May 30, 2024

Genome Assembly, Scaffolding, and Annotation in Sugar Beet

Forked from Genome Assembly, Scaffolding, and Annotation Pipeline in Sugar Beet

DOI
dx.doi.org/10.17504/protocols.io.bp2l621w5gqe/v1

Olivia Todd[1], kevin.dorn[1]

[1]USDA-ARS

USDA-ARS | USDA-ARS Sugar Beet G…

Olivia Todd
USDA-ARS

DOI: dx.doi.org/10.17504/protocols.io.bp2l621w5gqe/v1

Protocol Citation: Olivia Todd, kevin.dorn 2024. Genome Assembly, Scaffolding, and Annotation in Sugar Beet. **protocols.io https://dx.doi.org/10.17504/protocols.io.bp2l621w5gqe/v1**

Manuscript citation:
If this protocol is helpful to you, look out for the published paper to cite!

Here is the pre-print:
https://www.biorxiv.org/content/10.1101/2024.05.04.592522v1.abstract

Protocol status: Working
We use this protocol and it's working

Created: December 15, 2022

Last Modified: May 30, 2024

Protocol Integer ID: 99524

Keywords: Long read sequencing, genomics, de novo assembly, FC309, sugar beet genome

Abstract

This protocol was designed for use in *Beta vulgaris* and *Beta* crop wild relatives.
It uses several types of software and the online platform, GenSAS for assembly and annotation.
Data input sequencing types are HiFi, Iso-Seq and Omni-C with HiRise scaffolding.
The tissue collection protocol is available by searching the author profiles for "Sugar Beet Tissue Collection for Genome Assembly and Annotation with Long Read Sequencing".

This protocol is split into three pre-processing sections, as each input sequencing type requires some pre-processing. The fourth section uses generated input files from the preprocessing steps and explains where the file will be used and in which step.

Installing necessary software and modules

1 The computational steps are completed using USDA-ARS's High Performance Computing Platform, Ceres.

2 After logging in to the directory of choice, load the following modules or set up a conda virtual environment with the following software:
isoseq (v4.0.0)
lima (v2.9.0)
BBMap (v39.01)
hifiasm (v0.16.0)
pbmm2 (v1.11.99)

Alternatively, if you're a scinet user, you can activate a conda environment that contains this software using:

```
source activate /home/olivia.todd/.conda/envs/annotate_env
```

Iso-Seq data pre-processing

2.1 This protocol uses the file name "FC309" as the base name of the population being sequenced.

2.2 Use the following page as a guideline for the isoseq processing:
https://isoseq.how/clustering/cli-workflow.html

2.3 Starting with the recommended step 1 (Lima), run:

```
lima hifi.FC309.bam primers.fasta FC309_isoseq.bam --isoseq --
peek-guess
```

Where **hifi.FC309.bam** is the input file, and **FC309_isoseq.bam** is the base file name for the output file that will be used in step 6. The **primers.fasta** file is described on the isoseq workflow page.

2.4 Run isoseq refine using:

```
isoseq refine FC309_isoseq.NEB_5p--NEB_Clontech_3p.bam
primers.fasta
FC309.isoseq.flnc.bam --min-rq 0.9 --require-polya
```

Where **FC309_isoseq.NEB_5p--NEB_Clontech_3p.bam** is output from step 5, and input for step 6. **FC309.isoseq.flnc.bam** is the new output file. The other options are included, recommended by the documentation. This is the last pre-processing step before aligning the output to each haplome. First, we need to assemble the haplomes using the following steps.

OmniC data pre-processing

3 Upon receiving sequencing data, filter OmniC reads for adapter contamination and low quality bases using BBduk from BBMap with the following code, where **DTG_OmniC_792_R1.fastq.gz** (*_R1.fastq.gz) and **DTG_OmniC_792_R2.fastq.gz** are two separate input files:

```
for i in *_R1.fastq.gz
do
        base=$(basename "$i" _R1.fastq.gz)
        /path/to/software/bbmap/bbduk.sh -Xmx16g \
       in=${base}_R1.fastq.gz \
        in2=${base}_R2.fastq.gz \
        out=${base}_trimmed_R1.fastq \
        out2=${base}_trimmed_R2.fastq \
        ref=~/software/bbmap/resources/adapters.fa \
        k=23 ktrim=r qtrim=rl trimq=20 mink=11
done
```

The output files are **DTG_OmniC_792_trimmed_R1.fastq.gz** and **DTG_OmniC_792_trimmed_R2.fastq.gz** and are used in section 5, "Haplome assembly and scaffolding".

HiFi data pre-processing

4 After receiving HiFi reads, filter out adapters using HiFiAdapterFilt: https://github.com/sheinasim/HiFiAdapterFilt

Haplome assembly and scaffolding

5 Assemble filtered HiFi and OmniC reads using hifiasm (https://hifiasm.readthedocs.io/en/latest/index.html). Trimmed and filtered OmniC reads are included from the step 3 "OmniC pre-processing" for phasing (--h1 and --h2 flags).

FC309.asm is the output file prefix.
DTG_OmniC_792_trimmed_R1.fastq.gz and **DTG_OmniC_792_trimmed_R2.fastq.gz** are the OmniC reads. The three remaining **fastq.gz** files are three runs of HiFi sequencing.

```
hifiasm -o FC309.asm -t76 \
--h1 DTG_OmniC_792_trimmed_R1.fastq.gz \
--h2 DTG_OmniC_792_trimmed_R2.fastq.gz \
FC309_novogene.filt.fastq.gz \
FC309_hifi_1.filt.fastq.gz \
FC309_hifi_2.filt.fastq.gz
```

This process generates output files **FC309.asm.hic.hap1.p_ctg.fasta** and **FC309.asm.hic.hap2.p_ctg.fasta** which represent the primary phased OmniC/HiFi integrated assemblies.

5.1 The phased, contig level assemblies are run through the NCBI contamination tools "fcs-adaptor" and "fcs-gx" according to their document pages: https://github.com/ncbi/fcs?tab=readme-ov-file

5.2 Sort contigs in the **FC309.asm.hic.hap1.p_ctg.fasta** (haplome 1) and **FC309.asm.hic.hap2.p_ctg.fasta** (haplome 2) assembly by size (largest to smallest). Rename (ex: USDA_Bvulg_FC309_v0.3_hap1_contig_0000001), then rename the entire assembly file to **USDA_Bvulg_FC309_v0.3_hap1.fasta** and **USDA_Bvulg_FC309_v0.3_hap2.fasta**, respectively.

5.3 Submit **USDA_Bvulg_FC309_v0.3_hap1.fasta** and **USDA_Bvulg_FC309_v0.3_hap2.fasta** to DoveTail for the HiRise scaffolding process, used to construct pseudochromosomes.

Alignment of IsoSeq to scaffolded haplomes

6 Upon receipt of scaffolded fasta files **FC309V1.1.0.fasta** (haplome 1) and **FC309V1.2.0.fasta** (haplome 2), build an index for each fasta using:

```
pbmm2 index path/to/input/FC309v1.1.0.fasta
path/to/out/fc309v1.1.0.mm

and

pbmm2 index path/to/input/FC309v1.2.0.fasta
path/to/out/fc309v1.2.0.mm
```

For each haplome.

6.1 After indexing, use **FC309.isoseq.flnc.bam** from step 2.4 as the input for pbmm2. Align this file to each of the newly scaffolded haplomes.

```
pbmm2 align path/to/index/fc309v1.1.0.mmi 309isoseq.flnc.bam
/output/FC309.isoseq.v1.1.0_aligned.bam --preset ISOSEQ --sort

and

pbmm2 align path/to/index/fc309v1.2.0.mmi 309isoseq.flnc.bam
/output/FC309.isoseq.v1.2.0_aligned.bam --preset ISOSEQ --sort
```

Where **FC309.isoseq.v1.1.0_aligned.bam** and **FC309.isoseq.v1.2.0_aligned.bam** will be input for GenSAS at step 7.6.

6.2 Run isoseq3 collapse to collapse redundant isoforms using **FC309.isoseq.v1.1.0_aligned.bam** and **FC309.isoseq.v1.2.0_aligned.bam** as input. Output consists of unique isoforms in GFF format and secondary files containing information about the number of reads supporting each unique isoform.

```
isoseq3 collapse FC309.isoseq.v1.1.0_aligned.bam
FC309.isoseq.v1.1.0_align.collapse.fastq --do-not-collapse-extra-
5exons

and

isoseq3 collapse FC309.isoseq.v1.2.0_aligned.bam
FC309.isoseq.v1.2.0_align.collapse.fastq --do-not-collapse-extra-
5exons
```

Where the output gff, **FC309.isoseq.v1.1.0_align.collapse.fasta** and **FC309.isoseq.v1.2.0_align.collapse.fasta** will be used in GenSAS step 7.9 for refinement with PASA in their respective project files.

Using GenSAS for haplome annotation

7 Here we describe our pipline for annotation of the two haplomes using GenSAS. **https://www.gensas.org/**

7.1 Start a new project for haplome 1 after creating a username and password at **https://www.gensas.org/** with appropriate metadata. The analysis for haplome 1 and 2 will be the same, but have two separate project files.

7.2 Under the "Sequences" tab, upload the scaffolded haplome file, **FC309v1.1.0.fasta** that was a result of step 5.3. This file has unanchored contigs that will be annotated along with the 9 scaffolded chromosomes.

7.3 Under the "Repeats" tab, use the following parameters for the RepeatMasker job:

```
Search Engine: repeatmasker_engine_sel : ncbi
Speed / Sensitivity: repeatmasker_speed_sel : -q
Repeat Library: FC309v1.1.0-families.fa
version : 4.1.1
```

*Note:
To generate the **FC309v1.1.0-families.fa** repeat family fasta file needed for this job, run RepeatModeler according to the steps outlined at **https://www.repeatmasker.org/RepeatModeler/**. Build a database, then run the main command. Generate a repeat file for each haplome, and upload **FC309v1.1.0-families.fa** under the "Evidence" tab.

7.4 Under the "Masking" tab, the masked repeat consensus is created with the repeatmasker output from step 7.3

7.5 Under the "Align" tab, run two separate BLASTn jobs with two files from the EL10.2_2 genome release, and the RefBeet-1.2.2 genome release, available on Phytozome:
1. **Bvulgarisssp_vulgaris_782_EL10.2_2.transcript.fa.gz**
2. **GCF_000511025.2_RefBeet-1.2.2_genomic.fasta**

Upload these files under the "Evidence" tab.
Run each job with these specific parameters (default):

```
Expect: expect : 1e-50
Max Hits Per Region (culling limit): culling_limit : 5
Word Size: word_size : 11
Gap Open: gapopen : 5
Gap Extend: gapextend : 2
Maximum HSP Distinace: max_hsp_gap : 30000
version : 2.12.0
```

7.6 Under the "Structural" tab, give BRAKER2 the processed IsoSeq file from step 6.1, **FC309.isoseq.v1.1.0_aligned.bam**.
Upload this file to the project under the "Evidence" tab.

7.7 Under the "Consensus" tab, use the default weights for the two BLAST jobs, and the singular BRAKER job. Run EvidenceModeler.

```
Job Name (weight)

BRAKER: (1)
BLAST nucleotide (blastn): (10)
BLAST nucleotide (blastn)2: (10)
```

7.8 Under the "OGS" tab, run an option BUSCO job on the EvidenceModeler output. After the BUSCO job is complete, select the EvidenceModeler output and use that selected job as the OGS.

7.9 Under the "Refine" tab, use **FC309.isoseq.v1.1.0_align.collapse.fasta** from step 6.2 and the EvidenceModeler job to refine the identified genes.

```
Transcripts FASTA file: FC309.isoseq.v1.1.0_align.collapse.fasta
version : 2.4.1

Selected job : EvidenceModeler
```

7.10 Under the "Functional" step, run InterProScan and Pfam jobs to get functional gene annotations using the PASA refined gene set.

7.11 Under the "Publish" step, check the box for all available jobs, and select filename options. When the publishing job is finished, download all files into a zipped folder for local and cloud backup.

Congratulations!

8 Your genome assembly and annotation is complete!