Apr 26, 2024

De - Novo Genome Protocol from only ONT reads

This protocol is a draft, published without a DOI.

Chase Donnelly[1]

[1]University of Antwerpen

Chase Donnelly
University of Antwerpen

Protocol Citation: Chase Donnelly 2024. De - Novo Genome Protocol from only ONT reads. **protocols.io** **https://protocols.io/view/de-novo-genome-protocol-from-only-ont-reads-dciq2udw**

Protocol status: In development
We are still developing and optimizing this protocol

Created: April 22, 2024

Last Modified: April 26, 2024

Protocol Integer ID: 98608

Disclaimer

This protocol is still under development and has not been finalized.

Abstract

Draft of pipeline and code used for de novo genome assembly of non model plant specis

DNA Extraction

1 The first challenge to developing this protocol was determining a DNA extraction technique that produced long and high-quality DNA. This protocol used a modified version of the fever tree gdna protocol available via Nanopore's community website. These modifications have since been added to or surpassed in the published protocol and so the latest version on the community page should be used for DNA extraction and Library prep for sequencing before following the rest of this protocol.

Software Setup

2 After sequencing the first step is to set up a clean compute space to analyze the sequencing data to avoid issues with mixing versions of software.

3 To set up an environment with conda or mamba you can use the following code to create an environment and install the required programs:

```
#First create environment with mamba
mamba create -n genome_env
#Next activate our new environment
mamba activate genome_env
```

4 There are a few different ways to manage a compute environment, here conda/mamba was used to create a clean environment as it works with all of the downstream programs and has an extensive manual. Mamba is a newer alternative to conda that can install programs with greater speed.

conda doc: https://conda.io/projects/conda/en/latest/user-guide/getting-started.html

mamba doc: **https://mamba.readthedocs.io/en/latest/**

mamba and conda can both be used in the same way, just replace mamba with conda in the codes below.

Quality Trimming

5 After base calling using high accuracy settings with no modifications, the resultant data is then further quality checked with Nano Pack 2 (v), a toolset designed to perform multiple long read processing and analysis steps on ONT reads (De Coster and Rademakers 2023). Short reads <1500 bp in length and reads with a Q score below 10 were filtered out of the sequencing data prior to assembly.

```
#install program into environment
mamba install -c bioconda nanoplot
#trim and remove reads less than 1kb in size
zcat reads.fastq.gz |
    /path to Nano Pack/NanoLyse -r all.fastq. | \
    /path to Nano Pack/NanoFilt --headcrop 200 --tailcrop 200 | \
    /path to Nano Pack/NanoFilt -q 7 -l 1000 | \
    gzip > preliminary-filter.fastq.gz
```

De Novo Assembly

6 There are multiple assembly options that can work well on plant data (Table 1). As this study uses only long read data, only assemblers optimized for long read assembly, such as Flye, WTDBG2, and Canu were tested.

	Tool	Link
	Flye	https://github.com/fenderglass/Flye
	Canu	https://github.com/marbl/canu
	WTDBG2	https://github.com/ruanjue/wtdbg2
	NextDenovo	https://github.com/Nextomics/NextDenovo

Table 1: List of tools that can be used for de novo genome assembly and links to their repositories

```
#install programs to environment
mamba install flye
mamba install -c conda-forge -c bioconda -c defaults canu
#run flye with recommended parameters for ONT data
flye --nano-hq all.fastq.gz \
--read-error 0.03 \
--out-dir Flye_Out \
--genome-size 400m \ #set for genome based on size estimation
-t <n>

#run conda
canu \
 -p name -d ./genome_directory\
 genomeSize=400m \ #set for genome based on size estimation
 -pacbio-hifi all.fastq.gz \ #at the time of writting there is no
specific ONT setting in Canu and so long read pacbio setting is
used.
```

Quality Assessment

7 There are many ways to check the quality of an assembly, here we discuss three that will give a good general representation of assembly quality. These tools can be used throughout the protocol to determine the effect of each step of the process. The first tool Nano Pack 2 (v), a toolset designed to perform
multiple long read processing and analysis steps on ONT reads (De Coster and Rademakers 2023), is good for visualization of the data and comes with a large range of tools to deal with ONT reads.

Next Quast is the standard genome evaluation tool to compare contig number, length, and N50 of each assembly (Gurevich et al. 2013). Quast can provide multiple scores and measurements for genome assemblies with and without a reference genome available. When a reference is available quast can also show contig alignment and gaps in the two assemblies.

BUSCO is the most commonly used tool to determine the completeness and redundancy of the genome and is based on use of universal single copy orthologs, aka highly conserver genes (Manni et al. 2021). BUSCO looks for the presence or absences of a set of highly conservative genes based on the organism of study, with a score in the high 90% being considered good. Example code for all three tools is provided below.

```
#install nanoplot
mamba install -c bioconda nanoplot
#run with basic configuration
NanoPlot --summary sequencing_summary.txt --loglength -o summary-
plots-log-transformed

#install
mamba install bioconda::quast
#run quast with a reference genome to compare with
quast genome.fasta -o quast_results -r reference_genome.fna --
features genomic.gtf --threads <n>
#run quast with no reference
quast genome.fasta -o quast_results --threads <n>

#Check genomes with BUSCO
busco -l viridiplantae_odb10 -m genome -c 12 -i genome.fasta -o
output_file_name
# change viridiplantae_odb10 to the best lineage for your genome
```

Polishing

8 After initial assembly multiple polishing and correction are performed. The first correction step is polishing and there again are a variety of tools to choose from such as Racon, Pilon, and POLCA.

	Tool	Link
	Racon	https://github.com/isovic/racon
	Pilon	https://github.com/broadinstitute/pilon
	POLCA	https://github.com/alekseyzimin/masurca
	Medaka	https://github.com/nanoporetech/medaka

Table 2: List of tools that can be used for de novo genome polishing and links to their repositories

This protocol uses a combination of Racon and Medaka that is known to produce good results on plant assemblies (Lee et al. 2021). The code used is below, minimap2 is used to map reads for each step of polishing.

```
#install programs
mamba install bioconda::racon
mamba install bioconda::medaka
mamba install install minimap2

#run minimap2 with recommended parameters for ONT
minimap2 -ax map-ont -t <n>-K5g -I50g polished1.fasta
all_lotus.fastq.gz > polished1.sam
#run racon with recommended parameters for ONT
racon -m 8 -x 6 -g -8 -w 500 -t <n> all.fastq.gz polished1.sam
polished1.fasta > polished2.fasta
#repeat this (minimap2 into racon) x4
#run medaka with recommend ONT parameters, the model should be the
same model used for basecalling ONT reads
medaka_consensus -i all.fastq.gz -d polished4.fasta -o
medaka_consensus -t <n>-m r1041_e82_400bps_hac_v4.2.0
```

Remove Haplotigs

9 As discussed, plant genomes contain areas of high heterogeneity, this can cause issues when looking to recognize haplotype homology during genome assembly. To counteract this issue, Purge Haplotigs was used on the draft genome assembly during multiple steps to improve the genome assembly by labeling haplotigs. Purge Haplotigs uses Minimap2 alignments and mapped read coverage to discover contig pairs that are syntenic and assign one of the contigs to a haplotig “pool” and remove the haplotigs from the primary assembly

```
#
purge_haplotigs hist -b all_reads.bam -g consensus.fasta -t 12
#
purge_haplotigs contigcov -i all_reads.bam.gencov -l 0 -m 100 -h
200
#
purge_haplotigs purge -g consensus.fasta -c coverage_stats.csv -b
all_reads.bam -d -a 50 -t 12
```

Scaffolding

10 When high quality genomes of closely related species are available, tools such as RagTag can be used to scaffold assembled contigs to known chromosomes of the related species (Volarić et al. 2022).

```
#Install ragtag
mamba install -c bioconda ragtag
#Run ragtag with the recommended settings
ragtag.py scaffold reference_genome.fna genome.fasta --remove-small
-f 10000 -t 12 -u -e scaffold.lst
#Remove the _ragtag from the names for downstream analysis
sed -e 's/_RagTag//g' ragtag.genome.fasta > genome.fasta
#Check names
bioawk -c fastx '{print $name}' scaffold.fasta > scaffold.lst
```

Structural Annotation

11 After assembly is completed, the draft genome must then go through the process of structural gene annotation. This method finds the positions of genomic features such as protein-coding genes, promoters, and regulatory elements, which can be a challenge for plant genomes as they contain highly repetitive elements (M. E. Bolger, Arsova, and Usadel 2018). There are several popular protocols available for this process and all go through a similar workflow that use multiple tools and consists of three main stages: masking noncoding regions, predicting gene structure positions, and then finally assigning biological meaning to those predictions (Table 12; Jung et al. 2020).

12

	Tool	Link
	RepeatMasker	https://www.repeatmasker.org/
	RepeatModeler2	https://www.repeatmasker.org/RepeatModeler/
	Braker3	https://github.com/Gaius-Augustus/BRAKER
	Maker	https://github.com/Yandell-Lab/maker
	GeneMark-ETP	https://github.com/gatech-genemark/GeneMark-ETP
	AUGUSTUS	https://github.com/Gaius-Augustus/Augustus
	TSEBRA	https://github.com/Gaius-Augustus/TSEBRA
	Compleasm	https://github.com/huangnengCSU/compleasm

Table 3: List of tools for structural annotation of de novo genomes and links to their repositories

13 For plants, the general process of repeat masking is performed by two main packages RepeatMasker and Repeatmodeler2, both of which soft mask or obscure repeated regions to annotation software (Smit, Hubley, and Green 2015; Flynn et al. 2020). First we use repeatmodeler to create a database and

Below is example code for both programs.

```
#install program
mamba install repeatmodeler
#create database
BuildDatabase -name <nameofdatabase> <reference_genome.fa>
#run program
RepeatModeler -database <nameofdatabase> -pa <n> -LTRStruct >
out.log
#now we can use this output to run RepeatMasker to softmask our
genome
RepeatMasker -pa <n> -gff -lib <output_of_modeler> -dir
<output_directory_name> <genome.fasta>
```

14 Gene prediction on the remaining regions then takes place via evidenced based, evidence free, or a combination of the two methods. Evidence based methods use RNA-Seq data and protein sequence search algorithms, while evidence free methods rely on gene structure and signal based searches to predict genes (Z. Wang, Chen, and Li 2004). Here we will use BRAKER3, for on protein and RNA seq data to complete to most accurate annotation. BRAKER uses a combination of GenMark-ETP, AUGUSTUS and TESBRA to perform annotations. An example code is provided below.

```
wget https://repo.anaconda.com/archive/Anaconda3-2018.12-Linux-x86_64.sh
bash bin/Anaconda3-2018.12-Linux-x86_64.sh # do not install VS (needs root privileges)
mamba install -c anaconda perl
mamba install -c anaconda biopython
mamba conda install -c bioconda perl-app-cpanminus
mamba install -c bioconda perl-file-spec
mamba install -c bioconda perl-hash-merge
mamba install -c bioconda perl-list-util
mamba install -c bioconda perl-module-load-conditional
mamba install -c bioconda perl-posix
mamba install -c bioconda perl-file-homedir
mamba install -c bioconda perl-parallel-forkmanager
mamba install -c bioconda perl-scalar-util-numeric
mamba install -c bioconda perl-yaml
mamba install -c bioconda perl-class-data-inheritable
mamba install -c bioconda perl-exception-class
mamba install -c bioconda perl-test-pod
mamba install -c bioconda perl-file-which # skip if you are not comparing to reference annotation
mamba install -c bioconda perl-mce
mamba install -c bioconda perl-threaded
mamba install -c bioconda perl-list-util
mamba install -c bioconda perl-math-utils
mamba install -c bioconda cdbtools
mamba install -c eumetsat perl-yaml-xs
mamba install -c bioconda perl-data-dumper
mamba install braker3

#Change perl paths to work with conda
perl change_path_in_perl_scripts.pl "/home/chase/miniforge3/envs/braker3env/bin perl"

#run Braker, protiens cabe be taken OrthDB/Prothint, see https://github.com/gatech-genemark/ProtHint#protein-database-preparation for instructions
braker.pl --genome=<genome.fasta> --prot_seq=<proteins.fa> --bam=genome.bam --threads=<n>--busco_lineage=<fabales_odb10>

#get gtf statistics
/Path_to_ausgustus/agat_sp_statistics.pl -gff braker.gtf -g Lotus_use.fasta
```

Functional Annotation

15 Functional annotation was completed with EnTAP. EnTAP was chosen to add functional annotation to our draft genome due to its ability to work across multiple curated datasets, and due to its preexisting setup for the RNA protocols (Hart et al. 2020b). Using EnTAP allowed for the incorporation of the annotations from closely related species to increase the results of functional annotation. Example code is provided below.

16

```
#install EnTAP based on the site instructions
(https://entap.readthedocs.io/en/latest/)

#Run database configuration, this can be run for all databases that
your will search against eg: nr, swissprot, etc..
EnTAP --config -d  /Path_to_database_fasta/nr_database.fasta --out-
dir output_folder -t <n>--ini entap_config.ini
#Run EnTAP
EnTAP --runP -i protiens.pep -d /Path_to_database/nr.dmnd -t <n>--
ini Path_to_Entap/EnTAP-v0.10.8-beta/entap_config.ini
```