources and manpower required.

Once a high-quality reference genome is constructed, it becomes a valuable resource for researchers to perform a wide range of genetic analyses such as gene annotation, comparative genomics and variant calling [10]. Providing a high-quality reference genome is critical for downstream analyses, especially in the case of large-scale studies. Therefore, streamlining the genome assembly process, especially automating the scaffolding process, can help reduce the cost and time required to construct high-quality genomes, which is critical to advancing genomics research.

### 3) *What is de novo genome assembly?*

De novo genome assembly is the process of reconstructing a genome from its constituent sequencing reads without the aid of a reference genome [23], [45]. In recent years, the development of high-throughput sequencing technologies has made it possible to generate large amounts of sequence data in a relatively short amount of time, making de novo genome assembly an increasingly important and common task in genomics research. De novo genome assembly typically involves three main steps: read preprocessing, contig assembly, and scaffolding [44]. In the read preprocessing step, the raw sequencing reads are filtered and trimmed to remove low-quality reads and sequencing artifacts. In the contig assembly step, the filtered reads are assembled into contiguous sequences (contigs) using a variety of algorithms, including overlap-layout-consensus (OLC), de Bruijn graph (DBG), and hybrid approaches that combine both OLC and DBG methods [21], [31], [43], [53]. Finally, in the scaffolding step, the contigs are arranged and oriented into larger sequences (scaffolds) using additional information such as paired-end reads, mate-pair reads, optical mapping, and chromatin conformation capture (Hi-C).

### 4) *What is scaffolding?*

Scaffolding is an essential step in the process of constructing a high-quality reference genome [44]. The goal of scaffolding is to arrange the ordered and oriented contigs generated in the previous step of genome assembly into a more contiguous and complete assembly. Scaffolding is a critical step in the genome assembly process since it helps bridge the gaps between the

contigs, filling in missing information and producing a more accurate representation of the genome.

Scaffolding can be carried out using different approaches and techniques, including physical, genetic, and computational methods [44]. Physical approaches such as long-read sequencing or optical mapping provide long-range information about the genome's structure, allowing researchers to place contigs in the correct order and orientation. Genetic mapping can provide linkage information to anchor contigs onto the chromosome. These methods have been successful in scaffolding many genomes, especially those of organisms with smaller genomes.

### 5) Hi-C Technology

Recent advances in chromatin conformation capture (Hi-C) have revolutionized genome assembly and scaffolding, providing a new method for assembling chromosome-length sequences, which has proliferated in the last decade [5]. Hi-C provides long-range information by identifying physical interactions between distant genomic regions to help determine the relative position and orientation of contigs in genome assembly [48], [50]. By incorporating Hi-C data into the assembly process, researchers can reduce errors and gaps and improve the continuity and accuracy of the final genome. Hi-C technology is also useful for studying the three-dimensional structure of the genome and its dynamic nature, enabling researchers to gain insights into how the genome is organized and how it functions [36]. Various strategies for incorporating Hi-C data into *de novo* genome assembly have been developed, including using Hi-C data to guide scaffolding and allelic genome orientation, incorporating Hi-C data directly into the assembly process, and using methods designed specifically for incorporating Hi-C data into new genome assembly processes [24]. While other computational methods exist for scaffolding contigs, such as using information from paired-end sequencing or mate-pair sequencing, Hi-C provides a high-resolution, genome-wide view of the genome's structure, making it an essential tool for studying new species that require a high-quality reference genome. Several bioinformatics tools have been developed for scaffolding contig-level sequences and chromosomal assignment using Hi-C data, including Ychs [1], SALSA [27], LACHESIS [41], 3D-DNA [24], ALLHiC [18], instaGRAAL [15], EndHiC [4], and pin\_hic [11]. These programs use statistical approaches to calculate proximity scores, but they are primarily developed using human chromosome data and can be less sensitive when applied to non-model organisms, leading to a lack of scaffolded sequences. Nonetheless, Hi-C technology has the potential to provide unprecedented insights into the mechanisms underlying complex biological processes and is an essential tool for studying new species that require a high-quality reference genome.

The recent benchmarking paper for comparing a range of Hi-C scaffolders (as mentioned above, Ychs, SALSA, etc.) proposed that different tools perform differently on scaffolding datasets from different species [17]. The scaffolding accuracy also depends on the number of contigs, ploidy and the

complexity of the genome [28]. On the other hand, datasets generated from different library preparation protocols such as the traditional restriction enzyme-based Hi-C approach (i.e. cocktail enzyme combination from PhaseGenomics) and OmniC to obtain more complete contact matrices may lead to different scaffolding results [6]. Furthermore, the incorrectly joined sequences (incorrect orders and orientations) are frequently seen in the produced chromosomes of non-model organisms as Figure 1 (clear boards can be seen on the Hi-C heatmap when sequences are joined incorrectly) [26], especially plants, due to their genomic complexity, which are required to be assessed and curated. To encounter these obstacles, Juicebox [33] has been frequently used to visualize Hi-C heatmaps that facilitate the correction of scaffolding chimeras as well as the accomplishment of the final chromosomes; however, this manual curation method is time-consuming and may introduce human errors.

In order to overcome the aforementioned obstacles and streamline the genome scaffolding process, an automated pipeline has been implemented, built with Nextflow as introduced in the following section. The use of an automated pipeline can significantly improve the accuracy and efficiency of genome scaffolding by minimizing manual intervention, while also reducing the time required for the analysis. Thus, it enables researchers to process large amounts of data efficiently, while also reducing the risk of human error in the analysis process.

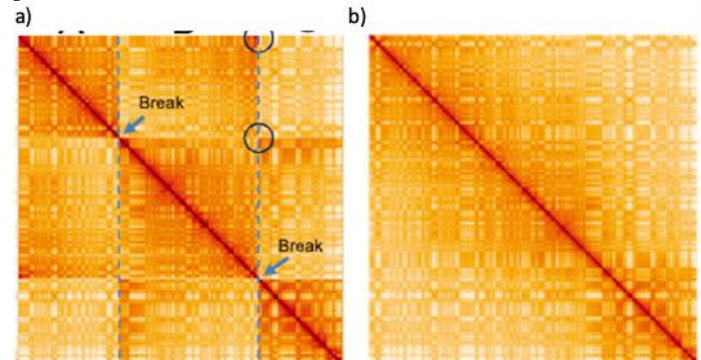


Figure 1. [26]

a) The high-intensity signal outside the diagonal (black circle) indicates the wrong combination of scaffold/contig.

b) Resolve the incorrect combination by breaking the scaffold at the coordinates marked by the arrows and then orienting the fragment correctly.

### 6) What is Nextflow? Why do we use it to build pipeline?

Nextflow is a powerful, flexible, and open-source workflow management system software tool that has gained popularity in the field of bioinformatics for building and running complex computational pipelines [20]. It utilises software containers to enable scalable and repeatable scientific workflows, particularly for bioinformatics experiments, greatly improving the reproducibility of bioinformatics experiments [16], [25], also, a valuable tool for researchers and developers who need to process and analyze large amounts of data.

In the context of workflow management, a task is a discrete computational step that is part of a larger workflow. The output of one task often serves as the input for another task [7]. Dependencies between tasks refer to the relationships that exist

between tasks, which dictate the order in which they need to be executed. One of the key features of Nextflow is its ability to manage these dependencies automatically, based on the input data and the dependencies between the tasks. This means that Nextflow can determine which tasks need to be run, and in what order, without requiring the researcher to manually specify these details, which saves a lot of time and effort for researchers who would otherwise have to manually manage these dependencies, potentially introducing errors or inefficiencies into the workflow. This is one of the most important features of Nextflow, which makes it easier for researchers to build and execute complex workflows, without having to worry about managing dependencies and ensuring that each task is executed in the correct order. Nextflow also comes with a wide range of built-in tools and features that make it easier to build and manage complex workflows, for example, it includes support for parallel execution, so that different parts of a workflow can be run simultaneously to speed up the overall execution time. It also includes support for caching and checkpointing, which can help to reduce the amount of time and resources needed to re-run workflows.

Its flexibility and portability make it suitable for use on different computing architectures such as cloud computing platforms, local workstations, and high-performance computing clusters. It's simple and intuitive domain-specific language (DSL) based on the Groovy [58] programming language makes it easy to use and learn. The DSL allows users to write workflow scripts in a way that is easy to read and understand, even for those with limited programming experience. Additionally, Nextflow's flexibility in supporting multiple programming languages, including Python [59], R [60], Bash [57], as well as Groovy [58], further broadens its utility, making it an excellent option for developers with varying programming language backgrounds.

Beyond these capabilities, Nextflow has also engendered a thriving community that is devoted to advancing scientific workflow management. The nf-core project is a community-led initiative aimed at developing and maintaining a collection of high-quality, reusable workflows on top of the Nextflow platform [14]. These workflows are extensively tested and validated by the community, ensuring users have access to reliable and high-quality workflows. The nf-core project also provides a forum for researchers and developers to share knowledge and collaborate on new workflow and pipeline development. This platform allows users to exchange best practices, seek advice, and collaborate on novel workflow designs tailored to their specific needs, driving insights and discoveries across scientific domains. Given its versatility and the expert support of the nf-core community, Nextflow has emerged as an indispensable tool for data-intensive scientific research.

### III. PROJECT GOAL

The goal of this project is to develop an automated pipeline to automate the scaffolding step of genome assembly. The pipeline will aim to select the best scaffolding tool suitable for

the input dataset, using a benchmarking process and optimized parameter settings that target non-model organisms. By automating this process, the pipeline should largely reduce the time required for manual curation.

The pipeline will include several stages, beginning with a data quality check, followed by mapping to contigs, scaffolding using multiple cutting-edge scaffolders, and producing final Hi-C heatmaps, which will serve as the scaffolding result. An automatic evaluation step based on Hi-C heatmaps will also be integrated into the pipeline.

The pipeline will be designed to be time-efficient, providing the best possible scaffolding result from computational methods. In addition, the pipeline will build and standardize an evaluation process as part of its workflow, and produce a comprehensive final report that includes computational usage.

By automating the scaffolding step of genome assembly, this project aims to reduce the potential for human error and minimize the time required for manual curation. By selecting the best scaffolding tool for the input dataset, the pipeline will improve the accuracy and efficiency of the scaffolding process, particularly for non-model organisms. The automatic evaluation step will further ensure that the final scaffolding result is of high quality, and the comprehensive final report will enable researchers to easily assess and interpret the results of the genome scaffolding analysis. Overall, this project aims to advance the field of genome assembly by developing an efficient, automated pipeline for scaffolding, which will facilitate research in various areas of genomics.

## IV. DESIGN AND METHOD

For this project, I utilized Nextflow to construct an automated pipeline. Groovy was utilized to write each process of the pipeline based on parallel programming. I utilized Python, R, and Bash for different processes based on the programming language the software tool(s) was developed with to achieve the desired functions. Each process was implemented as a function, employing distinct software tools within the process. To encapsulate all the tools used in the pipeline within one container file, I used a singularity container. The container was created using a stable Debian environment in Docker, which only need to be built once before running the pipeline, the automated pipeline streamlined chromosome-scale scaffolding process.

In the project, the Nextflow automated pipeline of chromosome-scale scaffolding of de novo genome assemblies are based on chromatin interactions (Hi-C data), which has been partitioned into three distinct stages to cater to diverse research requirements as Figure 2. Users are able to choose to stop at the end of any stage or run all the stages.

The initial stage involves the pre-processing of the raw dataset, wherein the quality assessment of the raw Hi-C data is performed. This process is primarily used to assess the success of the Hi-C library preparation and the worthiness of obtaining deep-sequencing data. The second stage involves the alignment of reads to contigs/scaffolds and scaffolding with specific scaffolders, as per the bioinformatician's preference. Finally,



the third stage comprises the scaffolder benchmarking process, which involves the utilization of multiple scaffolders to scaffold and evaluate the different results obtained. This stage provides information on the Hi-C scaffold that performs best under specific conditions for a particular species. After each execution of the automated pipeline, a comprehensive final report is generated. This report includes input data information (reads and contigs), such as file path, file size, statistical summary, heatmap evaluation score, etc.

The Hi-C method produces a dataset by sequencing genomic DNA fragments that have been crosslinked by formaldehyde, and these fragments are then processed into paired-end reads that are stored as two files, R1 and R2, are called long-range linked read pairs, in the .fastq.gz file format as shown in Figure 4. Both R1 and R2 files have the same format, as shown in Figure 5. The R1 and R2 files contain the forward and reverse reads, respectively, of the paired-end sequencing reads. Each read corresponds to a specific genomic region that is crosslinked to another region in the three-dimensional space of

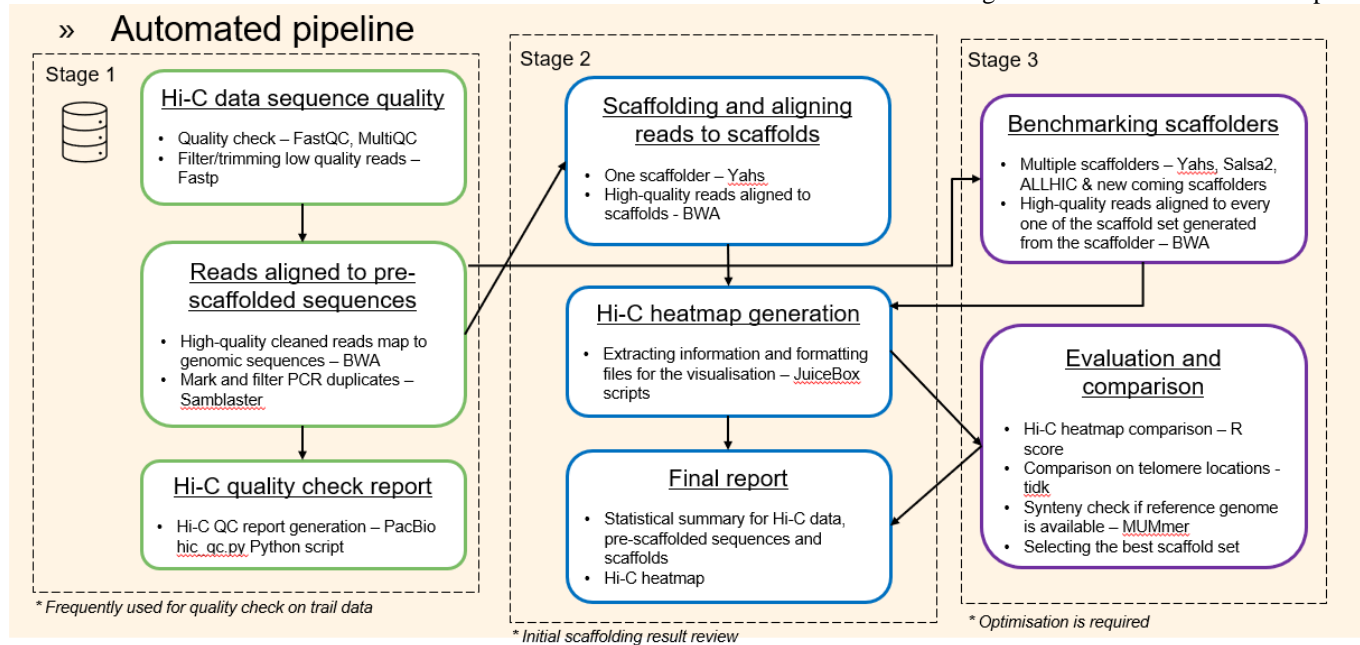


Figure 2. workflow of 3 stages in the automated pipeline

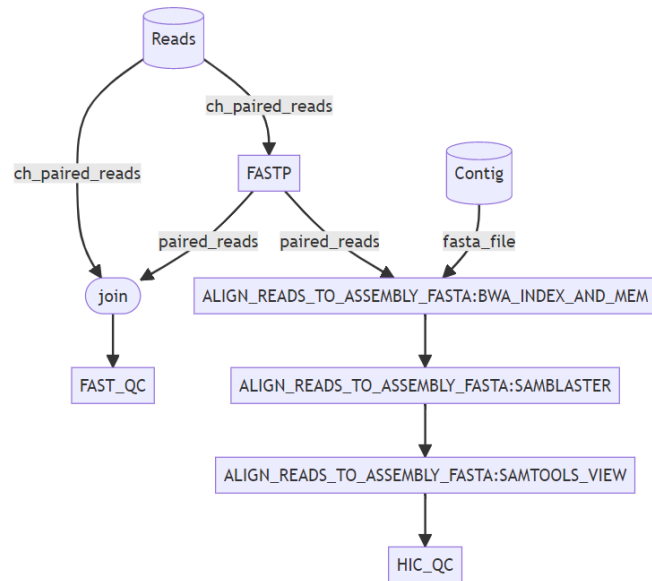


Figure 3. Stage 1 detailed processes in automated pipeline

### 1) Stage 1

Stage 1 is the pre-process for dataset, in detailed processes that have been run in the automated pipeline as shown in Figure 3. Start with 2 input files, R1 and R2 files. As mentioned before, Hi-C is a molecular biology technique that enables the study of three-dimensional chromatin interactions in a cell's nucleus.

the nucleus, as regions that are closer on the genome are generally closer in spatial structure. The paired-end reads are used to identify the crosslinked genomic regions and their interaction frequency, which can be represented as a contact matrix or heatmap. At stage 1 during Hi-C data analysis, the R1 and R2 files are first do quality check using FAST\_QC tool, then do pre-processed to remove low-quality reads and trim adapter sequences using FASTP tool as shown in Figure 3. With the input data R1 and R2 files, which first go through quality check process with FAST\_QC tool [52]. FASTQC is a widely used quality control tool in bioinformatics that performs a comprehensive analysis of high-throughput sequencing data [52]. The tool is designed to provide a detailed report on various quality control metrics to assess the quality of raw sequencing data before any downstream analyses are performed as shown in Figure 6. FASTQC takes the raw sequencing data in the .fastq or .fastq.gz format as input and generates a report that includes several metrics such as per-base sequence quality, sequence length distribution, GC content (The percentage of G and C nucleotides in the read sequence. GC content can impact the sequencing quality, as regions with extreme GC content may be difficult to amplify or sequence. Deviations from the expected GC content may also indicate contamination or bias in the data), overrepresented sequences, and adapter contamination (Adapter sequences are used in library preparation to enable sequencing, but they may also be present in the sequencing



[40]

```
[hraaxl@aklppj31 002.Fastp.trimming]$ zcat R1.cleaned.specifiedAdapter.short.fq.gz | head -n 10
```

	Header (read ID)	Read sequence
@A00121:176:HHVY2DRXX:1:2101:1642:1016 1:N:0:NTTGTA		CGATCTGATCCCTGCCACCGTCATCGACGCCCCACCTCTGGATTAAATTACAAGAAATCTTTTCAAC
+		FFF::FFFFFFFFFFFFFF:F
@A00121:176:HHVY2DRXX:1:2101:1660:1016 1:N:0:NTTGTA		CTTCTTGCTTGTTGAGCTTTTTGTATTATGGATGCTTTGCTCCTGCGGCCTCTTCTCTTTTCGACATAATATTAGTC
+		FFFFFFFFFFFFFF::FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF:FFF:F:,FFFFFFFFFFFFFF
@A00121:176:HHVY2DRXX:1:2101:1714:1016 1:N:0:NTTGTA		ATTTTGTCTAGGAATTCTTCTCATTCGTAGGTCAATTATGGCACGCGGGAAGGGCTCGTGACGCTGCAGCAGGATTTGAAAAGGAATTGATCGATCTTTCTTTTGCAATGCTTGTGATA
ACTCATTTGTAA		



Figure 5. input data R1/R2 file format

data and can affect the quality of downstream analyses, which can be identified and removed). These metrics are visualized as graphs, tables, and plots, which allow users to quickly identify potential issues with the data, such as low-quality reads, overrepresented sequences, or adapter contamination. The quality control analysis provided by FASTQC is an important step in the analysis of high-throughput sequencing data, as it can help researchers to identify and troubleshoot any issues with the data that may affect downstream analyses.

After we have detected the quality of raw dataset of R1 and R2, next process is improving the quality of those raw dataset using FASTP tool [21]. FASTP is designed to provide a fast and memory-efficient way to filter, trim, and correct sequencing reads, and to perform quality control analyses to improve the accuracy and reliability of downstream analyses. FASTP can perform a range of quality control analyses on the sequencing data, including evaluating sequence quality, sequence length distribution, and adapter content. It can also detect and remove

sequence or order of those fragments as shown in Figure 7. A contig refers to a sequence of DNA that has been constructed by assembling overlapping and oriented reads that share a subset or all of their nucleotide base pairs. This process involves combining two or more reads, aligning them to each other, and then extending the sequence to create a contiguous stretch of DNA. The goal is to create a complete, uninterrupted sequence that spans the region of interest, this process is typically performed using computational algorithms that take into account factors such as read quality, coverage, and overlap. Contigs are stored in .fasta format files as shown in Figure 8, the sequence data itself is then listed on subsequent lines, with each line containing a fixed number of nucleotides (usually 60 or 80). The sequence data is represented using single-letter codes that correspond to each of the four nucleotide bases: A, C, G, and T.

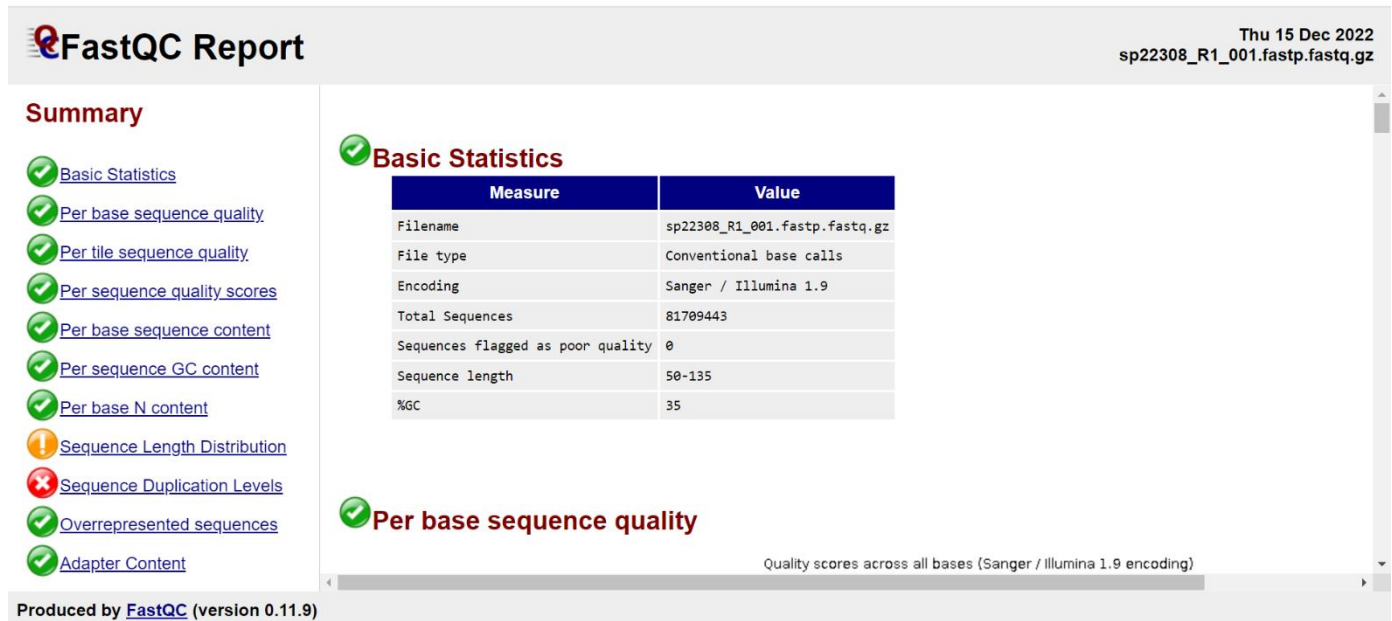


Figure 6. FASTQC quality check report

adapter sequences and low-quality reads from the data. Additionally, FASTP can correct sequencing errors, such as those introduced by sequencing platforms with high error rates. In contrast to other preprocessing tools, FASTP is designed to be highly efficient and can process sequencing data very quickly, even for very large datasets. It also supports parallel processing, allowing users to take advantage of multi-core CPUs and high-performance computing clusters to speed up preprocessing and analysis.

Then, after the FASTP step, which filtering out low quality data and improving the input dataset R1 and R2. We finally get cleaned R1 and R2 data, and we do the FAST\_QC quality check again, to make sure that the cleaned data quality is good enough to go to the next step.

Furthermore, the third input dataset is the contig/assembly file, which comes from De novo assembly method that is constructing genomes from a large number of (short- or long-) DNA fragments, with no a priori knowledge of the correct

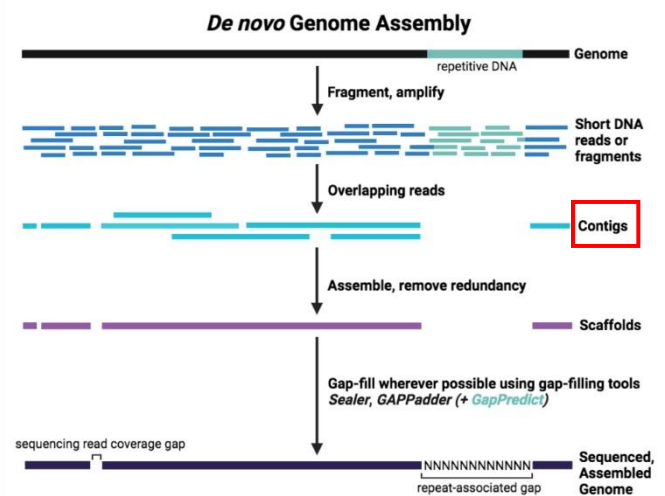
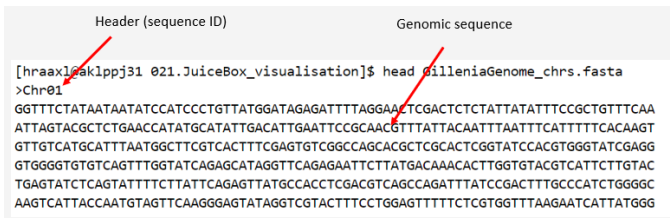


Figure 7. input data genomic contigs file generated from de novo genome assembly [12]



```
[hreaaxl@aklppj31 021.JuiceBox_visualisation]$ head milleniaGenome_chrs.fasta
>Chr01
GGTTTCTATAATAATCCATCCCTGTTATGGATAGAGATTTTAGGAACTCGACTCTCTATTATTTCCGCTGTTTCAA
ATTAGTACGCTCTGAACCATATGCATTTGACATTTGAATTCGCCAACGTTTATTACAATTTAATTTTCCACAGT
GTTGTGTCATGATTTAATGGCTTCGTCACTTCGAGTGTCCGCCAGCAGCTCGCACTCGGTATCCACGTGGGTATCGAGG
GTGGGGTGTGTCAGTTTGGTATCAGAGCATAGGTTTCAAGAGATTTCTATGACAAACACTTGGTGTACGTCATTTCTGTAC
TGAGTATCTCAGTATTTTCTTATTCAGAGTTATGCCACCTCGACGTCAGCCAGATTTATCCGACTTGGCCATCTGGGGC
AAGTCATTACCAATGTAGTTCAAGGGAGTATAGGTCGATTTCTCTGGAGTTTTCCTGCTGTTTAAAGATCATTATGGG
```

Figure 8. input data genomic contigs file format

With the input data of contig file, and cleaned dataset of R1 and R2 reads, the mapping process is carried out using BWA tool as shown in Figure 3. BWA (Burrows-Wheeler Aligner) is a software package to align DNA sequencing reads to assembly/contigs. BWA uses the Burrows-Wheeler Transform (BWT) to create an index of the assembly, which allows for fast and efficient searching of the assembly for matches to the sequencing reads. In the automated pipeline, we have used BWA MEM, which uses a series of heuristics and algorithms to efficiently, accurately and rapidly align the reads to the contig [42]. These include methods for handling gaps and mismatches in the read and the reference, as well as strategies for dealing with repetitive regions of the genome that can be difficult to align.



```
[hreaaxl@aklppj31 021.JuiceBox_visualisation]$ samtools view mapped_allhic_assembly.bam | head -n 10
A00121:176:HHVY2DRXX:1:2101:1136:1016 65 unique_mapped.Reduced.paired_only.counts_GATC.9g1 76922889 0
67M scaffold1,15927111,f1Z15927111 10170562 0 GAACGCACCAAGCTTTCGGTTCTAGGGCTATGCAATCTTATATGCAGCCAT
GTGGTCCGCC F:FFF:FFF::FFF,FFFFFFF,F:FFFFFFFFFFFF,FFFFFF,,F:FF:FFFFFFFFF::F NM:i:0 MD:Z:67 MC:Z:63M AS:i:63X
S:i:67 XS:i:67
A00121:176:HHVY2DRXX:1:2101:1136:1016 129 scaffold1,15927111,f1Z15927111 10170562 0 63M unique_m
apped.Reduced.paired_only.counts_GATC.9g1 76922889 0 TAAGATTGCATAGCCCTAGGCAGAAACCGAAAGCTTGGTGCCTTCACC
AAGTCCGGCGATC F:FF,FFF:F:FFF,F,FFF:FFFFFFF,F,FFFFFFF,FFFF,,F:FFF,FFF:,F:FF NM:i:0 MD:Z:63 MC:Z:67M AS:i:63X
S:i:63
A00121:176:HHVY2DRXX:1:2101:1298:1016 81 unique_mapped.Reduced.paired_only.counts_GATC.9g1 26007158 6
132M = 26007102 -188 TAGAACCAATAATCGAGTTGCATGCAATTAAGCGCATGATGCAATACATGATACATTGTCACCAAGATTGA
TCGATATTACTAAAAAGCAACCAATACAAACAAACATTGACAATGAACCTAAGTACTAT FFFFFFF:FFFFFFFF,,FFFF:FFFF,FFF,FFFF:F:,FFFFFFFFF:FF
:FFFFFF:FFFFFFF::FFFFFFFFFFFF::FF:,FFFF,FF:FFFFFFFFF::FFFFFF,FFFFFF:FFF NM:i:1 MD:Z:18G113 MC:Z:127M AS:i:127
S:i:127 XS:i:0
A00121:176:HHVY2DRXX:1:2101:1298:1016 161 unique_mapped.Reduced.paired_only.counts_GATC.9g1 26007102 6
127M = 26007158 188 TCGATCAATTTGTTTTGGCCAGAAACCAATAAATTCGTAAGGAAATAATTAGGTTAGAACCAATAATCG
AGGTGCATGCAATTAAGGCAGTGATGCAATACATGATGTCAGCAGATTG FFF:FF:FFFF:FFFFFF,,FF,F,,F,FFFF:FF:,FF:FF:,FFFFFFFF:F:F,,F
::FF:F,,FFFF,,FF:,F,,FFF,FFF::FFF:FFFF:FFFFFF,,FF:FFF NM:i:4 MD:Z:1T9T26C80C7 MC:Z:132M AS:i:110
XS:i:19
```

Figure 9. After cleaned R1 & R2 align to genomic contigs (combine 3 input files together)

Then, the next step is to identify duplicates (PCR duplicate is a term used in genomics research to describe multiple identical copies of a DNA fragment that result from the Polymerase Chain Reaction) from these files using SAMBLASTER tool, which can be a common problem in sequencing experiments, and need to be removed because they can artificially inflate the number of reads, leading to over-representation of certain genomic regions and skew downstream analyses [32], [39]. After obtaining the output .sam format file, we have applied SAMTOOLS to convert the .sam file to a deduplicated .bam file as shown in Figure 9, which includes options for filtering out reads likely to be PCR duplicates and can improve the accuracy and reliability of downstream analyses [63].

With the Hi-C data in .bam format file, the quality control process is carried out using HiQC tool, which is a method for studying the three-dimensional structure of genomes by measuring the frequency of interactions between different

genomic regions [13]. HiQC provides a comprehensive analysis of Hi-C data to identify technical artifacts and biases, which can be critical for downstream data analysis and interpretation [13]. The tool implements a set of standard quality control metrics, including the number of valid interaction pairs, the distribution of interaction distances, and the percentage of reads mapping to the genome. HiQC can also perform normalization and visualization of Hi-C data, which can help to identify systematic biases and ensure that data from different samples can be compared accurately.

This is the end of stage 1, with the HiQC report, before scaffolding process, which we can know how many linked R1 and R2 contigs are useful to do scaffolding. Bioinformaticians will make a decision whether to go to next stage according to Hi-C data's quality by looking at the report. The HiQC result clarified with whether the bad result (heatmap) after scaffolding is caused by the data itself, or the scaffolder.

## 2) Stage 2

Stage 2 is the scaffolding step with specific scaffolder(s) according to user's preferences as shown in Figure 10. In this case, we have choose Yabs as our Hi-C scaffolding tool, as

Yabs is a software tool used for scaffolding genome sequences using Hi-C data [1]. There are 5 input files need to be provided for Yabs, as shown the code of Yabs process from automated pipeline in Figure 11.

There are 5 input files required, including contig/assembly file, assembly index file, a Hi-C .bam file which is the output from stage 1 reads map and alignment, the optional parameters using in Yabs tool and a prefix variable for the output file. The contig/assembly file is the initial input file which has also been used at stage 1; The assembly index file is generated using the SAMTOOLS, the command FAIDX provides an index of the locations of each contig within the reference sequence, as well as their lengths; The Hi-C .bam file from stage 1 contains information about the frequency and strength of physical interactions between different regions of the genome; The variable of Yabs parameter settings are used inside the Yabs tool; The prefix for the output file specifies the name of the output file that Yabs will generate.

Yabs uses a Bayesian approach to infer the most likely ordering and orientation of the scaffolds based on the frequency

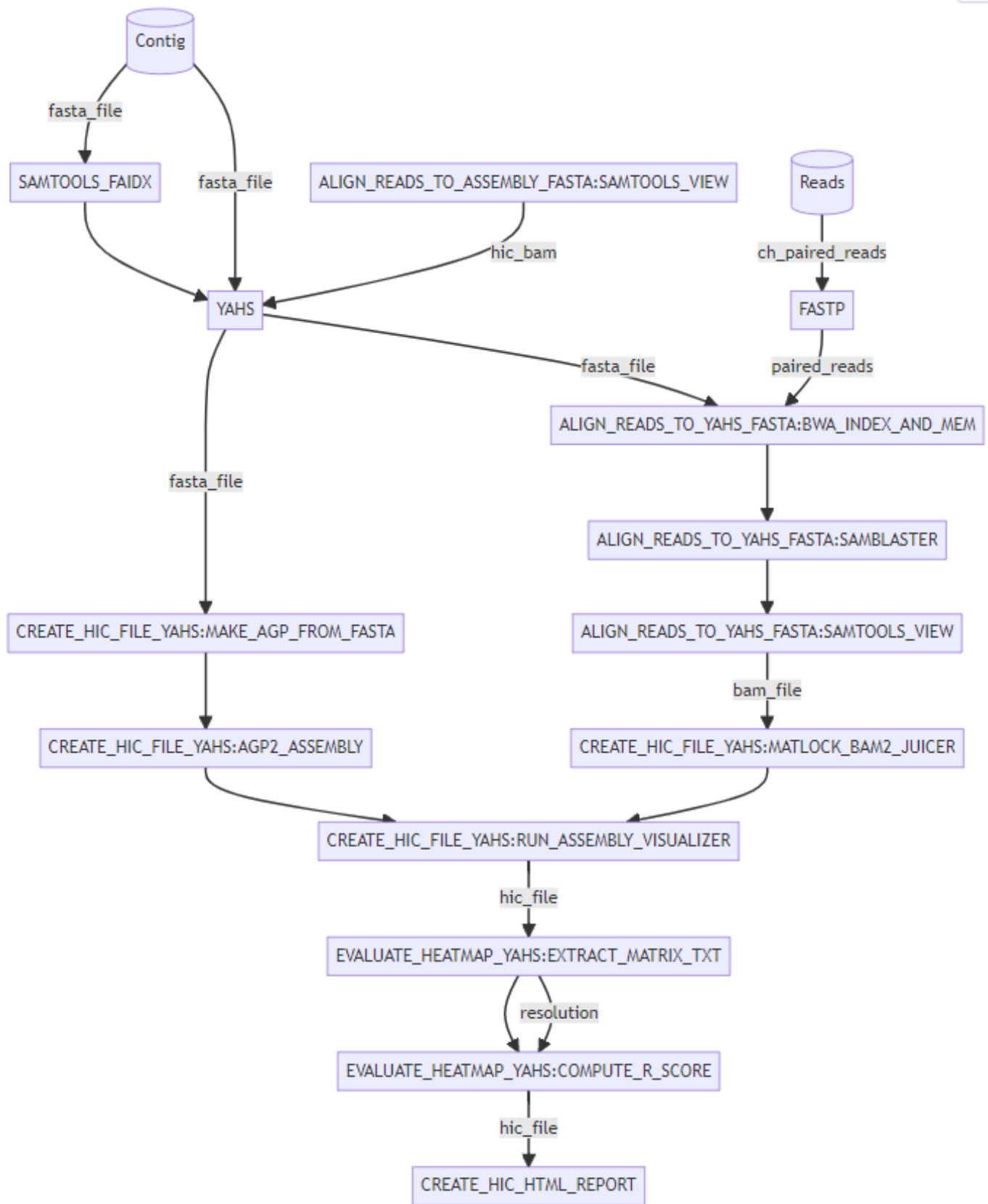


Figure 10. Stage 2 detailed processes in automated pipeline

```

23 lines (18 sloc) | 442 Bytes
1  nextflow.enable.dsl=2
2
3  process YAHs {
4
5      publishDir params.outdir.yahs, mode:'copy'
6      container "quay.io/biocontainers/yahs:1.2a.2--h7132678_0"
7
8      input:
9          path assembly
10         path assembly_index
11         file hic_bam
12         val yahs_params
13         val prefix
14
15         output:
16             path '*_final.agp', emit: agp_file
17             path '*_final.fa', emit: fasta_file
18
19         script:
20             """
21             yahs $assembly $hic_bam ${yahs_params} -o ${prefix}_yahs
22             """
23     }

```

Figure 11. Code of Yahs scaffolding

and strength of the Hi-C interactions between them. It uses the information from the Hi-C .bam file to identify contigs or scaffolds that are in close physical proximity and uses this information to order and orient them in the final genome assembly. It is also super-fast, for example, Yahs takes less than 5 minutes to reconstruct the human genome from an assembly of 5,483 contigs with ~45X Hi-C data. The output of Yahs is a more contiguous and accurate genome assembly, with fewer gaps and misassemblies than the initial assembly, which is in .fasta format.

After running Hi-C scaffolder Yahs to generate an improved genome assembly in the .fasta format, then we do the map and alignment cleaned reads of R1 and R2 again on to this better genome assembly file. This process involves converting the .fasta file into a BAM file, and then processing the BAM file to generate the Hi-C interaction matrix. Therefore, we use the BWA, SAMBLASTER and SAMTOOLS software tools again as we did at stage 1, BWA aligns the Hi-C reads to the improved genome assembly to generate a BAM file that contains the alignments of the Hi-C reads. The BAM file can then be processed using tools such as SAMBLASTER and SAMTOOLS\_VIEW to remove duplicate reads and filter out low-quality reads, which can improve the accuracy of the Hi-C contact map. Once the BAM file has been processed, the MATLOCK BAM JUICER tool can be used to generate a Hi-C interaction matrix, this involves counting the number of Hi-C read pairs that connect each pair of genomic loci and aggregating these counts into a matrix. Also, with .assembly file is generated from the output of the .agp file conversion step from the 3D-DNA pipeline using MAKE\_AGP\_FROM\_FASTA and AGP2ASSEMBLY tool [24]. The .agp file is a file format used to represent the order and orientation of contigs within a genome assembly from Yahs scaffolding output as shown in Figure 10. The .assembly file contains additional information required by the visualizer script to generate the 3D model as shown in Figure 12, such as the positions of centromeres and telomeres. Then, the resulting Hi-C interaction matrix can be visualized and further analyzed

using tools such as the RUN ASSEMBLY VISUALIZER.sh from 3D-DNA pipeline, which can generate a 3D representation of the genome structure based on the Hi-C contact map.

```

[hraaxl@aklppj31 021.JuiceBox_visualisation]$ head bilberry_allhic_assembly.assembly
>unique_mapped.REduced.paired_only.counts_GATC.9g1 1 121644197
>unique_mapped.REduced.paired_only.counts_GATC.9g2 2 5155772
>unique_mapped.REduced.paired_only.counts_GATC.9g3 3 1436272
>unique_mapped.REduced.paired_only.counts_GATC.9g4 4 1252963
>unique_mapped.REduced.paired_only.counts_GATC.9g5 5 915379
>unique_mapped.REduced.paired_only.counts_GATC.9g6 6 295782
>unique_mapped.REduced.paired_only.counts_GATC.9g7 7 50589
>unique_mapped.REduced.paired_only.counts_GATC.9g8 8 45483
>unique_mapped.REduced.paired_only.counts_GATC.9g9 9 42360

```

Figure 12. Contig information .assembly file

The Hi-C contact map/heatmap is the output from previous step saved as a matrix in .hic format. With the heatmap, there was no automated standard evaluation method or tool published, instead, user has to make the decision based on visualising Hi-C heatmaps as shown in Figure 13, which is the Hi-C contact map/heatmap from Karaka, which is the local plant/fruit from New Zealand [62]. To understand the heatmap, from following criteria:

- Interpret the axes: The x and y axes of a Hi-C contact map represent the positions of genomic regions along the genome. Typically, the genome is divided into evenly sized bins, and the x and y axes represent the centers of each bin. The size of the bins depends on the resolution of the Hi-C experiment, with higher resolution experiments having smaller bin sizes.
- Interpret the color scale: The color of each pixel in a Hi-C contact map represents the frequency of interactions between the corresponding genomic regions. The intensity of the color indicates the strength of the interaction between the regions, with darker shades of red indicating stronger interactions. White pixels in the Hi-C contact map generated by Juicebox represent regions where no interactions were detected. These are regions where the genomic loci are not in close physical proximity to each other, or where the interactions are too weak to be detected with the current experimental parameters.
- Identify features: When identifying features in a Hi-C map, it is important to search for biologically relevant patterns. In particular, we search for distinct patterns of "diagonal stripes" that extend from the upper left to the lower right of the map. These stripes correspond to frequent interactions between genomic regions that are in close proximity to each other. Additionally, we may observe "off-diagonal stripes" that represent interactions between distant genomic regions. Another key feature to look for is clusters or "blobs" of color that indicate specific interactions between genomic regions, such as interactions between enhancers and their target genes.



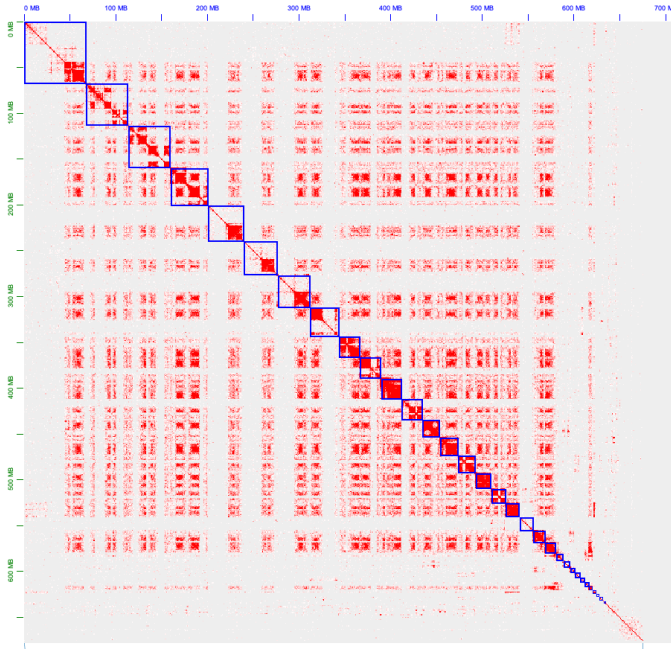
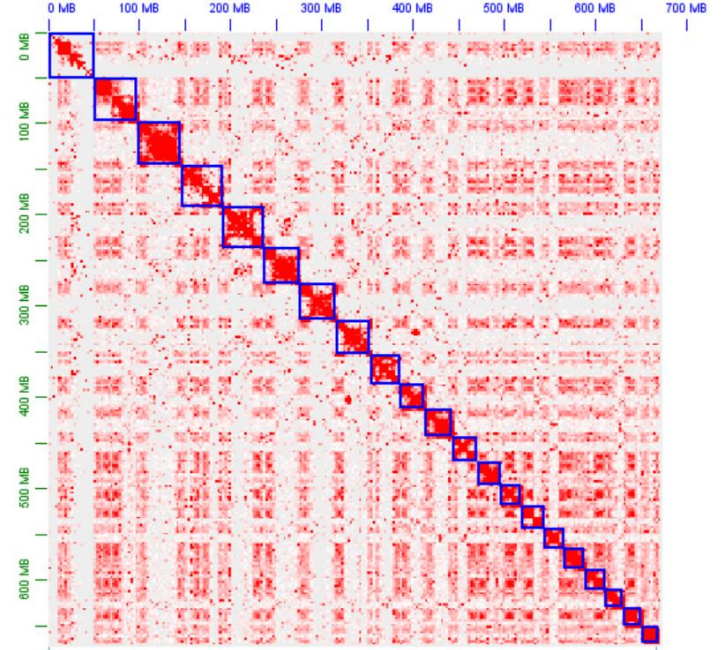


Figure 13. Left is the bad Hi-C heatmap from Karaka, right side is the better heatmap from Karaka

By utilizing the evaluation criteria designed for Hi-C heatmaps, we can discern the quality of each map, as depicted in Figure 13. Specifically, the heatmap on the left side is deemed inferior, given the presence of some prominent off-diagonal signals, which indicates a heightened frequency of interactions between the corresponding genomic regions, suggesting a proximity between them, those genetic regions should be put together into right order and orientation. On the other hand, the heatmap on the right side is considered superior, as it displays a stronger diagonal signal in conjunction with weaker off-diagonal signals, suggesting a high-quality scaffolding result.

To automate this Hi-C heatmap evaluation process, there is one potential algorithm can be applied, involves computing the correlation coefficient between the rows and columns of the heatmap matrix [38]. As the entries in the matrix represent frequencies, with proximity to the diagonal being an indicator of a high-quality heatmap, computing the correlation coefficient allows for assigning credit for being close to the diagonal. The matrix is generated by repeatedly generating a pair of numbers  $x$  and  $y$ , and incrementing the count of the matrix entry at position  $(x, y)$ . Viewing  $x$  and  $y$  as samples of random variables  $X$  and  $Y$ , respectively, the sample correlation coefficient  $r$  of  $X$  and  $Y$  ranges between  $-1$  and  $1$ . The coefficient is  $1$  if  $X$  and  $Y$  are perfectly correlated, and  $-1$  if they are perfectly anticorrelated. Specifically, the matrix entries tend to be near the diagonal when  $X$  and  $Y$  are perfectly correlated, which corresponds to a strong correlation. Importantly, this approach is robust, as the correlation coefficient remains unchanged even if the matrix is scaled. Notably, the formula for the correlation coefficient holds even when the matrix entries are non-negative real numbers.

Upon adapting the formulas presented in the aforementioned reference to the current situation, the resulting expressions can be expressed as follows. Let the heatmap matrix  $A$  be a  $d \times d$



matrix, let  $j$  be the  $d$ -long vector of all ones, and let  $r = (1, 2, \dots, d)$  and  $r_2 = (1^2, 2^2, \dots, d^2)$ . Then:

$$n = jA j^T \text{ (The sum of the entries of } A \text{)}$$

$$\sum x = rA j^T$$

$$\sum y = jAr^T$$

$$\sum x^2 = r_2A j^T$$

$$\sum y^2 = jAr_2^T$$

$$\sum xy = rAr^T$$

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{n \sum x^2 - (\sum x)^2} \sqrt{n \sum y^2 - (\sum y)^2}}$$

This algorithm is implemented by python code and added into the automated pipeline, so that we can the efficient batch-processing of a large number of Hi-C heatmaps and to automatically score them.

The final step of stage 2 as shown in Figure 10 involves the generation of a comprehensive report that furnishes users with requisite information and statistical summaries, inclusive of a comprehensive summary of the computational usage pertaining to the entire automated pipeline. Herein, an instance of the final report is illustrated, generated by utilizing WiX [55], <https://qzho906.wixsite.com/hic-pipeline-stage2>. Notably, the final report is produced in .html format, and a small full-stack web development program has been integrated into the automated pipeline to accomplish this. Unfortunately, I am unable to provide an actual example of the integrated program and actual final report due to confidentiality restrictions.

### 3) Stage 3

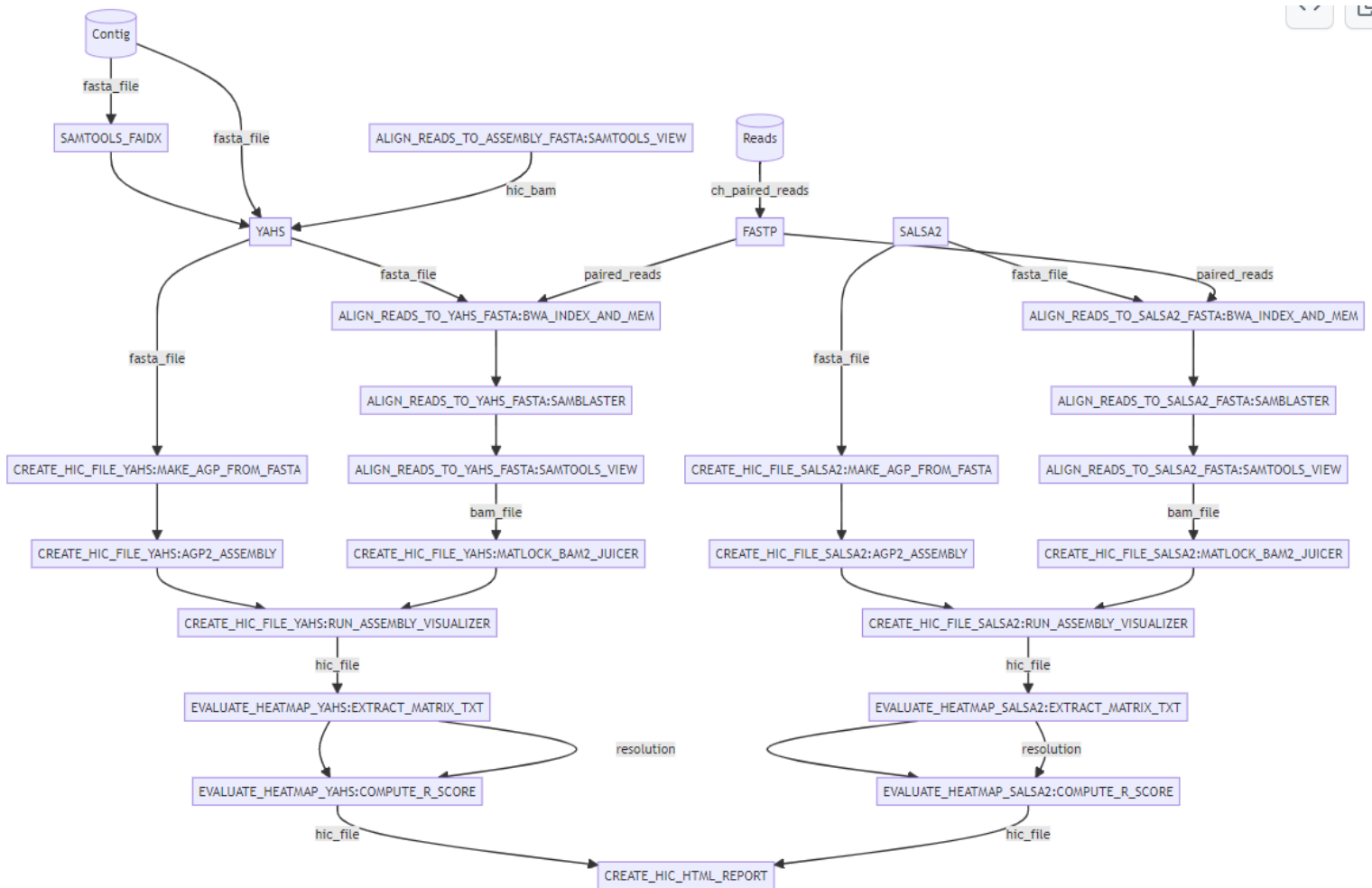


Figure 14. Stage 3 detailed processes in automated pipeline

In Stage 3 of the automated pipeline, scaffolding benchmarking is conducted using multiple scaffolders, including Yahs [1] and Salsa2 [19] as shown in Figure 14. The implementation of other scaffolders is similar to that of Yahs, which was introduced in Stage 2. It is important to note that the input and output data formats or types for different scaffolders may differ, and therefore require proper formatting prior to running other scaffolders and any subsequent steps following the scaffolding process.

Different Hi-C scaffolders, for example, Yahs, Salsa2, ALLHiC, etc. are genome scaffolding algorithms that utilize Hi-C data to determine the order and orientation of contigs in a genome assembly. However, they differ in their approach to this task. For example, Yahs uses a hierarchical clustering algorithm, which is a bottom-up approach to clustering where small clusters are progressively merged to form larger ones. In the case of Yahs, the algorithm clusters contigs based on their Hi-C interaction frequencies, which are a measure of the physical proximity of contigs in the 3D space of the nucleus. The algorithm starts by creating individual clusters for each contig and then merging them iteratively until a single cluster that contains all the contigs is formed. This hierarchical clustering approach allows Yahs to detect and resolve conflicts between contigs in the assembly that may arise due to repetitive

regions or misassemblies.

Whereas, Salsa2 uses a combinatorial optimization algorithm, which is an approach that searches for the optimal solution to a problem by considering all possible combinations of its constituent elements. In the case of Salsa2, the algorithm searches for the best arrangement of contigs that fits the Hi-C data by evaluating different contig orders and orientations and choosing the one that maximizes the number of Hi-C interactions between adjacent contigs. This optimization approach allows Salsa2 to accurately place contigs in the assembly even in regions with complex genomic structures or low Hi-C coverage.

The final report generated from the stage 3 is slightly different, compare with stage 2's. The benchmarking report add on information about scaffolders and relevant statistical summaries. An example of such a report can be viewed at <https://qzho906.wixsite.com/hi-c-pipeline-final>. Users can easily compare the output generated by different scaffolders and determine the most appropriate one for a particular species under specific conditions, such as varying scaffold settings and input data from different lab protocols.

The automated pipeline is running as Figure 15, we can see that the whole automation pipeline is successfully completed, also the final result of heatmap and evaluation score is generated in the final comprehensive report.



consistent and comparable across different studies, promoting

```

[hraaxz@aklppf31 hic-scaffolding-pipeline]$ ./main.nf -resume
N E X T F L O W ~ version 22.10.4
Launching `./main.nf` [adminiring_ptolemy] DSL2 - revision: 1cd5e7d880
[85/f618a6] process > FASTP (HiC8-lib_S1_L001_R) [100%] 1 of 1, cached: 1 ✓
[b1/b39e3f] process > FAST_QC (HiC8-lib_S1_L001_R) [100%] 1 of 1, cached: 1 ✓
[1c/6260b1] process > ALIGN_READS_TO_ASSEMBLY_FASTA:BWA_INDEX_AND_MEM (HiC8-lib_S1_L001_R) [100%] 1 of 1, cached: 1 ✓
[03/7b9d7d] process > ALIGN_READS_TO_ASSEMBLY_FASTA:SAMBLASTER (1) [100%] 1 of 1, cached: 1 ✓
[e4/a88fca] process > ALIGN_READS_TO_ASSEMBLY_FASTA:SAMTOOLS_VIEW (1) [100%] 1 of 1, cached: 1 ✓
[fb/cb99c1] process > HIC_QC (1) [100%] 1 of 1, cached: 1 ✓
[31/8a0a20] process > SAMTOOLS_FAIDX (1) [100%] 1 of 1, cached: 1 ✓
[6f/4f2be4] process > BEDTOOLS_BAM_TO_BED (1) [100%] 1 of 1, cached: 1 ✓
[fc/6e2be5] process > YAHS (1) [100%] 1 of 1, cached: 1 ✓
[0a/15d754] process > ALIGN_READS_TO_YAHS_FASTA:BWA_INDEX_AND_MEM (HiC8-lib_S1_L001_R) [100%] 1 of 1, cached: 1 ✓
[8d/dc515e] process > ALIGN_READS_TO_YAHS_FASTA:SAMBLASTER (1) [100%] 1 of 1, cached: 1 ✓
[92/ccb123] process > ALIGN_READS_TO_YAHS_FASTA:SAMTOOLS_VIEW (1) [100%] 1 of 1, cached: 1 ✓
[4c/2e891d] process > CREATE_HIC_FILE_YAHS:MAKE_AGP_FROM_FASTA (1) [100%] 1 of 1, cached: 1 ✓
[71/f0c870] process > CREATE_HIC_FILE_YAHS:AGP2_ASSEMBLY (1) [100%] 1 of 1, cached: 1 ✓
[10/e04880] process > CREATE_HIC_FILE_YAHS:MATLOCK_BAM2_JUICER (1) [100%] 1 of 1, cached: 1 ✓
[61/648b5f] process > CREATE_HIC_FILE_YAHS:RUN_ASSEMBLY_VISUALIZER (1) [100%] 1 of 1, cached: 1 ✓
[ea/d60fc0] process > SALSA2 (1) [100%] 1 of 1, cached: 1 ✓
[b1/4f3a80] process > ALIGN_READS_TO_SALSA2_FASTA:BWA_INDEX_AND_MEM (HiC8-lib_S1_L001_R) [100%] 1 of 1, cached: 1 ✓
[23/571233] process > ALIGN_READS_TO_SALSA2_FASTA:SAMBLASTER (1) [100%] 1 of 1, cached: 1 ✓
[09/3528da] process > ALIGN_READS_TO_SALSA2_FASTA:SAMTOOLS_VIEW (1) [100%] 1 of 1, cached: 1 ✓
[33/30a300] process > CREATE_HIC_FILE_SALSA2:MAKE_AGP_FROM_FASTA (1) [100%] 1 of 1, cached: 1 ✓
[e6/fa8b11] process > CREATE_HIC_FILE_SALSA2:AGP2_ASSEMBLY (1) [100%] 1 of 1, cached: 1 ✓
[b2/ea8b37] process > CREATE_HIC_FILE_SALSA2:MATLOCK_BAM2_JUICER (1) [100%] 1 of 1, cached: 1 ✓
[42/035d41] process > CREATE_HIC_FILE_SALSA2:RUN_ASSEMBLY_VISUALIZER (1) [100%] 1 of 1, cached: 1 ✓

Hi-C - N F P I P E L I N E
=====
reads      : /workspace/hraiyc/Kiwi/Ck69_01_monoploid/HiC8_MiSeq/220622_M01815_0436/HiC8-lib_S1_L001_R{1,2}_001.fastq.gz
assembly   : /output/genomic/plant/Actinidia/chinensis/CK69_01m/Genome/Assembly/LATEST/Fasta/CK69_01_v2.scaffolds.fsa
outdir     : /powerplant/workspace/hraaxz/HiC-heatmap-processing/hic-scaffolding-pipeline/results

Complete!

```

Figure 15. Nextflow pipeline of genome assembly using Hi-C technology

## V. CONTRIBUTION

The development of the Hi-C scaffolding benchmarking pipeline using Nextflow, along with tools like YaHS, SALSA2, and ALLHiC, has significant potential to contribute to the field of de novo genome assembly. The pipeline has been successfully applied to multiple datasets for different species, and has been approved to be highly time-efficient, provide more efficient computational resource allocation, and facilitate the comparison of results and selection of the best set of scaffolds. It can also provide advice for library protocols and input data volume to save costs, and generate a comprehensive final report with insights on scaffolding failing reasons and the selection of the best set of settings.

Moreover, the Hi-C scaffolding benchmarking pipeline can offer several advantages, including the ability to evaluate and compare the performance of different tools and strategies for Hi-C scaffolding, which can help researchers identify the most effective methods for their specific applications. This is particularly important as different organisms and genomes can have varying degrees of complexity, heterozygosity, and repeat content, which can impact the effectiveness of different scaffolding methods.

The development of a standardized and reproducible framework for Hi-C scaffolding can help ensure that results are

scientific progress. The use of such a benchmarking pipeline can provide researchers with confidence that their results can be replicated by others, facilitating further research.

The use of Nextflow in the pipeline enables efficient and scalable processing of large datasets, allowing for more comprehensive and accurate analysis of the genome. The scalability of Nextflow allows for the analysis of increasingly large datasets, given the increasing amounts of sequencing data being generated. The ability to process and analyze large datasets quickly and efficiently is essential to genome assembly and is a significant advantage of using Nextflow.

In addition to its efficacy, the automated pipeline is highly flexible and can be upgraded with ease. This ensures its long-term sustainability, as it can be easily adapted to accommodate future advancements in the field of genomics. More importantly, the pipeline offers valuable guidance for the design of future bioinformatics pipelines, providing researchers with a framework for developing customized pipelines that suit their research needs.

One of the most significant contributions of the automated pipeline is its potential to enable the efficient exploration of 3D genome structure. The three-dimensional organization of the genome plays a critical role in regulating gene expression, and understanding this organization is essential for unraveling the complexity of gene regulation. The pipeline's ability to process large-scale genomic data with high accuracy and efficiency

makes it an ideal tool for exploring 3D genome structure. This can facilitate the identification of novel regulatory elements and provide insights into the mechanisms that govern gene regulation.

Overall, the development of this Hi-C scaffolding benchmarking pipeline has the potential to improve the accuracy and completeness of de novo genome assembly, facilitating more detailed and comprehensive analysis of the genome. Accurate and comprehensive genome assembly is necessary for a range of applications, including comparative genomics, functional genomics, and evolutionary studies, and the development of a pipeline that can improve the quality and consistency of genome assembly is a significant contribution to the field.

## VI. FUTURE WORK AND LIMITATIONS

In terms of future work and limitations, it is worth noting that current scaffolders are primarily developed for modeled organisms, such as humans or other mammals. However, the performance of these scaffolders is often unsatisfactory when applied to non-model organisms. With the aid of an automated pipeline, identifying the most appropriate scaffolders for non-model organisms, particularly various plant families, could be achieved more readily by including or developing additional scaffolders and conducting benchmarking assessments.

Moreover, it may be possible to develop new scaffolders using neural network deep learning models, such as CNN or GAN. Another aspect of this pipeline is the ability to conduct numerous experiments to determine the minimum input data required to produce a reasonably accurate genome result, thus significantly reducing the cost.

Finally, the ultimate goal would be to automate and optimize the genome assembly process as much as possible to minimize the manual workload of bioinformaticians.

## VII. REFLECTION

From this industry project, I found the courses COMPSCI 345: Human-computer Interaction, COMPSCI 701: Creating Maintainable Software, and INFOSYS 220: Business Systems Analysis, which I have learned from the University of Auckland, during my bachelor's and master's is extremely useful.

### 1) *Knowledge applied in the internship*

- *Business systems and requirements analysis*

The process of designing a Nextflow automated pipeline requires a deep understanding of the underlying concepts and principles. As someone without a bioinformatics background, it was necessary for me to first gain knowledge and expertise in this area before designing the pipeline. After acquiring the necessary skills and knowledge, I was able to proceed with designing the pipeline, with close consultation and feedback from my supervisor, to ensure that the design aligned with both the project goals and the values of the company. From the course of INFOSYS 220: Business Systems Analysis, I have learnt that it is

instrumental to understand how software systems fit within an organization and how to design systems that meet the needs of business stakeholders. This course provides a foundation in principles and techniques used to analyze business systems and requirements and equips me with the skills necessary to understand the needs of clients or stakeholders, thereby enabling them to produce software products that meet their requirements. The insights gained from this course are thus critical for designing a Nextflow automated pipeline that meets the requirements of both the project and the stakeholders.

- *Agile development methodology*

In addition, the Infosys 220: Business Systems Analysis course also teaches me about the principles of agile methodology, including the importance of communication, collaboration, and continuous improvement. By following these principles, it is possible to develop software that is not only functional but also meets the changing needs of the end users. The use of agile methodology can help to ensure the success of the Nextflow automated pipeline, and its adoption by stakeholders, by ensuring that it meets their needs and is developed in an efficient and effective manner. Because Designing and developing the Nextflow automated pipeline can be a complex process. The agile methodology involves an iterative and collaborative approach to software development, with a focus on rapid prototyping, continuous testing, and customer feedback. In my own experience developing the Nextflow pipeline, I used an agile methodology to ensure that every step of the process was checked and verified with my supervisors and team members to ensure that they were satisfied. This approach helped to identify potential issues early on in the development process and allowed for quick resolution of any problems.

- *Maintainable software/automated pipeline creation*

When designing the Nextflow automated pipeline, there are several key considerations in order to ensure that it is both scalable and easy to maintain. One important factor is the use of parallel programming techniques, which can help to ensure that the pipeline can handle large volumes of data efficiently. Another important consideration is the use of modularity, abstraction, and encapsulation principles, which can help to ensure that each process is contained within a single file and that the pipeline is easy to debug. By using these principles, it is possible to create a pipeline that is not only reliable and scalable, but also easy to modify and update as needed. For example, encapsulation is one of software design principle I have applied to design the Nextflow automated pipeline. Encapsulation involves grouping related functionality together in order to limit the interactions between different parts of the system, which can make the pipeline easier to maintain and debug. One way to achieve encapsulation in the Nextflow pipeline I have applied is to use sub-workflows, which are workflows

that are designed to perform a specific function and can contain their own sub-workflows. This allows for the encapsulation of functionality into smaller, more manageable components that can be tested and debugged independently. By using sub-workflows, it is possible to create a highly modular and encapsulated pipeline that can be easily maintained and scaled as needed. The use of sub-workflows also helps to reduce the complexity of the pipeline, making it easier to detect and debug errors. Those knowledge are the focus of COMPSCI 701: Creating Maintainable Software, which teaches me how to design and engineer software that is both reliable and easy to maintain, with a particular focus on the principles of modularity, abstraction, and encapsulation. By understanding these principles, I can apply them to the design of the Nextflow automated pipeline that is both powerful and easy to maintain.

- *User-centered design*

The course COMPSCI 345: Human-computer Interaction primarily focuses on designing effective and efficient user interfaces, its principles can also be applied to the development of automated pipelines. The goal of designing an effective interface is to create a system that is easy to use and understand, while also meeting the needs of the end user. In the context of an automated pipeline, this means creating a system that is simple and straightforward for the user to interact with. By following the step-by-step instructions in the README documentation I wrote, the user should be able to easily install and run the pipeline with just a single line of code in the terminal. This ease of use is essential in ensuring that the pipeline is accessible to a wide range of users, regardless of their technical proficiency.

## 2) *Lessons learned from the industry project*

- *Soft skills Enhancement*

During my internship program, I had the opportunity to participate in two conferences, one on leadership and Mc Ginty's 2023: on current trends in the plant science and bioinformatics industry from PFR, which provided me with novel and valuable experiences for my professional growth.

### a) *From Leadership conference*

Attending the leadership conference with other summer interns from different departments of the company PFR has been a valuable experience for me. Through this conference, I gained a better understanding of myself according to my Herrmann Brain Dominance Instrument (HDBI) reference code, what is leadership, and how I can apply it to practical situations. I had the opportunity to learn about the latest theories and practical applications of leadership, which were presented by industry leaders who have years of experience in the field.

Based on my HDBI reference code of 1211, I possess a number of key personality traits that are typically associated with success and achievement. Specifically, I have a high degree of self-confidence, assertiveness, and goal-orientation,

as well as a strong competitive drive and achievement motivation. In addition, I am described as a highly independent and self-reliant individual who prefers to take charge of situations and make decisions independently. These traits are consistent with the results of previous research on the HDBI assessment, which is widely used in the field of personality psychology to assess a broad range of personality traits and characteristics. Individuals with a reference code of 1211 are often described as dominant and forceful, with a strong desire to succeed and be recognized for their achievements. They tend to be highly confident in their abilities and are not afraid to take risks or pursue ambitious goals. Despite my competitive and assertive nature, I am also highly principled and ethical, with a strong sense of fairness and justice. I am known for my high levels of motivation and energy and am often seen as a natural leader in various settings. However, I have also recognized that these traits can also have potential drawbacks, such as a tendency towards overconfidence or a lack of consideration for the opinions of others. After getting to my personality batter, can be beneficial in achieving success and achieving my goals, also be aware of my potential pitfalls associated with these traits and to work to mitigate them when necessary.

One of the most significant aspects that I learned about was the importance of emotional intelligence. Emotional intelligence refers to the ability to identify and manage one's emotions and the emotions of others. It is a critical leadership skill as it enables leaders to build strong relationships with their team members and to manage conflicts effectively. Effective communication was also emphasized as an essential leadership skill. The ability to communicate clearly, effectively and with empathy can help to foster positive relationships and inspire trust among team members.

In addition, I also learned about adaptive leadership, which is the ability to adjust one's leadership style to suit different situations. As leaders, we need to recognize that different situations require different approaches, and being able to adapt our leadership style accordingly is essential. This is particularly important in today's rapidly changing business environment, where leaders need to be flexible and able to respond quickly to new challenges.

The networking opportunities provided by the conference were also a significant benefit. I had the chance to interact with professionals from different industries and exchange ideas on leadership practices. This broadened my perspectives on this important aspect of organizational management and helped me to develop a more well-rounded view of the subject.

### b) *From Mc Ginty's 2023 conference*

Attending the conference on current trends in the plant science and bioinformatics industry held by PFR has allowed me to develop an important soft skill, which is the ability to think from a researcher's perspective and relate the concepts and skills presented by the speakers to my own IT knowledge. This skill enabled me to critically evaluate the presentations, understand their relevance to my field of study, and brainstorm how my IT skills could be utilized to optimize or develop solutions for current bioinformatics problems. As a result of this

skill, I was able to gain insights into how IT can contribute to the plant science and bioinformatics industry. By listening to the speakers, I learned about the latest technologies, tools, and techniques used in the field and how they can be applied to solve real-world problems. I also gained an understanding of the challenges that researchers face in the industry, such as data management, analysis, and visualization, and how IT solutions can aid in overcoming these challenges.

c) *From internship project*

During my internship, I had the opportunity to develop a range of soft skills, including teamwork, effective communication, and public speaking. Throughout the entire internship program, I was able to hone my skills by participating in various activities and projects, including two presentations that allowed me to apply my communication skills in a professional setting.

One of the most valuable lessons I learned was the importance of teamwork. Working with other members from different departments and backgrounds, I had to learn how to communicate effectively and collaborate efficiently to achieve our shared goals. Through teamwork, I learned how to delegate tasks, respect diverse perspectives, and leverage individual strengths to achieve common objectives. This experience taught me the value of diversity in teams and the importance of effective communication in building strong relationships and achieving successful outcomes.

Another valuable skill I developed was effective communication, which was essential in facilitating team collaboration and achieving project goals. I learned how to communicate my ideas clearly and concisely, adapt my communication style to different audiences and situations, and listen actively to others to understand their perspectives. I also learned how to provide constructive feedback and receive feedback graciously to improve the quality of our work.

Public speaking was another area in which I developed my skills during my internship. I had the opportunity to deliver two presentations, which allowed me to improve my public speaking skills and overcome my fear of speaking in front of a large audience. Through these experiences, I learned how to organize and structure my presentation effectively, tailor my message to the audience, and deliver my content with confidence and clarity.

In addition to these skills, I also developed my time management skills, adaptability, problem-solving skills, and professionalism. Time management skills were crucial in managing multiple tasks and meeting project deadlines. I also learned to adapt to changing work requirements, new technologies, and different work environments. Additionally, I learned how to approach problems systematically, gather relevant information, and generate creative solutions. Lastly, I learned how to act professionally in different settings, such as interacting with colleagues and clients, responding to feedback, and accepting criticism gracefully.

- *Technical skills enhancement*

During my internship program, I had the opportunity to acquire new technical skills, including programming languages

such as Groovy and R, as well as the use of Nextflow, a bioinformatics domain automation tool. Additionally, I learned how to debug code and how to use multiple programming languages simultaneously, including Groovy, R, C, and Bash. Moreover, I learned how to use Docker and parallel programming.

The use of Groovy and R programming languages was new to me, and I found these languages to be quite powerful and flexible for data analysis and visualization. I also learned how to automate workflows using Nextflow, which enabled me to process large amounts of data efficiently, and allowed me to focus on other tasks related to my internship. Debugging code was a new skill for me, and I found it to be an essential technique for identifying and correcting errors in code, which ultimately improved the efficiency and accuracy of my programs. Moreover, learning to use multiple programming languages simultaneously was a challenging but valuable experience, which allowed me to explore different approaches to problem-solving and expand my skillset. The use of Docker was also a new concept for me, and I found it to be a powerful tool for creating, deploying, and running applications in a consistent and reproducible manner. Additionally, I learned how to use parallel programming to improve the performance of my programs and enable them to take advantage of multiple CPU cores.

## VIII. CONCLUSION

In conclusion, this project has developed an automated pipeline using Nextflow to benchmark and evaluate different Hi-C scaffolding methods for de novo genome assembly. The pipeline has the potential to contribute significantly to the field of genomics by offering researchers a standardized and reproducible framework for Hi-C scaffolding, facilitating the comparison of different tools and strategies, and providing guidance for the design of customized pipelines that suit specific research needs. Also, it has several advantages, including its efficiency, scalability, and flexibility, and its potential to enable the efficient exploration of 3D genome structure. And the pipeline has been successfully applied to multiple datasets for different species, and its use can improve the accuracy and completeness of de novo genome assembly, facilitating more detailed and comprehensive analysis of the genome.

The project has also provided me with significant enhancement in both soft and technical skills, including business systems and requirements analysis, agile development methodology, maintainable automated pipeline creation, and user-centered design, also allowed me to develop valuable soft skills, including communication and teamwork, which are essential for collaborating with diverse teams and stakeholders. These skills are valuable assets in the industry, and the project has enabled me to apply them in a real-world setting.

## ACKNOWLEDGMENT

I would like to express my sincere gratitude to my university supervisor, Joerg Simon Wicker, for his patient guidance and



follow-up on my project. He provided valuable insights on how to use soft skills to solve workplace challenges and offered technical construction suggestions. We are also grateful to the Master of IT program at the University of Auckland for providing an internship opportunity that allowed us to successfully obtain this position.

And I am grateful to my mentor Chen Wu for her patience and quick teaching of bioinformatics knowledge throughout the project as my mentor. Using Hi-C data, especially assemblies, is still an art form and Chen Wu's experience with hundreds of libraries and species proved invaluable. Thanks to Usman Rashid for his patience and teaching as an experienced expert at Nextflow. Thanks to Ignacio Carvajal for the well-organized Jupyter notebook workflow, and for the answers to existing workflow questions that were very even. Thanks to Plant & Food Research for providing data on kiwifruit and other new species. This work was facilitated by the use of the advanced computing, storage, and networking infrastructure provided by the Plant & Food Research computing cluster system.

## REFERENCES

- [1] C. Zhou, S. A. McCarthy, and R. Durbin, "YaHS: yet another Hi-C scaffolding tool," *Bioinformatics*, vol. 39, no. 1, pp. btac808, 2023.
- [2] W. M. van Rengs, M. H. W. Schmidt, S. Effgen, D. B. Le, Y. Wang, M. W. A. M. Zaidan, B. Huettel, H. J. Schouten, B. Usadel, and C. J. Underwood, "A chromosome scale tomato genome built from complementary PacBio and Nanopore sequences alone reveals extensive linkage drag during breeding," *The Plant Journal*, vol. 110, no. 2, pp. 572-588, 2022.
- [3] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, and A. Gershman, "The complete sequence of a human genome," *Science*, vol. 376, no. 6588, pp. 44-53, 2022.
- [4] S. Wang, H. Wang, F. Jiang, A. Wang, H. Liu, H. Zhao, B. Yang, D. Xu, Y. Zhang, and W. Fan, "EndHiC: assemble large contigs into chromosome-level scaffolds using the Hi-C links from contig ends," *BMC bioinformatics*, vol. 23, no. 1, pp. 1-19, 2022.
- [5] D. L. Lafontaine, L. Yang, J. Dekker, and J. H. Gibcus, "Hi-C 3.0: improved protocol for genome-wide chromosome conformation capture," *Current Protocols*, vol. 1, no. 7, pp. e198, 2021.
- [6] K. Yamaguchi, M. Kadota, O. Nishimura, Y. Ohishi, Y. Naito, and S. Kuraku, "Technical considerations in Hi-C scaffolding and evaluation of chromosome-scale genome assemblies," *Molecular Ecology*, vol. 30, no. 23, pp. 5923-5934, 2021.
- [7] F. Lehmann, D. Frantz, S. Becker, U. Leser, and P. Hostert, "FORCE on Nextflow: Scalable Analysis of Earth Observation Data on Commodity Clusters," in *CIKM Workshops*, 2021, pp. 55-62.
- [8] I. Branco and A. Choupina, "Bioinformatics: new tools and applications in life science and personalized medicine," *Applied microbiology and biotechnology*, vol. 105, pp. 937-951, 2021.
- [9] J. Luo, Y. Wei, M. Lyu, Z. Wu, X. Liu, H. Luo, and C. Yan, "A comprehensive review of scaffolding methods in genome assembly," *Briefings in Bioinformatics*, vol. 22, no. 5, pp. bbab033, 2021.
- [10] A. Whibley, J. L. Kelley, and S. R. Narum, "The changing face of genome assemblies: Guidance on achieving high-quality reference genomes," *Wiley Online Library*, 2021.
- [11] D. Guan, S. A. McCarthy, Z. Ning, G. Wang, Y. Wang, and R. Durbin, "Efficient iterative Hi-C scaffolder based on N-best neighbors," *BMC Bioinformatics*, vol. 22, no. 1, pp. 1-16, Jan. 2021.
- [12] B. Cancer, "Filling in the gaps: GapPredict can complement repertoire of tools used to resolve missing DNA sequences in genome assemblies," 2021. [Online]. Available: <https://bcgsc.ca/news/filling-gaps-gappredict-can-complement-repertoire-tools-used-resolve-missing-dna-sequences>. [Accessed: Feb. 2023].
- [13] S. Sullivan H. and B. Nelson, "hic\_qc," 2021. [Online]. Available: [https://github.com/phasegenomics/hic\\_qc](https://github.com/phasegenomics/hic_qc). [Accessed: Jan. 2023].
- [14] P. A. Ewels, A. Peltzer, S. Fillinger, H. Patel, J. Alneberg, A. Wilm, M. U. Garcia, P. Di Tommaso, and S. Nahnsen, "The nf-core framework for community-curated bioinformatics pipelines," *Nature Biotechnology*, vol. 38, no. 3, pp. 276-278, Mar. 2020.
- [15] L. Baudry, N. Guiglielmoni, H. Marie-Nelly, A. Cormier, M. Marbouty, K. Avia, Y. L. Mie, O. Godfroy, L. Sterck, and J. M. Cock, "instaGRAAL: chromosome-level quality scaffolding of genomes using a proximity ligation-based scaffolder," *Genome Biology*, vol. 21, no. 1, pp. 1-22, Jan. 2020.
- [16] C. Arisdakessian, S. B. Cleveland, and M. Belcaid, "MetaFlow| mics: Scalable and Reproducible Nextflow Pipelines for the Analysis of Microbiome Marker Data," in *Practice and Experience in Advanced Research Computing*, 2020, pp. 120-124.
- [17] M. Kadota, O. Nishimura, H. Miura, K. Tanaka, I. Hiratani, and S. Kuraku, "Multifaceted Hi-C benchmarking: what makes a difference in chromosome-scale genome scaffolding?," *GigaScience*, vol. 9, no. 1, giz158, Jan. 2020.
- [18] X. Zhang, S. Zhang, Q. Zhao, R. Ming, and H. Tang, "Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data," *Nature Plants*, vol. 5, no. 8, pp. 833-845, Aug. 2019.
- [19] J. Ghurye, A. Rhie, B. P. Walenz, A. Schmitt, S. Selvaraj, M. Pop, A. M. Phillippy, and S. Koren, "Integrating Hi-C links with assembly graphs for chromosome-scale assembly," *PLoS Computational Biology*, vol. 15, no. 8, e1007273, Aug. 2019.
- [20] A. Federico, T. Karagiannis, K. Karri, D. Kishore, Y. Koga, J.D. Campbell, and S.M. Pipeliner: A Nextflow-Based Framework for the Definition of Sequencing Data Processing Pipelines. *Frontiers in Genetics*, 10:614, 2019.
- [21] 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.
- [22] A. Bayat, N.P. Deshpande, M.R. Wilkins, and S. Parameswaran, "Fast short read de-novo assembly using overlap-layout-consensus approach," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 1, pp. 334-338, 2018.
- [23] J.-i. Sohn and J.-W. Nam, "The present and future of de novo whole-genome assembly," *Briefings in Bioinformatics*, vol. 19, no. 1, pp. 23-40, 2018.
- [24] O. Dudchenko, S.S. Batra, A.D. Omer, S.K. Nyquist, M. Hoeger, N.C. Durand, M.S. Shamim, I. Machol, E.S. Lander, and A.P. Aiden, "De novo assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds," *Science*, vol. 356, no. 6333, pp. 92-95, 2017.
- [25] P. Di Tommaso, M. Chatzou, E.W. Floden, P.P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," *Nature Biotechnology*, vol. 35, no. 4, pp. 316-319, 2017.
- [26] G.G. Yardımcı and W.S. Noble, "Software tools for visualizing Hi-C data," *Genome Biology*, vol. 18, no. 1, pp. 1-9, 2017.
- [27] J. Ghurye, M. Pop, S. Koren, D. Bickhart, and C.-S. Chin, "Scaffolding of long read assemblies using long range contact information," *BMC Genomics*, vol. 18, no. 1, pp. 1-11, 2017.
- [28] 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," *Genome Research*, vol. 27, no. 5, pp. 722-736, 2017.
- [29] G.P. Thottathil, K. Jayasekaran, and A.S. Othman, "Sequencing crop genomes: a gateway to improve tropical agriculture," *Tropical Life Sciences Research*, vol. 27, no. 1, pp. 93, 2016.
- [30] S.M. Friedrich, H.C. Zec, and T.-H. Wang, "Analysis of single nucleic acid molecules in micro-and nano-fluidics," *Lab on a Chip*, vol. 16, no. 5, pp. 790-811, 2016.
- [31] I. Sović, K. Križanović, K. Skala, and M. Šikić, "Evaluation of hybrid and non-hybrid methods for de novo assembly of nanopore reads," *Bioinformatics*, vol. 32, no. 17, pp. 2582-2589, 2016.
- [32] M.T. Ebbert, M.E. Wadsworth, L.A. Staley, K.L. Hoyt, B. Pickett, J. Miller, J.S. Kauwe, P.G. Ridge, and AsDN Initiative, "Evaluating the necessity of PCR duplicate removal from next-generation sequencing data and a comparison of approaches," *BMC Bioinformatics*, vol. 17, pp. 491-500, 2016.
- [33] N.C. Durand, J.T. Robinson, M.S. Shamim, I. Machol, J.P. Mesirov, E.S. Lander, and E.L. Aiden, "Juicebox provides a visualization system for Hi-C contact maps with unlimited zoom," *Cell Systems*, vol. 3, no. 1, pp. 99-101, 2016.
- [34] M. Jain, H.E. Olsen, B. Paten, and M. Akeson, "The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community," *Genome Biology*, vol. 17, pp. 1-11, 2016.
- [35] A.D. Goldman and L.F. Landweber, "What is a genome?" *PLoS Genetics*, vol. 12, no. 7, pp. e1006181, 2016.
- [36] J. Fraser, I. Williamson, W.A. Bickmore, and J. Dostie, "An overview of genome organization and how we got there: from FISH to Hi-C,"

- Microbiology and Molecular Biology Reviews*, vol. 79, no. 3, pp. 347-372, 2015.
- [37] J.A. Reuter, D.V. Spacek, and M.P. Snyder, "High-throughput sequencing technologies," *Molecular Cell*, vol. 58, no. 4, pp. 586-597, 2015.
- [38] Tad, "Measure of 'how much diagonal' a matrix is," [Online]. Available: <https://math.stackexchange.com/q/1393907>.
- [39] G.G. Faust and I.M. Hall, "SAMBLASTER: fast duplicate marking and structural variant read extraction," *Bioinformatics*, vol. 30, no. 17, pp. 2503-2505, 2014.
- [40] J.O. Korbel and C.L. "Genome assembly and haplotyping with Hi-C," *Nature Biotechnology*, vol. 31, pp. 1099-1101, 2013.
- [41] J.N. Burton, A. Adey, R.P. Patwardhan, R. Qiu, J.O. Kitzman, and J. Shendure, "Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions," *Nature Biotechnology*, vol. 31, no. 12, pp. 1119-1125, Dec. 2013.
- [42] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM," *arXiv preprint arXiv:1303.3997*, Mar. 2013.
- [43] Z. Li, Y. Chen, D. Mu, J. Yuan, Y. Shi, H. Zhang, J. Gan, N. Li, X. Hu, and B. Liu, "Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph," *Briefings in Functional Genomics*, vol. 11, no. 1, pp. 25-37, Jan. 2012.
- [44] J.T. Simpson and R. Durbin, "Efficient de novo assembly of large genomes using compressed data structures," *Genome Research*, vol. 22, no. 3, pp. 549-556, Mar. 2012.
- [45] M. Baker, "De novo genome assembly: what every biologist should know," *Nature Methods*, vol. 9, no. 4, pp. 333-337, Apr. 2012.
- [46] M.C. Schatz, J. Witkowski, and W.R. McCombie, "Current challenges in de novo plant genome sequencing and assembly," *Genome Biology*, vol. 13, no. 4, pp. 1-7, Apr. 2012.
- [47] J.A. Stamatoyannopoulos, "What does our genome encode?" *Genome Research*, vol. 22, no. 9, pp. 1602-1611, Sep. 2012.
- [48] J.-M. Belton, R.P. McCord, J.H. Gibcus, N. Naumova, Y. Zhan, and J. Dekker, "Hi-C: a comprehensive technique to capture the conformation of genomes," *Methods*, vol. 58, no. 3, pp. 268-276, Mar. 2012.
- [49] D.M. Church, V.A. Schneider, T. Graves, K. Auger, F. Cunningham, N. Bouk, H.-C. Chen, R. Agarwala, W.M. McLaren, and G.R. Ritchie, "Modernizing reference genome assemblies," *PLoS Biology*, vol. 9, no. 7, e1001091, Jul. 2011.
- [50] N.L. Van Berkum, E. Lieberman-Aiden, L. Williams, M. Imakaev, A. Gnirke, L.A. Mirny, J. Dekker, and E.S. Lander, "Hi-C: a method to study the three-dimensional architecture of genomes," *JoVE (Journal of Visualized Experiments)*, no. 39, e1869, Nov. 2010.
- [51] Z. Duan, M. Andronescu, K. Schutz, S. McIlwain, Y. J. Kim, C. Lee, J. Shendure, S. Fields, C. A. Blau, and W. S. Noble, "A three-dimensional model of the yeast genome," *Nature*, vol. 465, no. 7296, pp. 363-367, 2010.
- [52] S. Andrews, "FastQC: a quality control tool for high throughput sequence data," *Babraham Bioinformatics*, Babraham Institute, Cambridge, United Kingdom, 2010.
- [53] J. R. Miller, S. Koren, and G. Sutton, "Assembly algorithms for next-generation sequencing data," *Genomics*, vol. 95, no. 6, pp. 315-327, 2010.
- [54] M. E. Hudson, "Sequencing breakthroughs for genomic ecology and evolutionary biology," *Molecular ecology resources*, vol. 8, no. 1, pp. 3-17, 2008.
- [55] Anon., "Wix," [Online]. Available: <https://www.wix.com/about/us>. [Accessed: Feb. 2023].
- [56] G. B. West and J. H. Brown, "The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization," *Journal of experimental biology*, vol. 208, no. 9, pp. 1575-1592, 2005.
- [57] C. Newham, "Learning the bash shell: Unix shell programming: " O'Reilly Media, Inc.", 2005.
- [58] Apache, "Groovy," [Online]. Available: <http://groovy-lang.org/>. [Accessed: Feb. 2023].
- [59] G. Van Rossum and F. L. Drake, "An introduction to Python," *Network Theory Ltd.*, Bristol, 2003.
- [60] R. Ihaka and R. Gentleman, "R: a language for data analysis and graphics," *Journal of computational and graphical statistics*, vol. 5, no. 3, pp. 299-314, 1996.
- [61] X. Huang, "A contig assembly program based on sensitive detection of fragment overlaps," *Genomics*, vol. 14, no. 1, pp. 18-25, Jan. 1992.
- [62] R. J. Costall, Y. Shimada, D. Anthony, and G. L. Rapson, "The endemic tree *Corynocarpus laevigatus* (karaka) as a weedy invader in forest remnants of southern North Island, New Zealand," *New Zealand Journal of Botany*, vol. 44, no. 1, pp. 5-22, Mar. 2010.
- [63] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, and R. Durbin, "The Sequence Alignment/Map format and SAMtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078-2079, Aug. 2009.