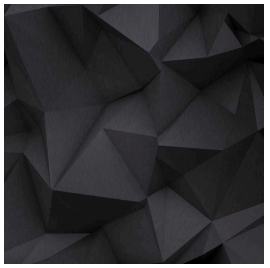
Mar 14, 2023 Version 2

ONT Sequencing IT/Compute Pop!_OS 22.04 Setup V.2



In 1 collection

DOI
https://dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v2

Stephen Douglas Russell[1]

[1]The Hoosier Mushroom Society

Mycota Lab

Stephen D Douglas Russell
Mycota Lab, Biodiverse, MycoMap

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v2

Protocol Citation: Stephen Douglas Russell 2023. ONT Sequencing IT/Compute Pop!_OS 22.04 Setup. **protocols.io** https://dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v2Version created by **Stephen D Douglas Russell**

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

Created: March 14, 2023

Last Modified: March 14, 2023

Protocol Integer ID: 78717

Keywords: Oxford Nanopore Technologies, DNA sequencing, DNA barcoding, MinION, Flongle, system76, linux, POP! OS, processing minion data, minion data, lab workflow, minion device, gpu during the analytical workflow, important component for fast processing, best vendor for laptop, fast processing, sequencing it, minion, analytical workflow, gpu, setup, careful attention to the gpu, linux system, it requirement

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.

Preparing a new CPU for MinION Sequencing

1 The final setup I went with can be found below. It was expensive (around 4000 for the laptop in early 2022), 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 *Seq Read: 7,000 MB/s, Seq Write: 5,000 MB/s*	$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 :

Command

new command name

```
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x8
6_64/cuda-ubuntu2204.pin
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-
pin-600
wget
https://developer.download.nvidia.com/compute/cuda/11.7.0/local_instal
lers/cuda-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_amd64.deb
sudo cp /var/cuda-repo-ubuntu2204-11-7-local/cuda-*-keyring.gpg
/usr/share/keyrings/
sudo apt-get update
sudo apt-get -y install cuda
```

3 Install Boost

Command

new command name

```
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.

Command

Add ONT Focal Repository (Pop!_OS 22.04)

```
# 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
```

Command

sudo apt update (Pop!_OS 22.04)

```
sudo apt update
```

Command

Check for access to ONT files (Pop!_OS 22.04)

```
apt policy minknow-core-minion-nc
```

Expected result

```
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:

Command

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/
```

Command

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

new command name

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

Expected result

```
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:

Command

new command name

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

Copy and paste the following into your file:

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

Check that is was created correctly

Command

new command name

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

Expected result

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

Command

new command name

```
sudo apt update
```

7 Install MinKNOW and required packages

Command

new command name

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

8 Install ONT Guppy

Command

new command name

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

Command

Check the paths once installed

new command name

```
which guppy_basecaller
```

Expected result

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

Command

new command name

```
guppy_basecaller --version
```

Expected result

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

9 Setup the MinKnow service

Command

new command name

```
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
```

Command

new command name

```
systemctl restart minknow.service
```

Command

new command name

```
systemctl status minknow.service
```

Command

new command name

```
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:

Command

new command name

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

Command

new command name

```
systemctl enable guppyd.service
```

Command

new command name

```
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"

Command

new command name

```
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
Install R: **https://cran.r-project.org/**
Install MinionQC: https://github.com/roblanf/minion_qc

Install R:

Command

new command name

```
# 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/"
```

Command

new command name

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

Command

new command name

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

12 Install Bioconductor:

In an R command window:

Command

new command name

```
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

Command

new command name

```
cd /tmp
```

Command

new command name

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

Command

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:

new command name

```
sha256sum anaconda.sh
```

Expected result

You'll receive output that looks similar to this:

```
fedf9e340039557f7b5e8a8a86affa9d299f5e9820144bd7b92ae9f7ee08ac6
0  anaconda.sh
```

Command

new command name

```
bash anaconda.sh
```

Expected result

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
>>>
```

Command

new command name

```
source ~/.bashrc
```

Command

new command name

```
conda list
```

Expected result

```
# 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
```

Command

new command name

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

Command

new command name

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

Command

new command name

```
conda activate my_env
```

Command

Verify Python is installed

new command name

```
python --version
```

Command

new command name

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

14 Install NGSpeciesID: https://github.com/ksahlin/NGSpeciesID

Command

new command name

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

Command

new command name

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

Command

new command name

```
conda activate NGSpeciesID
```

Command

Test the install

new command name

```
mkdir test_ngspeciesID
cd test_ngspeciesID
```

Command

Download the test fastq file called "sample_h1.fastq" (filesize 390kb)

new command name

```
curl -LO
https://raw.githubusercontent.com/ksahlin/NGSpeciesID/master/test/samp
le_h1.fastq
```

Command

Run the NGSpecies 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.

new command name

```
NGSpeciesID --ont --fastq sample_h1.fastq --outfolder ./sample_h1 --
consensus --medaka
```

15 You should now be ready to begin sequencing runs.

16

Updating MinKnow

17 Since this protocol was first written, MinKnow needed an update for the 10.4.1 Flongle cells. The steps to perform the update can be found below:

```
sudo apt install minknow-core-minion-nc=5.3.1-focal
sudo apt install ont-bream4-minion=7.3.5-1~focal
sudo apt install ont-configuration-customer-minion=5.3.8-1~focal
sudo apt install ont-kingfisher-ui-minion=5.3.6-1~focal

sudo apt-get update
sudo apt-get install minion-nc

sudo chmod -R a+rw /var/lib/minknow/

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


sudo apt install cuda
conda install -c bioconda seqkit

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