

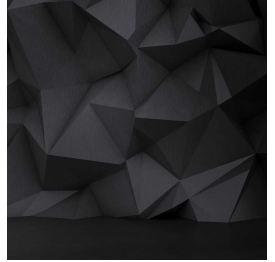
Mar 13, 2023

Version 2

ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes V.2

DOI

<https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v2>



Stephen Douglas Russell¹

¹The Hoosier Mushroom Society

Mycota Lab



Stephen D Douglas Russell

Mycota Lab, Biodiverse, MycoMap

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v2>

Protocol Citation: Stephen Douglas Russell 2023. ONT Basecalling, Demultiplexing, and Analysis for Fungal Barcodes. protocols.io <https://dx.doi.org/10.17504/protocols.io.dm6gpbm88lzp/v2> Version created by **Stephen D Douglas Russell**



License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: October 12, 2022

Last Modified: March 13, 2023

Protocol Integer ID: 71213

Keywords: analysis for fungal barcode, simplex read, fungal barcode, average quality scores from each simplex, simplex, flowcell, raw data to useable fasta file, flowcells with v14 chemistry, duplex component, duplex, minion run, raw data, useable fasta file, demultiplexing

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 containing consensus sequences for each of your fungal barcodes.

Note: This protocol assumes you are using 10.4.1 flowcells with V14 chemistry, so there is a process to filter out and combine your duplex reads with your simplex reads. It will also show the average quality scores from each simplex and duplex component.

Initial Post-Run Preparation

- 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 10 plates (960 specimens) can be found here:

 **NANOPORE TEMPLATE SEVENTH ...**

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**

My shorthand "quick" guide for all of the following commands can be found in the file below. I use this file whenever working through this protocol. It is saved on my desktop.



Just copy-and-paste the commands sequentially into the terminal.

Guppy Run Code - Final

SUP Basecalling with Guppy

11h 20m

4

2h

Run the basecalling command. The command below uses Super-accuracy mode with Guppy.

```
guppy_basecaller -x "cuda:all" -i ~/Desktop/FirstRun/fast5 -s
~/Desktop/FirstRun/simplex_calls --config
dna_r10.4.1_e8.2_400bps_sup.cfg --records_per_fastq 0 --
trim_adapters --trim_strategy dna --chunks_per_runner 256
```

For a Flongle cell with 1.15Gb of bases and 700 - 1.18M reads, this command takes about  02:00:00 to run. Example output:

Expected result

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 This command will install the duplex tools you will need for filtering them out and combining them with the simplex reads.

```
python -m venv venv --prompt duplex
. venv/bin/activate
pip install duplex_tools
```



- 6 Around 20-30% of the reads in your final results will be duplex reads - the single fragment is read twice - once in the forward direction and one in the reverse direction. These reads need to be identified with the command below.

```
duplex_tools pairs_from_summary
~/Desktop/FirstRun/simplex_calls/sequencing_summary.txt
~/Desktop/FirstRun/simplex_calls/pairs
duplex_tools filter_pairs
~/Desktop/FirstRun/simplex_calls/pairs/pair_ids.txt
~/Desktop/FirstRun/simplex_calls/pass
```

- 7 Rerun Guppy basecalling with the duplex reads alone that were parsed out in the previous step. This step may take about  01:15:00 .

1h 15m

```
guppy_basecaller_duplex \
-i ~/Desktop/FirstRun/fast5 \
-r -s ~/Desktop/FirstRun/duplex_calls \
-x 'cuda:all' -c dna_r10.4.1_e8.2_260bps_sup.cfg \
--chunks_per_runner 256 \
--duplex_pairing_mode from_pair_list \
--duplex_pairing_file
~/Desktop/FirstRun/simplex_calls/pairs/pair_ids_filtered.txt
```

- 8 Combine all FASTQ files into a single file for simplex and duplex calls, show the read counts, and zip them.



```
cat ~/Desktop/FirstRun/simplex_calls/pass/*runid*.fastq >
~/Desktop/FirstRun/simplex_calls/pass/basecall.fastq
cat ~/Desktop/FirstRun/simplex_calls/pass/basecall.fastq | wc -l |
awk '{print $1/4}'
gzip ~/Desktop/FirstRun/simplex_calls/pass/basecall.fastq
cat ~/Desktop/FirstRun/duplex_calls/pass/*runid*.fastq >
~/Desktop/FirstRun/duplex_calls/pass/basecall.fastq
cat ~/Desktop/FirstRun/duplex_calls/pass/basecall.fastq | wc -l |
awk '{print $1/4}'
gzip ~/Desktop/FirstRun/duplex_calls/pass/basecall.fastq
gzip ~/Desktop/FirstRun/duplex_calls/pass/*.fastq
```

9

Remove the uncombined FASTQ files.

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

Validate the QC of the Run

11h 20m

10 Compile quality control summary charts for the simplex and duplex reads.

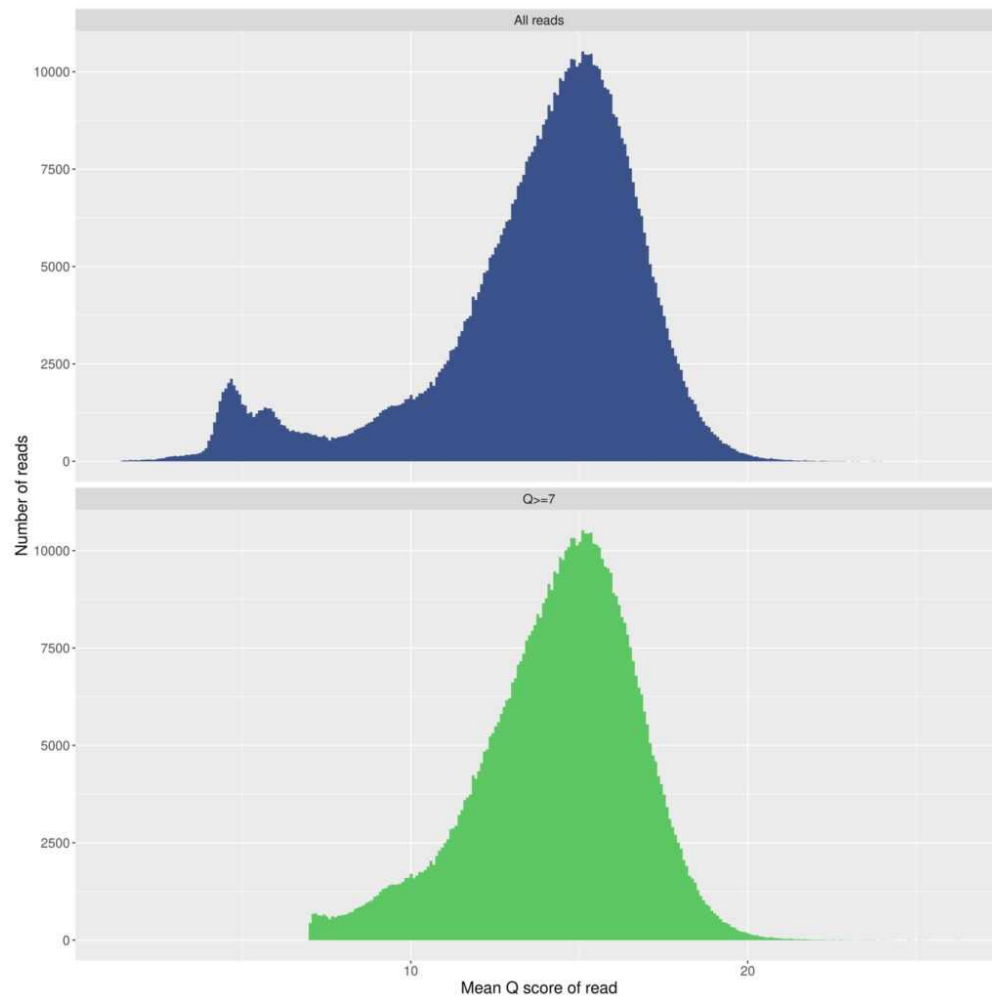
```
cd ~/Desktop/FirstRun/Programs
Rscript MinIONQC.R -i
~/Desktop/FirstRun/simplex_calls/sequencing_summary.txt -o
~/Desktop/FirstRun/QC-Reports-simplex
Rscript MinIONQC.R -i
~/Desktop/FirstRun/duplex_calls/sequencing_summary.txt -o
~/Desktop/FirstRun/QC-Reports-duplex
```



Expected result

```
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 10.4.1 Flongle with "Q20+" V14 chemistry, I typically get a peak in the 15-16 range.



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:



Merge the Simplex and Duplex Reads

11h 20m

- 11 Use seqkit to extract all the filenames of reads from the simplex run.

```
seqkit seq --name  
~/Desktop/FirstRun/simplex_calls/pass/basecall.fastq.gz >  
~/Desktop/FirstRun/simplex_calls/pass/simplex_ids.txt
```

- 12 Create additional file structure and move around necessary files.

```
mkdir ~/Desktop/FirstRun/combined_bases/  
cd ~/Desktop/FirstRun/combined_bases/  
cp ~/Desktop/FirstRun/simplex_calls/pass/basecall.fastq.gz  
~/Desktop/FirstRun/combined_bases/simplex_basecall.fastq.gz  
cp ~/Desktop/FirstRun/duplex_calls/pass/basecall.fastq.gz  
~/Desktop/FirstRun/combined_bases/duplex_basecall.fastq.gz  
cp ~/Desktop/FirstRun/simplex_calls/pass/simplex_ids.txt  
~/Desktop/FirstRun/combined_bases/simplex_ids.txt  
cp ~/Desktop/FirstRun/simplex_calls/pairs/pair_ids_filtered.txt  
~/Desktop/FirstRun/combined_bases/pair_ids_filtered.txt
```

- 13 Replace the reads in the simplex with the duplex reads.

```
{ sed 's/ /\n/'  
~/Desktop/FirstRun/combined_bases/pair_ids_filtered.txt | \  
seqkit grep -v -f -  
~/Desktop/FirstRun/combined_bases/simplex_basecall.fastq.gz ; \  
zcat ~/Desktop/FirstRun/duplex_calls/pass/*.fastq.gz ; } \  
| gzip - > combined.fastq.gz
```

Note: This code was derived from a post in the [ONT Community Forums](#). It may be in need of some future revision.

- 14 Manually unzip the file here: ~/Desktop/FirstRun/combined_bases/combined.fastq.gz
- 15 Review the total number of reads in your final file and move some files for additional analysis.



```
cat ~/Desktop/FirstRun/combined_bases/combined.fastq | wc -l | awk
'{print $1/4}'
cp ~/Desktop/FirstRun/combined_bases/combined.fastq
~/Desktop/FirstRun/NGSpeciesID/combined.fastq
cp ~/Desktop/FirstRun/Programs/minibar.py
~/Desktop/FirstRun/NGSpeciesID/minibar.py
cp ~/Desktop/FirstRun/Programs/primers.txt
~/Desktop/FirstRun/NGSpeciesID/primers.txt
cp ~/Desktop/FirstRun/Programs/Index.txt
~/Desktop/FirstRun/NGSpeciesID/Index.txt
cp ~/Desktop/FirstRun/Programs/summarize.py
~/Desktop/FirstRun/NGSpeciesID/summarize.py
```

Demultiplex the Reads

11h 20m

- 16 Demultiplex your samples using MiniBar. This should take less than 00:05:00 .

5m

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

- 17 Remove several large files that are not necessary for most use cases and will just make your final analysis take longer.

```
rm combined.fastq
rm sample_Multiple_Matches.fastq
rm sample_unk.fastq
```

Create the Final Consensus Sequences

11h 20m

- 18 Utilize NGSpecies ID to generate your final consensus sequences from your demultiplexed samples. This can take 08:00:00 . It would be great if someone could get this to run through the GPU.

8h



```
conda activate NGSpeciesID
for file in *.fastq; do
bn=`basename $file .fastq`
NGSpeciesID --ont --consensus --sample_size 500 --m 730 --s 400 --
medaka --primer_file primers.txt --fastq $file --outfolder ${bn}
done
```

Summarize the Data

11h 20m

- 19 Create a summary file for your results.

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