Jul 17, 2022

ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes

In 1 collection

Stephen Douglas Russell[1]

[1]The Hoosier Mushroom Society

1 *Works for me* Share

dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1

The Hoosier Mushroom Society

Stephen Douglas Russell

ABSTRACT

This protocol assumes that your MinION run has been completed and the data from the run has been saved. It should take you from raw data to useable FASTA files for each of your fungal barcodes.

DOI

dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1

PROTOCOL CITATION

Stephen Douglas Russell 2022. ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes. **protocols.io** https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1

COLLECTIONS

ONT DNA Barcoding Fungal Amplicons w/ MinION &amp; Flongle

LICENSE

CREATED

Jun 14, 2022

LAST MODIFIED

Jul 17, 2022

Citation: Stephen Douglas Russell ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1

PROTOCOL INTEGER ID

64537

PARENT PROTOCOLS

Part of collection

ONT DNA Barcoding Fungal Amplicons w/ MinION &amp; Flongle

1 This protocol assumes the experiment name is "FirstRun."

Create a new working folder on the desktop. Ex - FirstRun. Within that create a new folder called "fast5," another called "Programs," and a final one called "NGSpeciesID."

I will start by copying all of the fast5 files from:

/var/lib/minknow/data/./FirstRun/CellName/long_unique_name/fast5 to the newly created fast5 folder on the desktop

2 Create an index file from your extraction template papers. This will allow you to link all of your reads with the individual specimens. A template for 7 plates (672 specimens) can be found here:

NANOPORE TEMPLATE THIRD RUN.xlsx

This .xlsx is formatted to utilize the Lab Code and iNaturalist # columns as the only inputs. It will combine these and all of the other columns into a single cell - concatenating them all into the final file name. For the Lab Code, I will typically put these into the iNaturalist "Voucher Number(s)" Observational Field, and then export them all at once into a .csv from iNat. This allows me to simply copy and paste many iNat numbers at once, without ever needing to input any of the numbers manually.

After editing, save as a tab-delimited text file in the NGSpeciesID folder. You will need to remove most of the final columns from the template. The final output should be saved like this:

Index.txt

3 Copy these Python scripts into the Programs folder you just created.

minibar.py

summarize.py

primers.txt

Citation: Stephen Douglas Russell ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1

4

Run Guppy Basecalling

```
guppy_basecaller -x "cuda:all" -i ~/Desktop/FirstRun/fast5 -s ~/Desktop/FirstRun/basecalling --flowcell FLO-FLG001 --kit SQK-LSK110 --records_per_fastq 0 --trim_adapters --trim_strategy dna
```

This command runs basecalling for a Flongle flowcell using the LSK110 kit.

Pop!_OS 22.04

For a Flongle cell with 1.15Gb of bases and 1.18M reads, this command takes about 37 minutes to run. Example output:

```
Init time: 681 ms

0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
***************************************************
Caller time: 2216974 ms, Samples called: 13852213920, samples/s: 6.24825e+06
Finishing up any open output files.
Basecalling completed successfully.
```

Sometimes after the run I need to restart the CPU before this command runs successfully.

5

Combine all FASTQ files into a single file

```
cat ~/Desktop/FirstRun/basecalling/pass/*runid*.fastq > ~/Desktop/FirstRun/basecalling/pass/basecall.fastq
```

Pop!_OS 22.04

6

Citation: Stephen Douglas Russell ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes https://dx.doi.org/10.17504/protocols.io.dm6gpbm8lzp/v1

Validate the number of reads in your file

```
cat ~/Desktop/FirstRun/basecalling/pass/basecall.fastq | wc -l | awk
'{print $1/4}'
```

Pop!_OS 22.04

850800

7

Remove the uncombined FASTQ files

```
rm ~/Desktop/FirstRun/basecalling/pass/*runid*.fastq
```

Pop!_OS 22.04

8

Move your fastq file and demultiplexer to a second location

```
cp ~/Desktop/FirstRun/basecalling/pass/basecall.fastq
~/Desktop/FirstRun/NGSpeciesID/basecall.fastq
cp ~/Desktop/FirstRun/Programs/minibar.py
~/Desktop/FirstRun/NGSpeciesID/minibar.py
cp ~/Desktop/FirstRun/Programs/summarize.py
~/Desktop/FirstRun/NGSpeciesID/summarize.py
cp ~/Desktop/FirstRun/Programs/primers.txt
~/Desktop/FirstRun/NGSpeciesID/primers.txt
```

Pop!_OS 22.04

9

MinIONQC.R

```
cd ~/Desktop/FirstRun/Programs
Rscript MinIONQC.R -i
~/Desktop/SecondRun/basecalling/sequencing_summary.txt -o
~/Desktop/SecondRun/basecalling/pass/summary/
```

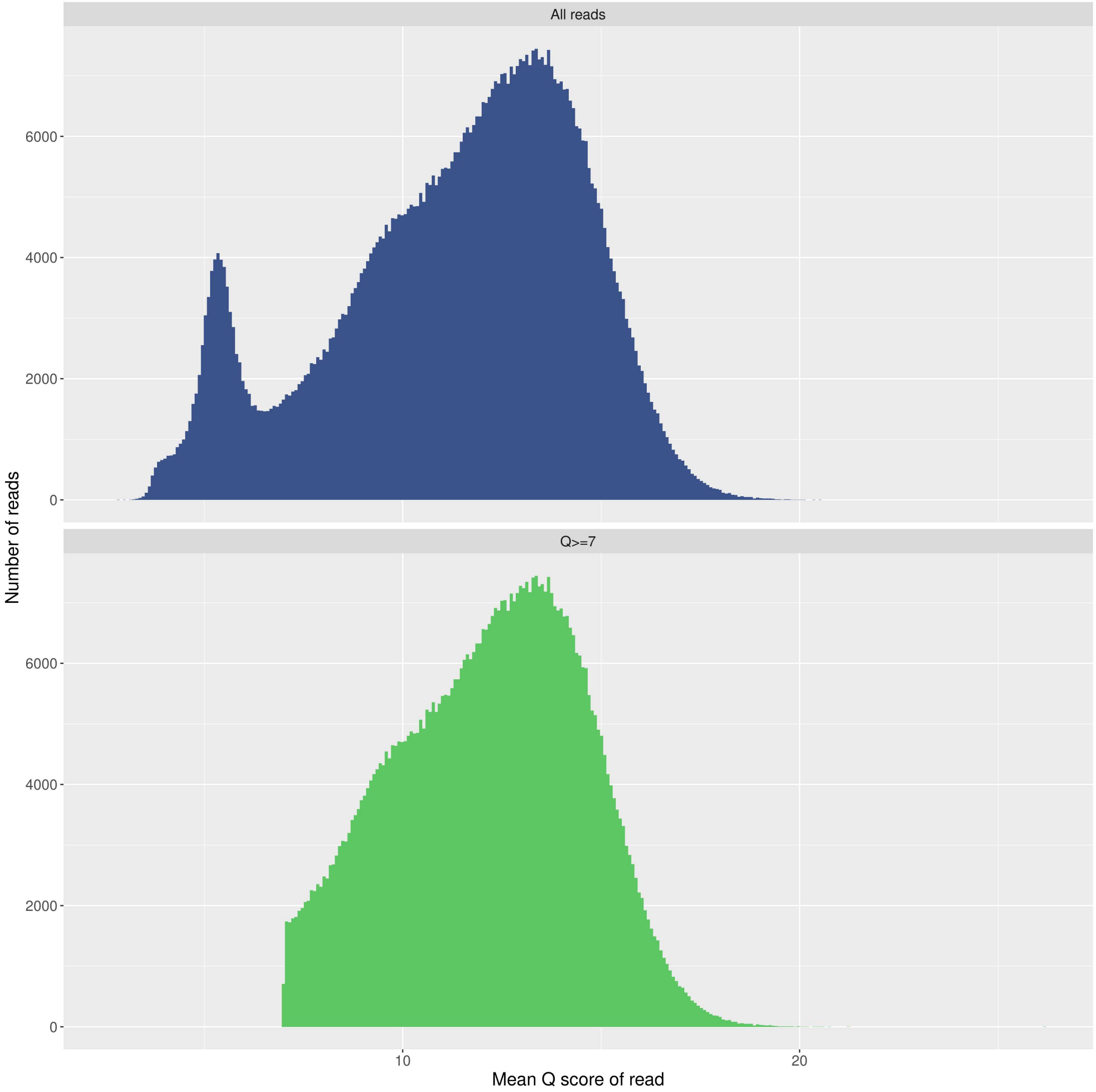
Runs summary charts for the sequencing run.

Pop!_OS 22.04

```
INFO [2022-07-03 17:16:12] Loading input file:
/home/user/Desktop/FirstRun/basecalling/sequencing_summary.txt
INFO [2022-07-03 17:16:12] MinION flowcell detected
INFO [2022-07-03 17:16:19] basecalling: creating output
directory:/home/userDesktop/FiirstRun/basecalling/pass/summary//basecalling
INFO [2022-07-03 17:16:19] basecalling: summarising input file for flowcell
INFO [2022-07-03 17:16:19] basecalling: plotting length histogram
INFO [2022-07-03 17:16:21] basecalling: plotting mean Q score histogram
INFO [2022-07-03 17:16:22] basecalling: plotting flowcell overview
INFO [2022-07-03 17:16:42] basecalling: plotting flowcell yield over time
INFO [2022-07-03 17:16:50] basecalling: plotting flowcell yield by read length
INFO [2022-07-03 17:16:56] basecalling: plotting sequence length over time
INFO [2022-07-03 17:17:10] basecalling: plotting Q score over time
INFO [2022-07-03 17:17:23] basecalling: plotting reads per hour
INFO [2022-07-03 17:17:25] basecalling: plotting read length vs. q score scatterplot
INFO [2022-07-03 17:17:35] basecalling: plotting flowcell channels summary
histograms
INFO [2022-07-03 17:17:35] basecalling: plotting physical overview of output per
channel
```

Review the images that are generated. Ensure the quality scores of your run are in an appropriate range. For a 9.4.1 Flongle with Q20+ (V12) K12 chemistry, I typically get a peak in the 12-13 range.

Citation: Stephen Douglas Russell ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1



Mean Q scores for all of the reads in the run. You want to see the peak well above 10. The lower the Q score, the more errors your results will have.

Example of all outputs from this command: MinIONQC.zip

10

Go to your Programs folder and use MiniBar for Demultiplexing

```
cd ~/Desktop/FirstRun/NGSpeciesID
./minibar.py -F Index.txt basecall.fastq
```

Pop!_OS 22.04

This should take about 3-4 minutes to run.

Citation: Stephen Douglas Russell ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v1

```
850800 seqs: H 734497 HH 581546 Hh 72882 hh 53453 IDs 707881 Mult_IDs
105306 (193.4800s)
```

11

Run NGSpeciesID for read filtering, clustering, consensus generation and polishing

```
conda activate NGSpeciesID
for file in *.fastq; do
bn=`basename $file .fastq`
NGSpeciesID --ont --consensus --sample_size 500 --m 800 --s 400 --
medaka --primer_file primers.txt --fastq $file --outfolder ${bn}
done
```

Pop!_OS 22.04

This program will take about 3-6 hours to complete.

12

Create a summary of your NGSpeciesID consensus data

```
python summarize.py ~/Desktop/FirstRun/NGSpeciesID
```

Pop!_OS 22.04