bio-101

Contigs Scaffolding with Hi-C for Plant Genomes

Hong An[2, *], Qing Xiao[1], Zhibo Jia[1], J. Chris Pires[3] and Bin Yi[1, *]

[1]National Key Lab of Crop Genetic Improvement, Huazhong Agricultural University, Wuhan, Hubei, P. R. China
[2]Bond Life Sciences Center, University of Missouri, Columbia, MO, USA
[3]New York Botanical Garden, New York, NY, USA
*For correspondence: anho@missouri.edu; yibin@mail.hzau.edu.cn

Abstract

Hi-C is a chromosome conformation capture method originally developed to detect genome-wide chromatin interactions. Nowadays, it is widely applied in scaffolding *de novo* assembled contigs into chromosome-scale genome sequences. Multiple open-source software has been developed to perform genome scaffolding with Hi-C data. The input data is *de novo* assembled contigs using long-read or short-read sequencing. Then, Hi-C data is mapped to these contigs, and the interact matrix is computed by software to scaffold contigs into chromosome-scale sequences. Different tools have specific algorithms to calculate the interact matrix and correct misassemblies and misjoins and may require different dependent packages or running environments. Here, we describe a step-by-step protocol for genome scaffolding using Hi-C data with a comprehensive pipeline: compute interact matrix with Juicer, scaffold contigs with 3D-DNA pipeline, and then visualize and modify scaffolding with Juicebox. This is the first detailed protocol showing how to do Hi-C scaffolding using this pipeline in plants. Compared to many other pipelines, this protocol only requires primarily assembled contigs and raw Hi-C data as inputs. Moreover, it is also compatible with multiple enzymes, and provides visualization and the possibility for manual correction. Currently, more and more genomes are sequenced combining Hi-C; this step-by-step protocol may be applied widely in mass large eukaryotic genome scaffolding.

Keywords: Hi-C, Scaffolding, Genome assembly, Bioinformatics, Plant, Next-generation sequencing

Background

A plant genome provides valuable information to researchers for all kinds of molecular biological studies. In recent years, the development of sequencing technology has allowed faster and more affordable genome sequencing. Nevertheless, chromosome-scale genome sequences are still hard to obtain with only next-generation sequencing (NGS) or long-read sequencing due to some complicated genomic structures, like long interspersed repeats or highly homologous genome blocks. To conquer this, a genetic linkage map or optical map has been applied, to order and orient contigs into chromosome-scale sequences (Yamaguchi *et al*., 2021). However, the genetic linkage map is labour- and time-consuming in the Plantae kingdom. Meanwhile, the optical map requires a large quantity and high quality of high molecular weight DNA, which makes its production relatively difficult. In contrast, the quickly developed Hi-C scaffolding only requires 100 mg of plant tissue, and short-read sequencing on the NGS platform. This makes the Hi-C scaffolding both tissue- and cost-affordable. However, we need to be aware of the Hi-C library preparation, which will determine the success of the genome scaffolding. Young leaf tissue is commonly used in Hi-C library preparation for plants. Multi-round quality control is recommended during library preparation (Kadota *et al.*, 2020). In particular, small-scale sequencing is highly recommended to evaluate the quality of the library, including the proportion of valid interaction reads, and estimation of the proper read pairs for further deep sequencing. Hi-C scaffolding has become one of the main solutions to obtain chromosome-scale scaffolds, having been widely utilized in recent plant genome sequencing projects. Meanwhile, multiple open-source software have been developed to compute the interact matrix, and order and orient assembled contigs into scaffolds (Table 1). Among these tools, the 3D-DNA pipeline is a widely used software that supports interactively visualizing and manually modifying the scaffolds.

Table 1. Overview of the major Hi-C scaffolding software

Program	Input format	Other information	Literature
3D-DNA	Juicer mapper format	Compatible with multiple enzymes; results can be visualized and modified by Juicebox	(Dudchenko *et al.*, 2017)
LACHESIS	Generic bam format	No function to correct misjoins; developer's support discontinued	(Burton *et al*., 2013)
HiRise	Generic bam format	Used in Dovetail Chicago/Hi-C service; no open-source update available since 2015	(Putnam *et al*., 2016)
SALSA2	Generic bam (bed) file, assembly graph, unitig, 10× link files	Compatible with multiple enzymes; results can be visualized by Juicebox	(Ghurye *et al*., 2019)
ALLHiC	Hi-C reads; gene annotation or closely related chromosome-scale reference genome	Designed for scaffolding plant polyploid genome	(Zhang *et al*., 2019)
HiCAssembler	Hi-C matrix in h5 format created by HiCExplorer	Assembly errors can be manually corrected by specifying the position in the software	(Renschler *et al*., 2019)
instaGRAAL	Hi-C matrix created by	Requires NVIDIA CUDA and GPU	(Baudry *et al*., 2020)

	hicstuff or HiC-Box	environment

Software

1. Trimmomatic (Bolger *et al*., 2014) (http://www.usadellab.org/cms/?page=trimmomatic)
2. Juicer (Durand *et al.* 2016) (https://github.com/aidenlab/juicer/)
3. 3D-DNA pipeline (Dudchenko *et al.* 2017) (https://github.com/aidenlab/3d-dna)
4. Juicebox (version 1.11.08) (https://github.com/aidenlab/Juicebox)
5. BWA (Li and Durbin, 2009) (http://bio-bwa.sourceforge.net/)
6. Samtools (Li *et al.*, 2009) (http://www.htslib.org/)
7. Miniconda (https://docs.conda.io/en/latest/miniconda.html)
8. BUSCO (Seppey *et al*., 2019) (https://gitlab.com/ezlab/busco)
9. Java 1.8 JDK (https://www.oracle.com/java/technologies/downloads/#java8)

Note: We recommend users to use the latest version of each software listed above, except for Juicebox (v. 1.11.08).

Equipment

1. Linux server or cluster
2. PC or Mac with at least 16GB RAM for handling big genomes (>1GB)

Input data

1. *De novo* assembly contigs file in FASTA format
2. Raw Hi-C sequencing data in FASTQ format

Procedure

1. Install and configure Juicer
   Juicer is the software that maps Hi-C paired-end reads to assembled contigs and generates the Hi-C interact matrix for downstream analysis.
   a. Download Juicer from the official GitHub repository.
      ```
      $ mkdir hic; cd hic
      $ git clone https://github.com/theaidenlab/juicer.git
      ```
   b. Configure Juicer.
      ```
      $ ln -s juicer/SLURM/scripts/ scripts
      $ cd scripts; wget
      https://hicfiles.tc4ga.com/public/juicer/juicer_tools.1.9.9_jcuda.0.8.
      jar; ln -s juicer_tools.1.9.9_jcuda.0.8.jar juicer_tools.jar; cd ../
      $ mkdir references
      $ mkdir restriction_sites
      ```
   c. Make sure `samtools` and bwa are in your `$PATH`.
      ```
      $ export PATH=your_samtools/samtools:$PATH
      $ export PATH=your_bwa/bwa:$PATH
      ```
      Note: Juicer can be run on AWS, LSF, Univa Grid Engine (UGER), SLURM, and even a single CPU, but users may need to change the command line "ln -s juicer/SLURM/scripts/ scripts" in the "b. Configure Juicer" section to fit their system. For example, use "ln -s juicer/AWS/scripts/ scripts" for the AWS

scheduler. For macOS users, `curl https://hicfiles.tc4ga.com/public/juicer/juicer_tools.1.9.9_jcuda.0.8.jar --output ./` can be used instead of wget. The same can also be applied to all the wget in this protocol.

2. Prepare input data for Juicer
   a. Copy your contigs.fasta file (or make soft link) into reference path, and index it with bwa index.
   ```
   $ ln -s your_path/your_contigs.fasta ./references
   $ cd ./references; bwa index your_contig.fasta; cd ..
   ```
   b. Prepare enzyme site file for your_contigs.fasta.
   ```
   $ cd restriction_sites
   $ wget
   https://raw.githubusercontent.com/aidenlab/juicer/main/misc/generate_site_positions.py
   Use vi or vim to edit generate_site_positions.py, insert the following line in line 25:
    'your_contigs': '../references/your_contigs.fasta',
   $ python generate_site_positions.py your_enzyme your_contigs
   $ awk 'BEGIN{OFS="\t"}{print $1, $NF}' your_contigs_your_enzyme.txt > your_contigs.chrom.sizes
   $ cd ..
   ```
   c. Filter and clean raw Hi-C sequencing data.
   ```
   $ wget
   https://github.com/usadellab/Trimmomatic/files/5854859/Trimmomatic-0.39.zip
   $ unzip Trimmomatic-0.39.zip
   $ java -jar ./Trimmomatic-0.39/trimmomatic-0.39.jar PE -threads
   your_threads -phred33 -trimlog trimmomatic.log your_hic_R1.fastq.gz
   your_hic_R2.fastq.gz your_hic_pair_R1.fastq.gz
   your_hic_unpair_R1.fastq.gz your_hic_pair_R2.fastq.gz
   your_hic_unpair_R2.fastq.gz ILLUMINACLIP:TruSeq3-PE.fa:2:30:10
   LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
   ```
   Note: `your_contigs`, `your_enzyme`, and `your_threads` are variable; users need to name their own contig file, choose the specific enzyme they used, and specify how many threads they would like to use. `your_hic_R1.fastq.gz`, `your_hic_R2.fastq.gz` are the sequenced raw Hi-C paired-end data.
3. Run Juicer to obtain the interact matrix
   ```
   $ mkdir your_contigs_hic; cd your_contigs_hic
   $ mkdir fastq
   $ ln -s ../your_hic_pair* ./fastq/
   $ sh ../scripts/juicer.sh -D $PWD/hic -g your_contigs -s your_enzyme
   -p ../restriction_sites/your_contigs.chrom.sizes
   -y ../restriction_sites/your_contigs_your_enzyme.txt
   -z ../references/your_contig.fasta -Q 2-00:00 -L 7-00:00 -q your_queue_name
   -l your_long_queue_name -t your_threads -A your_account --assembly
   $ cd ..
   ```
   Note: Check juicer.sh, and make sure all the Partition, Account, QOS, and Threads fit your cluster's scheduler. To be safe, add these parameters to your command line. Juicer will submit jobs to the cluster through the scheduler automatically. After all the jobs are done, the file named merged_nodups.txt in your_contigs_hic/aligned will be used by 3D-DNA pipeline.
4. Run 3D-DNA pipeline
   3D-DNA pipeline is designed to correct misassembles and scaffold contigs based on the Hi-C interact matrix. It will generate the scaffolds fasta file and .hic and .assembly files for visualization in Juicebox.

a. Download 3D-DNA pipeline and uncompress.

```
$ wget
https://github.com/aidenlab/3d-dna/archive/refs/tags/201008.tar.gz
$ tar -zxf 201008.tar.gz
$ chmod 554 ./3d-dna-201008/*.sh
```

b. Run 3D-DNA to scaffold the contigs using the interact matrix information.

```
$ cd your_contigs_hic
$ ../3d-dna-201008/run-asm-pipeline.sh ../references/your_contigs.fasta
 ./aligned/merged_nodups.txt
```

After the job is done, two files named `your_contigs.rawchrom.assembly` and `your_contigs.rawchrom.hic` will be used by Juicebox.

Note: If the scaffolding results are not ideal, try different `-round (or -r)`*, different edit round, and slightly increase* `--editor-repeat-coverage` *misjoin editor threshold repeat coverage. In this case study, we use* `-r5 --editor-repeat-coverage 3`*.*

5. Visualize and modify the scaffolding with Juicebox

a. Based on the system, the corresponding Juicebox 1.11.08 version can be downloaded at https://github.com/aidenlab/Juicebox/wiki/Download.

b. Download `your_contigs.rawchrom.assembly` and `your_contigs.rawchrom.hic` to your PC.

c. Run Juicebox, then load your_contigs.rawchrom.hic and your_contigs.rawchrom.assembly in turn (Figure 1).

d. Correct scaffolding manually (Figures 2 and 3).

1) Shift+left-click to choose the region that needs to be edited.

2) Right-click to choose to remove or add chr boundaries.

3) Move the mouse to the upper-right corner of the selected region until a circle appears, and then left-click to rotate the selected region.

Note: A demo video made by the Juicebox developer can be found on the GitHub repository.

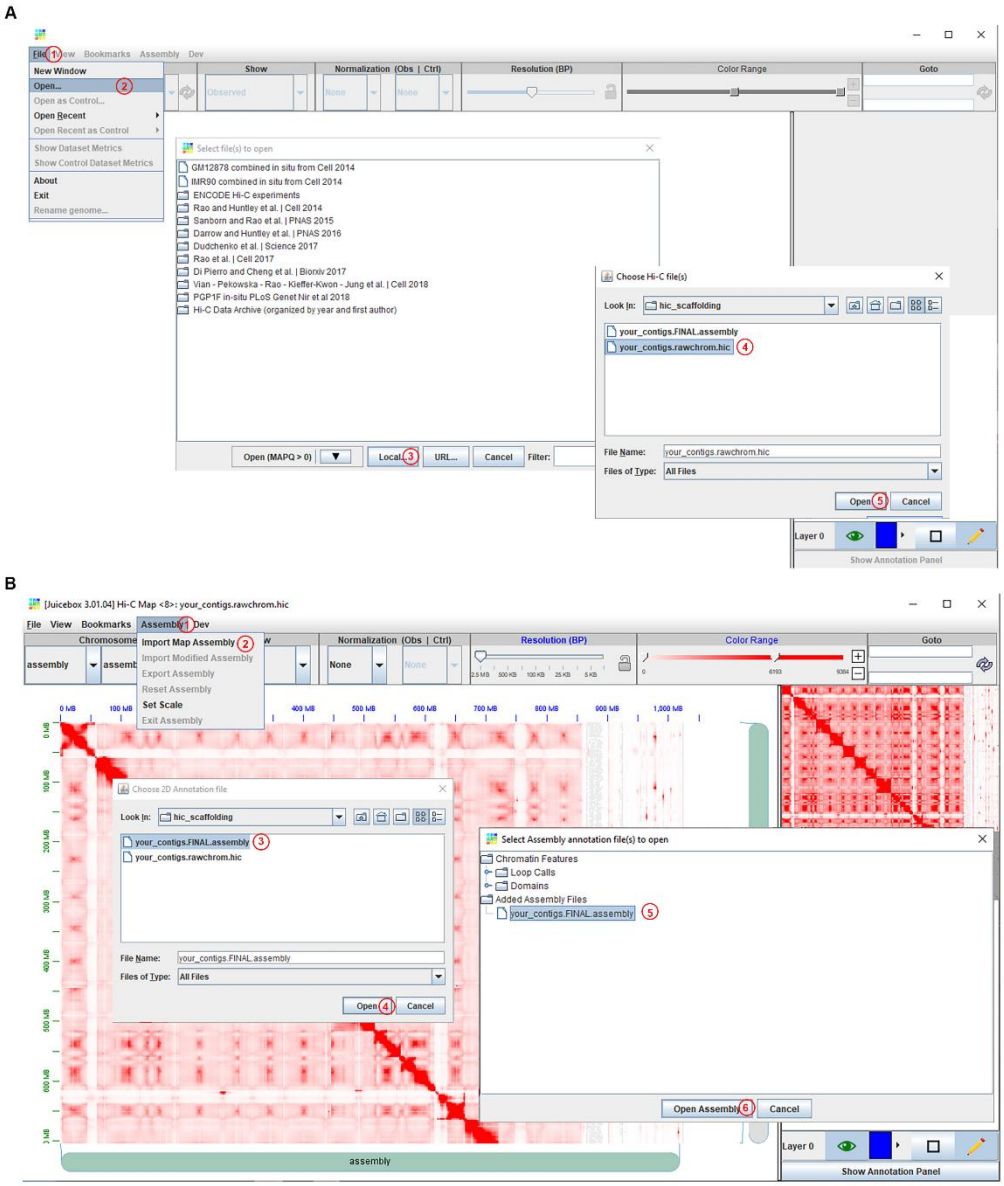

Figure 1. Steps to load .hic and .assembly file to Juicebox.
A. load the .hic file; B. load the .assembly file.

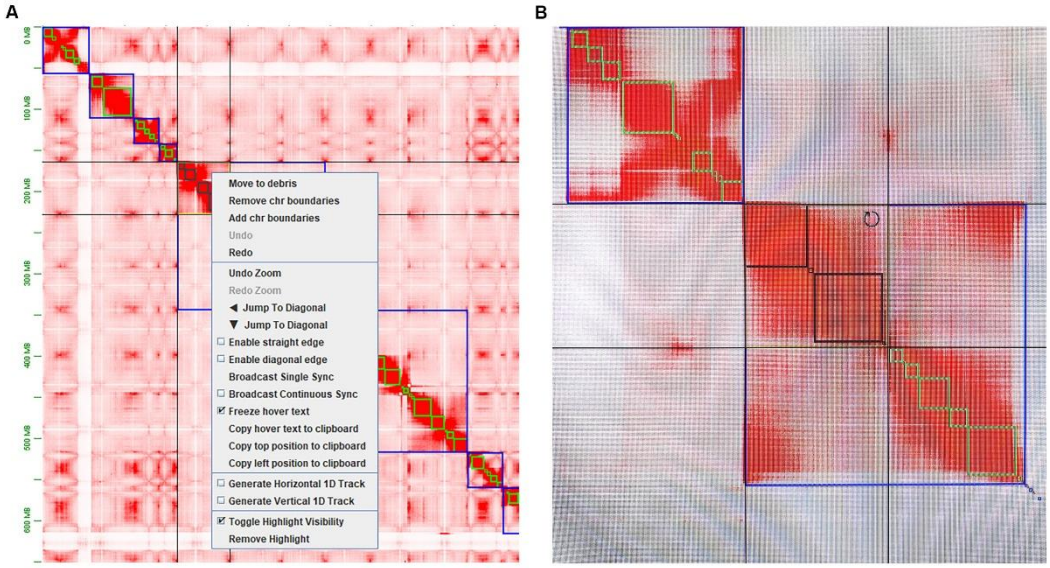

Figure 2. Examples of how to edit misjoin and misorientation.
A. edit misjoin via add and remove chr boundaries. B. edit misorientation by rotating selected contigs.

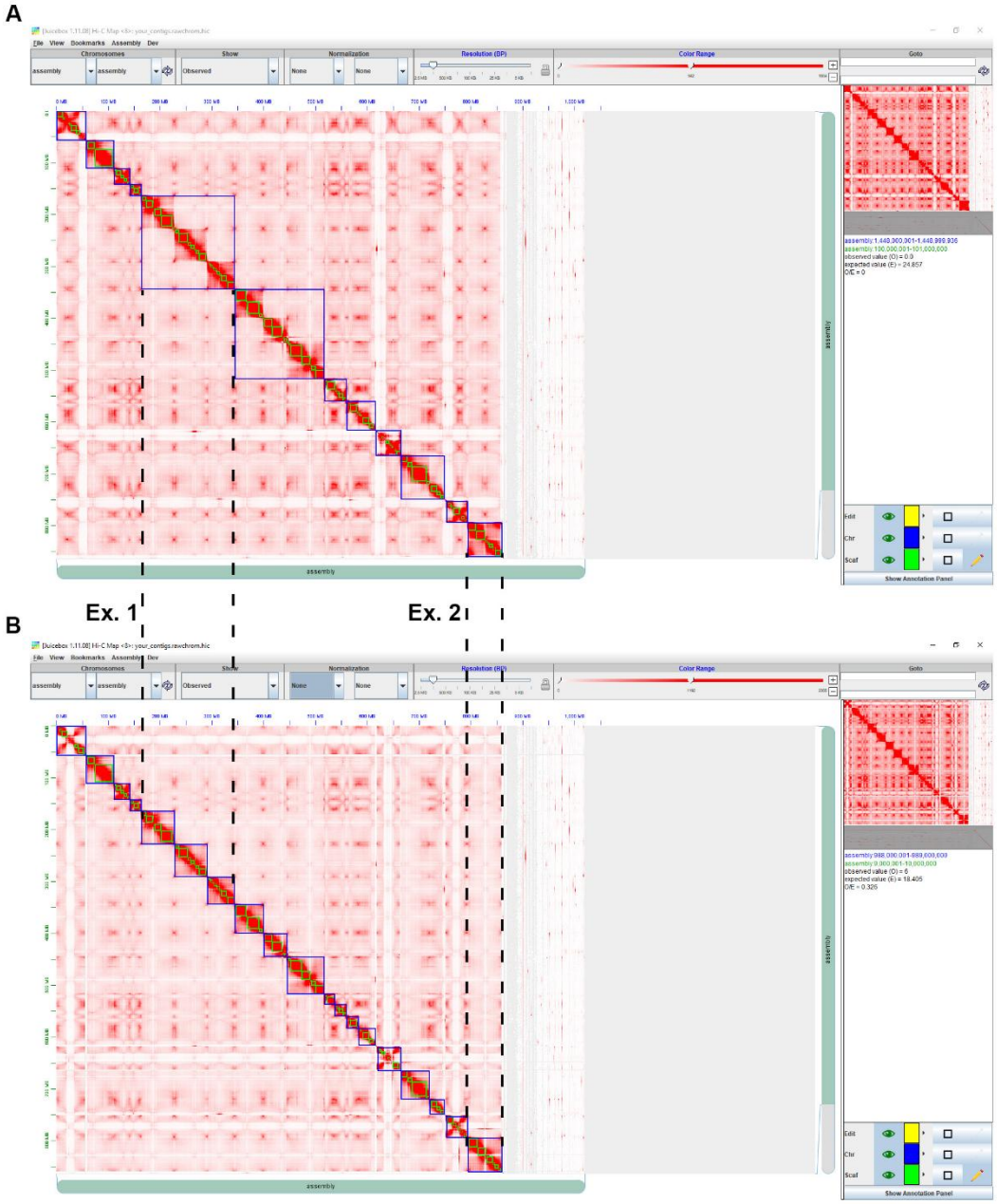

Figure 3. Manually correct the scaffolding with Juicebox.
A. the original scaffolding visualization. B. manually corrected scaffolding. Ex. 1: misjoin, Ex. 2: misorientation.

6. Run the 3D-DNA pipeline again to update the manual modification
   a. Upload `your_contigs.rawchrom.edit.assembly` to cluster, and place it in `your_contigs_hic`.
   b. Run the 3D-DNA pipeline to obtain your edited scaffolds fasta file.
   ```
   $ ../3d-dna/run-asm-pipeline-post-review.sh -r
   your_contigs.rawchrom.edit.assembly ../references/your_contigs.fasta
   aligned/merged_nodups.txt
   ```
7. Primarily estimate the scaffolding with BUSCO
   a. Install BUSCO.

```
$ conda install -c bioconda busco
```

b. Download the dataset for BUSCO.

```
$ mkdir busco_evalue; cd busco_evalue
$ wget
https://busco-data.ezlab.org/v4/data/lineages/embryophyta_odb10.2020-0
9-10.tar.gz
$ tar -zxf embryophyta_odb10.2020-09-10.tar.gz
```

c. Run BUSCO.

```
$ busco -c your_threads -m genome
-i ../your_contigs_hic/your_contigs_arrow_nextpolish_HiC.fasta -o
your_contigs_hic_busco -l ./embryophyta_odb10
```

d. Check BUSCO value and components (Figure 4).

```
--------------------------------------------------
|Results from dataset embryophyta_odb10          |
--------------------------------------------------
|C:99.6%[S:8.6%,D:91.0%],F:0.1%,M:0.3%,n:1614    |
|1607   Complete BUSCOs (C)                      |
|138    Complete and single-copy BUSCOs (S)      |
|1469   Complete and duplicated BUSCOs (D)       |
|1      Fragmented BUSCOs (F)                    |
|6      Missing BUSCOs (M)                       |
|1614   Total BUSCO groups searched              |
--------------------------------------------------
```

Figure 4. BUSCO summary information.

Result interpretation

Table 2. Comparison of input contigs and Hi-C scaffolds

	Contigs	Scaffolds
Count	1,277	1,756
Total	1,021,027,667	1,021,751,667
Max	33,995,119	71,758,703
Min	518	518
N25	16,298,438	62,920,139
L25	12	4
N50	8,776,215	52,692,430
L50	33	9
N75	3,298,835	23,729,965
L75	80	16

Acknowledgments

This project is supported by National Key Research and Development Program of China (2021YFD1600500).

Competing interests

The authors declare no conflict of interest.

References

Baudry, L., Guiglielmoni, N., Marie-Nelly, H., Cormier, A., Marbouty, M., Avia, K., Mie, Y. L., Godfroy, O., Sterck, L., Cock, J. M., *et al*. (2020). instaGRAAL: chromosome-level quality scaffolding of genomes using a proximity ligation-based scaffolder. *Genome Biol* 21(1): 148.

Bolger, A. M., Lohse, M. and Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 30(15): 2114-2120.

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

Dudchenko, O., Batra, S. S., Omer, A. D., Nyquist, S. K., Hoeger, M., Durand, N. C., Shamim, M. S., Machol, I., Lander, E. S., Aiden, A. P., *et al*. (2017). *De novo* assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds. *Science* 356(6333): 92-95.

Durand, N. C., Shamim, M. S., Machol, I., Rao, S. S., Huntley, M. H., Lander, E. S. and Aiden, E. L. (2016). Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments. *Cell Syst* 3(1): 95-98.

Ghurye, J., Rhie, A., Walenz, B. P., Schmitt, A., Selvaraj, S., Pop, M., Phillippy, A. M. and Koren, S. (2019). Integrating Hi-C links with assembly graphs for chromosome-scale assembly. *PLoS Comput Biol* 15(8): e1007273.

Kadota, M., Nishimura, O., Miura, H., Tanaka, K., Hiratani, I. and Kuraku, S. (2020). Multifaceted Hi-C benchmarking: what makes a difference in chromosome-scale genome scaffolding? *Gigascience* 9(1): giz158.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25(14): 1754-1760.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R. and Genome Project Data Processing, S. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25(16): 2078-2079.

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

Renschler, G., Richard, G., Valsecchi, C. I. K., Toscano, S., Arrigoni, L., Ramirez, F. and Akhtar, A. (2019). Hi-C guided assemblies reveal conserved regulatory topologies on X and autosomes despite extensive genome shuffling. *Genes Dev* 33(21-22): 1591-1612.

Seppey, M., Manni, M. and Zdobnov, E. M. (2019). BUSCO: Assessing Genome Assembly and Annotation Completeness. *Methods Mol Biol* 1962: 227-245.

Yamaguchi, K., Kadota, M., Nishimura, O., Ohishi, Y., Naito, Y. and Kuraku, S. (2021). Technical considerations in Hi-C scaffolding and evaluation of chromosome-scale genome assemblies. *Mol Ecol* 30(23): 5923-5934.

Zhang, X., Zhang, S., Zhao, Q., Ming, R. and Tang, H. (2019). Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data. *Nat Plants* 5(8): 833-845.

Supplementary information

1. Data and code availability: All data and code have been deposited to GitHub: https://github.com/Bio-protocol/Plant_genome_Hi-C_scaffolding.git.