Apr 30, 2024

FUNDIS ONT Sequencing IT/Compute Pop!_OS 22.04 Setup

Forked from ONT Sequencing IT/Compute Pop!_OS 22.04 Setup

This protocol is a draft, published without a DOI.

Stephen Douglas Russell[1,2], Harte Singer[3]

[1]Mycota Lab; [2]The Hoosier Mushroom Society; [3]FUNDIS

The Hoosier Mushroom S…

Harte Singer
FUNDIS

OPEN ACCESS

Protocol Citation: Stephen Douglas Russell, Harte Singer 2024. FUNDIS ONT Sequencing IT/Compute Pop!_OS 22.04 Setup. **protocols.io https://protocols.io/view/fundis-ont-sequencing-it-compute-pop-os-22-04-setu-dcwy2xfw**

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

Created: January 06, 2024

Last Modified: April 30, 2024

Protocol Integer ID: 99000

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

Abstract

These are the steps to setting up a Linux system running Ubuntu for ONT Nanopore sequencing, basecalling, and consensus-building of fungal amplicon sequences. Adapted from **dx.doi.org/10.17504/protocols.io.14egn7kzmv5d/v3**

Includes an R-based run QC script

Preparing a new CPU for MinION Sequencing

1 Ideally you will be running a linux machine with a Ubuntu distro with an NVIDIA graphics card with at least 8GB of ram. (I am running a 4070Ti) You should have at least 32GB of system ram and a new processor.

Minimum IT requirements for MinION from ONT: minion-it-reqs.pdf

2

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

Command

```
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

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

4 Install MinKNOW from https://community.nanoporetech.com/downloads?from=support

Specifically download the GPU version of the software with the following name. It is a .deb file and your computer will install it when you double-click it

MinKNOW Software for the MinION Mk1B and the PromethION 2 Solo (P2 Solo) Devices

5 Install Guppy from the "Archived Software" section at the bottom of this page https://community.nanoporetech.com/downloads?from=support

Guppy v6.5.7

Make sure you install the Ubuntu 20 GPU version, just double click the installer and it should work.

6 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

```
# 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

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

Open an R terminal window for the following command

Command

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

7 Install Bioconductor:

In an R command window:

Command

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

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

Command

```
cd /tmp
```

Command

```
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:

```
sha256sum anaconda.sh
```

Expected result

You'll receive output that looks similar to this:

```
fedf9e340039557f7b5e8a8a86affa9d299f5e9820144bd7b92ae9f7ee08ac6
0  anaconda.sh
```

Command

```
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

```
source ~/.bashrc
```

Command

```
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

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

Command

Verify Python is installed

```
python --version
```

Command

Install bioconda

```
conda install -c bioconda seqkit
```

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

Command

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

Command

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

Command

```
conda activate NGSpeciesID
```

Command

Test the install

```
mkdir test_ngspeciesID
cd test_ngspeciesID
```

Command

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

```
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.

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

10 You should now be ready to begin sequencing runs.

Installing Dorado

11 We will eventually be moving to Dorado basecalling.

Installing Dorado:

```
sudo -i

sudo apt-get update && apt-get install -y --no-install-recommends
\
        curl \
        git \
        ca-certificates \
        build-essential \
        nvidia-cuda-toolkit \
        libhdf5-dev \
        libssl-dev \
        libzstd-dev \
        cmake \
        autoconf \
        automake

git clone https://github.com/nanoporetech/dorado.git dorado
cd dorado
cmake -S . -B cmake-build
cmake --build cmake-build --config Release -j
ctest --test-dir cmake-build



pip install pre-commit
pre-commit install
```