



Jul 17, 2022

# ONT Sequencing IT/Compute Pop!\_OS 22.04 Setup

In 1 collection

Stephen Douglas Russell<sup>1</sup><sup>1</sup>The Hoosier Mushroom Society

1 Works for me

Share

[dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v1](https://dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v1)

The Hoosier Mushroom Society

Stephen Douglas Russell

## ABSTRACT

The IT requirements for processing MinION data should be carefully reviewed before purchasing a MinION device. You will want to go with a Linux system. [System76](#) is really the primary/best vendor for laptops. Pay careful attention to the GPU. It is probably the most important component for fast processing of the data. [Here is is a link](#) to a Facebook thread of some discussion when first considering the specs required.

Setting up all of the programs/dependencies, particularly for utilizing the GPU during the analytical workflows is the next important step. You will want to get all of this in place before you start with the lab workflows, as there are many things that could go wrong or that you will need to work through in order for you to be able to actually begin a run.

## DOI

[dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v1](https://dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v1)

## PROTOCOL CITATION

Stephen Douglas Russell 2022. ONT Sequencing IT/Compute Pop!\_OS 22.04 Setup. **protocols.io**  
<https://dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v1>

## COLLECTIONS ⓘ

 **ONT DNA Barcoding Fungal Amplicons w/ MinION & Flongle**

## KEYWORDS

Oxford Nanopore Technologies, DNA sequencing, DNA barcoding, MinION, Flongle, system76, linux, POP! OS

#### 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

#### CREATED

Jul 17, 2022

#### LAST MODIFIED

Jul 17, 2022

#### PROTOCOL INTEGER ID

66899

#### PARENT PROTOCOLS

Part of collection

[ONT DNA Barcoding Fungal Amplicons w/ MinION & Flongle](#)

- 1 The final setup I went with can be found below. It was expensive (around 💰**4000** for the laptop), but should be able to achieve live basecalling for two MinION devices at the same time. Overall specs of my laptop:

Pop!\_OS 21.10 (64-bit) with full disk-encryption

4.6 GHz i7-11800H - up to 4.6 GHz - 24MB Cache - 8 Cores - 16 Threads)

64 GB Dual Channel DDR4 at 3200 MHz (2x 32GB)	\$549.00
---	----------

1 TB NVMe <i>Seq Read: 7,000 MB/s, Seq Write: 5,000 MB/s</i>	\$329.00
--	----------

No Additional Storage

1 Year Limited Parts and Labor Warranty

Normal Assembly Service

16 GB RTX 3080 W/ 6144 CUDA Cores	\$649.00
-----------------------------------	----------

17.3" Matte 144Hz Full HD 1080p	\$79.00
---------------------------------	---------

United States QWERTY Keyboard

WiFi + Bluetooth

Specs of the System 76 Oryx Pro laptop this protocol uses for ONT sequencing.

Minimum IT requirements for MinION from ONT: [minion-it-reqs.pdf](#)

- 
- 2 The remainder of this protocol assumes you have completed all of the preliminary setup steps that are common with any new CPU.

Install CUDA toolkit - <https://developer.nvidia.com/cuda-toolkit> :

```
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204
ubuntu2204.pin
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-p
wget
https://developer.download.nvidia.com/compute/cuda/11.7.0/local_insta
repo-ubuntu2204-11-7-local_11.7.0-515.43.04-1_amd64.deb
sudo dpkg -i cuda-repo-ubuntu2204-11-7-local_11.7.0-515.43.04-1_amd
sudo cp /var/cuda-repo-ubuntu2204-11-7-local/cuda-*-keyring.gpg /usr/
sudo apt-get update
sudo apt-get -y install cuda
```

### 3 Install Boost

```
sudo apt install libboost-all-dev
```

- 4 The process at this link was instrumental to this protocol. It is recreated and simplified here. [ORIGINAL PROTOCOL](#). It was written for Pop!\_OS 21.04. The following protocols also work with Pop!\_OS 22.04. I would follow the steps at the link rather than here so you get a broader context of the actions you are performing on your system.

Add ONT Focal Repository

```
# update packages list
sudo apt-get update
# check for and install wget if needed
if [ $(dpkg-query -W -f='${Status}' wget 2>/dev/null | grep -c "ok
installed") -eq 0 ];
then
    sudo apt --yes install wget;
fi
# add the key
wget -O- https://mirror.oxfordnanoportal.com/apt/ont-repo.pub | sudo
apt-key add -
# add the focal repo
echo "deb http://mirror.oxfordnanoportal.com/apt focal-stable non-
free" | sudo tee /etc/apt/sources.list.d/nanoporetech.sources.list
```

Pop!\_OS 22.04

sudo apt update

**sudo apt update**

Pop!\_OS 22.04

Check for access to ONT files

**apt policy minknow-core-minion-nc**

Pop!\_OS 22.04



minknow-core-minion-nc:  
Installed: 4.3.4-focal

```
Candidate: 4.3.4-focal
Version table:
4.3.4-focal 100
10 http://mirror.oxfordnanoportal.com/apt focal-stable/non-free amd64
Packages
100 /var/lib/dpkg/status
```

## 5 Add the Focal repos:

Create a new file and edit in nano

```
sudo nano /etc/apt/sources.list.d/system-focal.sources
```

Copy and paste the following into your file:

```
X-Repolib-Name: Pop_OS System Sources
Enabled: yes
Types: deb deb-src
URIs: http://us.archive.ubuntu.com/ubuntu/
Suites: focal focal-security focal-updates focal-backports
Components: main restricted universe multiverse
X-Repolib-Default-Mirror: http://us.archive.ubuntu.com/ubuntu/
```

```
cat system-focal.sources
```

Check that the file exists and contains the right information.  
Pop!\_OS 22.04



```
X-Repolib-Name: Pop_OS System Sources
Enabled: yes
Types: deb deb-src
URIs: http://us.archive.ubuntu.com/ubuntu/
Suites: focal focal-security focal-updates focal-
backports
Components: main restricted universe multiverse
X-Repolib-Default-Mirror:
http://us.archive.ubuntu.com/ubuntu/
```

- 6 Pin the Focal repos. Start by creating another new file with nano:

```
sudo nano /etc/apt/preferences.d/focal-default-settings
```

```
Pop!_OS 22.04
```

Copy and paste the following into your file:

```
Package: *
Pin: release n=focal*
Pin-Priority: 10
```

Check that it was created correctly

```
cat focal-default-settings
```



```
Package: *  
Pin: release n=focal*  
Pin-Priority: 10
```

```
sudo apt update
```

## 7 Install MinKNOW and required packages

```
sudo apt install \  
  minknow-core-minion \  
  ont-kingfisher-ui-minion \  
  ont-bream4-minion \  
  ont-configuration-customer-minion \  
  ont-jwt-auth \  
  ont-vbz-hdf-plugin
```

## 8 Install ONT Guppy

```
sudo apt install ont-guppy
```

```
which guppy_basecaller
```

Check the paths once installed





```
/usr/bin/guppy_basecaller
```

**guppy\_basecaller --version**



```
: Guppy Basecalling Software, (C) Oxford Nanopore  
Technologies, Limited. Version 5.0.11+2b6dbff
```

## 9 Setup the MinKnow service

```
sudo /opt/ont/minknow/bin/config_editor --conf application \  
--filename /opt/ont/minknow/conf/app_conf \  
--set  
guppy.server_executable="/opt/ont/guppy/bin/guppy_basecall_server"  
\  
--set  
guppy.client_executable="/opt/ont/guppy/bin/guppy_basecall_client" \  
--set guppy.gpu_calling=1 \  
--set guppy.num_threads=16 \  
--set guppy.ipc_threads=2
```

```
systemctl restart minknow.service
```

```
systemctl status minknow.service
```

```
sudo nano /lib/systemd/system/guppyd.service
```

Copy the following to your new file:

```
[Unit]
Description=Service to manage the guppy basecall server.
Documentation=https://community.nanoporetech.com/protocols/Guppy-protocol/v/GPB_2003_v1_revQ_14Dec2018

[Service]
Type=simple
ExecStart=/opt/ont/guppy/bin/guppy_basecall_server --log_path
/var/log/guppy --config dna_r9.4.1_450bps_fast.cfg --port 5555 -x
cuda:all
Restart=always
User=root
MemoryLimit=8G
MemoryHigh=8G
CPUQuota=200%

[Install]
Alias=guppyd.service
WantedBy=multi-user.target
```

Check the file:

```
cat /lib/systemd/system/guppyd.service
```

```
systemctl enable guppyd.service
```

```
systemctl restart guppyd.service
```

MinKNOW GUI should now be available in your programs. Validate that it opens correctly.

- 10 Changes to MinKNOW file permissions at the bottom here:  
<https://gringer.gitlab.io/presentation-notes/2021/10/08/gpu-calling-in-minknow/>

"For my computer, there's an issue with MinKNOW not being able to access or create files. As a "nuclear" option, Miles Benton suggested changing the user and group for the minknow service to root"

```
sudo service minknow stop  
sudo perl -i -pe 's/(User|Group)=minknow/$1=root/'  
/lib/systemd/system/minknow.service  
sudo systemctl daemon-reload  
sudo service minknow start
```

- 11 Per this document: [https://denbi-nanopore-training-course.readthedocs.io/en/latest/read\\_qc/MinionQC.html](https://denbi-nanopore-training-course.readthedocs.io/en/latest/read_qc/MinionQC.html)  
Install R: <https://cran.r-project.org/>  
Install MinionQC: [https://github.com/roblanf/minion\\_qc](https://github.com/roblanf/minion_qc)

Install R:

```

# update indices
sudo apt update -qq
# install two helper packages we need
sudo apt install --no-install-recommends software-properties-common
dirmngr
# add the signing key (by Michael Rutter) for these repos
# To verify key, run gpg --show-keys
/etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# Fingerprint: E298A3A825C0D65DFD57CBB651716619E084DAB9
wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | sudo tee -a
/etc/apt/trusted.gpg.d/cran_ubuntu_key.asc
# add the R 4.0 repo from CRAN -- adjust 'focal' to 'groovy' or 'bionic'
as needed
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"

```

```

sudo apt install --no-install-recommends r-base

```

```

install.packages(c("data.table",
                  "futile.logger",
                  "ggplot2",
                  "optparse",
                  "plyr",
                  "readr",
                  "reshape2",
                  "scales",
                  "viridis",
                  "yaml"))

```

## 12 Install Bioconductor:

In an R command window:

```
if (!require("BiocManager", quietly = TRUE))  
  install.packages("BiocManager")  
BiocManager::install(version = "3.15")
```

## 13 Install Anaconda:

from: <https://www.digitalocean.com/community/tutorials/how-to-install-the-anaconda-python-distribution-on-ubuntu-22-04>

```
cd /tmp
```

```
curl https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh --output anaconda.sh
```

```
sha256sum anaconda.sh
```

You can now verify the data integrity of the installer with cryptographic hash verification through the SHA-256 checksum. You'll use the sha256sum command along with the filename of the script:



You'll receive output that looks similar to this:

```
fedf9e340039557f7b5e8a8a86affa9d299f5e9820144bd7b92ae9f7e  
e08ac60  anaconda.sh
```

```
bash anaconda.sh
```



Press ENTER/yes as needed

```
Welcome to Anaconda3 2021.11
```

```
In order to continue the installation process, please  
review the license  
agreement.
```

```
Please, press ENTER to continue
```

```
>>>
```

```
source ~/.bashrc
```

```
conda list
```



```
# packages in environment at /home/user/anaconda3:
#
# Name                                Version                                Build
Channel
_ipyw_jlab_nb_ext_conf                0.1.0                                py39h06a4308_0
_libgcc_mutex                         0.1                                  main
_openmp_mutex                         4.5                                  1_gnu
alabaster                             0.7.12                              pyhd3eb1b0_0
anaconda                             2022.05                              py39_0
```

```
conda search "^python$"
```

```
conda create --name my_env python=3
```

```
conda activate my_env
```

```
python --version
```

Verify Python is installed

```
conda install --name my_env35 numpy
```

## 14 Install NGSpeciesID: <https://github.com/ksahlin/NGSpeciesID>

```
conda create -n NGSpeciesID python=3.6 pip  
conda activate NGSpeciesID
```

```
conda install --yes -c conda-forge -c bioconda medaka==0.11.5  
openblas==0.3.3 spoa racon minimap2  
pip install NGSpeciesID
```

```
conda activate NGSpeciesID
```

```
mkdir test_ngspeciesID  
cd test_ngspeciesID
```

Test the install

```
curl -LO  
https://raw.githubusercontent.com/ksahlin/NGSpeciesID/master/test/sample\_h1.fastq
```

Download the test fastq file called "sample\_h1.fastq" (filesize 390kb)



**NGSpeciesID --ont --fastq sample\_h1.fastq --outfolder ./sample\_h1 --consensus --medaka**

Run the NGSspecies command on test file. Outputs will be saved in "/test\_ngspeciesID/sample\_h1/", where the final polished consensus file ("consensus.fasta") is located in the "/test\_ngspeciesID/sample\_h1/medaka\_cl\_id\_" directory.

15 You should now be ready to begin sequencing runs.