Basestack

Release 2.0.0

Brian Merritt

Sep 06, 2023

CONTENTS

CHAPTER

ONE

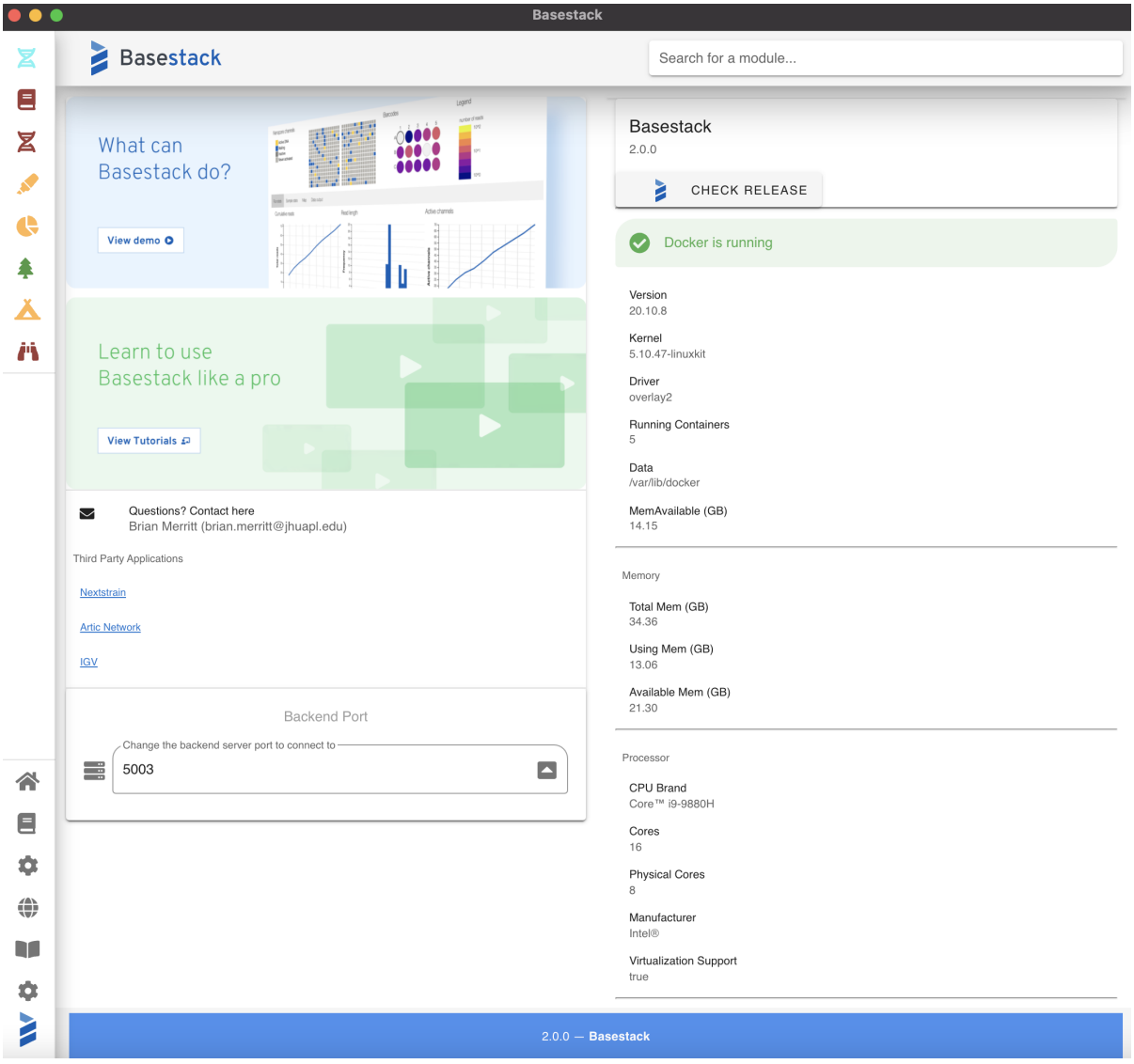
GETTING STARTED

CHAPTER

TWO

ABOUT

Get Basestack



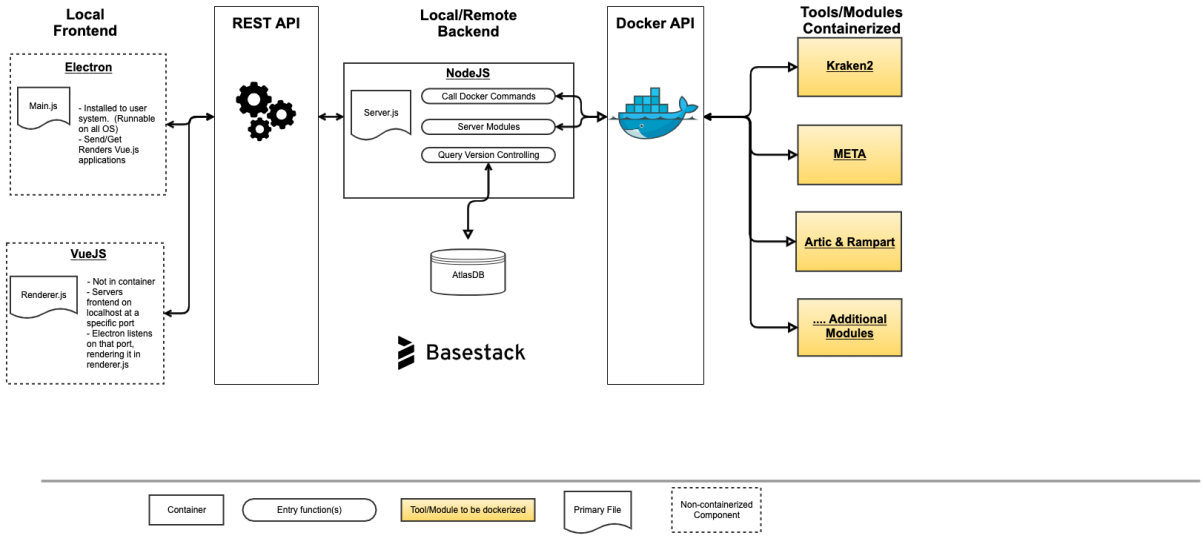
Basestack is a platform for rapid and real time analysis of Nanopore generated sequencing data.

Basestack comes bundled with a series of both cutting edge public tools, some modified by JHU built phylogenetic and genomic pipelines, all integrated into a user interface (that is quickly and seamlessly installable on all standard laptops on any of the major operating systems used today. Ultimately, the suite provides users with a

means of quickly and accurately generating consensus sequences from viral data within minutes of starting. In addition, we provide a system for automated updates once the application is installed for any future changes or additions to the application that is fundamentally built upon the Docker Container ecosystem.

Please follow the sidebar links to review both installation steps for your distribution and additional information on the underlying pipelines.

Basestack's recent releases can be found at: https://github.com/jhuapl-bio/Basestack/releases/



CHAPTER

THREE

CONTENTS

3.1 Installation

Get The Latest Basestack Release

3.1.1 Windows

Docker

Docker Download + Install

1. Head over to the Docker website to download the necessary package

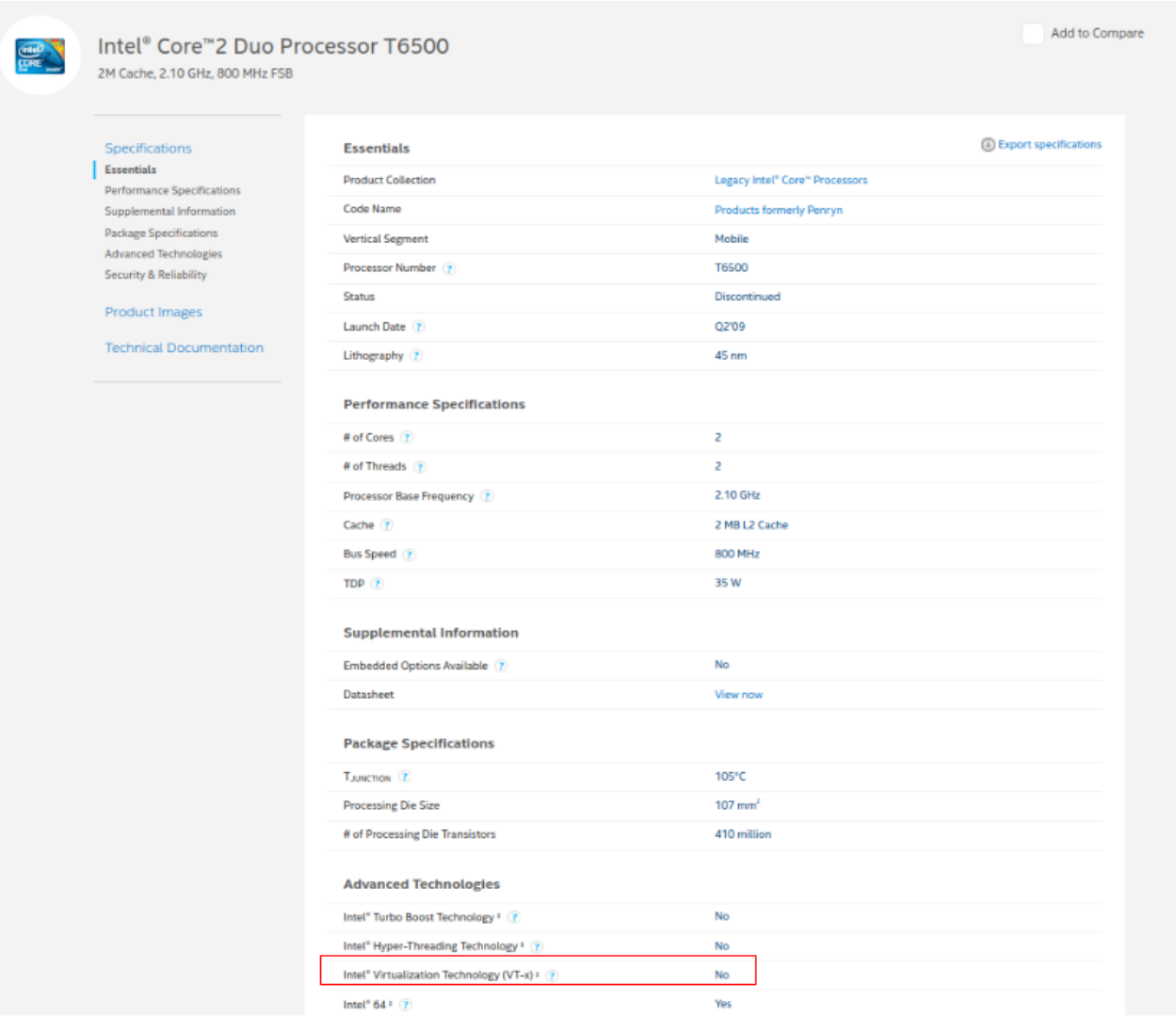
In order to run docker you must be able to support virtualization from your CPU. This feature must also be enabled within your BIOS and Windows Features.

See more of *Virtualization Disabled - Windows* for more details

If you're unsure whether docker is supported by your specific cpu, please visit and input your specific model number:

- Intel
- AMD

Type Your Model Number, e.g. T6500 into the product search bar



Intel® Core™2 Duo Processor T6500

2M Cache, 2.10 GHz, 800 MHz FSB

Add to Compare

Specifications
Essentials
Performance Specifications
Supplemental Information
Package Specifications
Advanced Technologies
Security & Reliability
Product Images
Technical Documentation

Export specifications

Essentials

Product Collection	Legacy Intel® Core™ Processors
Code Name	Products formerly Penryn
Vertical Segment	Mobile
Processor Number	T6500
Status	Discontinued
Launch Date	Q2'09
Lithography	45 nm

Performance Specifications

# of Cores	2
# of Threads	2
Processor Base Frequency	2.10 GHz
Cache	2 MB L2 Cache
Bus Speed	800 MHz
TDP	35 W

Supplemental Information

Embedded Options Available	No
Datasheet	View now

Package Specifications

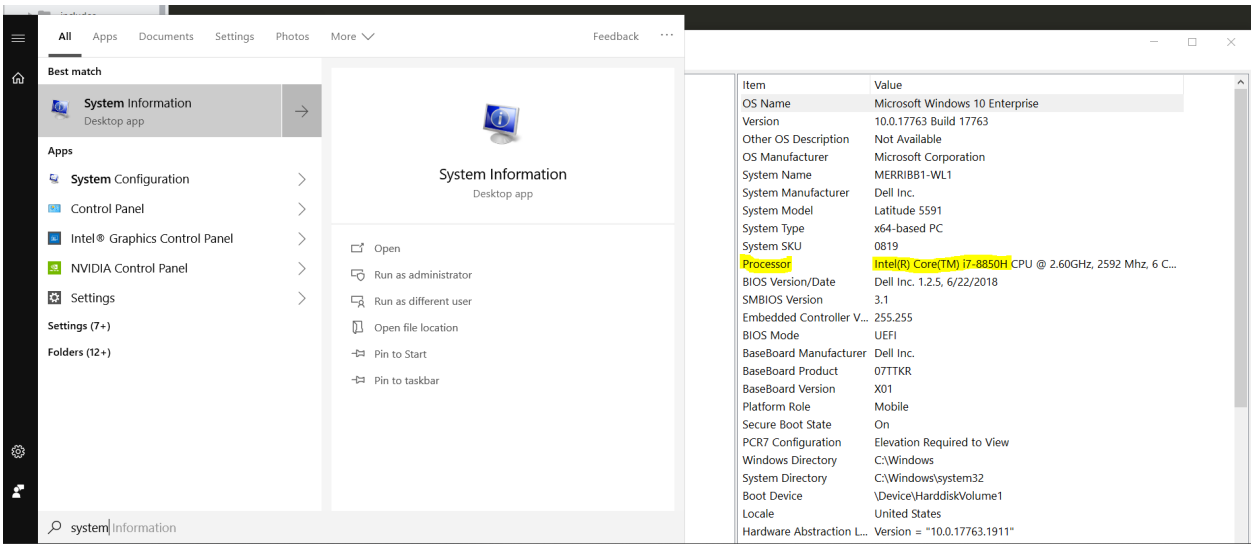
$T_{JUNCTION}$	105°C
Processing Die Size	107 mm^2
# of Processing Die Transistors	410 million

Advanced Technologies

Intel® Turbo Boost Technology ‡	No
Intel® Hyper-Threading Technology ‡	No
Intel® Virtualization Technology (VT-x) ‡	No
Intel® 64 ‡	Yes

In this example above, you can see that Vt-x (Virtualization) is not supported. This will be a **Yes** if it is supported.

To find the cpu model on Windows:



Install Docker Desktop on Windows

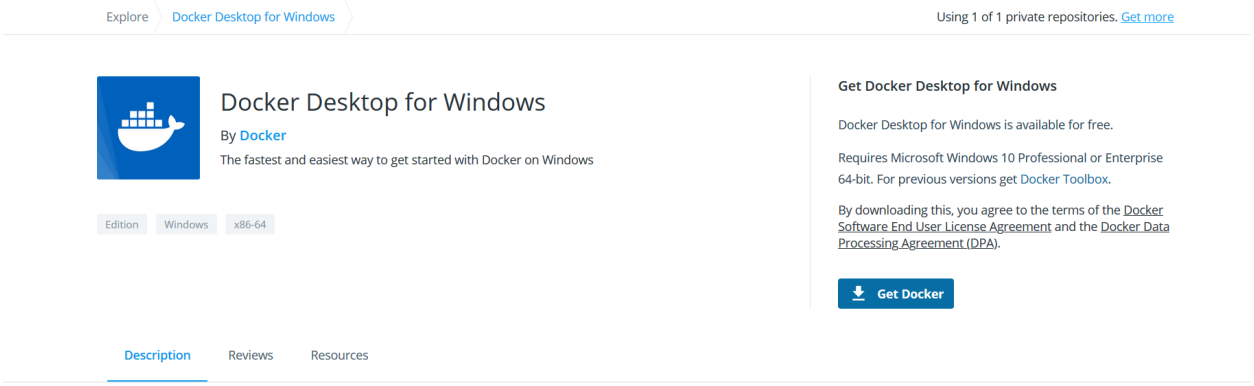
Estimated reading time: 6 minutes

Docker Desktop for Windows is the Community version of Docker for Microsoft Windows. You can download Docker Desktop for Windows from Docker Hub.

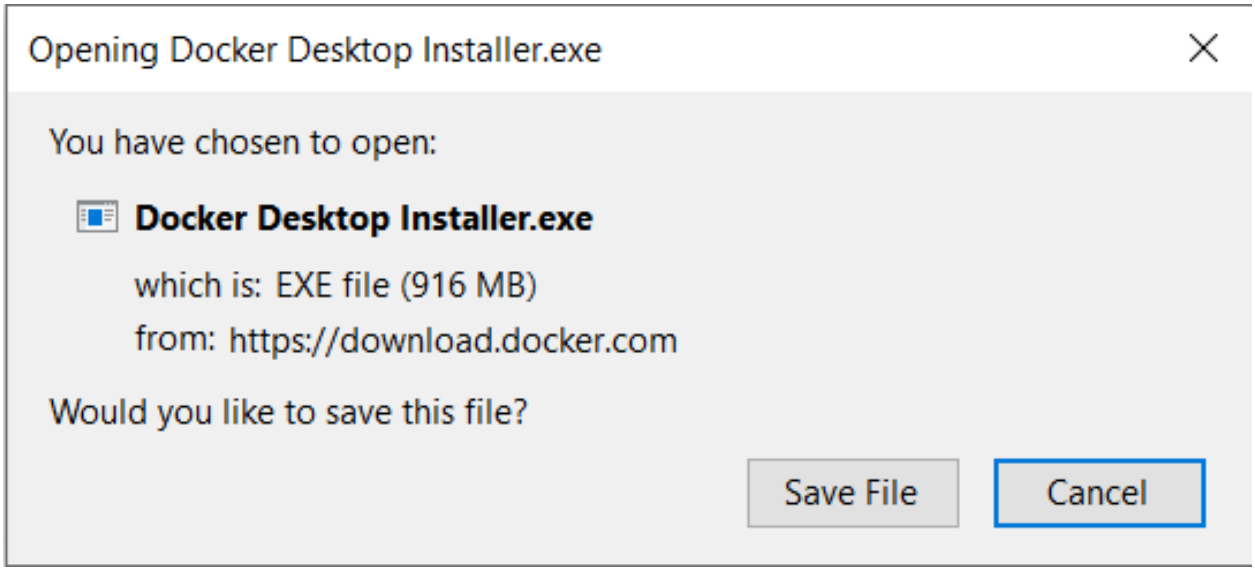
Download from Docker Hub

By downloading Docker Desktop, you agree to the terms of the Docker Software End User License Agreement and the Docker Data Processing Agreement.

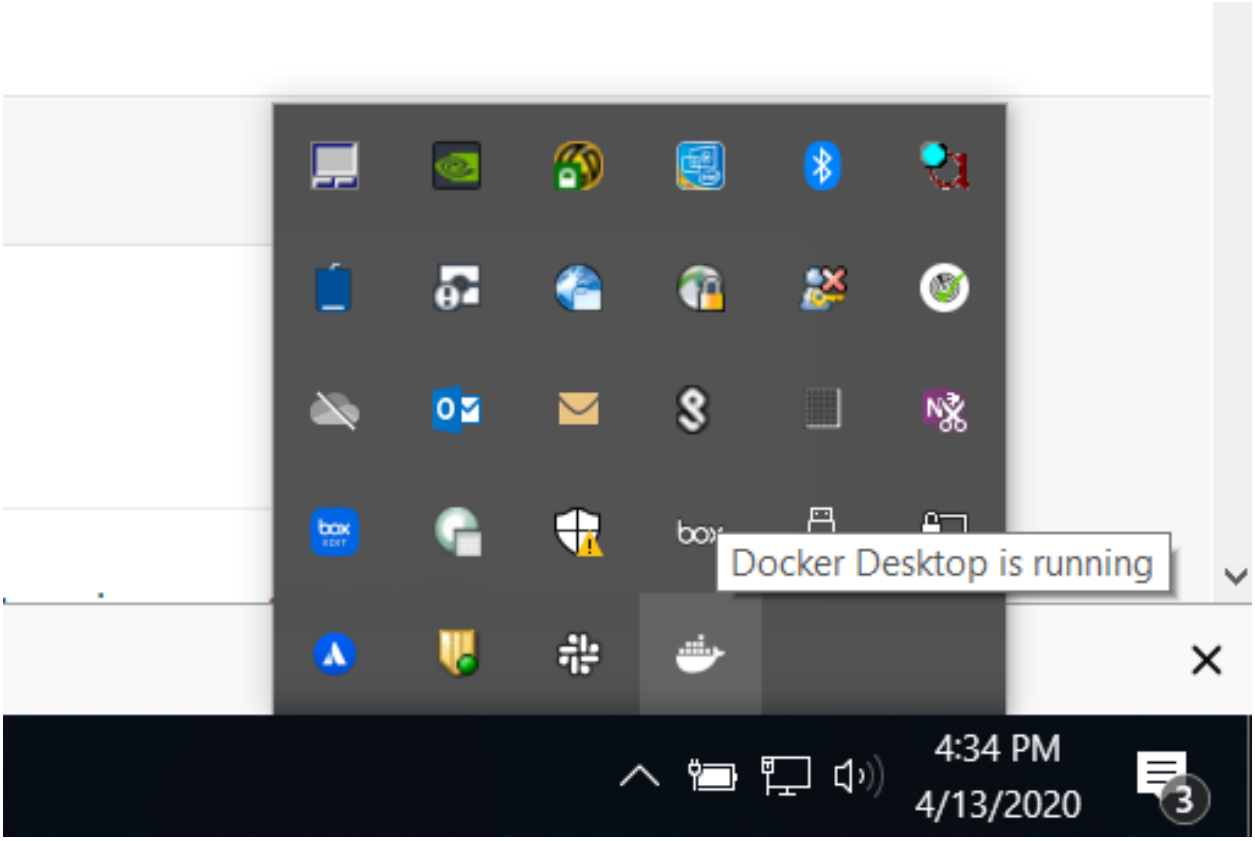
2. Choose **Get Docker**



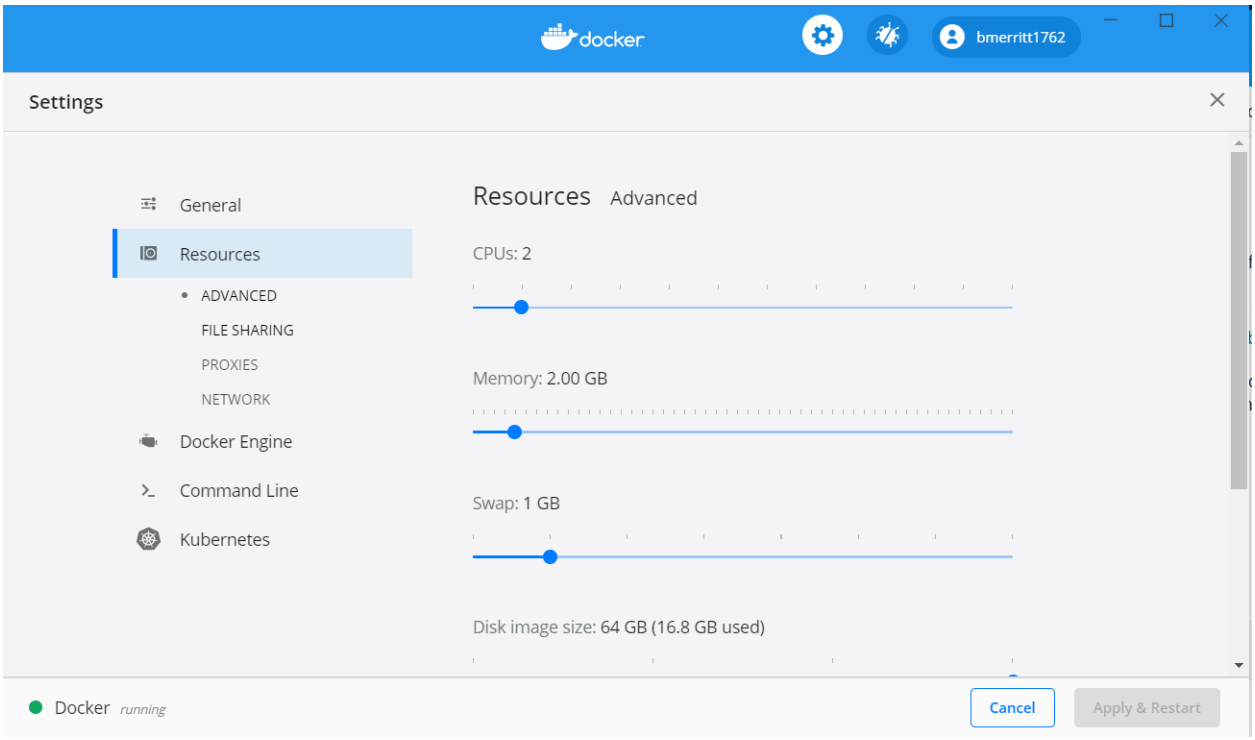
3. Choose **Save File** from the prompt



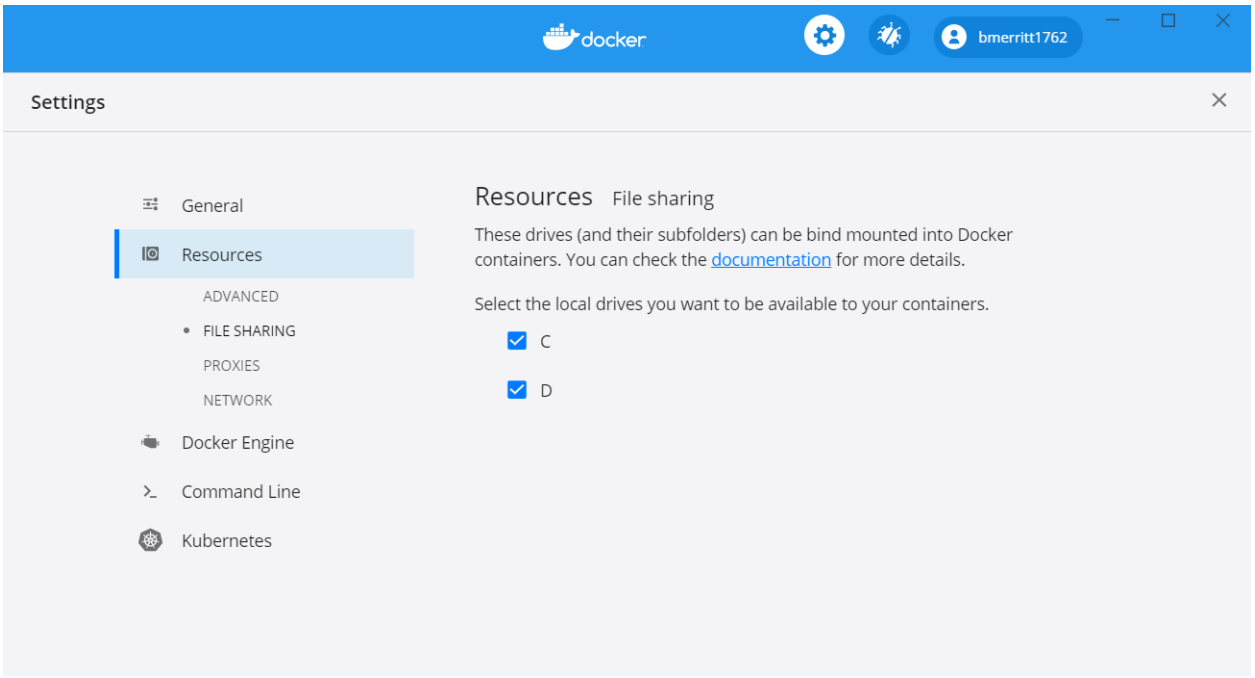
4. Once you've installed docker for Windows, you can start it at the **Quick Launch** by search **Docker**. You can also view it on your right-hand-bottom tray by right-clicking



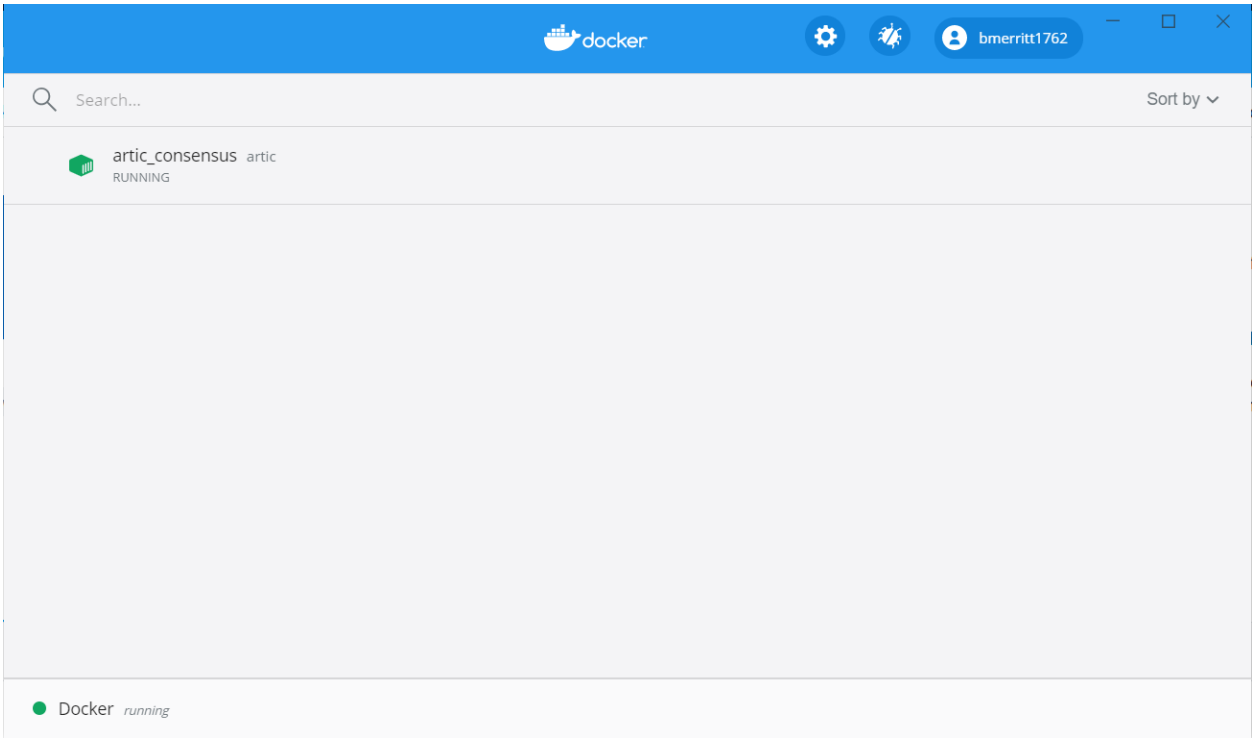
5. Here Docker provides a GUI environment to manage your system. You can allocate or limit resources to your containers as well as set networking settings if you'd like. **We use default values for our app**



6. **OPTIONAL** Choose Local drives to share with containers. Useful if you're storing data on an external drive.

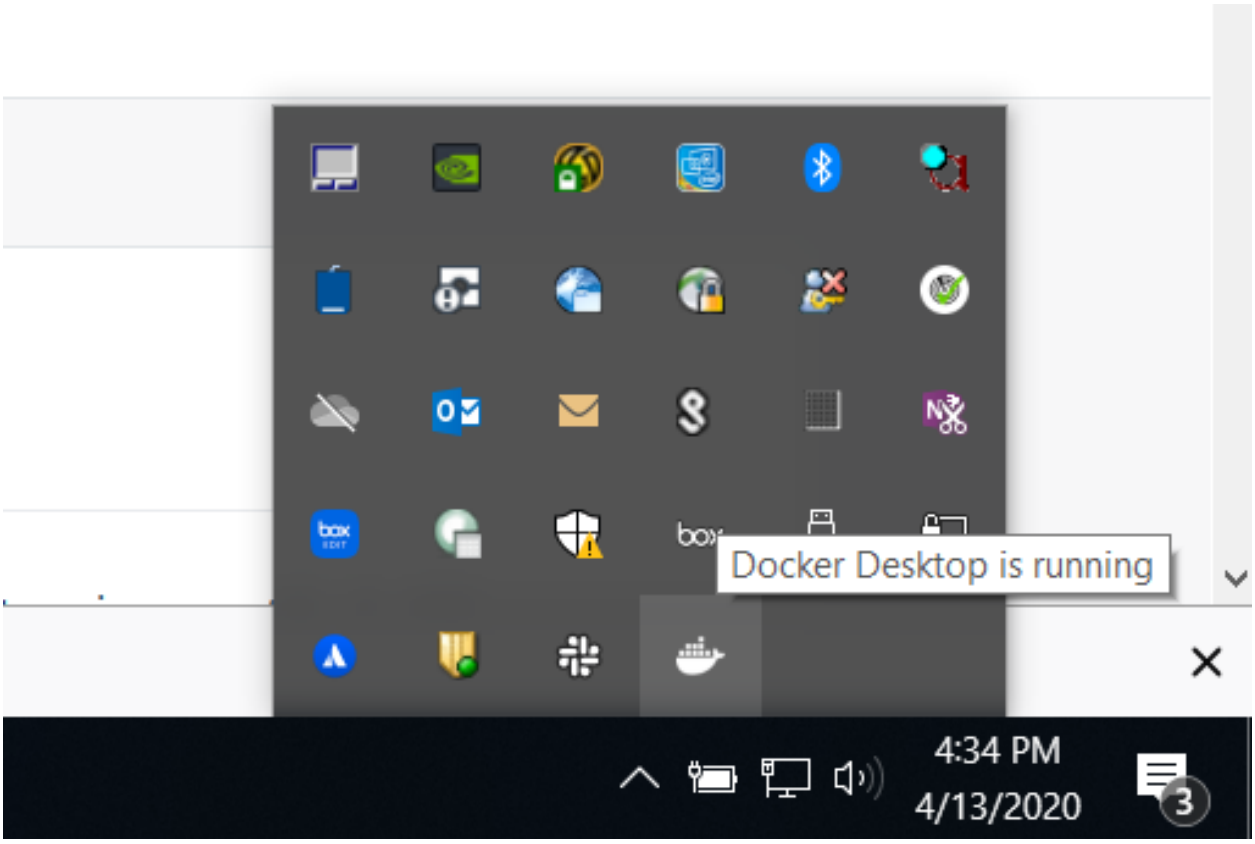


7. Main image that allows you to manage specific containers

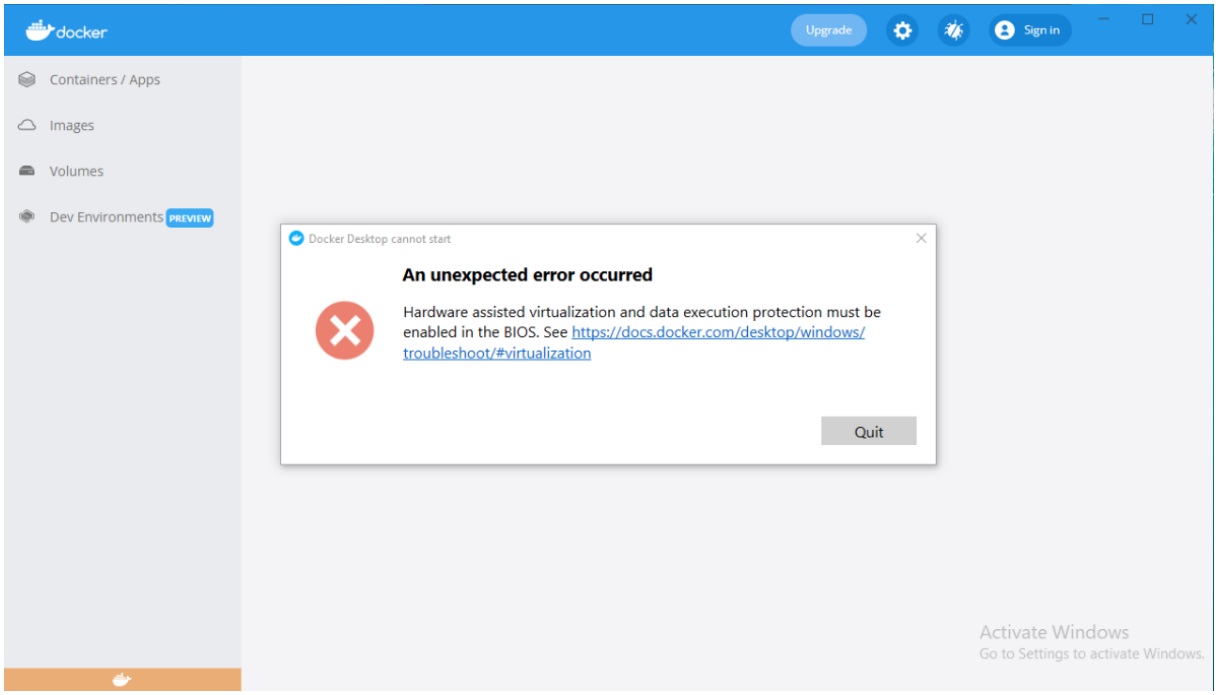


Confirm Docker is Running

In your taskbar (lower-right), if you hover over the icon you should see the message displayed below. Right-clicking will give additional options



Warning: You might experience an error about BIOS not having virtiualization enabled



Follow these steps from https://bce.berkeley.edu/enabling-virtualization-in-your-pc-bios.html

See here *Virtualization Disabled - Windows* for more information

Reboot your computer

- Right when the computer is coming up from the black screen, press Delete, Esc, F1, F2, or F4. Each computer manufacturer uses a different key but it may show a brief message at boot telling you which one to press. If you miss it the first time, reboot and try again. It helps to tap the key about twice a second when the computer is coming up. If you are not able to enter the BIOS via this method, consult your computer's manual.
- In the BIOS settings, find the configuration items related to the CPU. These can be in under the headings Processor, Chipset, or Northbridge.
- Enable virtualization; the setting may be called VT-x, AMD-V, SVM, or Vanderpool. Enable Intel VT-d or AMD IOMMU if the options are available.
- Save your changes and reboot.
- Delete any existing VMs (Machine > Remove ** and select ** Delete all files) and re-import the .ova file (following step 4 and subsequent steps of the installation instructions).
- Check if your system supports Virtualization
- If you are unable to find the Virtualization settings in your BIOS it may mean that your laptop does not support it. If you want to try to find this out yourself, then you can try:

On Windows, download and run a Microsoft utility. You can also download utilities to check if your CPU is capable of virtualization, if not enabled. Hyper-V must be disabled in order for VirtualBox to run 64-bit guest operating systems. Visit the "turn Windows feature on or off" application and make sure Hyper-V is not checked.

On Linux, open a terminal window and run:

```
egrep -q 'vmx|svm' /proc/cpuinfo && echo yes || echo no
```

Basestack

Install Main

Download Basestack from Releases

- You will select the item labeled `<Basestack-Version>.Setup.exe`

1. Double-click `Basestack-Version.Setup.exe`
2. Follow the prompts for installing the software. Choose defaults unless otherwise needed.

3.1.2 MacOS

Docker

Docker for Mac

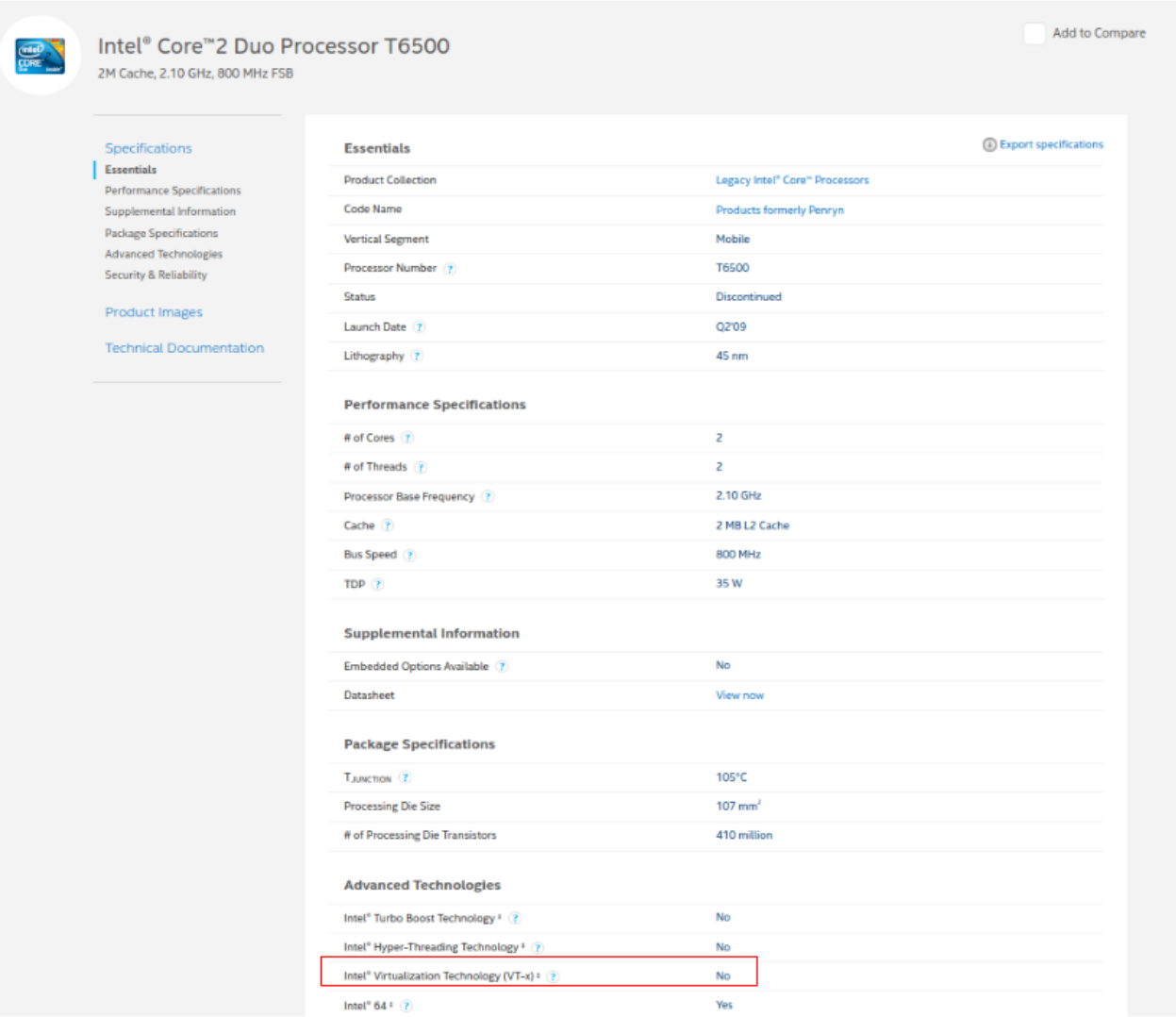
In order to run docker you must be able to support virtualization from your CPU. This feature must also be enabled within your BIOS and Windows Features.

If you're unsure whether docker is supported by your specific cpu, please visit and input your specific model number:

- Intel

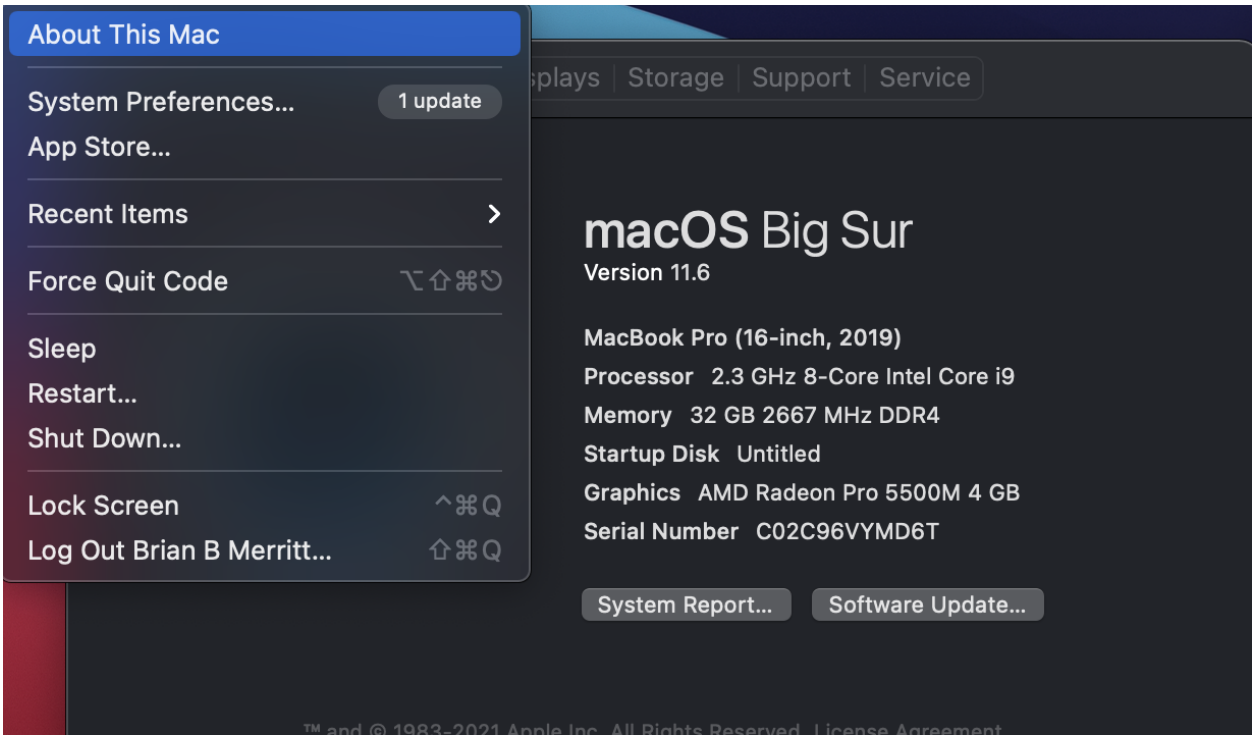
- AMD

Type Your Model Number, e.g. T6500 into the product search bar



In this example above, you can see that Vt-x (Virtualization) is not supported. This will be a **Yes** if it is supported.

On Mac you can find this value by



Instructions

https://docs.docker.com/docker-for-mac/install/

Basestack

Download Basestack from Releases

- You will select the item labeled `<Basestack-Version>.dmg`

1. Double-click `<Basestack-Version>.dmg`
2. Follow the prompts for installing the software. Choose defaults unless otherwise needed.

3.1.3 Linux

Docker

Instructions

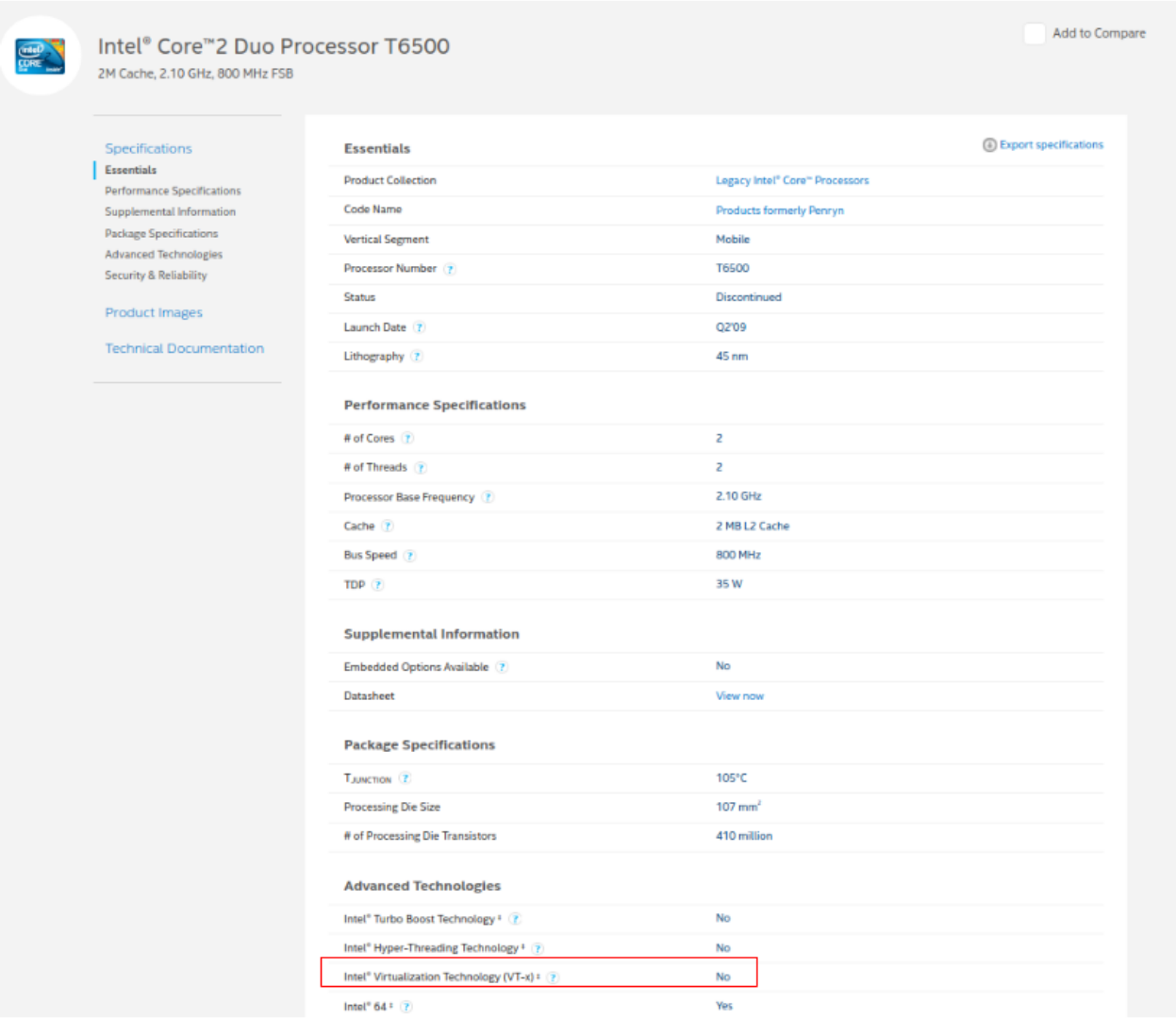
Install Docker

In order to run docker you must be able to support virtualization from your CPU. This feature must also be enabled within your BIOS and Windows Features.

If you're unsure whether docker is supported by your specific cpu, please visit and input your specific model number:

- Intel
- AMD

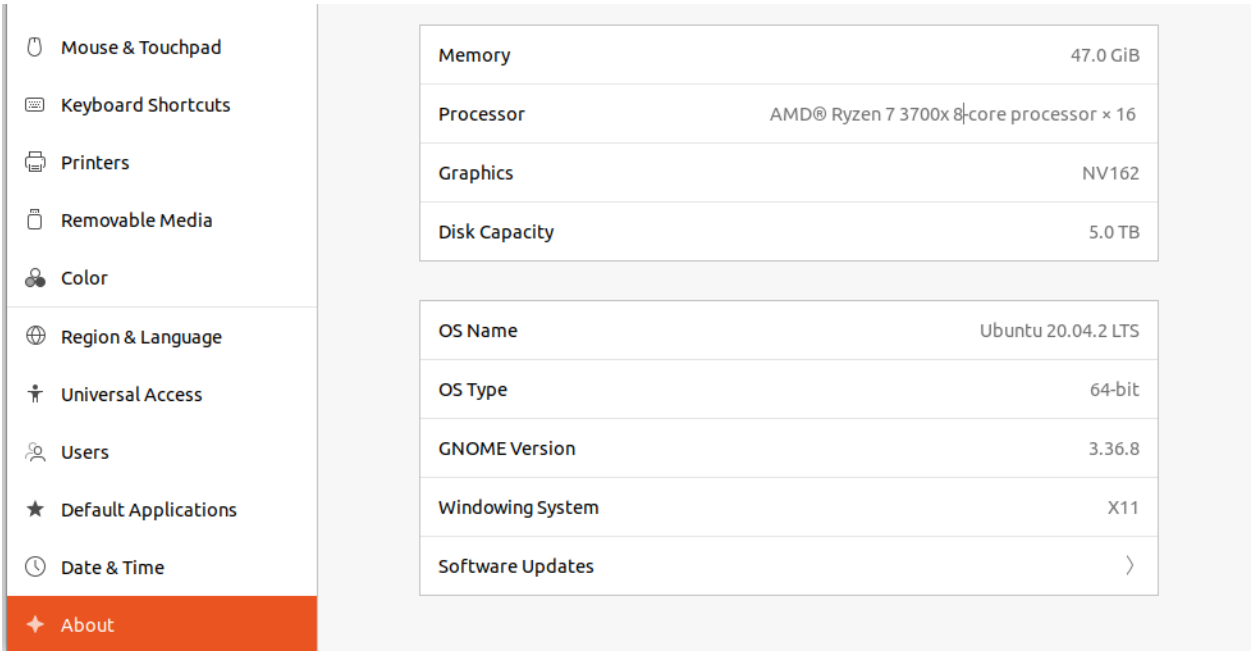
Type Your Model Number, e.g. T6500 into the product search bar


Intel® Core™2 Duo Processor T6500
2M Cache, 2.10 GHz, 800 MHz FSB
Add to Compare
Specifications
Essentials
Performance Specifications
Supplemental Information
Package Specifications
Advanced Technologies
Security & Reliability
Product Images
Technical Documentation
Export specifications
Essentials
Product Collection
Legacy Intel® Core™ Processors
Code Name
Products formerly Penryn
Vertical Segment
Mobile
Processor Number
T6500
Status
Discontinued
Launch Date
Q2'09
Lithography
45 nm
Performance Specifications
# of Cores
2
# of Threads
2
Processor Base Frequency
2.10 GHz
Cache
2 MB L2 Cache
Bus Speed
800 MHz
TDP
35 W
Supplemental Information
Embedded Options Available
No
Datasheet
View now
Package Specifications
TJUNCTION
105°C
Processing Die Size
107 mm²
# of Processing Die Transistors
410 million
Advanced Technologies
Intel® Turbo Boost Technology ‡
No
Intel® Hyper-Threading Technology ‡
No
Intel® Virtualization Technology (VT-x) ‡
No
Intel® 64 ‡
Yes

width
100%

In this example above, you can see that Vt-x (Virtualization) is not supported. This will be a **Yes** if it is supported.

On Linux you can find the processor by typing on the command line:



Make sure that you select the appropriate distribution for your machine. If you are unsure of your distribution use *lsb_release -a* from the command line to check your distro.

Note: You can retrieve a basic install script for a fresh environment that can be found here It is primarily intended for newly flashed Ubuntu systems. This script will work for both ARM64 and AMD64 processors

It will set up Docker, Download Basestack, CUDA, and MinKNOW

```
> lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.2 LTS
Release:        20.04
Codename:       focal
```

Docker Installation

Choose **ONE** option

- **A. Rootless - RECOMMENDED**
  - **https://docs.docker.com/engine/security/rootless/**
    - If you already have *docker* installed, see documentation on [*docker context*](https://docs.docker.com/engine/security/rootless/#client) to switch between rootless and rootful
- **B. Rootful (gives root access, use if you already have docker installed or use it regularly)**
  - **https://docs.docker.com/engine/install/ubuntu/**
    - Required to map you user permissions appropriately for generated files.
    - Recommended for most rootful-specific personal systems running Docker
  - **Post-Installation Steps:**
    1. **Create Docker group**
       - `sudo groupadd docker`
    2. **Add your user to the docker group**
       - `sudo usermod -aG docker $USER`
    3. **Ensure all root-created files map as your user id in docker containers and volumes (Do both of these)**
       - **1.** `sudo sed -i "1s/^/$USER:$(id -u):1\n/" /etc/subuid`
       - **2.** `sudo sed -i "1s/^/$USER:$(id -g):1\n/" /etc/subgid`
    4. **Create Docker container namespace CHOOSE ONE**
       - **a.** ***echo $(jq –arg user "$USER" '. += {"userns-remap": $user}' /etc/docker/daemon.json) > ~/daemon.json && sudo mv ~/daemon.json /etc/docker/daemon.json***
         - If you dont have the file already created (isn't created by default)
       - **b. Manually add your user by following the instructions here: https://docs.docker.com/engine/security/userns-remap/.**
         - You can disable the *userns-remap* functionality by deleting the *daemon.json* file described above or removing the line attributed to your user
    5. **Check that the subgid and subuid files are correct. Order of these lines matters in that the *<username>:<uid>:1* must come first in each file**
       - **1.** `cat /etc/subuid`
         `-<username>:<uid>:1 -<username>:100000:65536`
       - **2.** ***cat /etc/subgid***
         `-<username>:<uid>:1 -<username>:100000:65536`
    6. **Restart Docker**
       - **a**. `sudo service docker restart`
       - **b**. OR Restart your computer/session
    7. **Ensure that permissions are appropriate**
       - **1**. `docker run -v /tmp:/opt/tmp nginx touch /opt/tmp/test.txt`

        * **2. `ls -lht /tmp/test.txt`**
            - ^ ensure that ownership is your uid/gid or username:group

Listing 1: Full block of code for Option B (Rootful)

```
sudo usermod -aG docker $USER
sudo sed -i "1s/^/$USER:$(id -u):1\n/" /etc/subuid
sudo sed -i "1s/^/$USER:$(id -u):1\n/" /etc/subgid

if [[ -s "/etc/docker/daemon.json" ]]; then
        cat "/etc/docker/daemon.json" | jq --arg USERNS $USER '."userns-remap" = $USERNS
↪' > /tmp/daemon.json
        sudo mv /tmp/daemon.json /etc/docker/daemon.json
else
        echo "{\"userns-remap\": \"$USER\"}"  | sudo tee -a /etc/docker/daemon.json
fi
```

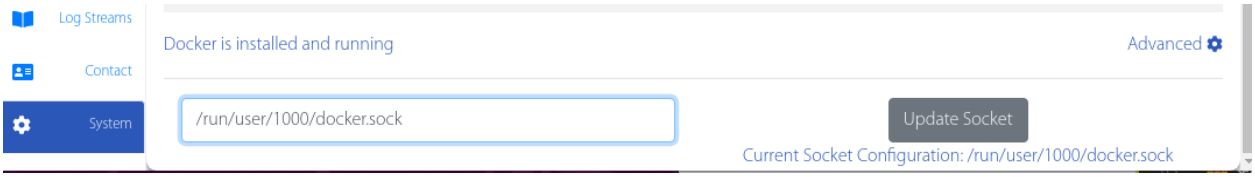
Open a terminal and type *docker info*. You should see information about your *docker* service

```
(base) brian-home@ubuntu:~$ docker info
Client:
 Debug Mode: false

Server:
 Containers: 14
  Running: 0
  Paused: 0
  Stopped: 14
 Images: 5
 Server Version: 19.03.8
 Storage Driver: overlay2
  Backing Filesystem: <unknown>
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc version: dc9208a3303feef5b3839f4323d9beb36df0a9dd
 init version: fec3683
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 4.15.0-118-generic
 Operating System: Ubuntu 18.04.1 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 5.086GiB
 Name: ubuntu
 ID: 3WKY:JXQC:F6AY:XAWE:QHWP:SPYH:JN4Y:VNER:XBIM:YELW:W6AJ:NGCD
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Username: bmerritt1762
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

WARNING: No swap limit support
(base) brian-home@ubuntu:~$
```

Rootful: - */var/lib/docker* is the Docker Root Dir. YOU MUST correctly utilize the *userns-remap* configuration described above for this to work

Rootless: - *$HOME/.local/share/docker* (or something similar in *$HOME*) will be the Docker Root Dir.

Additionally, for Docker Rootless only, you'll need to adjust the socket that Basestack is connecting to directly within the System tab of the application. This value will be wherever your *docker.sock* file is made.



If you're unsure where that is run: `docker context ls` and it will be the DOCKER ENDPOINT value sans the `unix://`

```
> docker context ls
NAME        DESCRIPTION                               DOCKER ENDPOINT                  KUBERNETES ENDPOINT   ORCHESTRATOR
default *   Current DOCKER_HOST based configuration   unix:///run/user/1000/docker.sock                       swarm
```

Basestack

Install Main

Download Basestack from Releases

- You will select the item labeled `<Basestack-Version>.<arch>.AppImage`

1. Double-click `<Basestack-Version>.<arch>.AppImage`
2. Follow the prompts for installing the software. Choose defaults unless otherwise needed.
3. chmod ugo+x `<Basestack-Version>.<arch>.AppImage`

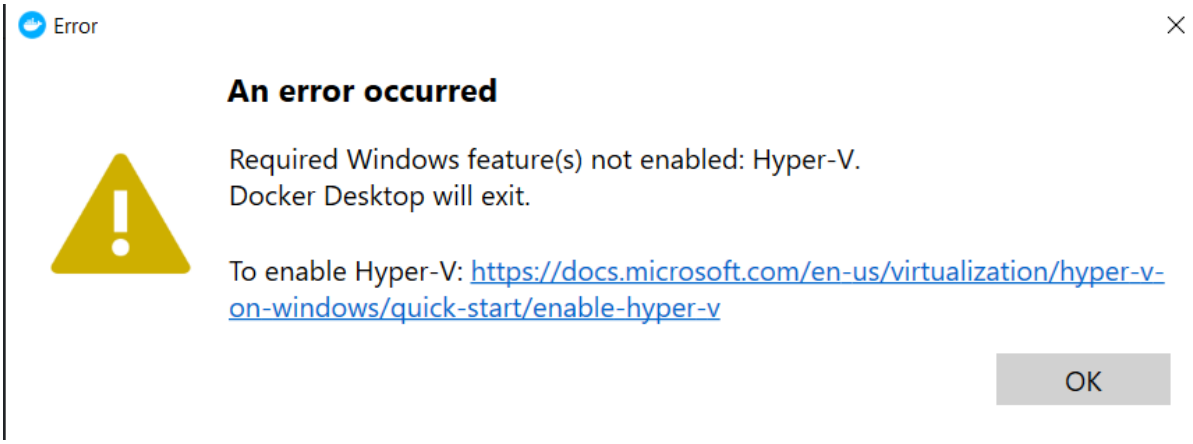
3.1.4 Troubleshooting

General

- Request or view feature changes at our [issue tracker](https://github.com/jhuapl-bio/Basestack/issues)
- **If you run into issues with the online install, you may want to download (or otherwise obtain) the offline install package**
  - Using the above download links, download the appropriate docker images you'd like e.g. `basestack_consensus.tar.gz` (~5.2GB)
  - Run `docker load < basestack_consensus.tar.gz` from the command line
- See below Appendices for more detailed installation instructions.

Windows

Hyper-V Not Enabled - Windows

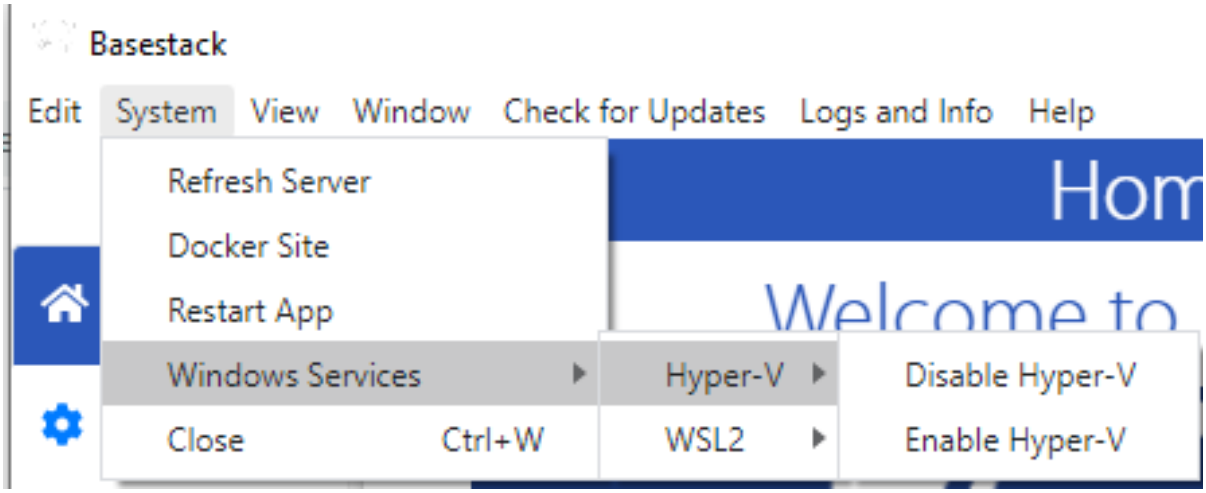


If you are on older Windows distributions, you may experience an error when attempting to start docker on how HyperV is not enabled.

Enable Hyper-V in Basestack

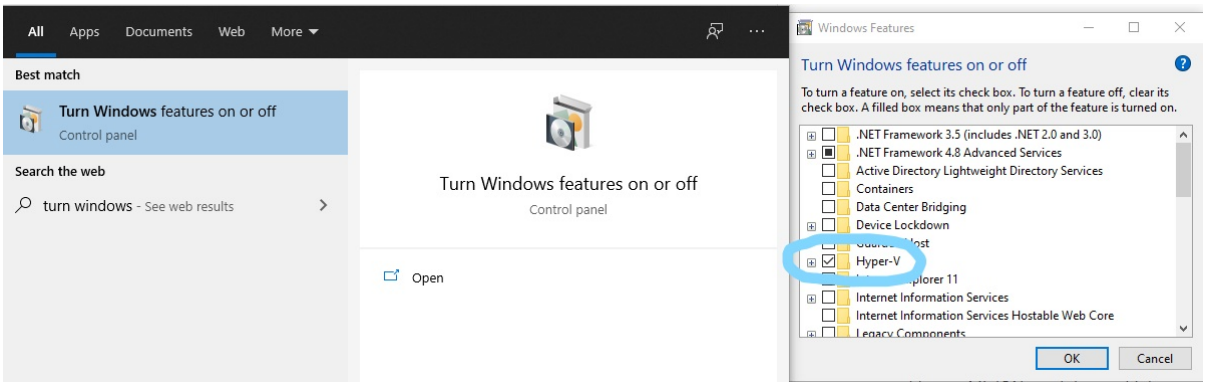
To enable it within Basestack select: `System -> Windows Services -> Hyper-V -> Enable Hyper-V`.

A window will appear prompting admin rights and then it will automatically being the enable process. See more below.



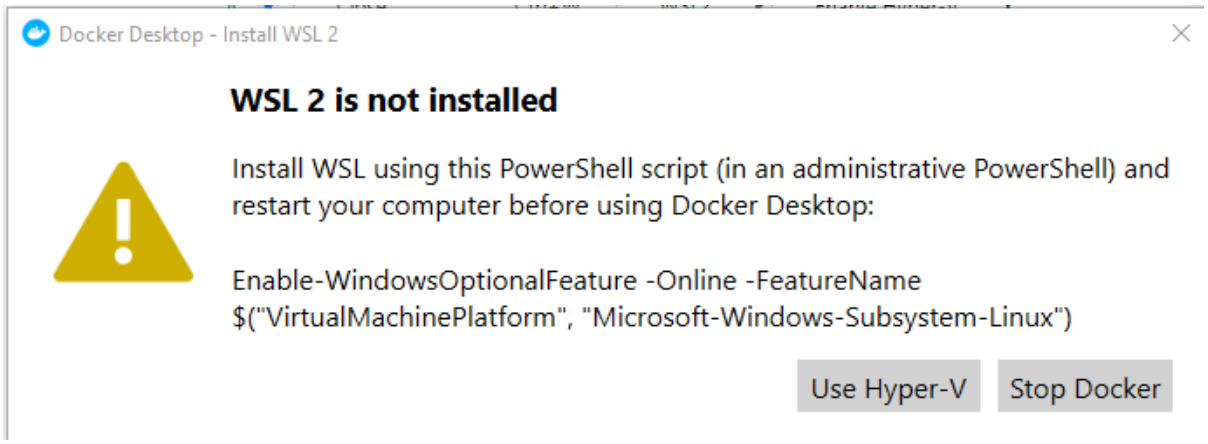
Enable Hyper-V in Windows System

Alternatively you can enable it within the Host system itself by searching for "Turns Windows features on or off" and selecting "Hyper-V". This will require a computer restart



WSL2 Not Installed - Windows

The error (seen below) is often shown for newer Windows OS types. If this occurs, you may have different variants. In the included example, I have the option to enable WSL or use Hyper-V.



Sometimes, another window will appear regarding installing WSL.

Install WSL2 from External Sources

Please follow that link

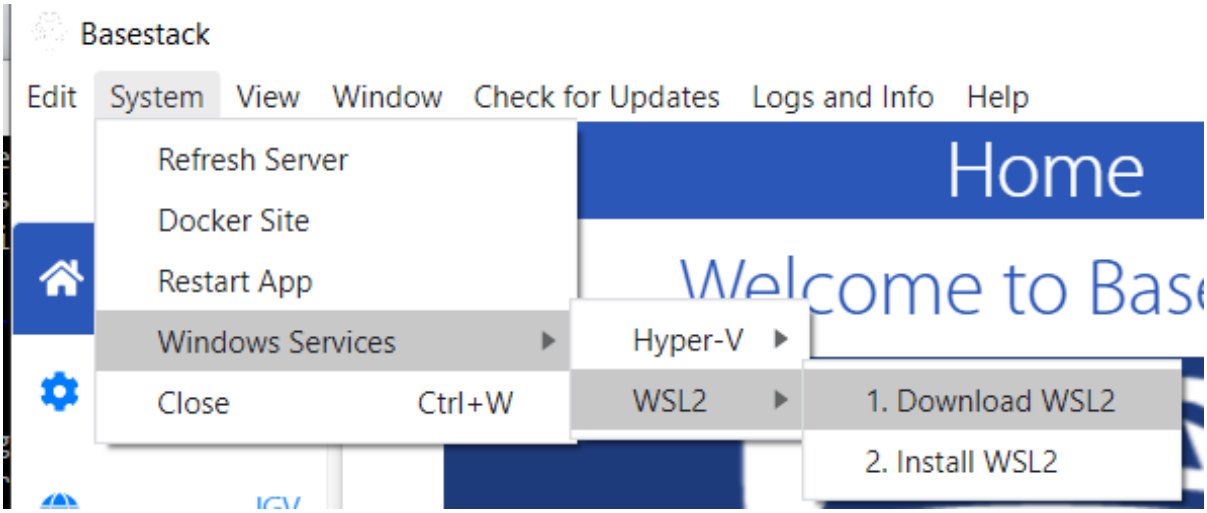
Make sure to perform **AT LEAST step 4**. Once WSL2 is installed/enabled, please restart Docker Desktop

Install WSL2 in Basestack

Alternatively Basestack allows users to download WSL directly.

To Download then Install it within Basestack do:

1. `System -> Windows Services -> WSL2 -> Download WSL2`
2. `System -> Windows Services -> WSL2 -> Install WSL2`

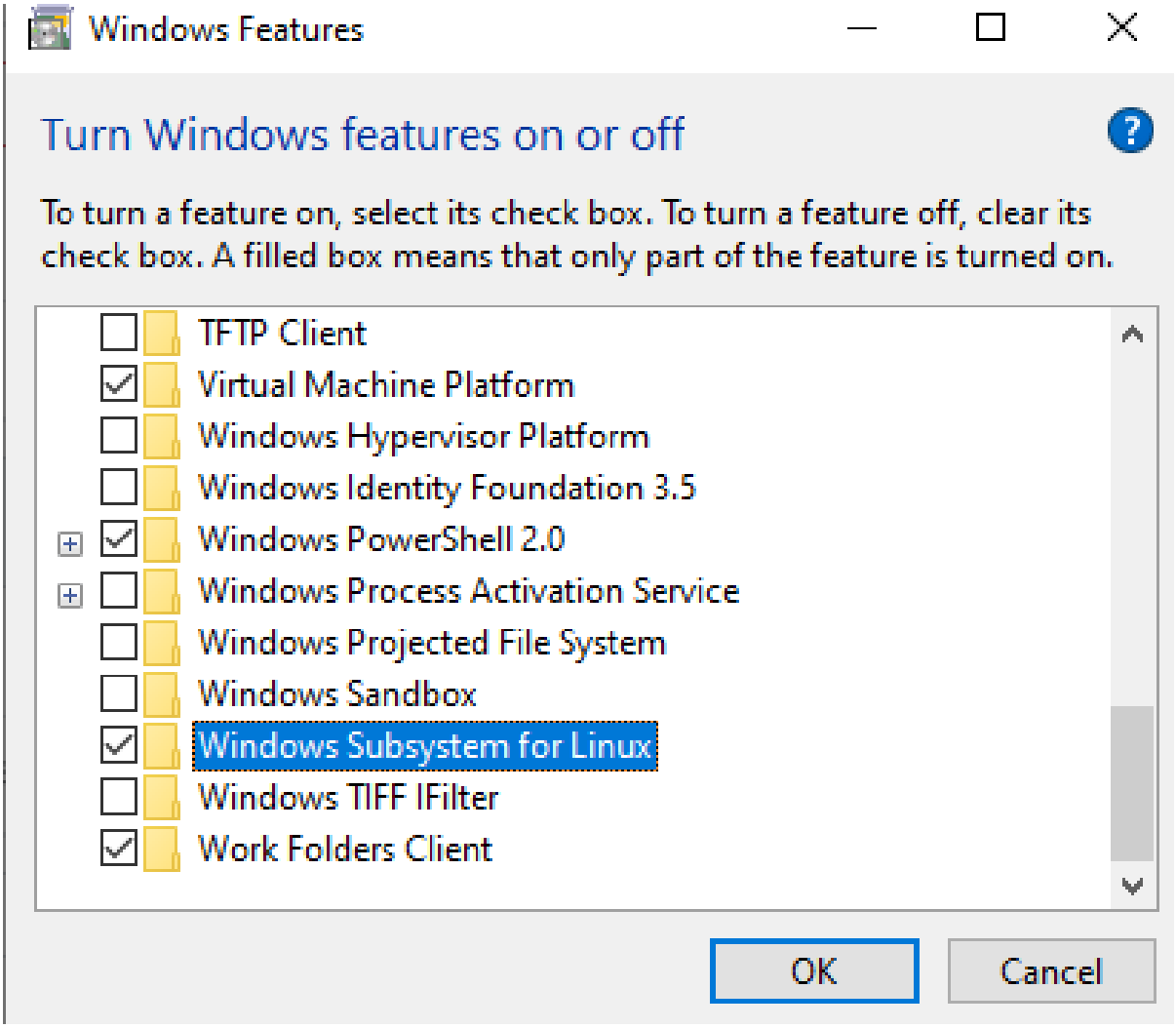


You can then attempt to restart Docker Desktop. This also may require a system restart.

If you are still experiencing issues, attempt to enable virtualization from Basestack:

3. `System -> Windows Services -> WSL2 -> Turn WSL On`
4. `System -> Windows Services -> WSL2 -> Enable Virtualization`
5. `System -> Windows Services -> WSL2 -> Set WSL2`

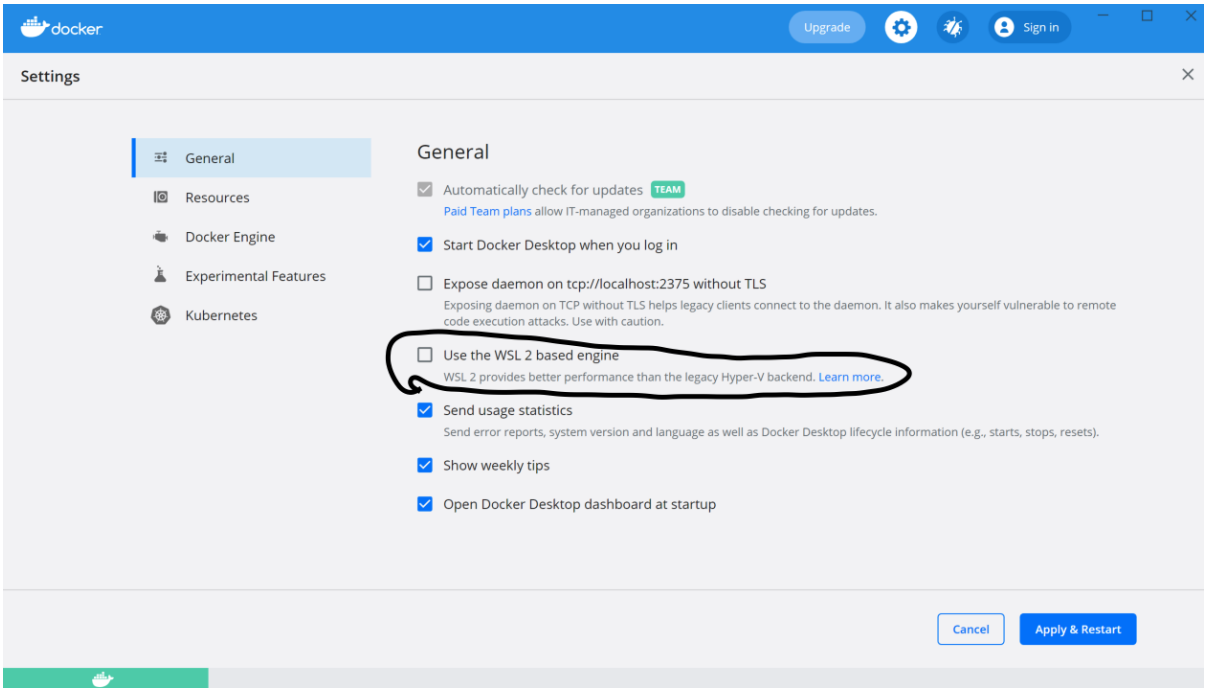
Or from "Turn Windows features on or off". This is also a good way double check that it is now enabled.



You will need to restart your PC/Laptop after doing this!

Switching between HyperV and WSL2 instance

When inside Docker-Desktop, hit the settings (cog) icon at the top-right of the page. Then, select General Tab and tick/un-tick the *Use the WSL 2 based engine*. Be aware that when using HyperV you may need to adjust resources to accommodate your system appropriately.



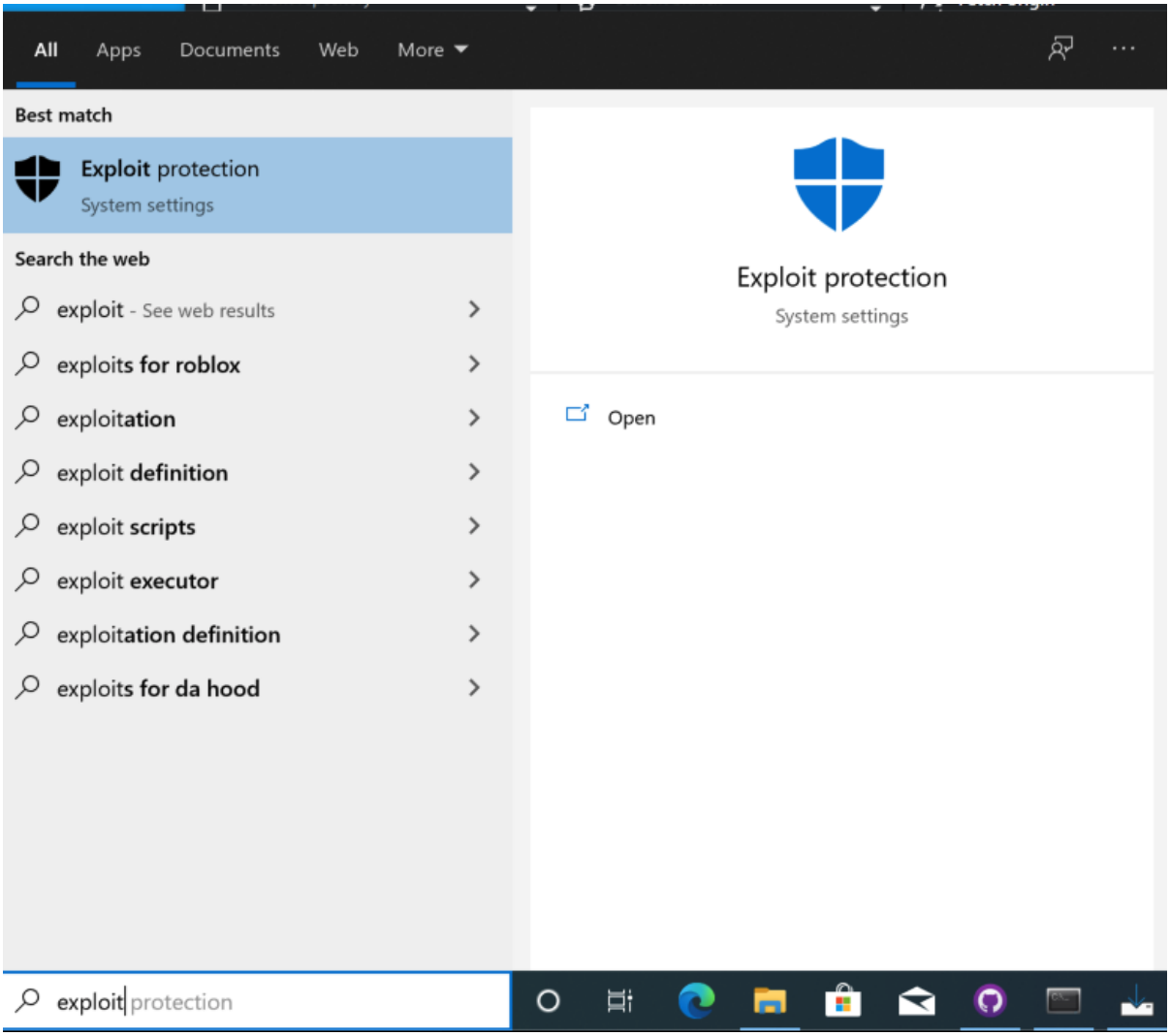
WSL2 error on Docker Start

If you're still experiencing issues after install WSL2 (also making sure you've attempted to use both installation methods), there may be an issue with your system's firewall configuration. To alleviate this, you can modify some settings within Windows by following:

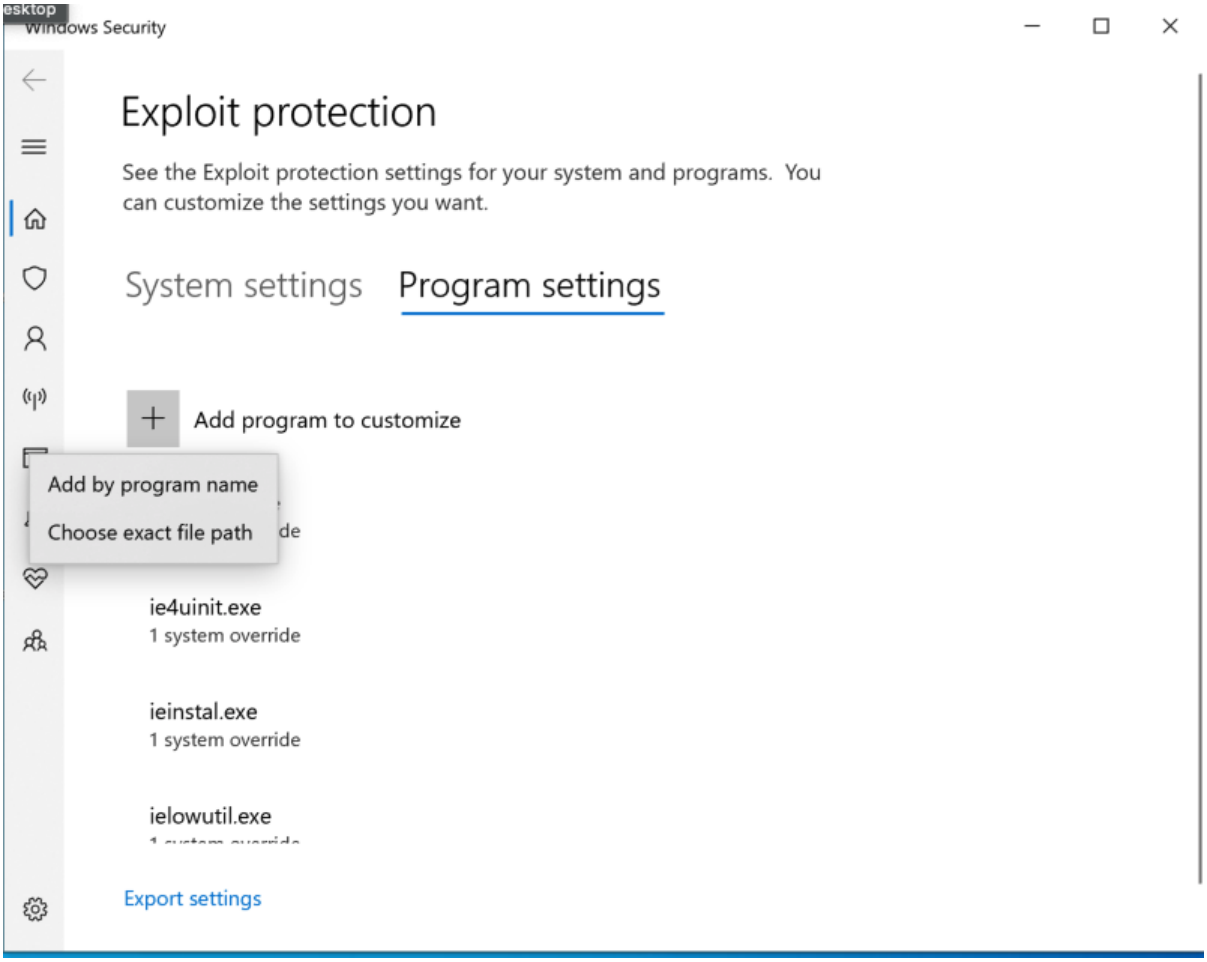
1. navigate to "Start" button,
2. type in "Exploit protection" and run it as administrator,

3. once in, nawigate to: "Program settings" "Add program to customise", adding the two below separatelly, in each case, disabling by unticking: "Code flow guard (CFG)"

```
C:\WINDOWS\System32\vmwp.exe
C:\WINDOWS\System32\vmcompute.exe
```

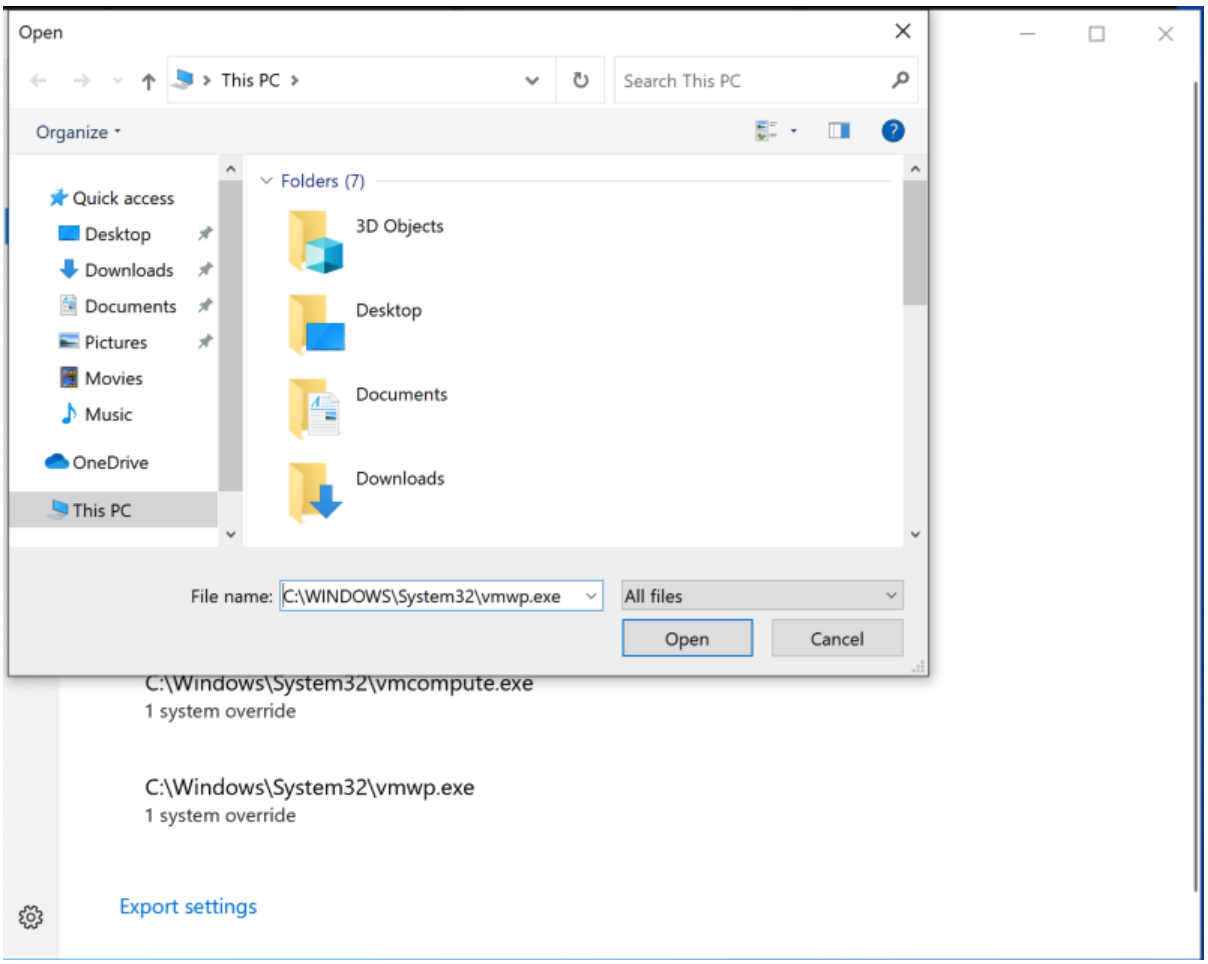


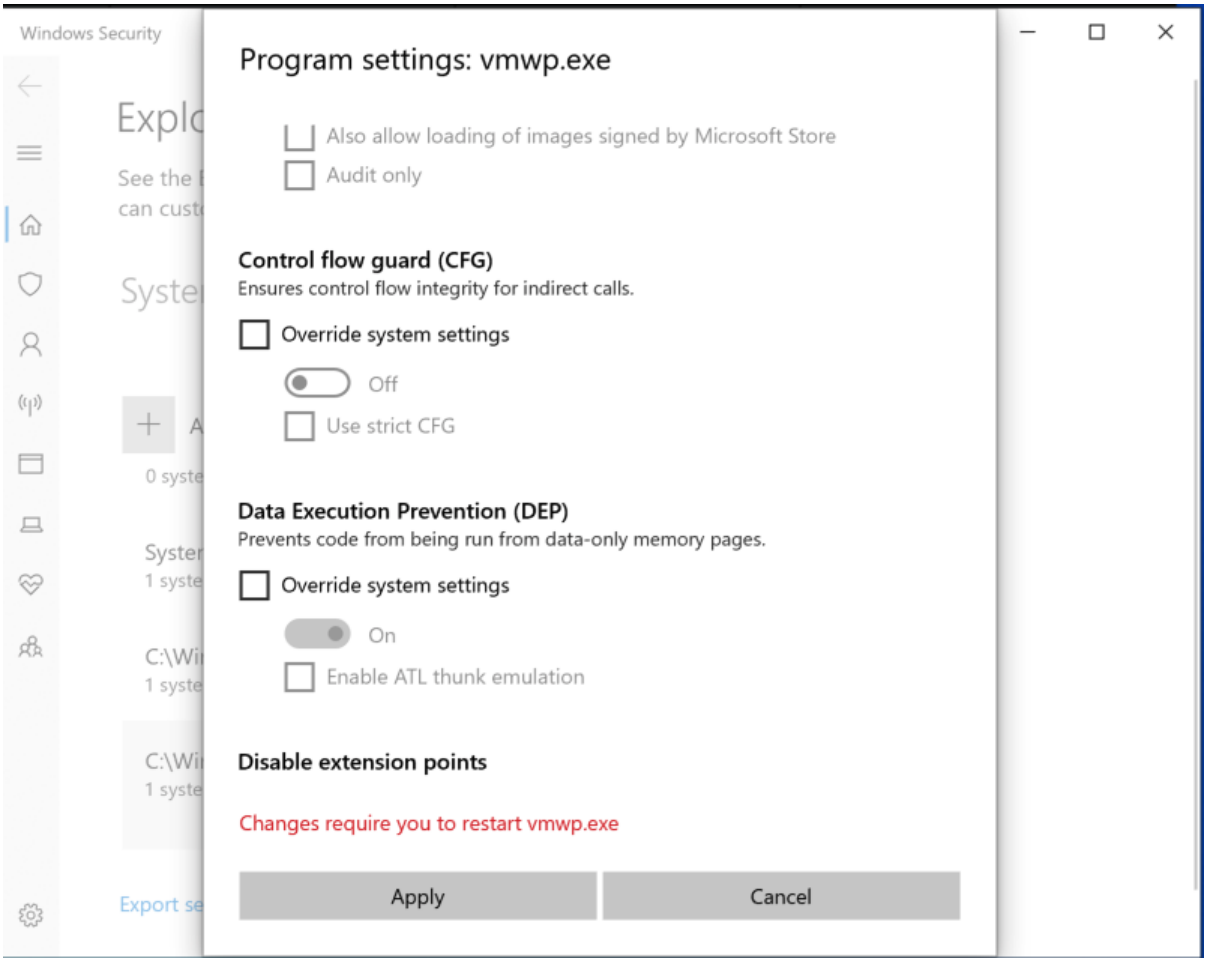
Select *choose exact file path*



Copy + paste these 2 commands one-by-one then apply changes

```
C:\WINDOWS\System32\vmwp.exe
C:\WINDOWS\System32\vmcompute.exe
```


Open
This PC
Search This PC
Organize
Quick access
Desktop
Downloads
Documents
Pictures
Movies
Music
OneDrive
This PC
Folders (7)
3D Objects
Desktop
Documents
Downloads
File name: C:\WINDOWS\System32\vmwp.exe
All files
Open
Cancel
C:\Windows\System32\vmcompute.exe
1 system override
C:\Windows\System32\vmwp.exe
1 system override
Export settings



Finally, restart **Docker Desktop**

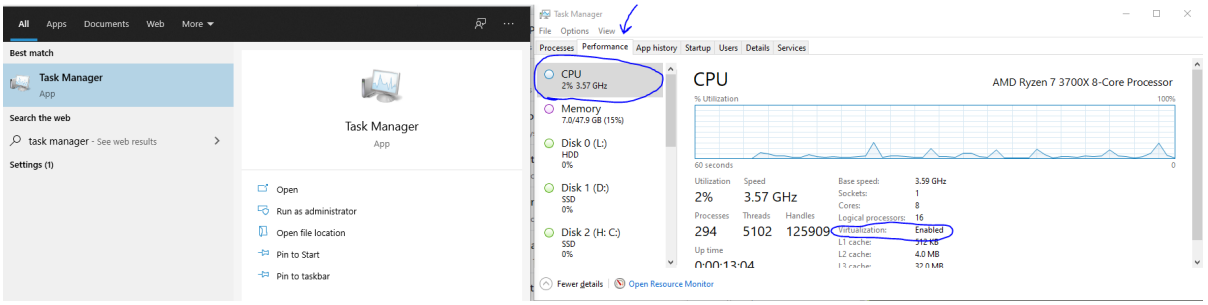
*Credit to this solution

Virtualization Disabled - Windows

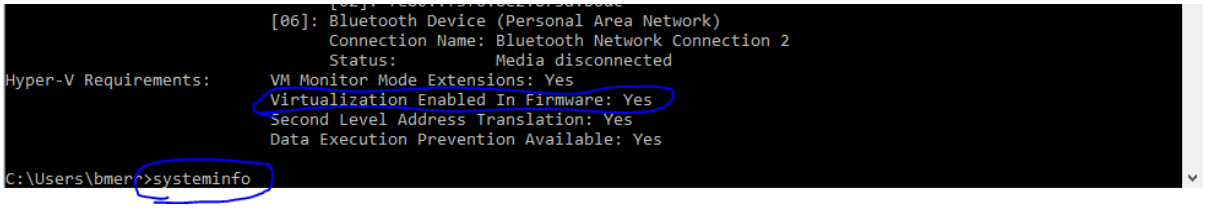
In order for either of the above to work, you need to ensure that **virtualization** is enabled in your firmware. Some processors do so by default, others do not. If you are having issue with starting Docker despite following either of the options above, please see below.

First, check that your CPU can support virtualization by viewing the model on Intel/AMD product page(s)

You can first check if it is enabled by going into the **Task Manager** and seeing if the Virtualization attribute is enabled.

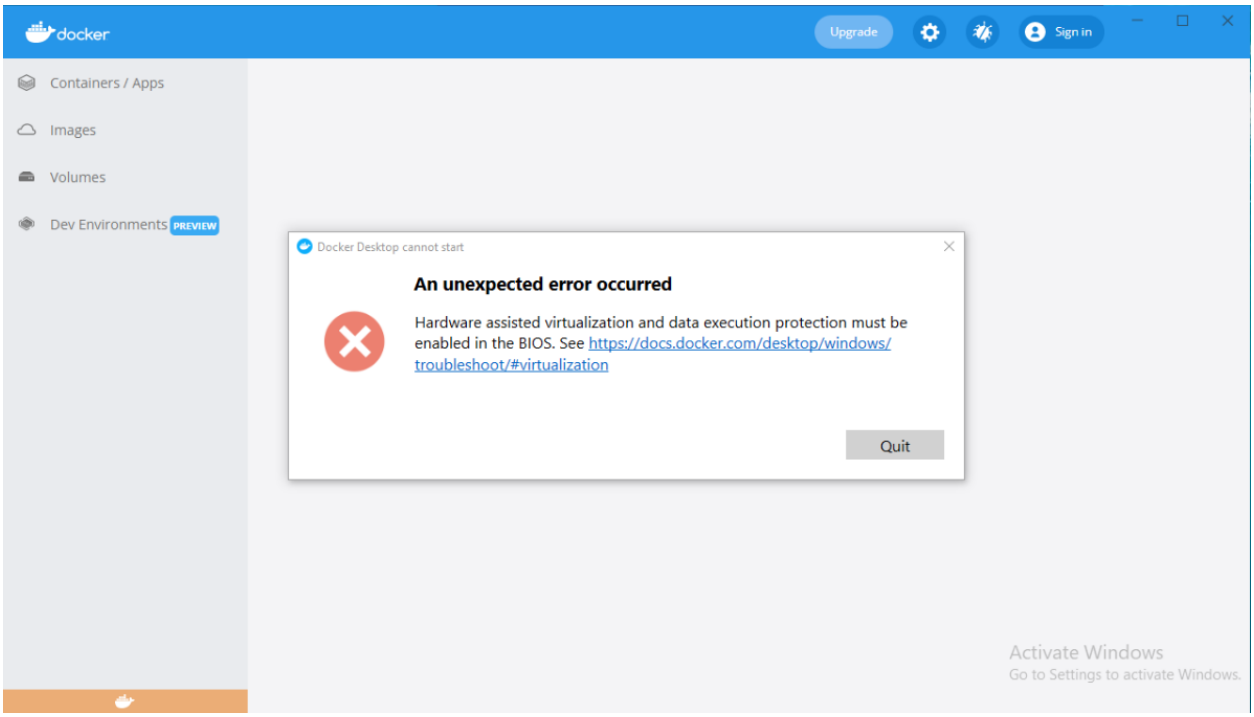


If it is not, open up **Command Terminal** and type: `systeminfo`. Scroll to the bottom of the output and check if the Firmware has it enabled for Hyper-V requirements.



If not, you will need to enable Virtualization in your BIOS. This process will look different based on everyone's system. You should try to follow the instructions in this [link](https://www.thewindowsclub.com/disable-hardware-virtualization-in-windows-10). Choose your manufacturer type.

You might also experience an error about BIOS not having virtiualization enabled as soon as Docker attempts to start



Follow these steps from https://bce.berkeley.edu/enabling-virtualization-in-your-pc-bios.html:

Reboot your computer

- Right when the computer is coming up from the black screen, press Delete, Esc, F1, F2, or F4. Each computer manufacturer uses a different key but it may show a brief message at boot telling you which one to press. If you miss it the first time, reboot and try again. It helps to tap the key about twice a second when the computer is coming up. If you are not able to enter the BIOS via this method, consult your computer's manual.
- In the BIOS settings, find the configuration items related to the CPU. These can be in under the headings Processor, Chipset, or Northbridge.
- Enable virtualization; the setting may be called VT-x, AMD-V, SVM, or Vanderpool. Enable Intel VT-d or AMD IOMMU if the options are available.
- Save your changes and reboot.

- Delete any existing VMs (Machine > Remove ** and select ** Delete all files) and re-import the .ova file (following step 4 and subsequent steps of the installation instructions).
- Check if your system supports Virtualization
- If you are unable to find the Virtualization settings in your BIOS it may mean that your laptop does not support it. If you want to try to find this out yourself, then you can try:

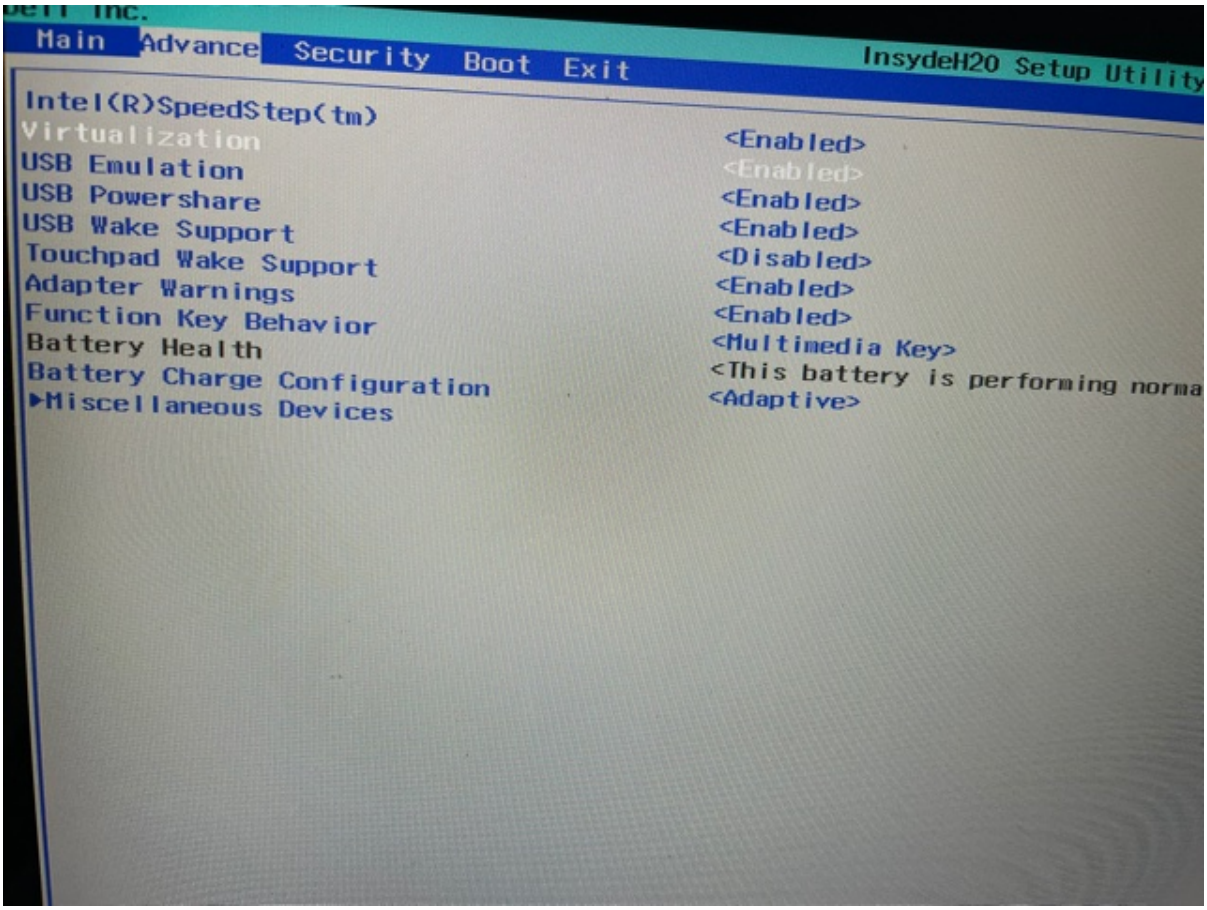
On Windows, download and run a Microsoft utility. You can also download utilities to check if your CPU is capable of virtualization, if not enabled. Hyper-V must be disabled in order for VirtualBox to run 64-bit guest operating systems. Visit the "turn Windows feature on or off" application and make sure Hyper-V is not checked.

On Linux, open a terminal window and run:

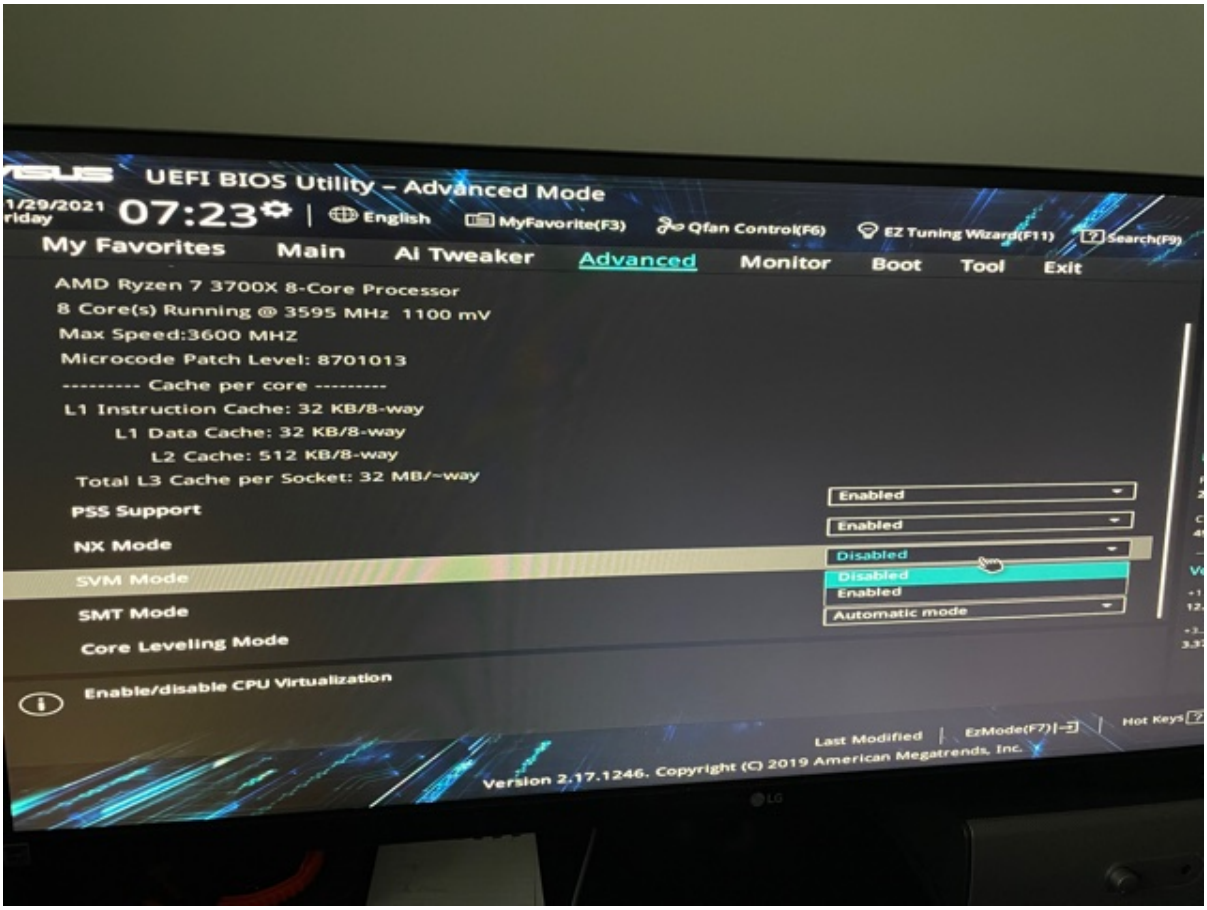
```
egrep -q 'vmx|svm' /proc/cpuinfo && echo yes || echo no
```

Typically, though, to enter BIOS you must restart the computer and while it is booting hit **DEL** or **F2** or sometimes **F12**. This process is usually very quick so be ready. When it is booting, you may be able to catch the necessary keys flash.

The default BIOS should look like the one below. In there, head to the **Advanced** tab and check if **Virtualization** is present. If so, enable it, save changes, and restart. If not, try to search in other tabs or open up some options that have further submenus within them as there is no guarantee it will be directly on the base **Advanced** tab.



On AMD CPU's if you don't see virtualization it may be labeled as **SVM** in the **Advanced** tab



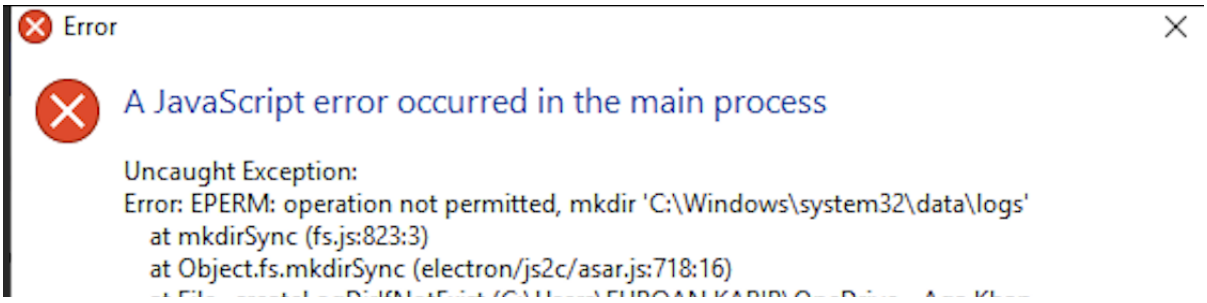
If the option is not present in the BIOS that means that your CPU does not support Virtualization and Docker **won't be able to properly run on your system.**

docker_support

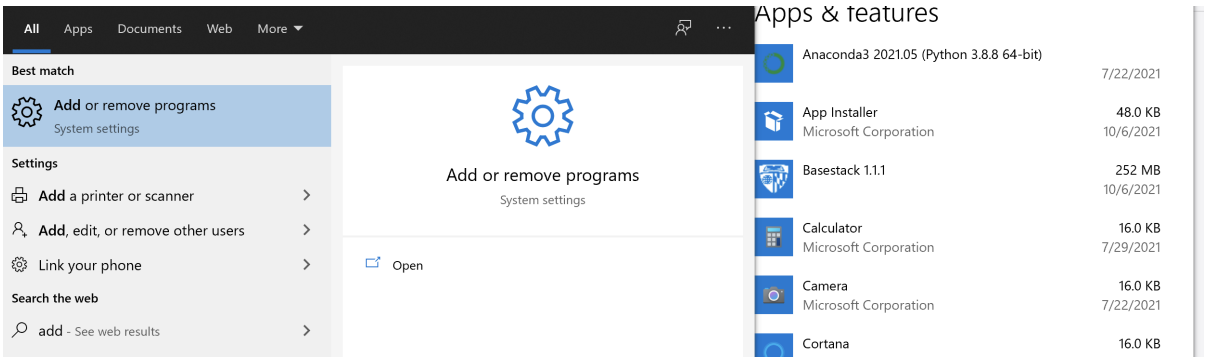
Operation not permitted - Windows

If you receive an error about operation not being permitted and you're reinstalling or updating Basestack (see image), attempt to uninstall Basestack and reinstall it.

Operation not permitted error



Uninstall Basestack

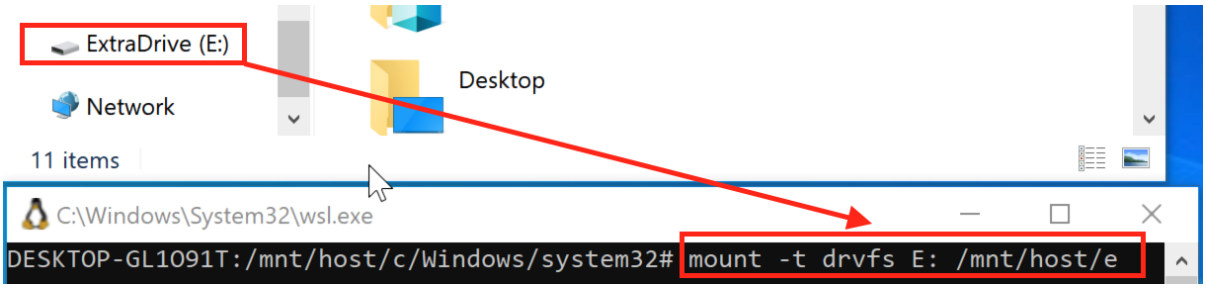


Docker Pipelines Do Not Run With External Drives

If you need to use an external Drives within the modules AND are using Docker-Desktop for Windows with WSL2, you will need to likely mount your external drive within WSL first.

This is done in 2 steps: 1. Start wsl from the command line by typing wsl into the quicklaunch and starting it - Make sure Docker is running on your system before starting this - A terminal window will appear at the start. Done fear you can follow step 2 easily by copying it in the terminal and just changing the <drive_letter> (See below for more info)

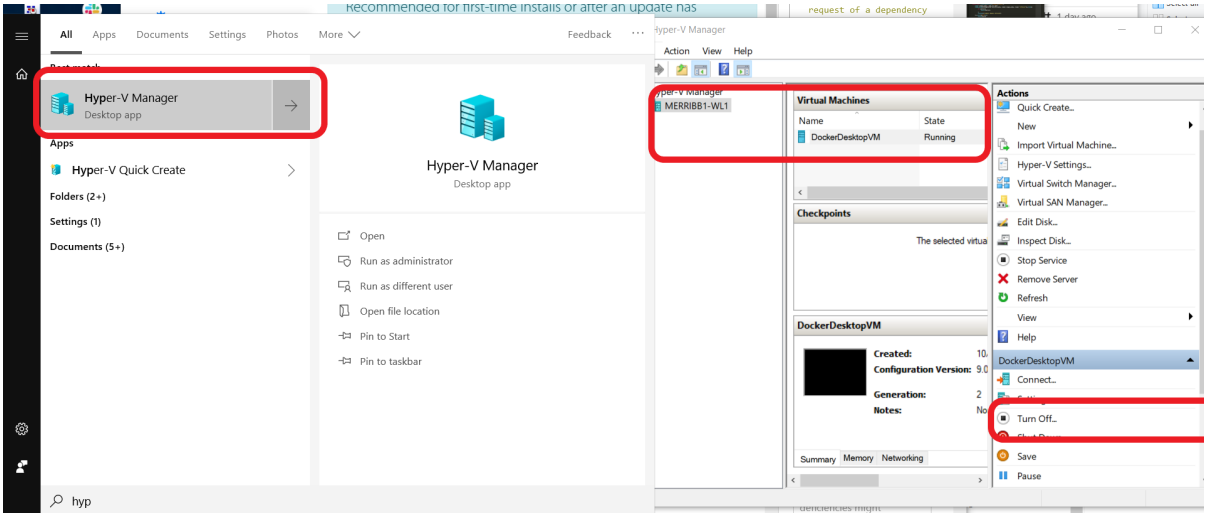
2. `mkdir -p /mnt/host/<drive_letter> && mount -t drvfs <drive_letter> /mnt/host/<drive_letter>` - This process maps your new drive letter to the exact letter in windows. For instance in the example my flash drive is E: and /mnt/host/e is what is it mapped to - `<drive_letter>` is whatever the letter is from your system. For example the example shown here is E: but yours may (likely to) differ



Alternatively, you can switch WSL2 instance to HyperV by following this [step](#switching-between-hyperv-and-wsl2-instance)

I/O timeout

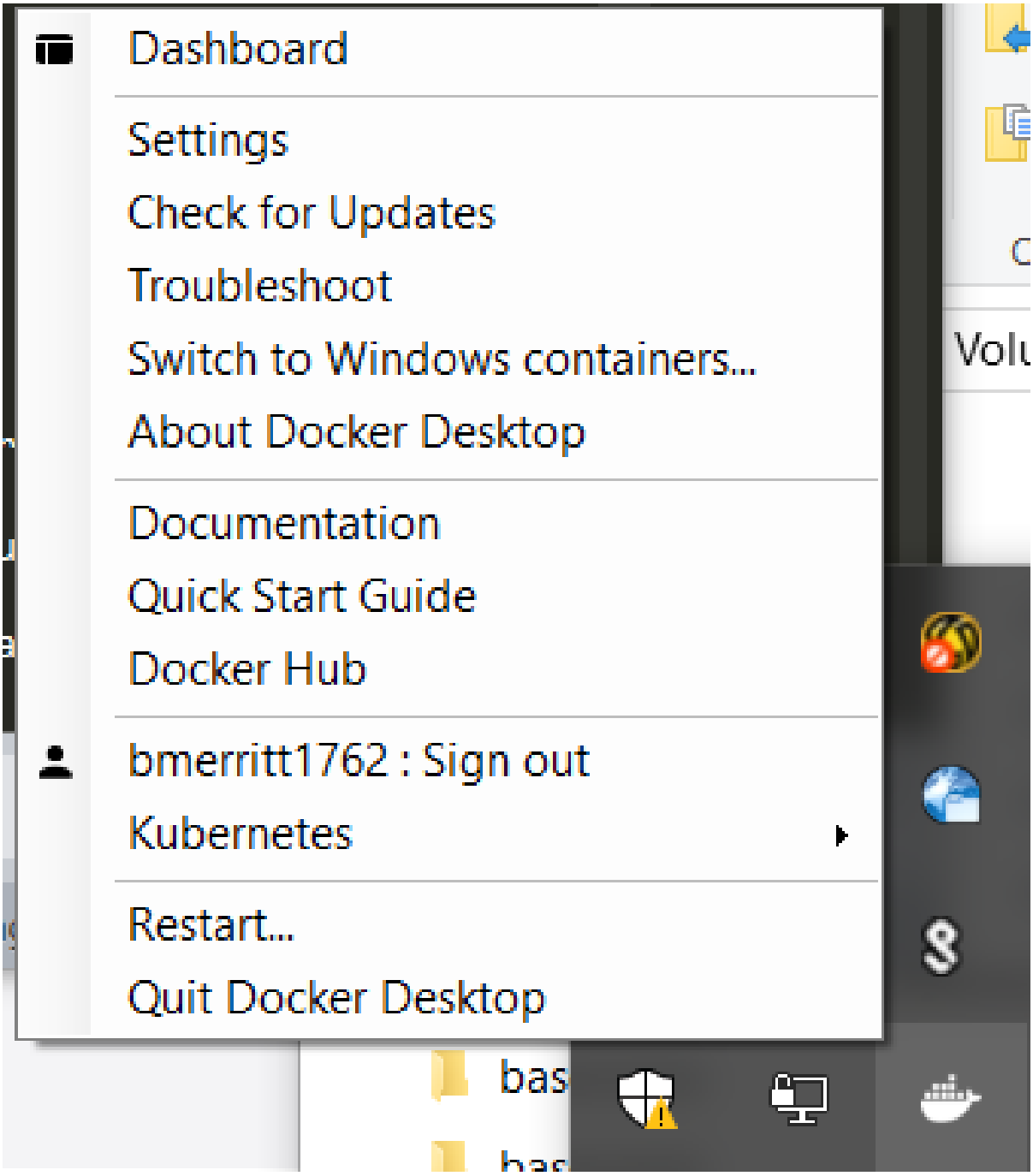
If installing as the offline method, sometimes you may retrieve and I/O exception as an error message. This is typically because you've tried to send docker to build too many images from large files in a short period of time. To fix this, you'll need to rerun docker a specific way



- Simply seach for Hyper-V Manager in your toolbar, select the VM (usually your username is in its name), and then *Turn Off*. Docker will then shut down and you can restart it

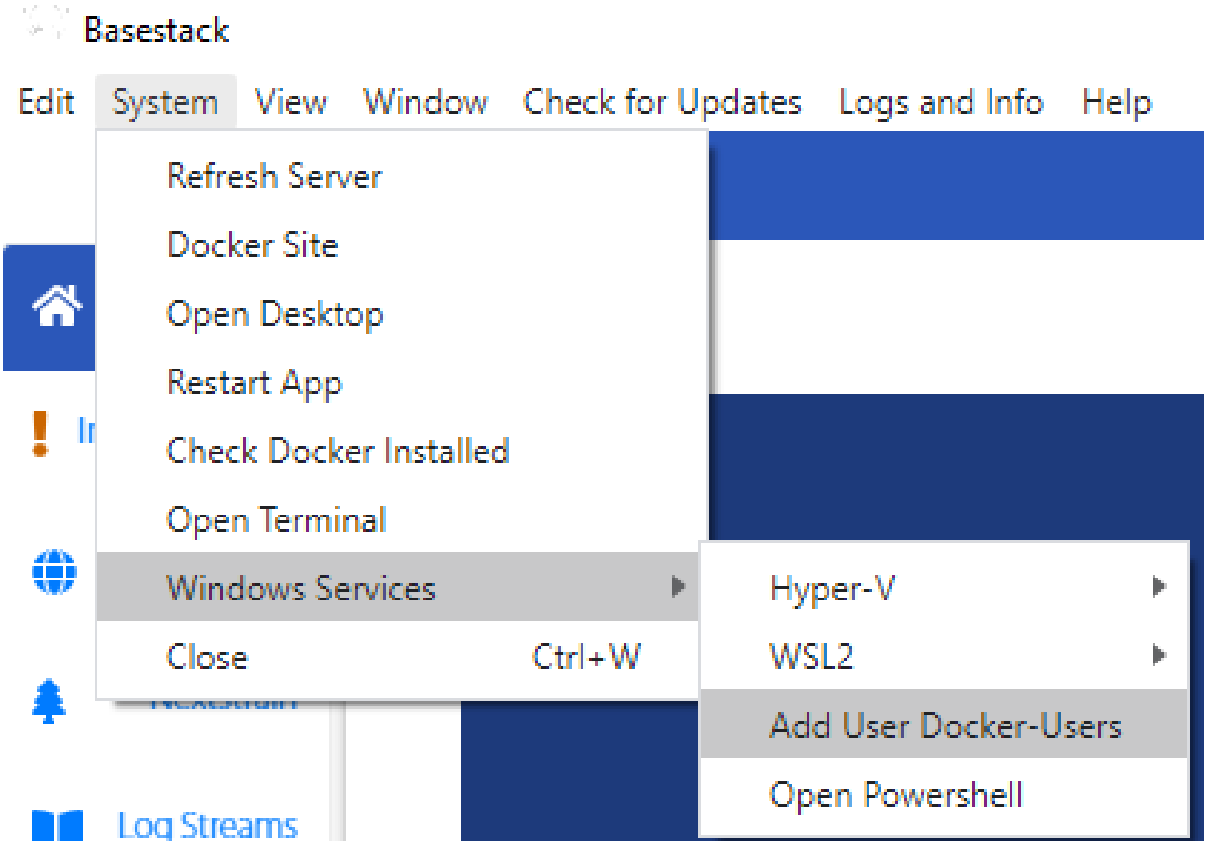
Docker Connection

If you receive an error that you couldn't connect to docker, please try to restart via the taskbar

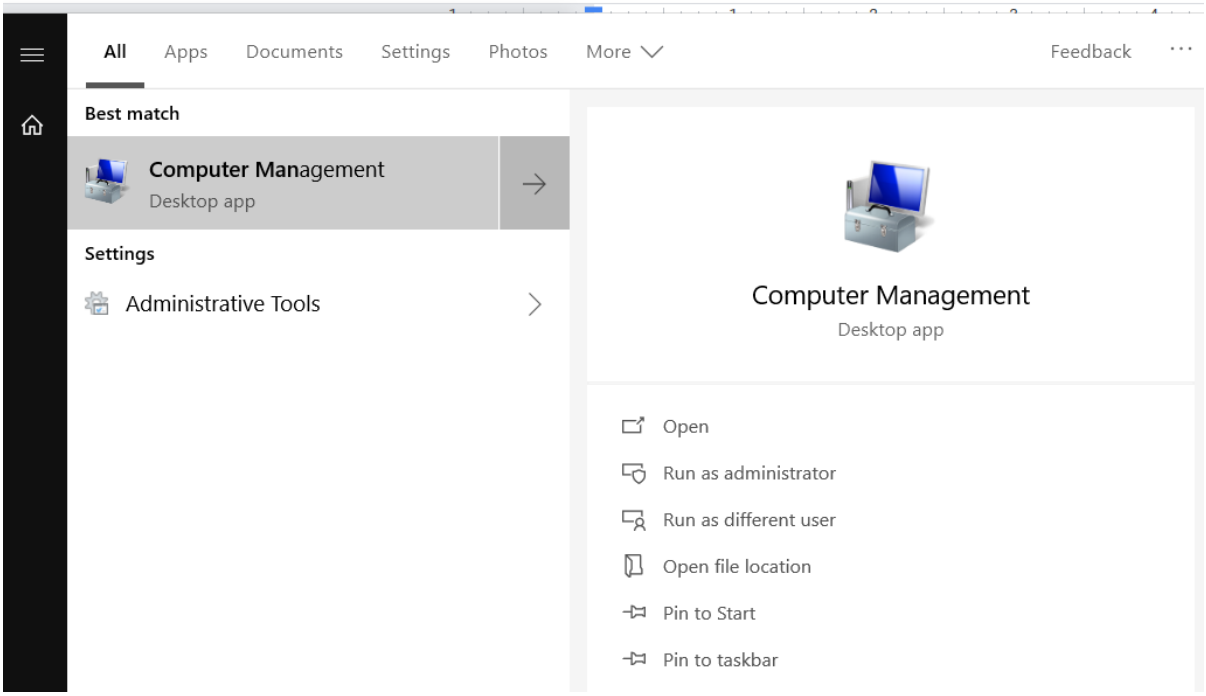

Dashboard
Settings
Check for Updates
Troubleshoot
Switch to Windows containers...
About Docker Desktop
Documentation
Quick Start Guide
Docker Hub
bmerritt1762 : Sign out
Kubernetes
Restart...
Quit Docker Desktop

'You are not allowed to use Docker, you must be in the "docker-users" group' - Windows

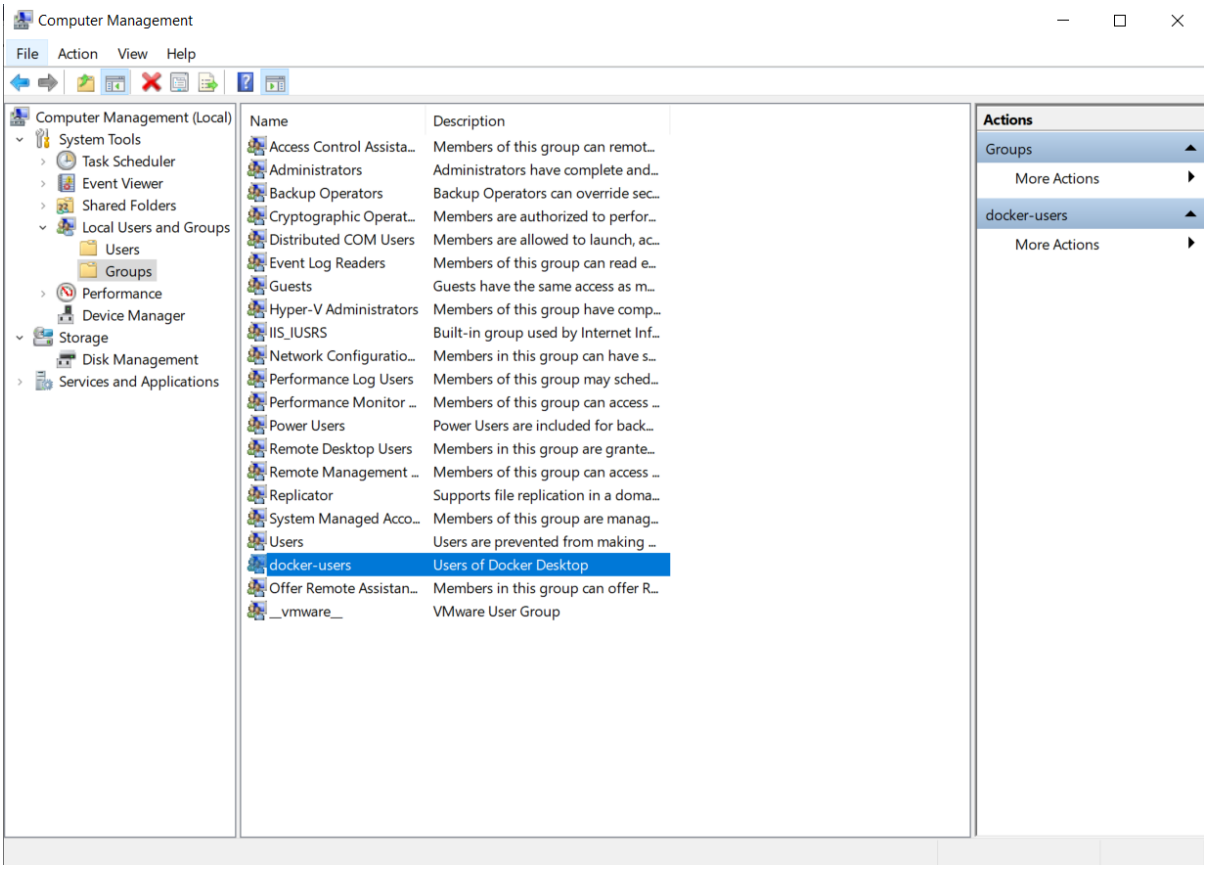
In Basestack select `System -> Windows Services -> Add User Docker-Users`. When completed you should see that either you're already a part of that group **OR** you've been successfully added.



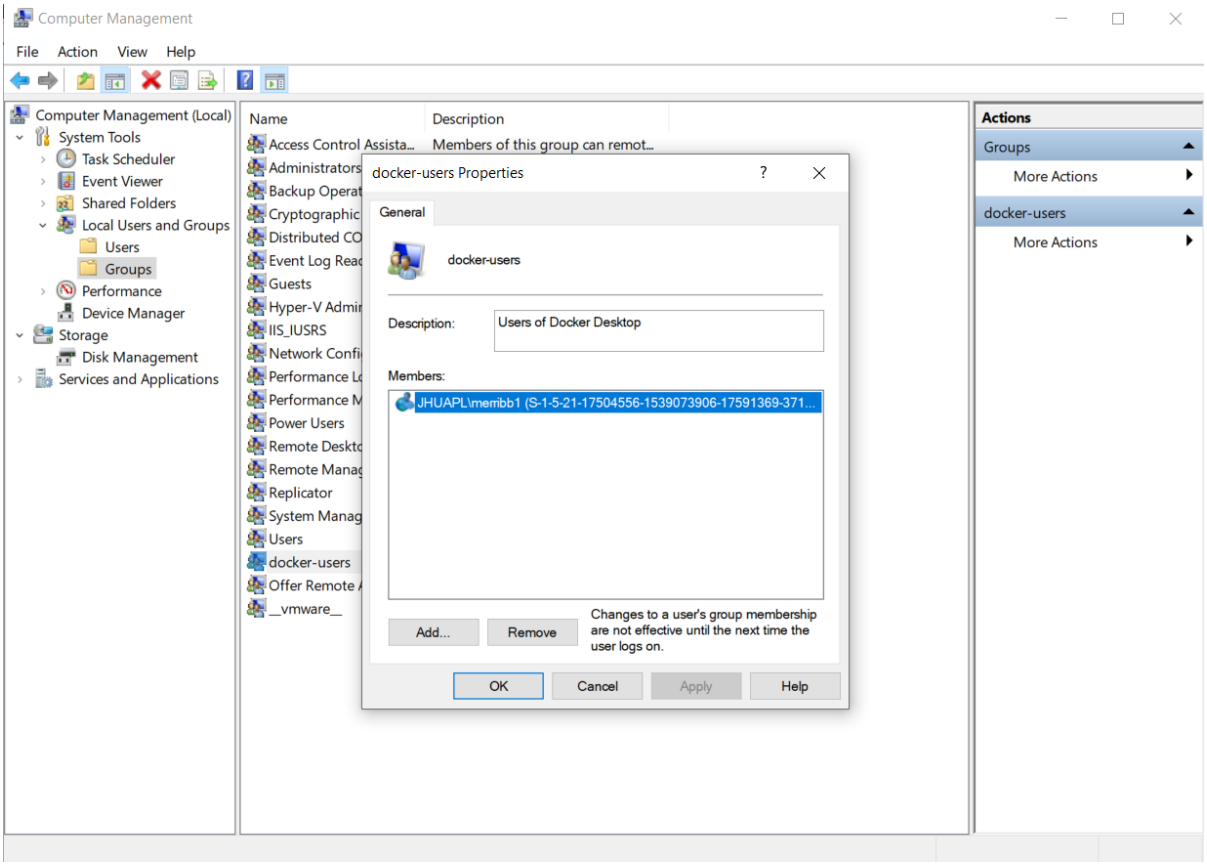
Alternatively if the above does not work try the following:



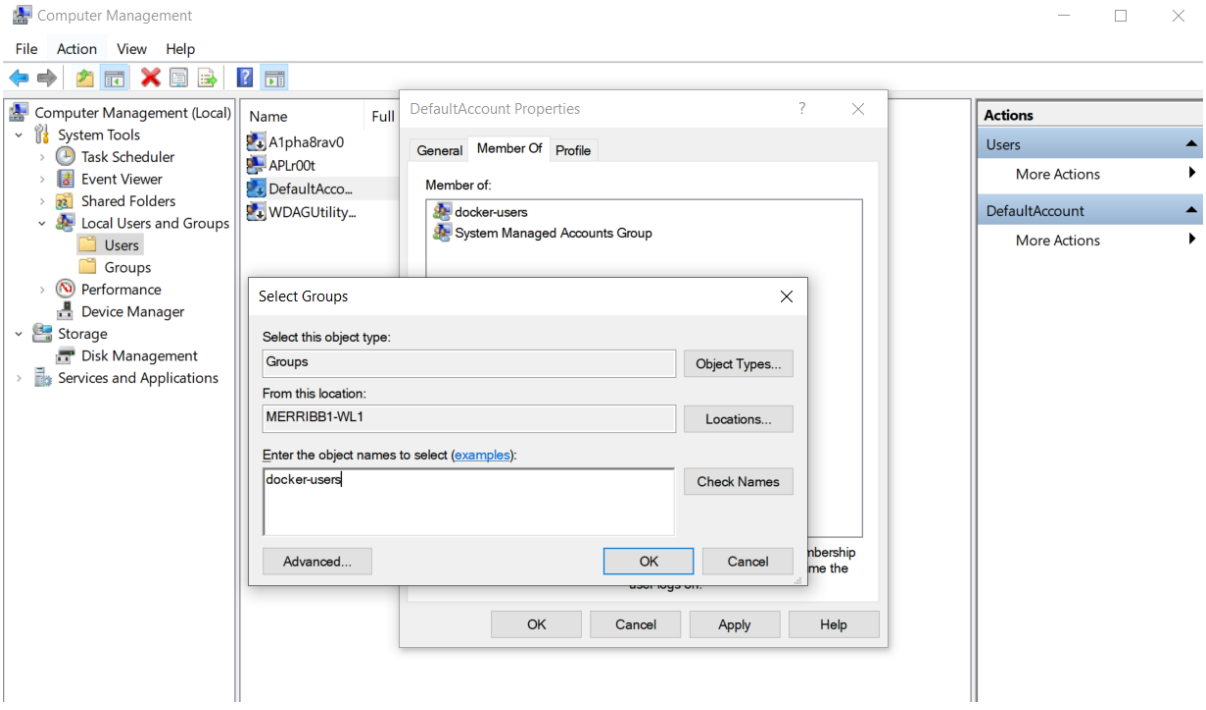
- In the Windows search taskbar (bottom left icon), find `Computer Management`



- Select (left-side) System Tools -> Local Users and Groups -> Groups



- Double click `docker-users` and see if your name is there, if not:



- Select the `Users` folder right about where you clicked `Groups`
- Select the name of your user

- **enter `docker-users` into the object field and add.**
  - You will need to log out and back into your account for this to take effect

Linux

Permisson denied (Linux)

Please ensure that you follow the correct *Docker Installation* here to using *userns-remap*

Note that this will map all of your processes INSIDE the docker containers to your user id if used properly. You will need sudo to delete any files or folders that are causing issues.

3.1.5 Uninstall

Uninstall

Docker Image(s):

- In Basestack, go to `Module Install`
- **Select: `Remove Docker Images` (trash-bin icon)**
  - This will only remove Basestack-specific images

Basestack

Windows

```
Add or remove programs -> Select Basestack -> Uninstall
```

Mac

Drag `~/Library/Application Support/Basestack` to the Bin

Linux

1. Remove the Basestack.AppImage Folder or Executable
2. Remove the directory: *~/.config/Basestack*

User Data

- Mac OS: `~/Library/Application Support/Basestack (taken from the name property in package.json)`
- Windows: `C:\Users\<you>\AppData\Local\Basestack Name`
- Linux: `~/.config/Basestack`

3.2 Usage

Note: Test data for most modules can be found here from a Gdrive location

3.2.1 Modules

Pavian

Pavian is a tool designed to analyze report files from a variety of classification modules, namely, for Basestack, Mytax uses Kraken2

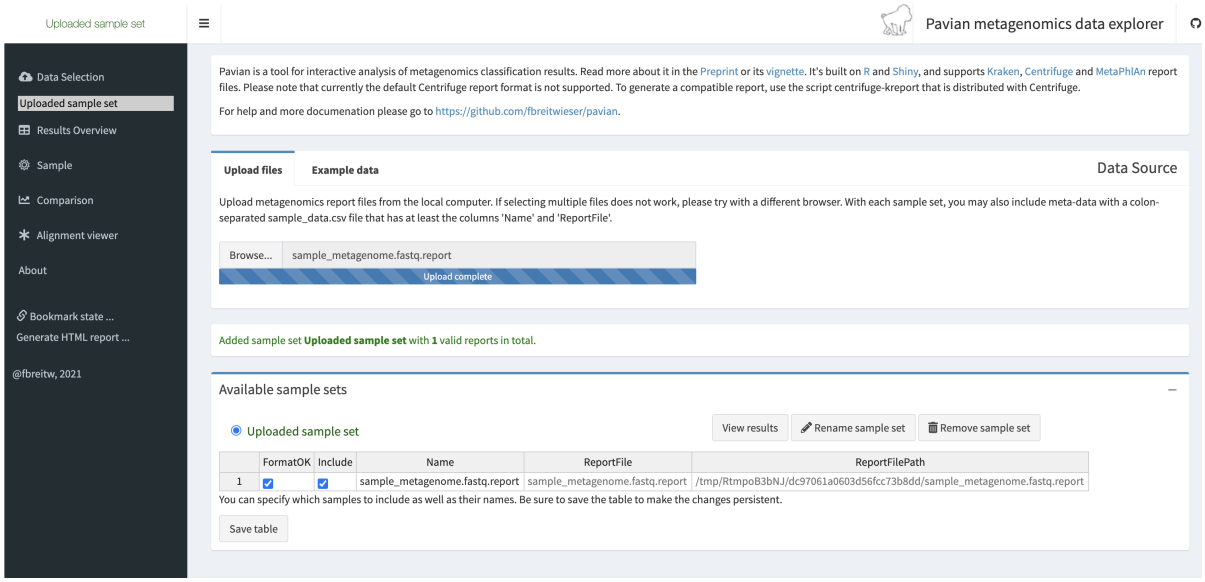
Note: This module contains a UI to be displayed from the rendering button in the variables table

Parameters

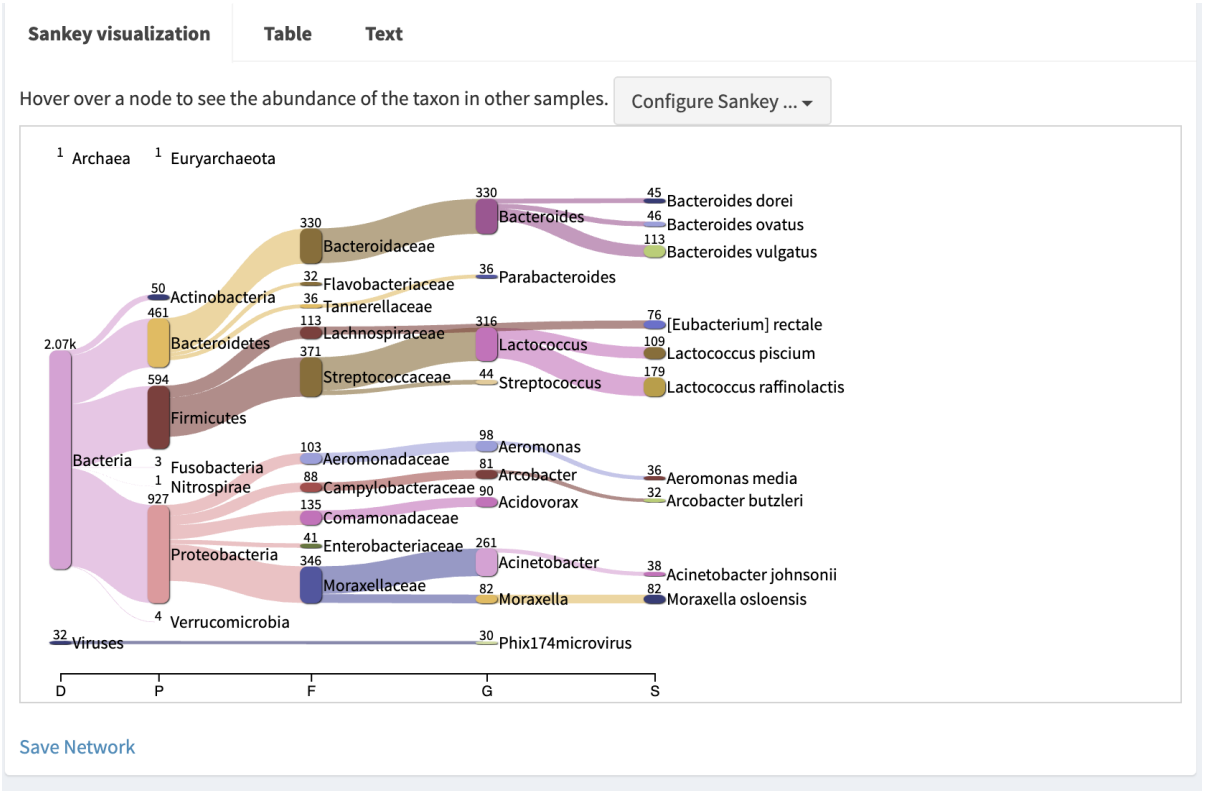
report file
[*File*] Report file from Kraken2 run. See *here*. To be input INSIDE the running visualization. See images below

Returns

Pavian Visualization
[*User interface*] The tool simply requires you to input your kraken.report (from mytax or otherwise) into the home-page.



Select `Sample` in the left-hand navigation drawer to view diagrams and information of your report



NCBI Scrubber

Note: Only works on interleaved R1/R2 fastq file OR NanoPore reads.

NCBI scubber removes human reads from sequencing data

Parameters

input
[*File*] Your input fastq file. Can be compressed but you must check "decompressed" if so

interleaved
[*Boolean*] Is the file interleaved? (TRUE/FALSE, Illumina)

decompressed: *Boolean*
Check if your file is compressed. You can decompress it directly in the UI here to the name: decompressed.fastq in the same path as the input file

Returns

Filtered File
[*.filtered.fastq*] removed of human reads

Mytax

Mytax is a tool designed to utilize the *Kraken2 <https://ccb.jhu.edu/software/kraken2/>* classification tool in order to identify (and visualize) taxonomic designations of fastq files. It includes a flu-kraken (from JHUAPL) and minikraken (from developers of kraken) set of databases in the installed images.

Note: This module contains a UI to be displayed from the rendering button in the variables table

Parameters

fastq file
[*File*] FastQ file (single) obtained from any sequencing run. You can consolidate multiple fastq files into one if you want to run a long sequencing run.

nodes
[*Directory*] Location of the taxonomy files. Default is contained within the `jhuaplbio/basestack_mytax` image and pre-loaded on run

database
[*Directory*] Directory to use as the database for Kraken2. Defaults to Flukraken pre-loaded in the Docker image.

memory-mapping`
[*Option*] Choose to pre-load the database (fast, requires high RAM) or run on the local filesystem (slow, only use with low RAM available)

Returns

Kraken-Report File
[*.report*]

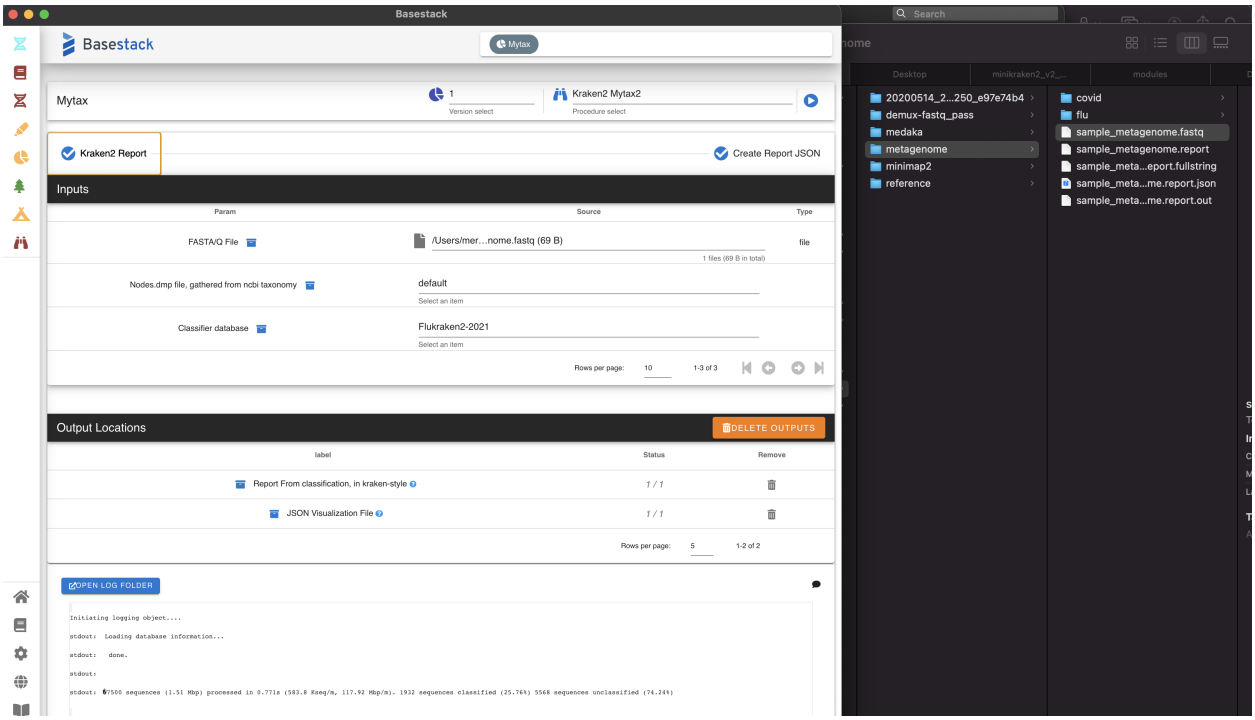
- Used in Pavian and for additional 3rd-party applications
- Contains a summary of the run

Kraken-JSON
[*.json*]

- Formatted file uploaded/used in the Mytax Sunburst Visualization Procedure.

There are 2 primary procedures for this module:

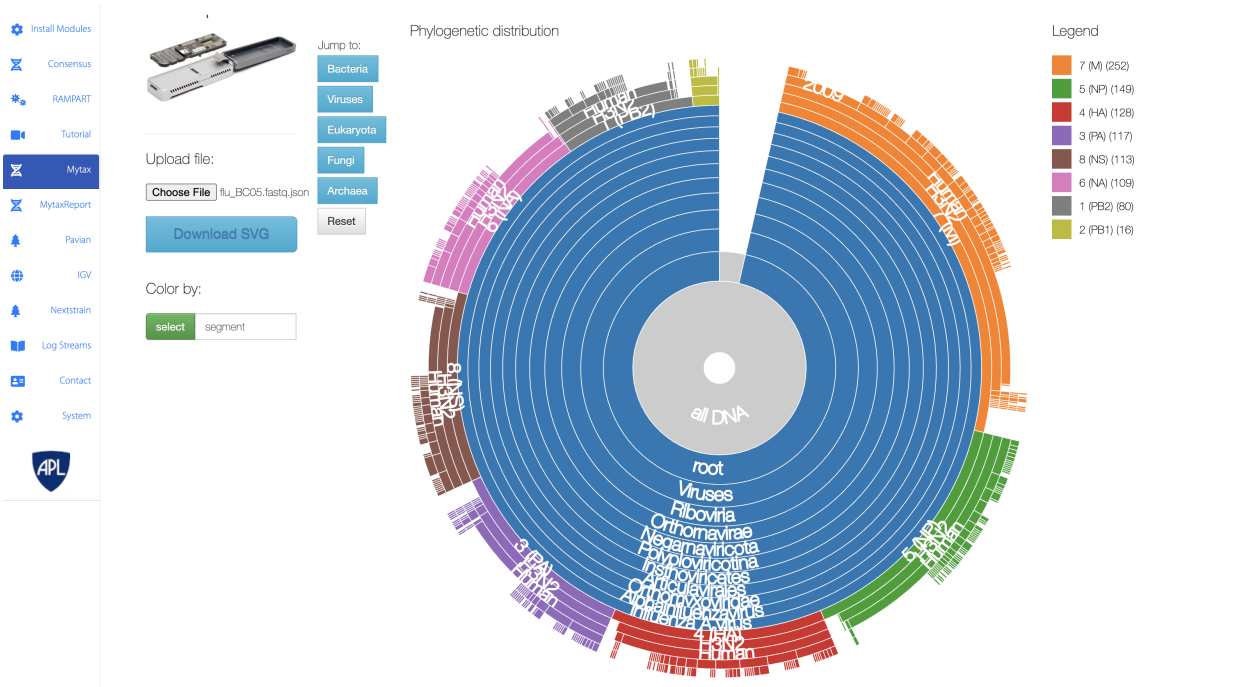
1. **Create Report Files**
    - Run kraken-classify to create kraken.report and kraken.json files from a single fastq file
2. **View the Visualization**
    - View the output JSON file from Step 1



WIP Procedures

- Download and create/update custom databases
- **Utilize other classifiers or aligners**
    1. Centrifuge
    2. Kraken2
    3. BLAST

Display



Mytax version 2 (Metagenomics)

> **Warning:** This module is under construction and is in alpha-release. Scheduled full release of v1.0 in Oct. 2022

Please see relevant links in the listed modules for more information on the underlying mechanisms and corresponding papers (if existent)

Parameters

- Samplesheet (.csv): *file*

  Contains a mapping of metadata and a single sample per row. Explanations of the possible columns for Basestack are seen below:

Table 1: Samplesheet Description

Column Name	Description
sample	Custom sample name. This entry will be identical for multiple sequencing libraries/runs from the same sample. Spaces in sample names are automatically converted to underscores (_).
format	[directory, run, file] Is it a run directory of files that need to be demux'd, an already full directory of files for Oxford, or a single file (or paired files)
path_1	Full path to FastQ file for Illumina short reads 1 OR OXFORD reads. File has to be gzipped and have the extension ".fastq.gz" or ".fq.
path_2	Full path to FastQ file for Illumina short reads 2. File has to be gzipped and have the extension ".fastq.gz" or ".fq.
kits	What default barcode kit to use for demux. Only applies to those with the "run" format
pattern	Regex matching for the names of the folders that are made on demux. Default is barcode[0-9]+
platform	Platform used, [illumina, oxford]
database	Kraken2 database path (root level folder)
compressed	TRUE/FALSE for gunzipped files

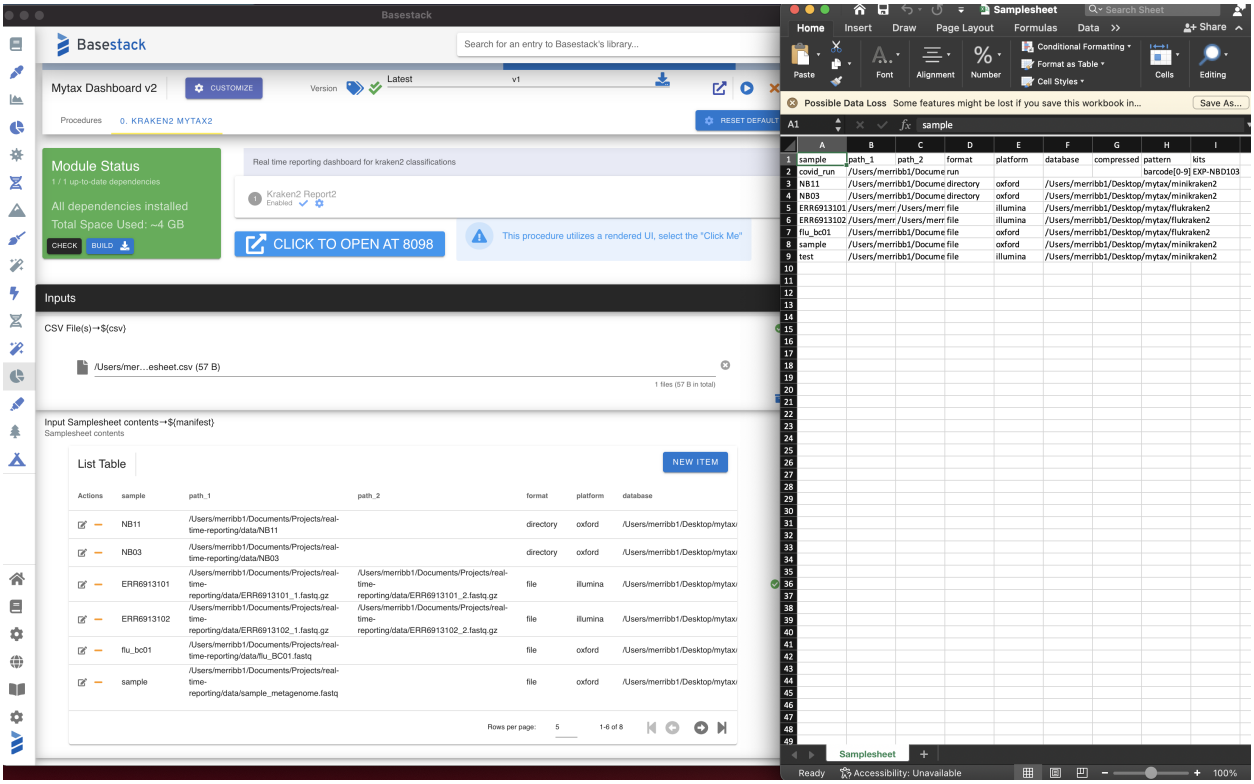
Table 2: Example Samplesheet

sample	path_1	path_2	format	platform	database	compressed	pattern	kits
covid_run	fastq_pass		run	oxford			barcode[0-9]+	EXP-NBD103
NB03	./NB11		directory	oxford	minikraken2	FALSE		
ERR123	ERR123_R1.fastq.gz	ERR123_R2.fastq.gz	file	illumina	flukraken2	TRUE		

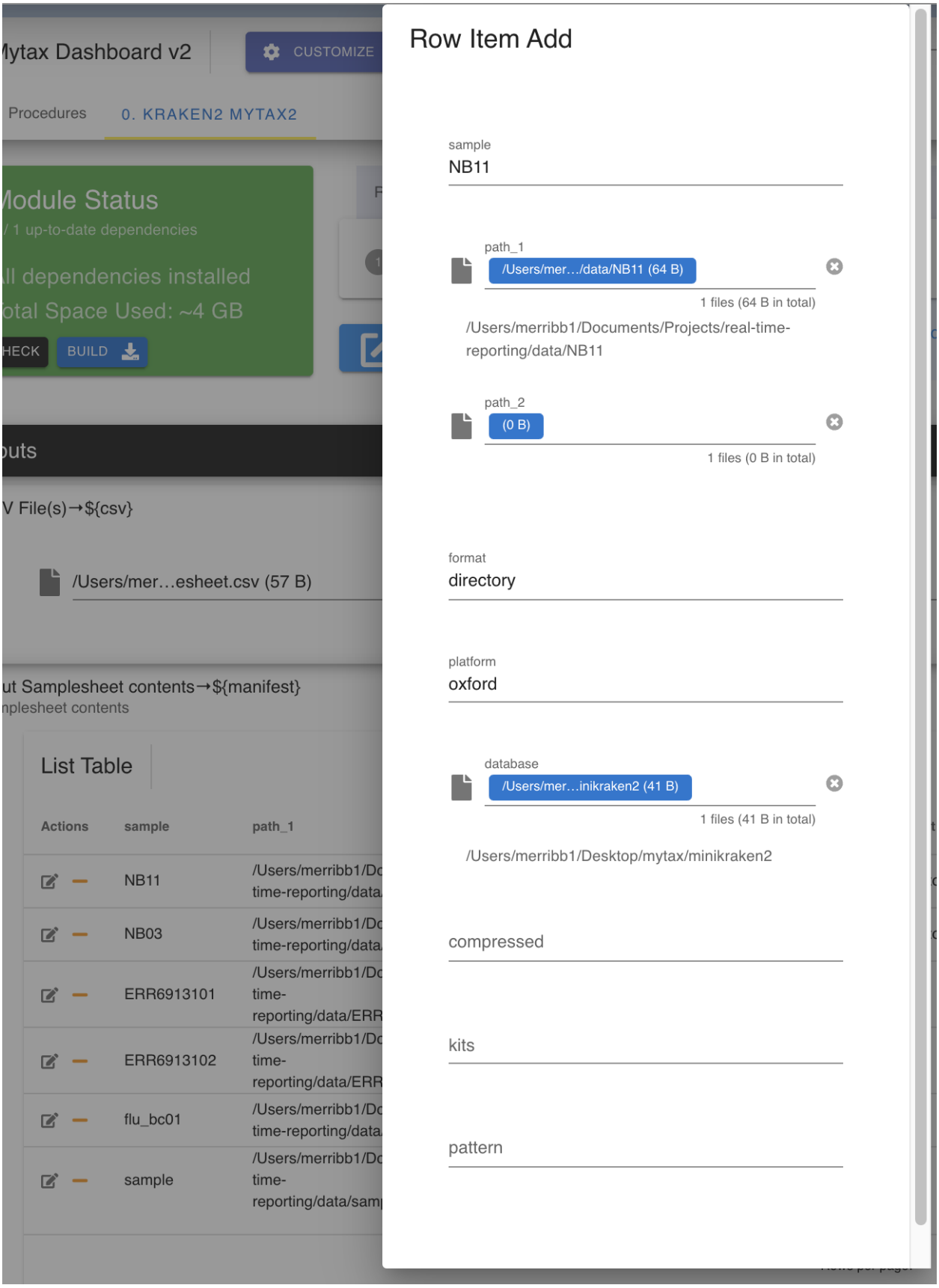
Returns

1. Barcode Folders OPTIONAL. This will only be generated if you input a run in the data sheet and opt to set barcoding (demux) on it from the Data Sheet view
2. Kraken2 Classifiction Reports. Each will be output in the same folder as the sample of interest as full.report. Full.report is the aggregation of all reports for all fastqs attributed to a sample

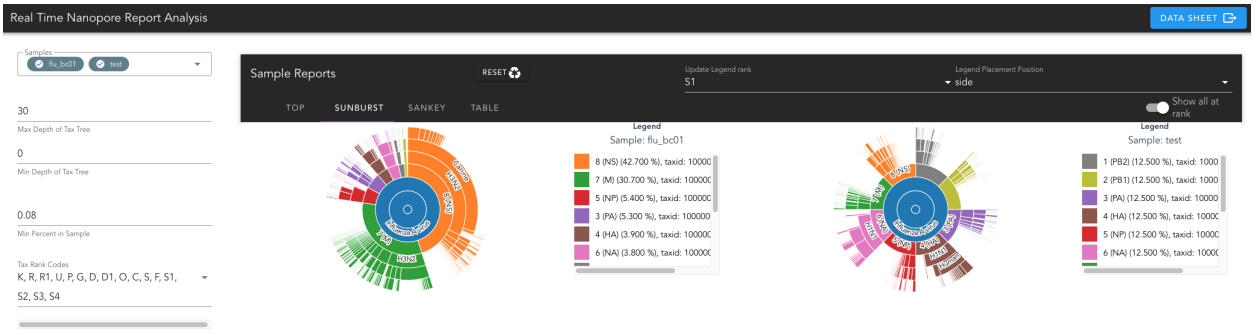
Display



If you need to edit a specific field directly within the UI, you can do so by selecting the pencil icon or adding a New Item


Mytax Dashboard v2
CUSTOMIZE
Procedures
0. KRAKEN2 MYTAX2
Module Status
/ 1 up-to-date dependencies
ll dependencies installed
otal Space Used: ~4 GB
HECK
BUILD
puts
V File(s)→${csv}
/Users/mer…esheet.csv (57 B)
ut Samplesheet contents→${manifest}
nplesheet contents
List Table
Actions
sample
path_1
NB11
NB03
ERR6913101
ERR6913102
flu_bc01
sample
Row Item Add
sample
NB11
path_1
/Users/mer…/data/NB11 (64 B)
1 files (64 B in total)
/Users/merribb1/Documents/Projects/real-time-reporting/data/NB11
path_2
(0 B)
1 files (0 B in total)
format
directory
platform
oxford
database
/Users/mer…inikraken2 (41 B)
1 files (41 B in total)
/Users/merribb1/Desktop/mytax/minikraken2
compressed
kits
pattern

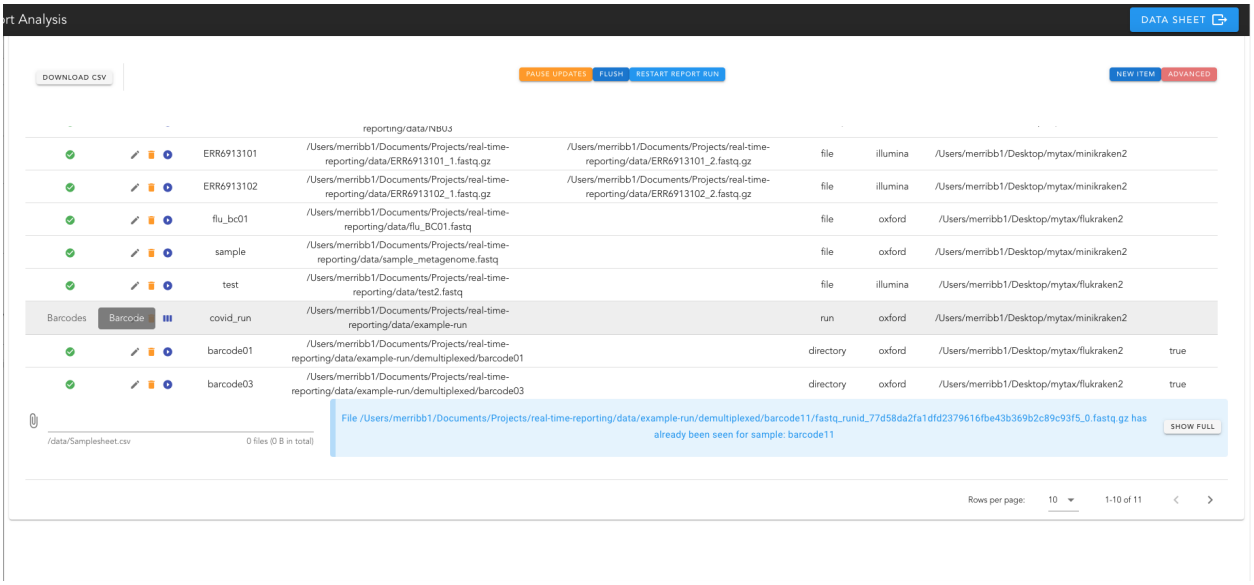
Within the dashboard, you can view all reports being generated in real time as the sunburst, sankey, table, or tops tab. These will be automatically started as soon as you access the website url and basestack has loaded the model as running.



Warning: If the app doesn't start automatically classifying, try to refresh the page

At the top of the page on the right, you can adjust your samplesheet.

Warning: If run with Basestack, this will be limited in scope since you are running on data mounted with Docker. Any changes must be made with Basestack regarding adjusting data that is to be seen not from the Data Sheet directly.



Basestack Consensus

Parameters

Long Read Run Directory
[*Dir*] Run directory from a MinION, GridION, Mk1C, etc. Must contain several files or folders in it. See `manifest`, `sequencing summary` and `fastq_pass` below

primers
[*Text/Dir*] Choice of Custom (directory) or pre-loadable options for primer set used (Artic only supported pre-loaded primer set currently)

barcoding
[*Text/File*] Select barcoding configuration used during demultiplexing. If demultiplexing didn't take place, any are allowed

basecalling
[*Text/File*] Select one of many supported basecalling configurations during the Basecalling step (creating fastq files from fast5)

fastq_pass
[*Dir*] Directory of fastq files (can be demultiplexed or not). All fastq files to be analyzed MUST be decompressed (no .gz or .zip format allowed)

manifest
[*List*] Contains your ID to barcode mapping in a .tsv (tab-separated) format.

Sequencing Summary
[*File*] Not inputable. Ensure that it is in the top-level directory (root, same level as the Run Directory). It is required to run some portions. It is output at the end of every Basecalling step from Guppy

---

Note: Within manifest, one entry must contain the NTC (case-sensitive, no template control). If you don't have a NTC, select NB00 as the barcode and NTC as the id.

Returns

Consensuses
[*./artic-pipeline/4-draft-consensus/*]

- Complete FASTA files will be output as `...complete.fasta for each barcode`
- This folder and other sibling folders will contain other file formats such as `.vcf` and `.bam` for other downstream analysis pipelines.

Report of Run
[*./artic-pipeline/report.pdf*]

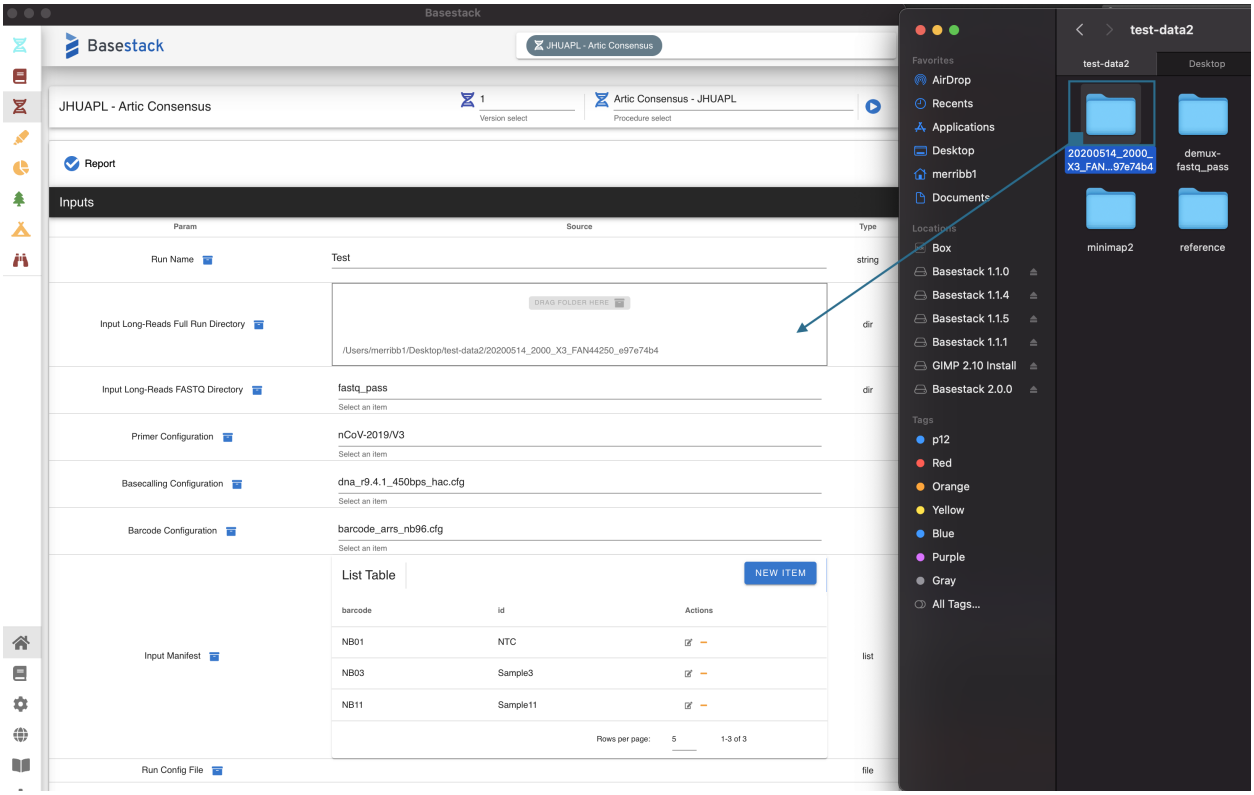
- Contains important information about your samples (each barcode), lineage information, mutations, etc.

Running Consensus Generation and Reporting

Consensus Generation is the main feature of this application and is used to generate a report of a run directory that was generated from a MinION run. It has multiple steps but is designed to be very automated once a job is submit for analysis.

If you haven't already done so, download and unzip the Test data folder

Starting a Run



1. **Select the appropriate test folder first. This folder is either included in the *test-data* folder in the source of this application OR you can retrieve it within the install location of the app. For example, in *C:Program FilesBasestackclientdatatest-data*.**
    - You can either drag + drop it into the *Run Folder* field or select it by left clicking and browser to the directory location on your computer
2. Once a folder has been input, you should see the text fields populate and turn green. If any field is marked as read, it is invalid and should be looked at further for proper formatting. These files should be valid for the test dataset. If you want to use your own data please follow the formatting in *run_config.txt*, *run_info.txt*, and *manifest.txt*. These formats are like:

Note: You may skip this portion if you'd like

---

1. `run_config.txt`

- This should be 3 rows that dictate the primer (first), basecalling workflow (second), and barcoding cfg (third/last). Separate by tab.

Example:

Target (Unlisted)	Config (Unlisted)
primers	nCov-2019/V3
basecalling	dna_r9.4.1_450bps_hac.cfg
barcoding	barcode_arrs_nb96.cfg

2. `manifest.txt`

Example:

Barcode	Sample
NB01	NTC (always required)
NB02	MDHP-00058
NB03	MDHP-00059
.	.
.	.

Note: If you don't have an NTC (NOT RECOMMENDED EXPERIMENTALLY), set `NTC` as `NB00`

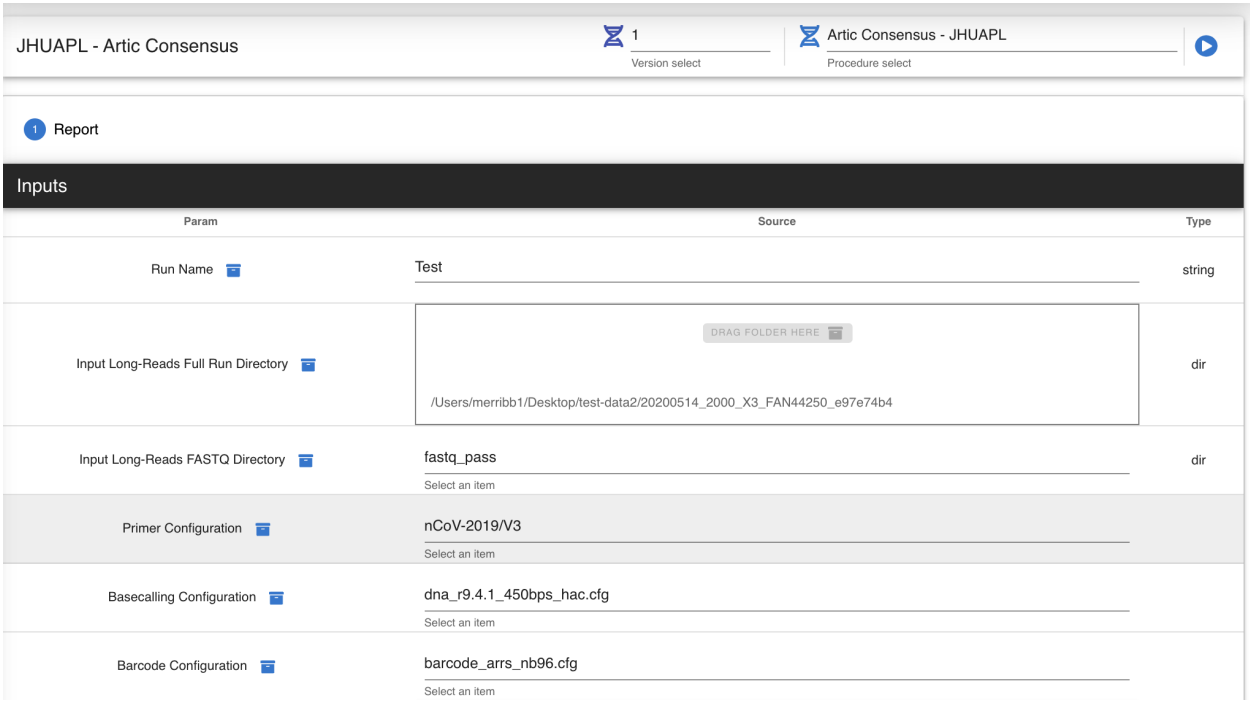
- This should be any number of rows that contain barcode on the left ALWAYS and the sample code on the right. A no-template-control (NTC) must always be specified for a report to be completed. Separate by tab.
- **You are allowed to input your own custom values for each of the 3 files where the app will overwrite that corresponding file on a job submission. That means you can populate these fields by either directly modifying the files OR by inputting them into the input fields**
  - These values are currently not validating to their greatest extent so take care to correctly input values and delimit them with tabs if doing this manually.

Lastly, there are three files that are made following a successful sequencing (and basecalling) run. These three are

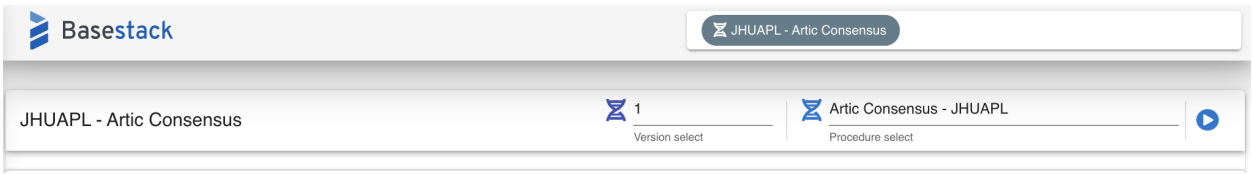
1. **Sequencing Summary REQUIRED**
   - This file is made following **basecalling**. It contains the mapping and summary stats of all fast5 to fastq generations and must be present in the run directory for report generation
   - If using CLI or stand-alone **basecalling** you will likely need to move this file from the fastq output savepath **to** the base run directory.
2. Throughput....csv **OPTIONAL**
3. Drift Correction **OPTIONAL**

Note: Future updates of Basestack will prevent the job from commencing if the sequencing summary is not present

Starting the process

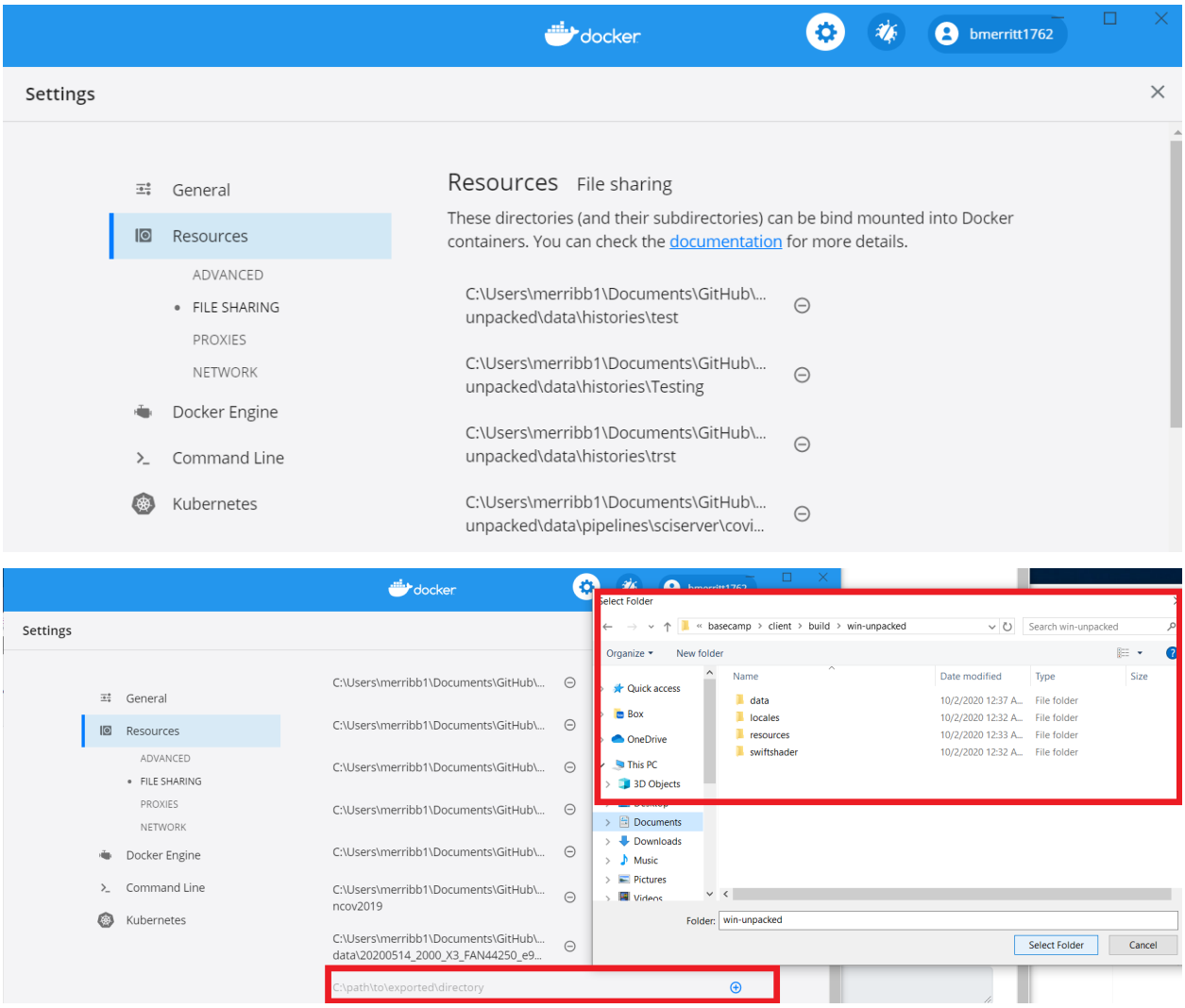


1. Once everything is staged, you should see all items update accordingly based on information in the directory.



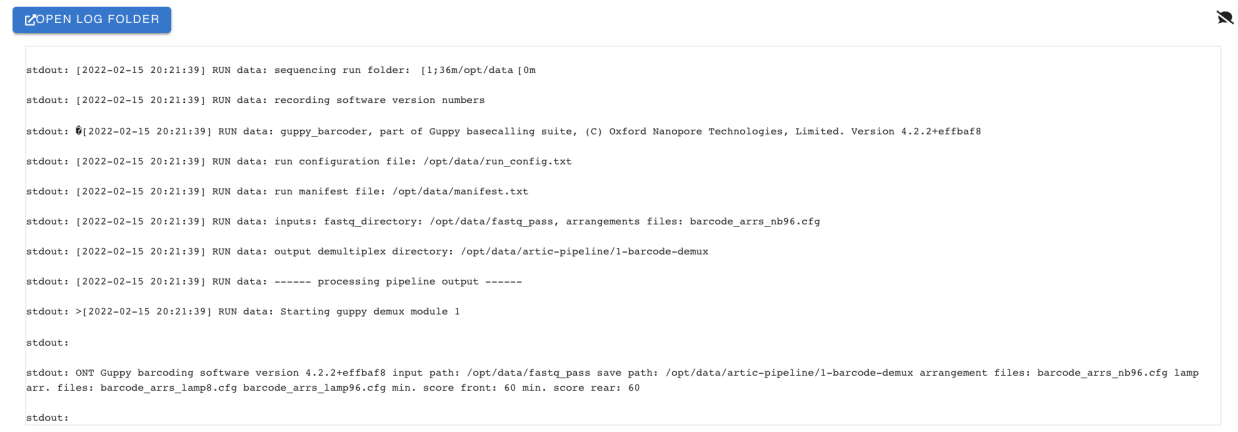
2. Hit *Start* in the upper right-hand corner to start consensus generation.

Note: Depending on your method of installing Docker on Windows, you may receive a notification for docker to share a folder. Hit okay to allow the pipeline to continue. If you run Basestack as an admin, this error will be avoided. You can also opt to share the Basestack folder and sub-folders in the Docker Desktop on Windows as well (see how to do this in the next 2 images)



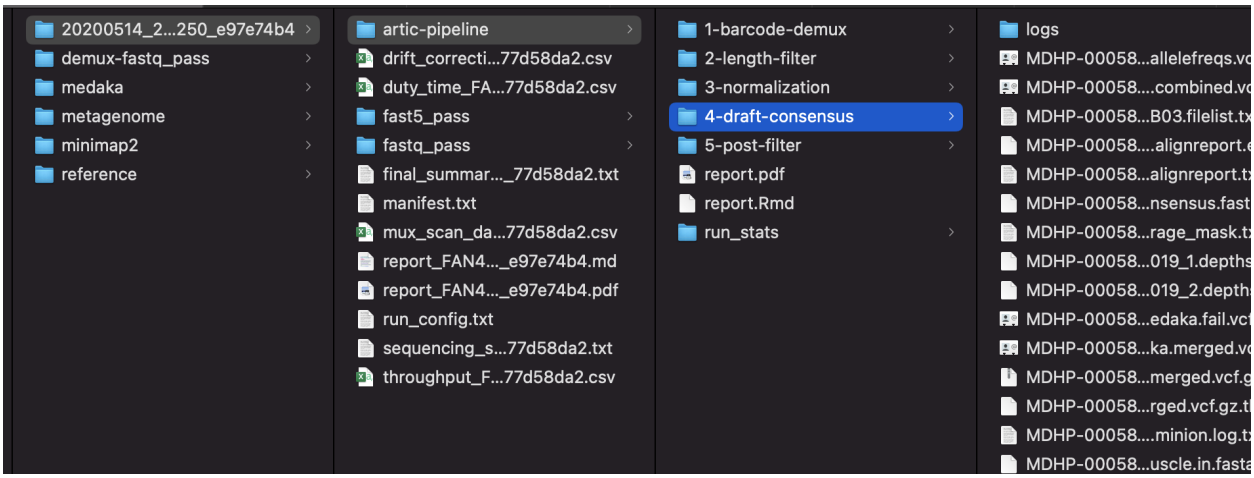
Note: Simply select the folder that contains the *Basestack.exe* file by selecting the plus-mark and navigation and selecting it within the browser. In this example it is: *. . . buildwin-unpacked*

Checking Logs and Status



Note: You can see the output of the run in the *Log Window* container on the bottom of the page. You can also see the *Output(s)* table begin to change as modules are completed for your run. The final module is the report generation module and should always be *1/1* when complete

Final Report



Note: Once complete, you can view the pdf report by clicking the *pdf* file icon link underneath the final row's status of *1/1*. You can also traverse to any of the module directories by hitting the link text on the first column for each module. In this example, I've chosen *Report Generation* as my link which is a top-level view of all modules, as well as the *report.pdf* location. Open this pdf to see your report either from the folder or the *pdf* link on the left-most column to see your results!

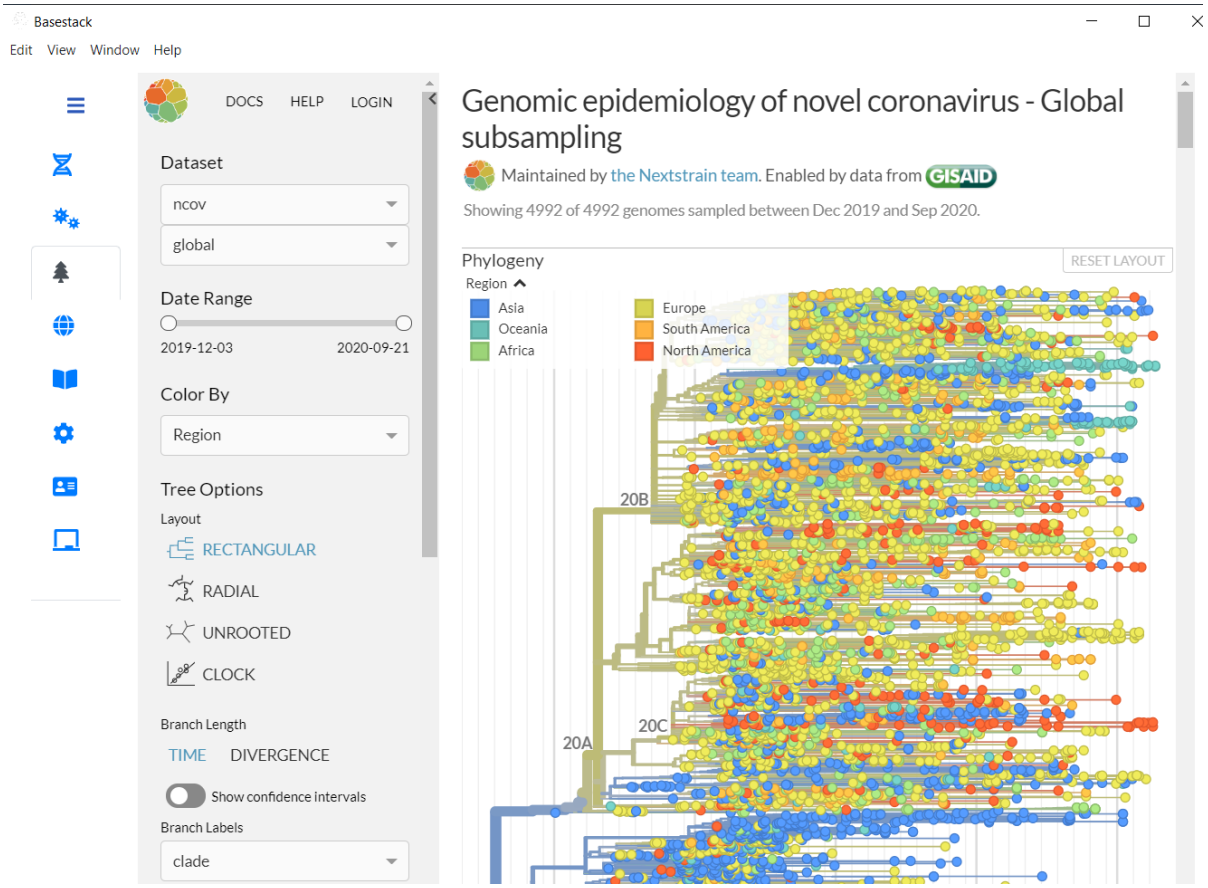
Nextstrain

After the app has generated consensus genomes from the previous 2 commands, you can view mutations or SNPs in your run by selecting the Nextstrain tab at the left-hand side. These mutations can be viewed around `Page 11` or `section 5.3`

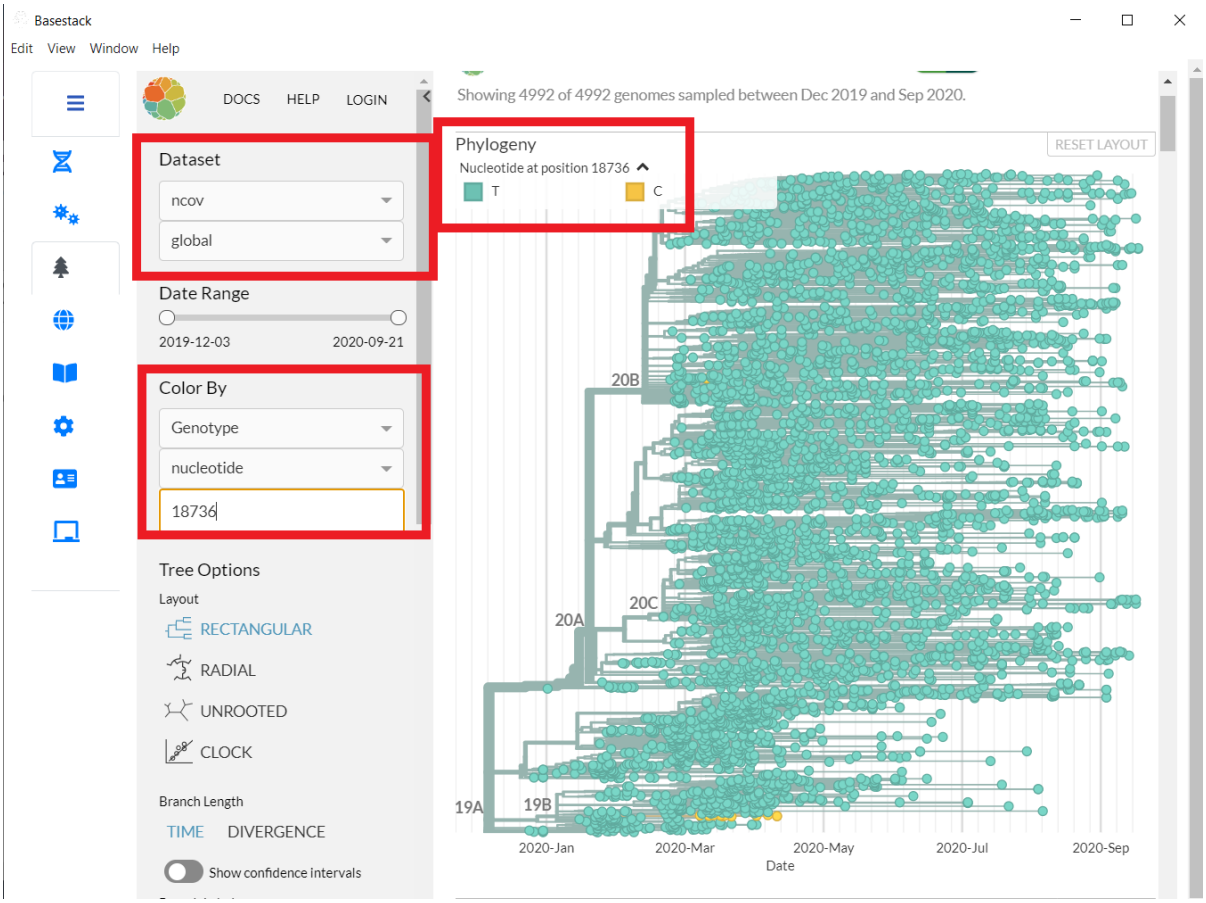
Let's go into our report we just generated and traverse to `section 5.3` or `page 10-11`. The tables provided are all reported mutations against the reference.

Note: Page and Section references refer to reports generated from basestack_consensus

---



- Position 18736 (for example) is a reported SNP for T to C mutation. The annotatin is a missense_variant (see all annotation types in the description of this figure for the report). Though nextstrain we can view this mutation across all samples available that have been sequences and input into nextstrain's website.
- Now that we've selected our mutation to view let's move forward through the interactive site



To View a specific mutation (and this one for example) reported in the table, select underneath **Color By**

Genotype -> nucleotide -> [Your number here]

Also, be sure that the dataset is ncov and global. You can change the layout of the visualization(s) with the Tree Options parameters

---

Note: Requires Internet. Also available at https://nextstrain.org/ncov/global

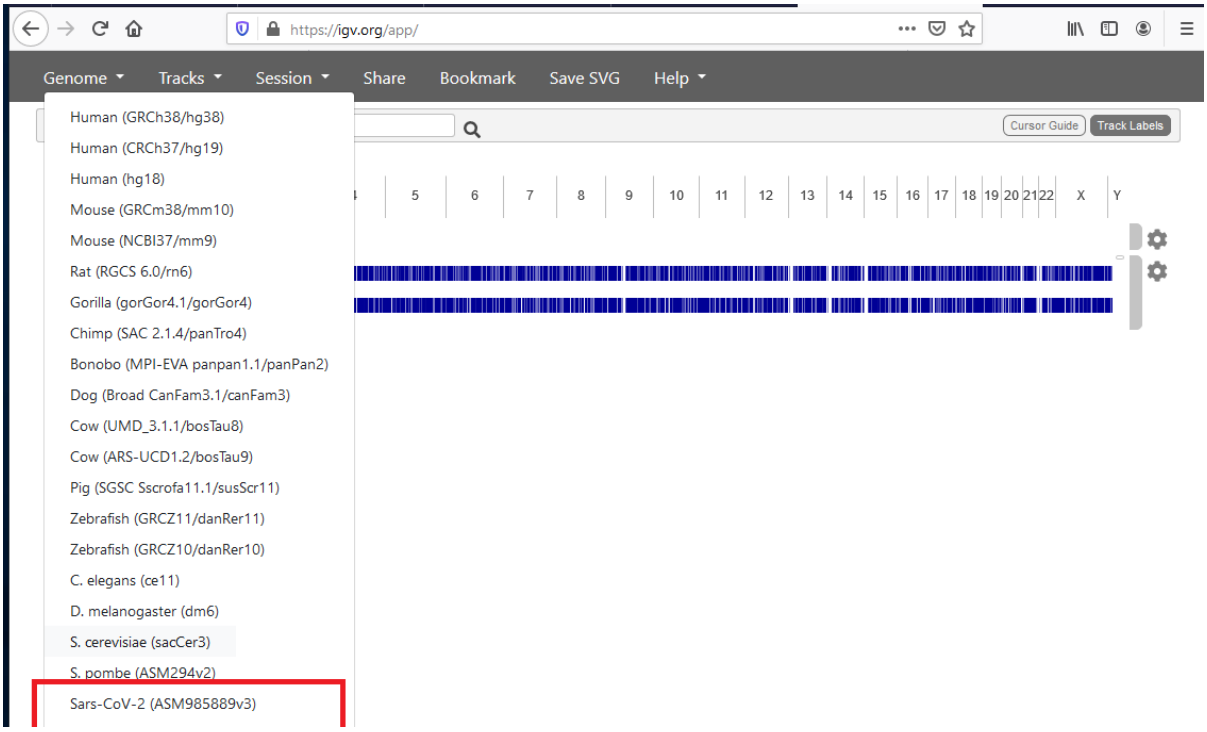
---

IGV

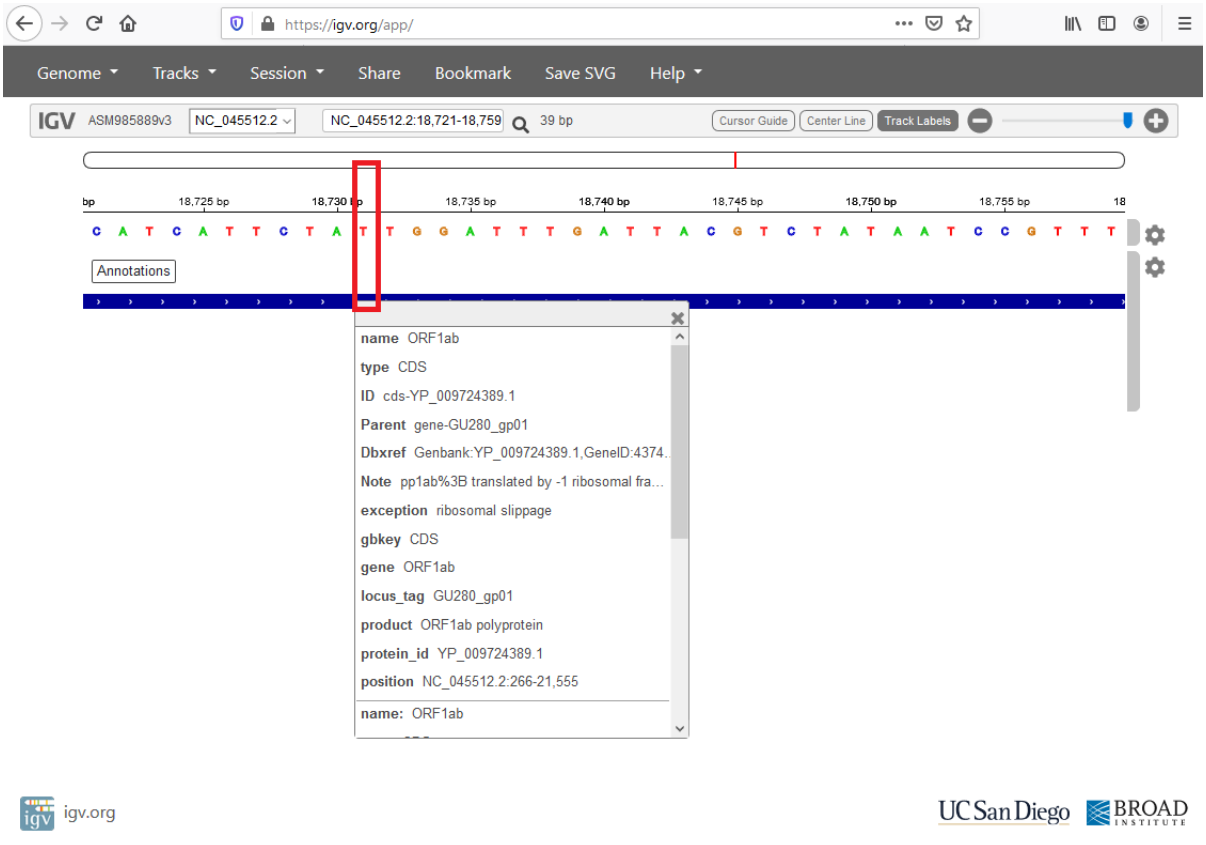
Note: Requires Internet. Also available at https://igv.org/app/

IGV is an interactive environment that allows you to view a genome and see any annotations at a specific position.

These plots are also available in the Consensus Generation Pipeline Reports



1. First, we are going to select our genome of interest, Sars-CoV-2.



2. For our example at position 18736, we can see that it belongs to ORF1ab from the reference and it has a T. As we saw in the report that T was change to a C when compared to the reference. We can also see what protein this

position is a part of as well as supplemental information like the *protein_id* or *Dbxref* (references for external database resource)

GAMMA

GAMMA is a tool designed to analyze gene allele mutations for microbes, primarily bacterial genomes.

Note: This module outputs only a text file for analysis

Parameters

FASTA file
[*File*] FASTA or genome file to check for AMR alleles

output
[*String*] Text string to have as a prefix for the output file(s)

Gene Database file
[*File*] `.fsa` file that contains the mapping of genes linked to AMR

Returns

GAMMA file : *Tab-separated file*

Note: Below output gathered from here

The default output of GAMMA is a tab-delimited file with a ".gamma" extension with 15 columns:

1. Gene – The name of the closest matching gene (target) from the database. If there are ambiguous gene matches (i.e., multiple target matches with the same number of non-degenerate codon changes, basepair changes, and transversions), the gene match will be appended with a "‡".
2. Contig – The name of the contig on which the match was found.
3. Start – The start position of the sequence matching the gene on the contig.
4. Stop – The end position of the sequence matching the gene on the contig.
5. Match_Type – The type of the gene match based on the translation of the sequence (i.e., the protein sequence). Can be native (for identical amino acid sequences to the target), mutant (for nonsynonymous mutations), truncation (for nonsense mutations), indels (for insertions/deletions), nonstop (for a missing stop codon), contig edge (for matches that are truncated at the start or stop of a contig), or a combination of multiple types (i.e., indel truncation).
6. Description – A short description of the match type.
7. Codon_Changes – The count of the non-degenerate codon changes in the sequence versus the closest match from the datbase.
8. BP_Changes - The count of the basepair changes in the sequence versus the closest match from the datbase.
9. Transversions - The count of basepair changes that are transversions (i.e., purine to pyrimidine or vice versa, such as an A -> C or a T -> G)
10. Codon_Percent – The percent (expressed as a decimal value) of the degenerate codon similarity between the query and match sequence. Gene matches with large insertions may show a negative value.

11. BP_Percent - The percent (expressed as a decimal value) of the basepair similarity between the query and match sequence. Gene matches with large insertions may show a negative value.
12. Percent_Length - The percent (expressed as a decimal value) of the length of the target covered by the matching sequence, maximum of 1.
13. Match_Length – The length (in basepairs) of the matching sequence.
14. Target_Length - The length (in basepairs) of the target sequence.
15. Strand – The sense of the strand (+ or -) on which the match is found.

Rampart

Rampart is an annotation tool provided by the Artic Network that gives quick, but less accurate reports for each of your demuxed samples. It is ideal for a quick look into what the annotations for each of your barcodes are but should not be fully relied on, instead opt for the pipeline that was just discussed.

Note: This module contains a UI to be displayed from the rendering button in the variables table

---

Parameters

Fastq Dir
[*Dir*] Run annotation of SARS-nCoV-2 sequences within the RAMPART suite

Annotated Clear
[*Option*] Remove your annotated folder if you want to do a fresh run. Annotations will be removed from the same directory as the `Fastq Dir`

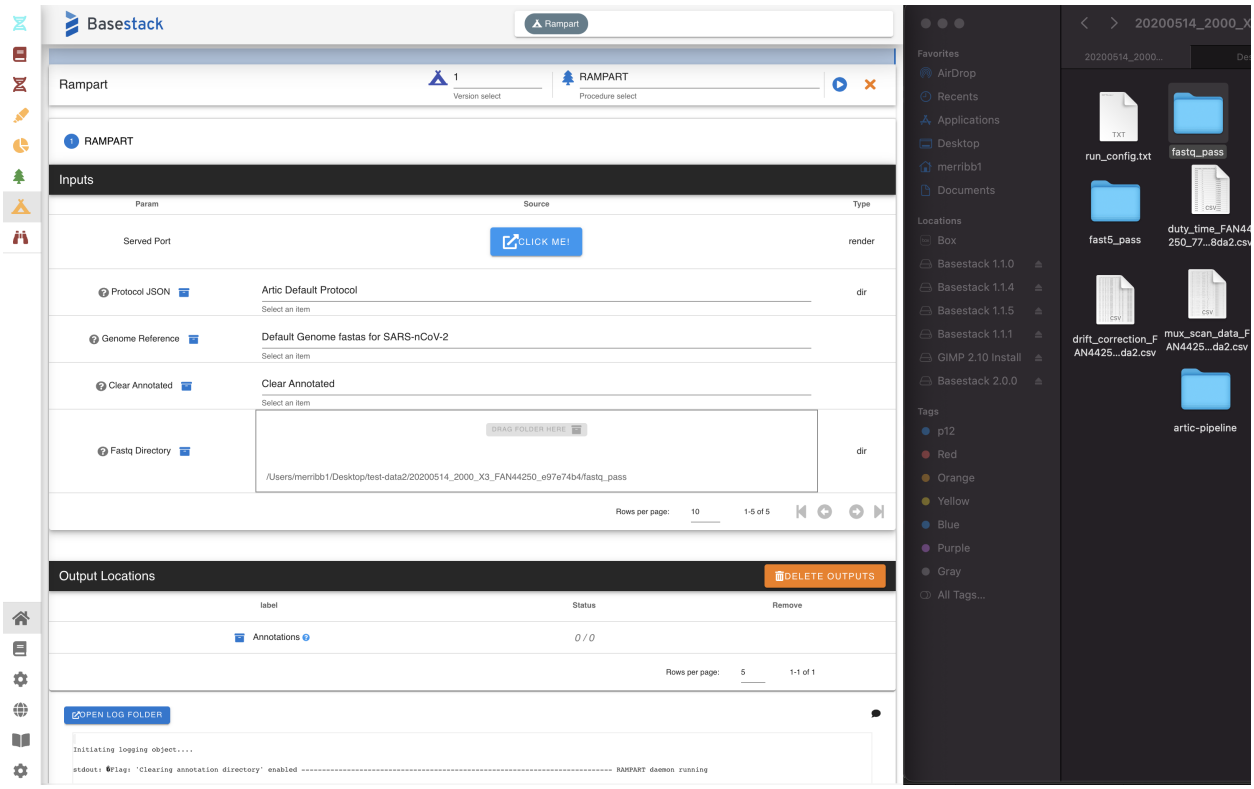
Returns

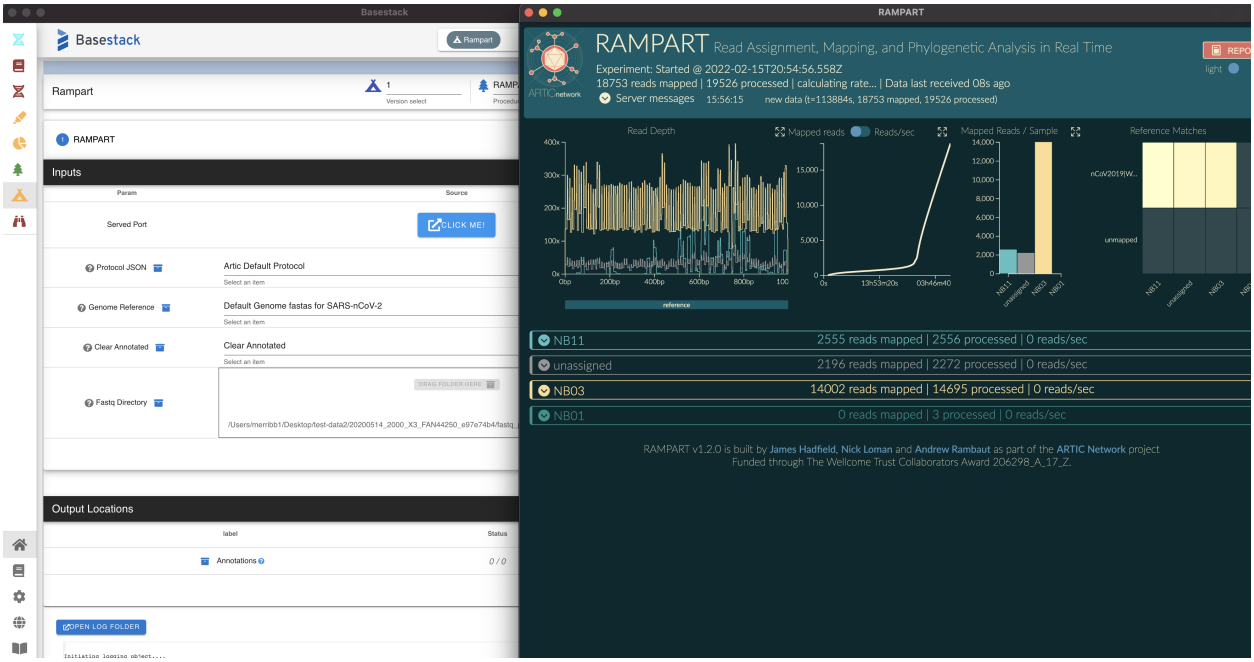
Annotations
[*./annotations*]

- Select `Click Me!` button to render RAMPART in the UI

---

1. Ensure you've loaded a run that has been bookmarked from the previous tab (described above).
2. Select one of the included primer-schemes from the drop-down list. For this example, the data is ncov-related so we will choose `Default Genome fasta for SARS-nCoV-2`.
3. Select the `Click Me!` button to render Rampart in a new window


Basestack
Rampart
RAMPART
Inputs
Param
Source
Type
Served Port
CLICK ME!
render
Protocol JSON
Artic Default Protocol
dir
Genome Reference
Default Genome fastas for SARS-nCoV-2
Clear Annotated
Fastq Directory
/Users/merribb1/Desktop/test-data2/20200514_2000_X3_FAN44250_e97e74b4/fastq_pass
Output Locations
DELETE OUTPUTS
label
Status
Remove
Annotations
0 / 0
OPEN LOG FOLDER
Initiating logging object....
RAMPART daemon running
Favorites
AirDrop
Recents
Applications
Desktop
Documents
Locations
Box
Basestack 1.1.0
Basestack 1.1.4
Basestack 1.1.5
Basestack 1.1.1
GIMP 2.10 Install
Basestack 2.0.0
Tags
p12
Red
Orange
Yellow
Blue
Purple
Gray
All Tags...
run_config.txt
fastq_pass
fast5_pass
artic-pipeline


Basestack
RAMPART
Rampart
Read Assignment, Mapping, and Phylogenetic Analysis in Real Time
Experiment: Started @ 2022-02-15T20:54:56.558Z
18753 reads mapped | 19526 processed | calculating rate... | Data last received 08s ago
Server messages
new data (t=1138845, 18753 mapped, 19526 processed)
Read Depth
Mapped reads
Reads/sec
Mapped Reads / Sample
Reference Matches
NB11
2555 reads mapped | 2556 processed | 0 reads/sec
unassigned
2196 reads mapped | 2272 processed | 0 reads/sec
NB03
14002 reads mapped | 14695 processed | 0 reads/sec
NB01
0 reads mapped | 3 processed | 0 reads/sec
RAMPART v1.2.0 is built by James Hadfield, Nick Loman and Andrew Rambaut as part of the ARTIC Network project
Funded through The Wellcome Trust Collaborators Award 206298_A_17_Z.

FastQC

FastQC is a tool designed to view the quality and do control on raw sequencing data, primarily for illumina reads

Note: This module outputs an html file that is a report of your raw sequencing data

Parameters

FASTQ-containing Directory
[*Directory*] Directory that contains one or more `fastq` files

Returns

Report file : *HTML file*

FastQC Report

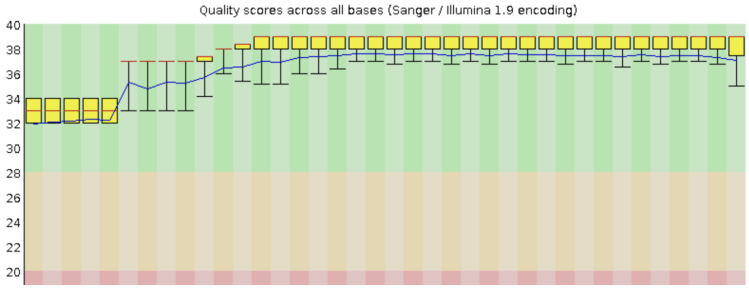
Summary

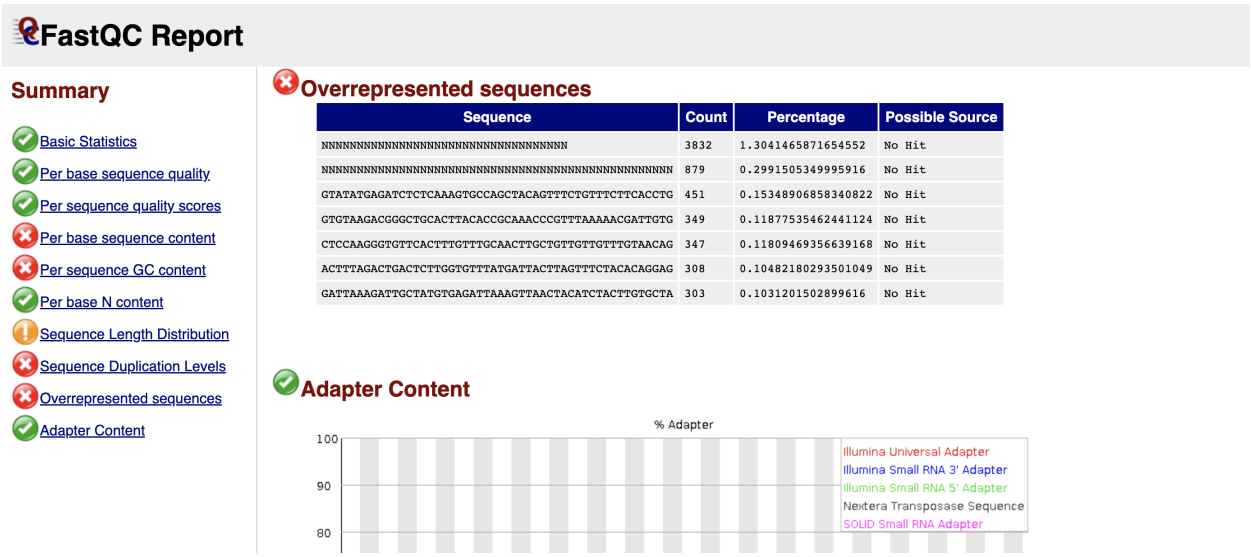
- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	full.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	293832
Sequences flagged as poor quality	0
Sequence length	35-151
%GC	38

Per base sequence quality





FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	3832	1.3041465871654552	No Hit
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	879	0.2991505349995916	No Hit
GTATATGAGATCTCTCAAAGTGCCAGCTACAGTTTCTGTTTCTTCACCTG	451	0.15348906858340822	No Hit
GTGTAAGACGGGCTGCACTTACACCGCAAACCCGTTTAAAAACGATTGTG	349	0.11877535462441124	No Hit
CTCCAAGGGTGTTCACTTTGTTTGCAACTTGCTGTTGTTGTTTGTAACAG	347	0.11809469356639168	No Hit
ACTTTAGACTGACTCTTGGTGTTTATGATTACTTAGTTTCTACACAGGAG	308	0.10482180293501049	No Hit
GATTAAAGATTGCTATGTGAGATTAAAGTTAACTACATCTACTTGTGCTA	303	0.1031201502899616	No Hit

Adapter Content

NanoPlot

NanoPlot is a tool designed to view the quality and do control on raw sequencing data from long-read runs

Note: This module outputs an html file that is a report of your raw sequencing data

Parameters

- **input file:** *file*
  Can be one of these list:
  1. **fastq file [file ...]**
     Data is in one or more default fastq file(s).
  2. **fasta file [file ...]**
     Data is in one or more default fasta file(s).
  3. **fastq_rich file [file ...]**
     Data is in one or more fastq file(s) generated by albacore or MinKNOW with additional information concerning channel and time.
  4. **fastq_minimal file [file ...]**
     Data is in one or more fastq file(s) generated by albacore or MinKNOW with additional information concerning channel and time. Minimal data is extracted swiftly without elaborate checks.
  5. **summary file [file ...]**
     Data is in one or more summary file(s) generated by albacore or guppy.
  6. **bam file [file ...]**
     Data is in one or more sorted bam file(s).

Note: Input information taken from the `--help` output of `NanoPlot`

- **barcoding file:** ***optional - file***
  Must only be given when using the `summary file` option

Returns

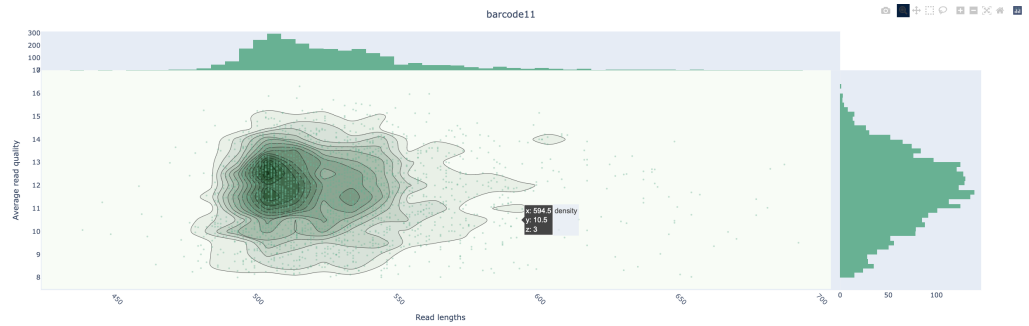
Report file : *HTML file*

Summary Statistics | Plots | Report issue on Github

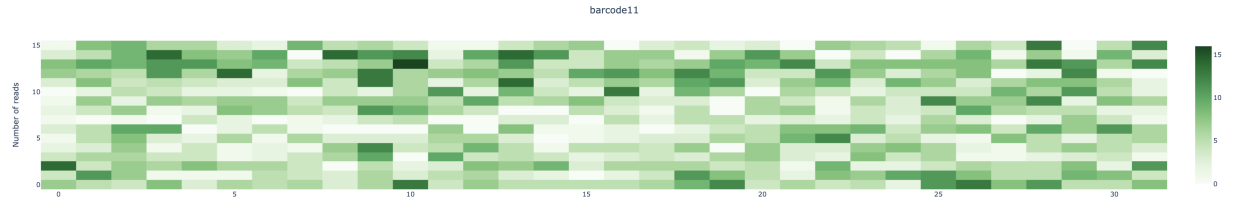
NanoPlot reports

Summary statistics

General summary	barcode11	unclassified	barcode03	barcode01	None	None	None
Active channels	479.0	490.0	500.0	3.0	None	None	None
Mean read length	526.4	495.5	513.3	422.7	None	None	None
Mean read quality	11.7	12.2	12.4	10.8	None	None	None
Median read length	519.0	492.0	508.0	423.0	None	None	None
Median read quality	11.8	12.3	12.4	10.8	None	None	None
Number of reads	2,683.0	4,192.0	12,648.0	3.0	None	None	None
Read length N50	521.0	493.0	508.0	423.0	None	None	None
STDEV read length	29.9	23.1	33.6	7.8	None	None	None
Total bases	1,412,359.0	2,077,197.0	6,492,511.0	1,268.0	None	None	None
Number, percentage and megabases of reads above quality cutoffs					None	None	None
>Q5	2683 (100.0%) 1.4Mb	4192 (100.0%) 2.1Mb	12648 (100.0%) 6.5Mb	3 (100.0%) 0.0Mb	None	None	None
>Q7	2683 (100.0%) 1.4Mb	4192 (100.0%) 2.1Mb	12648 (100.0%) 6.5Mb	3 (100.0%) 0.0Mb	None	None	None
>Q10	2278 (84.9%) 1.2Mb	3888 (92.7%) 1.9Mb	12073 (95.5%) 6.2Mb	2 (66.7%) 0.0Mb	None	None	None
>Q12	1199 (44.7%) 0.6Mb	2426 (57.9%) 1.2Mb	7933 (62.7%) 4.1Mb	0 (0.0%) 0.0Mb	None	None	None
>Q15	33 (1.2%) 0.0Mb	93 (2.2%) 0.0Mb	281 (2.2%) 0.1Mb	0 (0.0%) 0.0Mb	None	None	None
Top 5 highest mean basecall quality scores and their read lengths					None	None	None



Number of reads generated per channel



Viral Recon

nf-core/viralrecon is a bioinformatics analysis pipeline used to perform assembly and intra-host/low-frequency variant calling for viral samples. The pipeline supports both Illumina and Nanopore sequencing data. For Illumina short-reads the pipeline is able to analyse metagenomics data typically obtained from shotgun sequencing (e.g. directly from clinical samples) and enrichment-based library preparation methods (e.g. amplicon-based: ARTIC SARS-CoV-2 enrichment protocol; or probe-capture-based). For Nanopore data the pipeline only supports amplicon-based analysis obtained from primer sets created and maintained by the ARTIC Network.*

*Pulled from [https://nf-co.re/viralrecon{}](https://nf-co.re/viralrecon)

Note: The modules runs nextflow on the backend and thus utilizes Docker within Docker.

Parameters

Fastq Dir
[*Dir*] Basecalled Fastq files

Fast5 Dir
[*Dir*] Fast5 files directory from which you received the basecalled fastq directory of files from

Returns

Consensus
[*./viralrecon/medaka|nanopolish*]

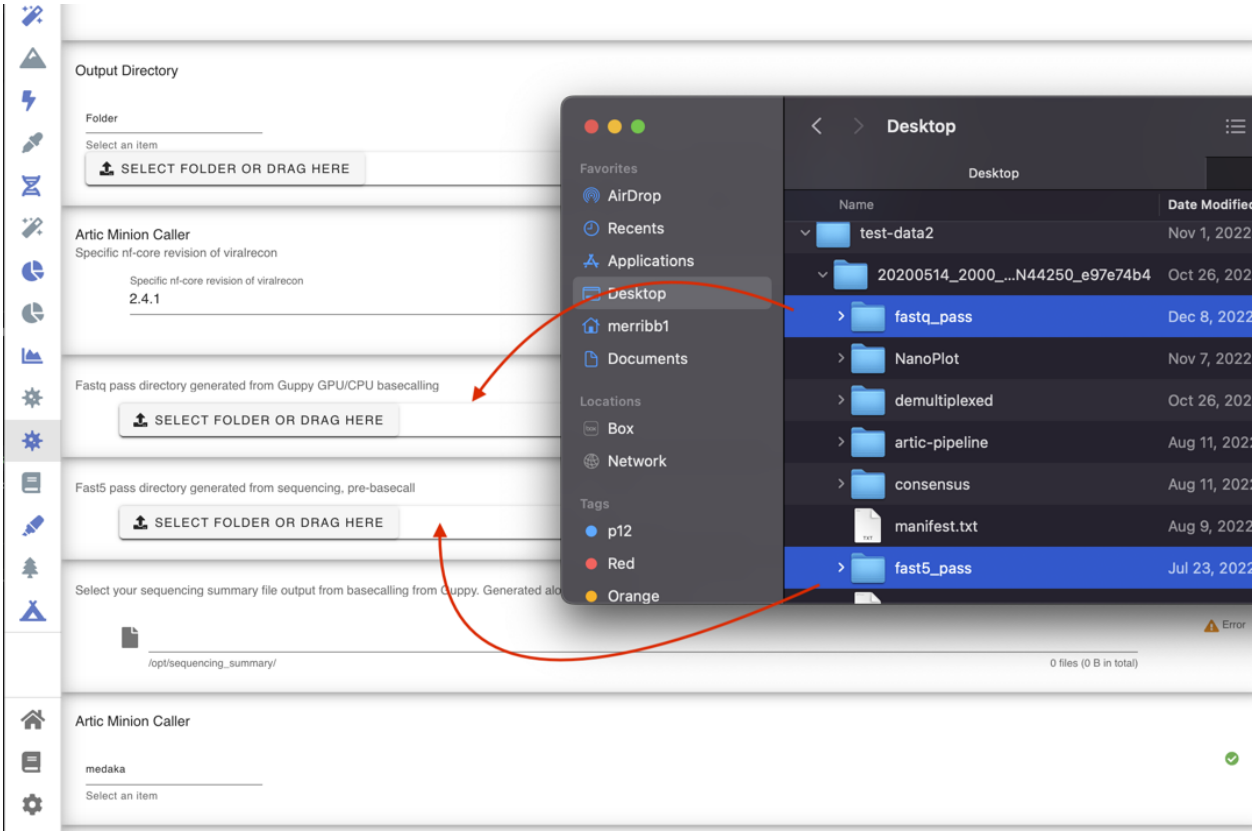
- Consensus FASTA files are made for both assembly processes

MultiQC Report: ***./viralrecon/multiqc/multiqc_report.html***

- HTML files that has information of your run

---

1. Ensure you've loaded a run with a fastq and fast5 directory specified



2. Select one of the included primer-schemes from the drop-down list. For this example, the data is ncov-related so we will choose `Default Genome fasta for SARS-nCoV-2`.
3. Select one of the basecaller options *medaka* or *nanopolish*
4. Select the `Play button` button to start the pipeline

Bamstats

Bamstats is a tool designed compute mapping stats from a BAM file, often output from alignment methods

Note: This module outputs a zipped folder that contains your necessary information

Parameters

- input BAM: *file*

  This is often output from alignment methods like `bwa`, `bowtie2`, or `minimap2`. Additionally, you can receive BAM files from modules within Basestack such as:

  1. *Basestack Consensus*
  2. *Medaka*
  3. *Nanopolish*

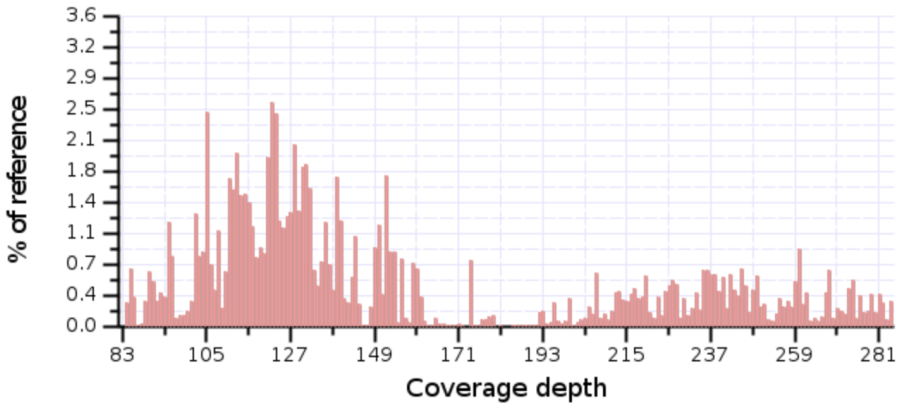
Returns

Bamstats Report zipped : *bamstats_report.zip, HTML-containing directory*

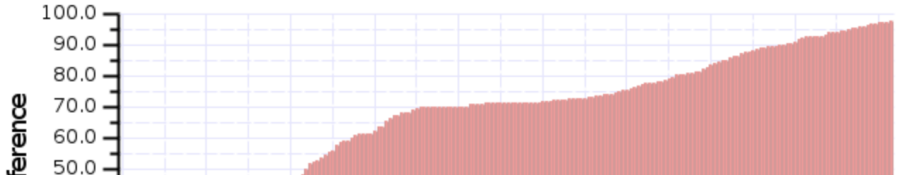
MN908947.3: Coverage distribution

Id	N	Mean	Median	sd	q1	q3	2.5% percentile	97.5% percentile	Min	Max
MN908947.3	29,903	157.91	132.00	60.61	115.00	214.00	84.00	284.00	0.00	320.00

Histogram



Cumulative histogram



Artic Field Bioinformatics

Artic is a tool designed for analyzing and creating consensuses from viral nanopore sequencing reads. These also are tied to the use of amplicon schemes (tiled)

Note: This module will output multiple consensuses per barcode of interest. It takes in a raw fastq_pass directory and outputs *.vcfs*, *.fasta*, and *.bam* files for further use

---

Medaka

Parameters

- input run folder: *directory*

  Must contain a `fastq_pass` directory or a custom inputted one (See below)

- Primer: *option or Directory*

  Artic Primer set or a custom one (directory) which contain the `genome.fasta` and necessary `BED` file for the primer set

- Normalize Coverage: *Number*

- Barcode Configuration: *option*

  Which barcode kit you used for demux. Select any for non-barcoded sample

- FASTQ Dir: *Directory, optional*

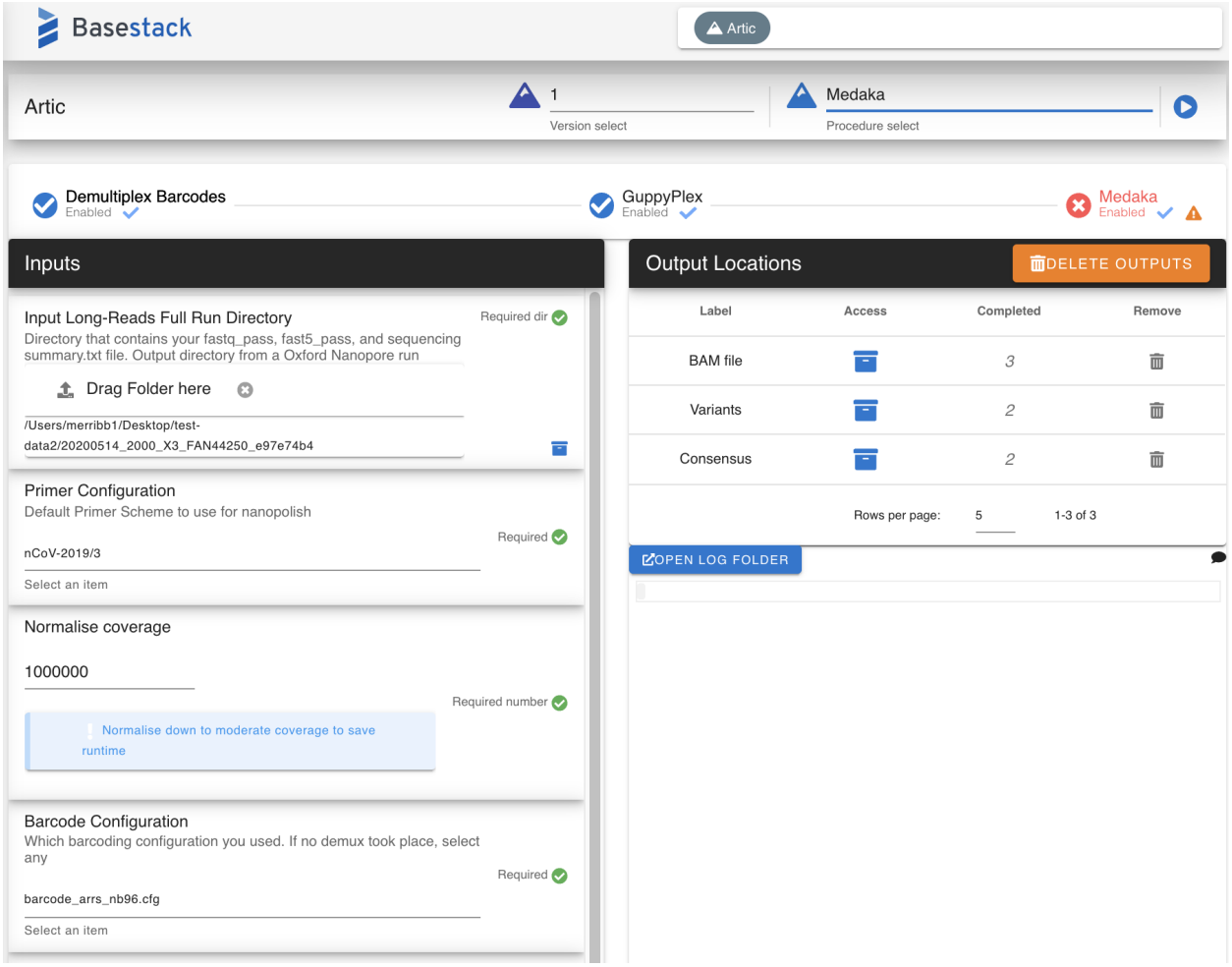
  Select your own custom *fastq_pass* directory to analyze and demux

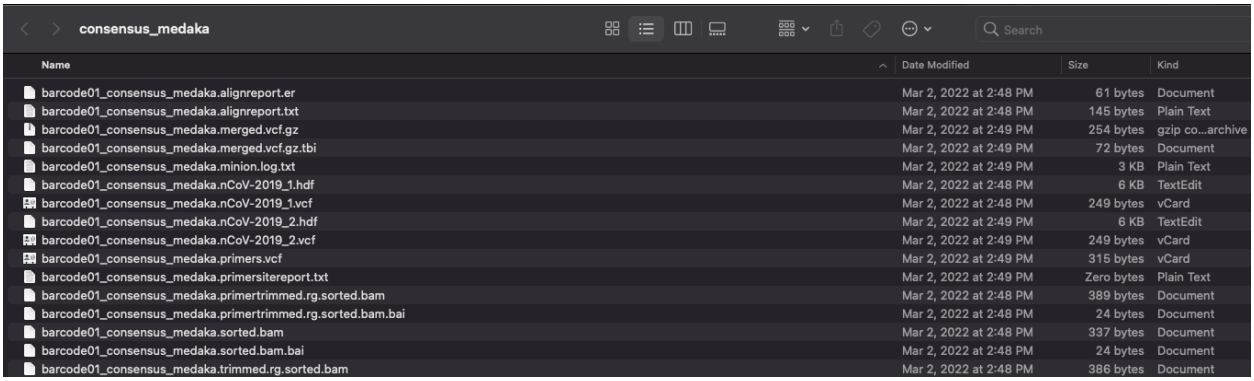
- Medaka Model: *option*

  FAST or HAC used during basecalling

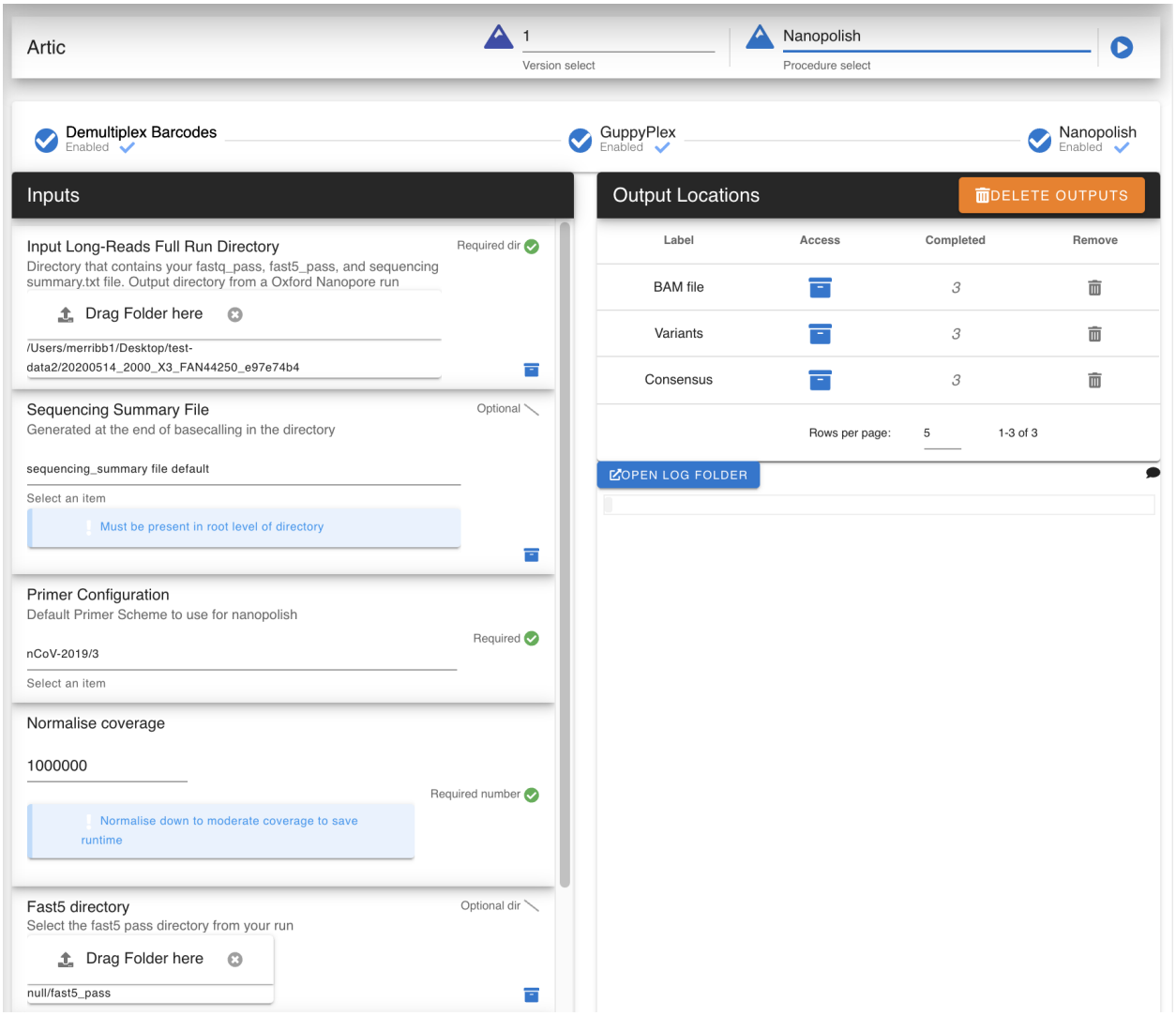
Returns

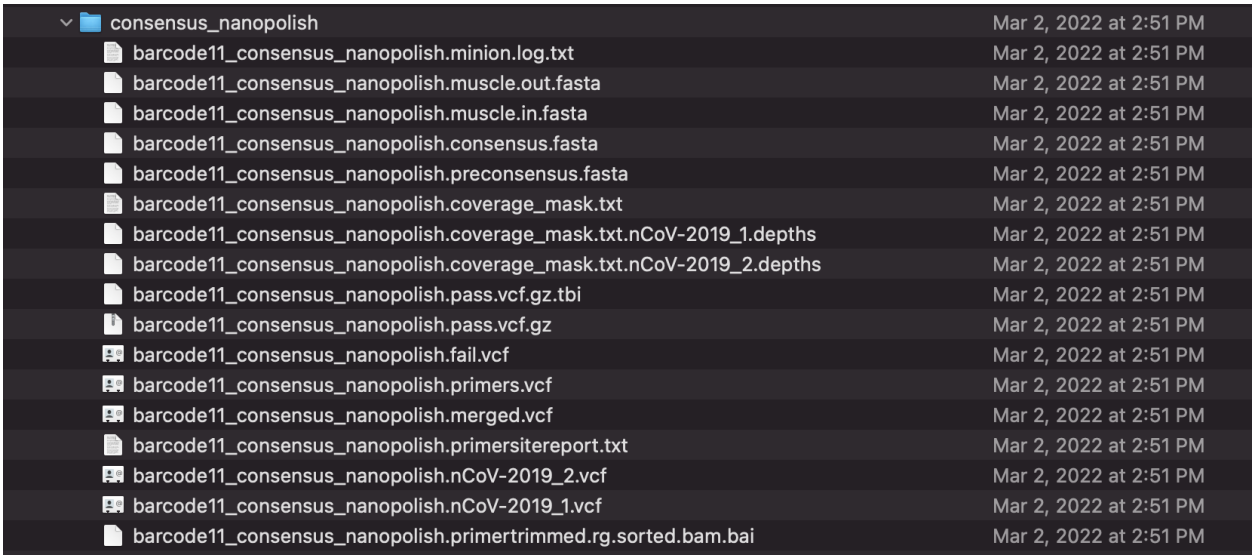
1. Medaka Consensus files: *FASTA files containing your consensuses for each barcode*
2. Medaka VCF files: *Variant files containing your variant calls for each barcode*
3. Medaka BAM files: *BAM files containing your alignment information for each barcode*


Basestack
Artic
Artic
1
Version select
Medaka
Procedure select
Demultiplex Barcodes
Enabled
GuppyPlex
Enabled
Medaka
Enabled
Inputs
Input Long-Reads Full Run Directory
Directory that contains your fastq_pass, fast5_pass, and sequencing summary.txt file. Output directory from a Oxford Nanopore run
Required dir
Drag Folder here
/Users/merribb1/Desktop/test-data2/20200514_2000_X3_FAN44250_e97e74b4
Primer Configuration
Default Primer Scheme to use for nanopolish
Required
nCoV-2019/3
Select an item
Normalise coverage
1000000
Required number
Normalise down to moderate coverage to save runtime
Barcode Configuration
Which barcoding configuration you used. If no demux took place, select any
Required
barcode_arrs_nb96.cfg
Select an item
Output Locations
DELETE OUTPUTS
Label
Access
Completed
Remove
BAM file
3
Variants
2
Consensus
2
Rows per page:
5
1-3 of 3
OPEN LOG FOLDER


consensus_medaka
Search
Name
Date Modified
Size
Kind
barcode01_consensus_medaka.alignreport.er
Mar 2, 2022 at 2:48 PM
61 bytes
Document
barcode01_consensus_medaka.alignreport.txt
Mar 2, 2022 at 2:48 PM
145 bytes
Plain Text
barcode01_consensus_medaka.merged.vcf.gz
Mar 2, 2022 at 2:49 PM
254 bytes
gzip co...archive
barcode01_consensus_medaka.merged.vcf.gz.tbi
Mar 2, 2022 at 2:49 PM
72 bytes
Document
barcode01_consensus_medaka.minion.log.txt
Mar 2, 2022 at 2:49 PM
3 KB
Plain Text
barcode01_consensus_medaka.nCoV-2019_1.hdf
Mar 2, 2022 at 2:48 PM
6 KB
TextEdit
barcode01_consensus_medaka.nCoV-2019_1.vcf
Mar 2, 2022 at 2:48 PM
249 bytes
vCard
barcode01_consensus_medaka.nCoV-2019_2.hdf
Mar 2, 2022 at 2:49 PM
6 KB
TextEdit
barcode01_consensus_medaka.nCoV-2019_2.vcf
Mar 2, 2022 at 2:49 PM
249 bytes
vCard
barcode01_consensus_medaka.primers.vcf
Mar 2, 2022 at 2:49 PM
315 bytes
vCard
barcode01_consensus_medaka.primersitereport.txt
Mar 2, 2022 at 2:49 PM
Zero bytes
Plain Text
barcode01_consensus_medaka.primertrimmed.rg.sorted.bam
Mar 2, 2022 at 2:48 PM
389 bytes
Document
barcode01_consensus_medaka.primertrimmed.rg.sorted.bam.bai
Mar 2, 2022 at 2:48 PM
24 bytes
Document
barcode01_consensus_medaka.sorted.bam
Mar 2, 2022 at 2:48 PM
337 bytes
Document
barcode01_consensus_medaka.sorted.bam.bai
Mar 2, 2022 at 2:48 PM
24 bytes
Document
barcode01_consensus_medaka.trimmed.rg.sorted.bam
Mar 2, 2022 at 2:48 PM
386 bytes
Document

Nanopolish


Artic
1
Version select
Nanopolish
Procedure select
Demultiplex Barcodes
Enabled
GuppyPlex
Enabled
Nanopolish
Enabled
Inputs
Output Locations
DELETE OUTPUTS
Input Long-Reads Full Run Directory
Required dir
Directory that contains your fastq_pass, fast5_pass, and sequencing summary.txt file. Output directory from a Oxford Nanopore run
Drag Folder here
/Users/merribb1/Desktop/test-data2/20200514_2000_X3_FAN44250_e97e74b4
Sequencing Summary File
Optional
Generated at the end of basecalling in the directory
sequencing_summary file default
Select an item
Must be present in root level of directory
Primer Configuration
Default Primer Scheme to use for nanopolish
Required
nCoV-2019/3
Select an item
Normalise coverage
1000000
Required number
Normalise down to moderate coverage to save runtime
Fast5 directory
Optional dir
Select the fast5 pass directory from your run
Drag Folder here
null/fast5_pass
Label
Access
Completed
Remove
BAM file
3
Variants
3
Consensus
3
Rows per page:
5
1-3 of 3
OPEN LOG FOLDER


consensus_nanopolish
Mar 2, 2022 at 2:51 PM
barcode11_consensus_nanopolish.minion.log.txt
barcode11_consensus_nanopolish.muscle.out.fasta
barcode11_consensus_nanopolish.muscle.in.fasta
barcode11_consensus_nanopolish.consensus.fasta
barcode11_consensus_nanopolish.preconsensus.fasta
barcode11_consensus_nanopolish.coverage_mask.txt
barcode11_consensus_nanopolish.coverage_mask.txt.nCoV-2019_1.depths
barcode11_consensus_nanopolish.coverage_mask.txt.nCoV-2019_2.depths
barcode11_consensus_nanopolish.pass.vcf.gz.tbi
barcode11_consensus_nanopolish.pass.vcf.gz
barcode11_consensus_nanopolish.fail.vcf
barcode11_consensus_nanopolish.primers.vcf
barcode11_consensus_nanopolish.merged.vcf
barcode11_consensus_nanopolish.primersitereport.txt
barcode11_consensus_nanopolish.nCoV-2019_2.vcf
barcode11_consensus_nanopolish.nCoV-2019_1.vcf
barcode11_consensus_nanopolish.primertrimmed.rg.sorted.bam.bai

Parameters

- input run folder: *directory*

  Must contain a `fastq_pass` directory or a custom inputted one (See below)

- Primer: *option or Directory*

  Artic Primer set or a custom one (directory) which contain the `genome.fasta` and necessary `BED` file for the primer set

- Normalize Coverage: *Number*

- Barcode Configuration: *option*

  Which barcode kit you used for demux. Select any for non-barcoded sample

- FASTQ Dir: *Directory, optional*

  Select your own custom *fastq_pass* directory to analyze and demux

- Sequencing Summary File: *File, exists*

  Selects the Sequencing summary file in the root of your run directory

  Note: Oftentimes, this file can be found in the `fastq_pass` directory. Move it one level up to the root run directory

Returns

1. Nanopolish Consensus files: *FASTA files containing your consensuses for each barcode*
2. Nanopolish VCF files: *Variant files containing your variant calls for each barcode*
3. Nanopolish BAM files: *BAM files containing your alignment information for each barcode*

IVAR

IVAR

iVar is a computational package that contains functions broadly useful for viral amplicon-based sequencing. Additional tools for metagenomic sequencing are actively being incorporated into iVar. While each of these functions can be accomplished using existing tools, iVar contains an intersection of functionality from multiple tools that are required to call iSNVs and consensus sequences from viral sequencing data across multiple replicates. We implemented the following functions in iVar: (1) trimming of primers and low-quality bases, (2) consensus calling, (3) variant calling - both iSNVs and insertions/deletions, and (4) identifying mismatches to primer sequences and excluding the corresponding reads from alignment files.

Note: Definition of IVAR gathered directory from the above-mentioned github link

Parameters

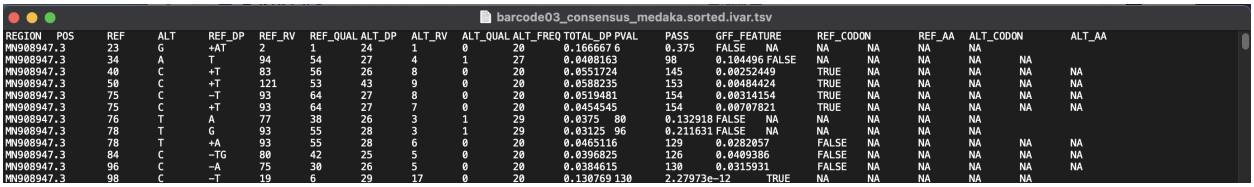
- BAM file: *file*
  - **Alignment BAM file to search for coding regions from. Example pipelines/commands that generate a *bam* file are**
    - *Minimap2*
    - *Medaka*
    - *Nanopolish*
    - *Basestack Consensus*
- Reference GFF3: *option or File*
  - Coding region list that you want to use the alignment searching against
  - Optional, only used for the `Variant` service
- **Reference FASTA: *File***
  - Your original reference FASTA file during alignment. Must also be linked to the GFF3 described above

Returns

1. IVAR Consensus FASTA: *file*
2. IVAR Output: *Tab-separated file*

Note: Contains list of GFFs that map to variants that were called in the BAM file

---



Field	Description
REGION	Region from BAM file
POS	Position on reference sequence
REF	Reference base
ALT	Alternate Base
REF_DP	Ungapped depth of reference base
REF_RV	Ungapped depth of reference base on reverse reads
REF_QUAL	Mean quality of reference base
ALT_DP	Ungapped depth of alternate base.
ALT_RV	Ungapped deapth of alternate base on reverse reads
ALT_QUAL	Mean quality of alternate base
ALT_FREQ	Frequency of alternate base
TOTAL_DP	Total depth at position
PVAL	p-value of fisher's exact test
PASS	Result of p-value <= 0.05
GFF_FEATURE	ID of the GFF feature used for the translation
REF_CODON	Codong using the reference base
REF_AA	Amino acid translated from reference codon
ALT_CODON	Codon using the alternate base
ALT_AA	Amino acid translated from the alternate codon

Note: Description of each field was gathered from the Manual

Minimap2

Minimap2

Minimap2 performs alignment for genomic and spliced nucleotide files

Parameters

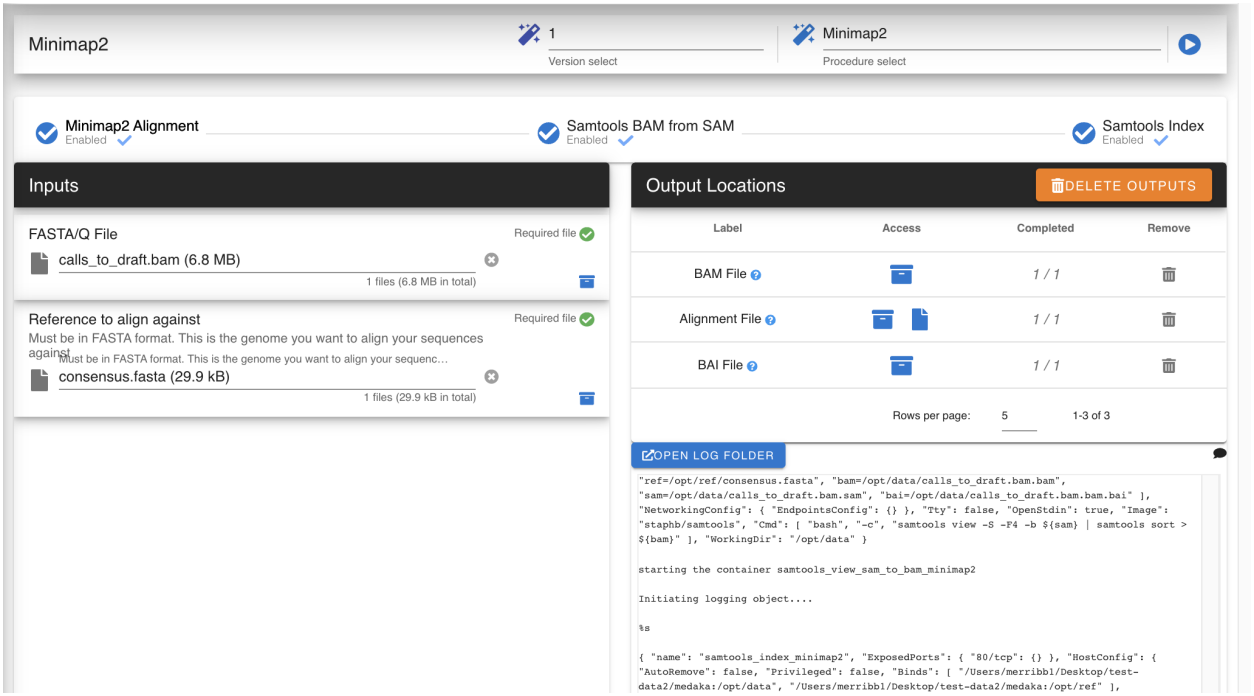
- FASTQ/A file: *file*

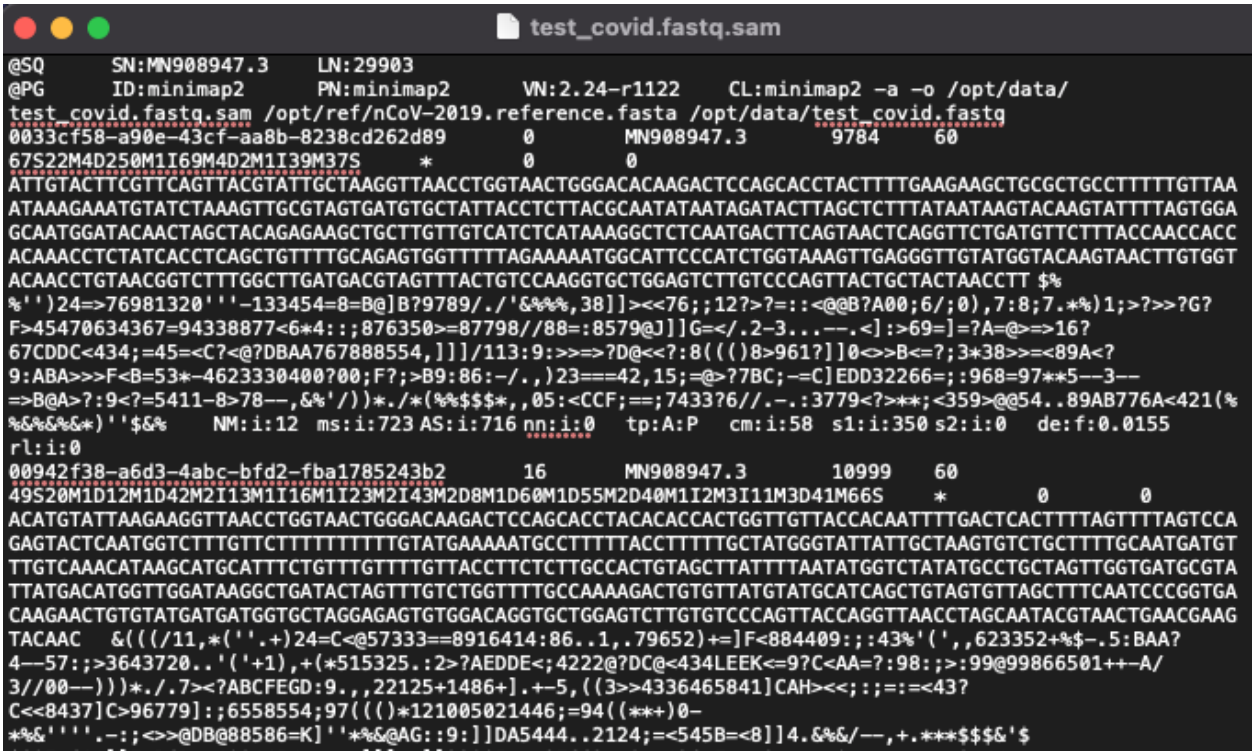
  Raw Reads file to run alignment against a reference on.

- **Reference FASTA:** ***File***
  Your original reference FASTA file during alignment. Must also be linked to the GFF3 described above

Returns

1. BAM: *file*
2. SAM: *file*





Pangolin

Pangolin

Software package for assigning SARS-CoV-2 genome sequences to global lineages.

Parameters

- FASTA file: *file*

  Genome file, can be a multi-sequence FASTA file

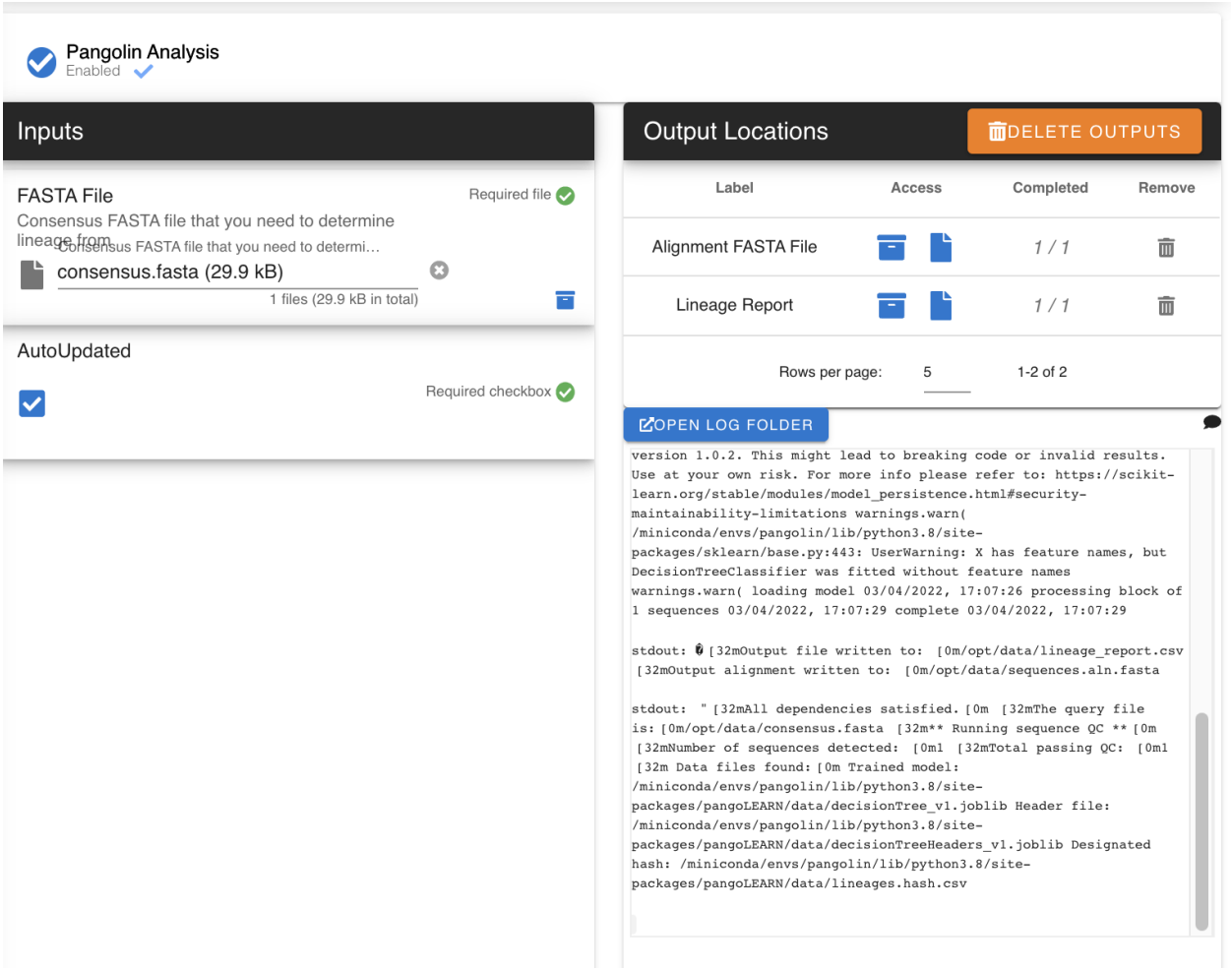
- AutoUpdate: *option*

Note: Pangolin updates so frequently, opt to check for updates before running. Requires Internet access

---

Returns

1. Pangolin Lineage Report: *comma-delimited file*
2. alignment FASTA: *sequences.aln.fasta, contains your alignment from pangolin*



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	taxon	lineage	conflict	ambiguity_s	scorpio_call	scorpio_supp	scorpio_conf	version	pangolin_ver	pangoLEARN	pango_versi	status	note
2	MN908947.3	A.1	0	1				PLEARN-v1.2	3.1.20	2/2/22	v1.2.124	passed_qc	
3													
4													

MASH

MASH

MASH is for uncovering metagenomic distances using MinHash

Parameters

- FASTA genome file: *file*

  Whatever genome file you want to uncover genomic distances from. Can be multipe entries. Can also be a consensus file

- **Reference Genomes:** ***File***
  Can be Sketch format `.msh` or FASTA file.

- Sketch File 1: *option*

Speeds up the process for the input file 1

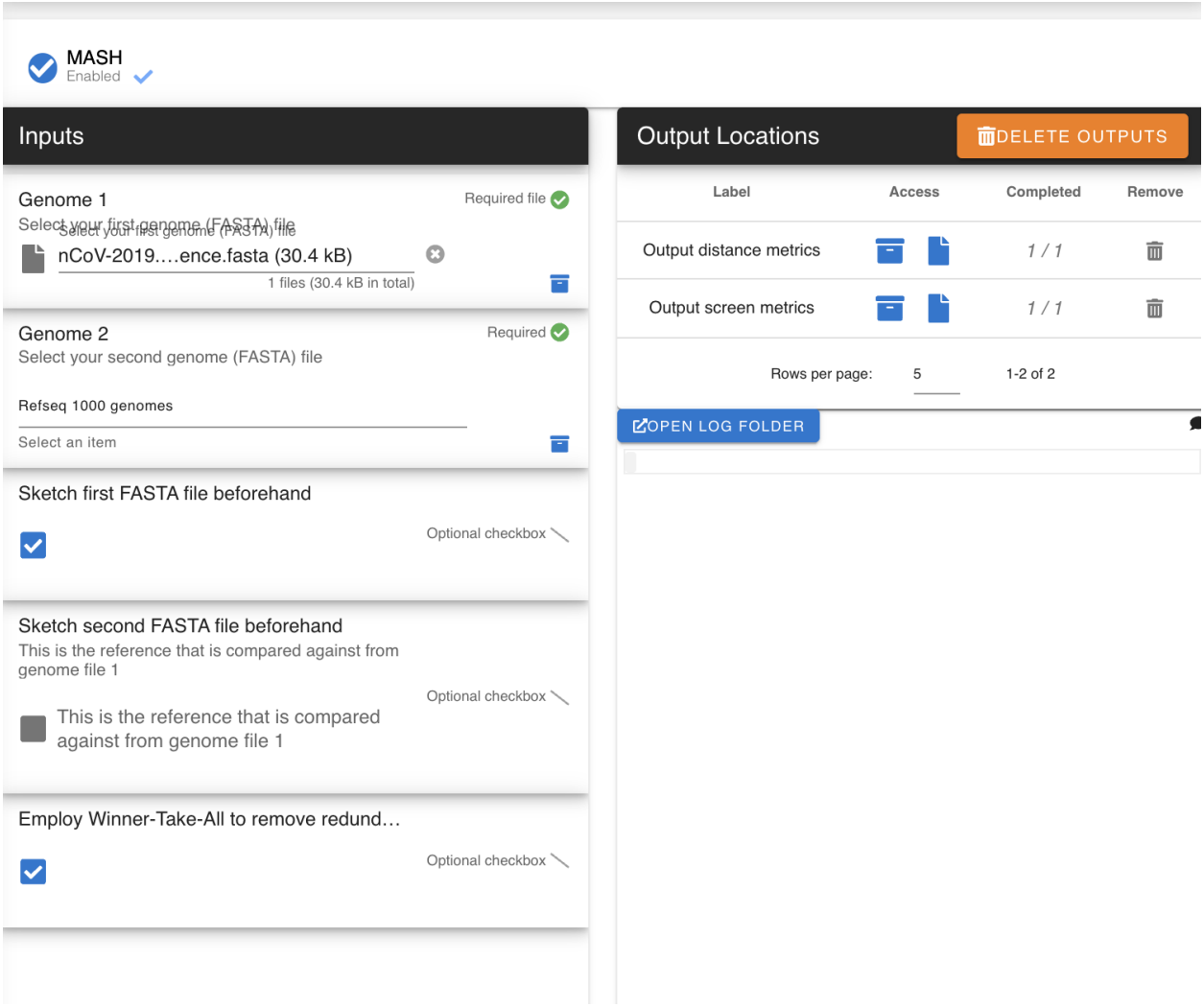
- Sketch File 2: *option*

  Speeds up the process for the input file 2

- Winnder Take All: *option*

  Only take the highest (best) distance annotations against the reference for your input file.

Returns



1. Distance: *tab-separated file*

Note:

Columns:

1. Reference-ID
2. Query-ID
3. Mash-distance

4. P-value
5. Matching-hashes

See more information on calculaions here

mash.dist.tsv

```
GCF_000001215.4_Release_6_plus_ISO1_MT_genomic.fna.gz      /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001405.36_GRCh38.p10_genomic.fna.gz /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001515.7_Pan_tro_3.0_genomic.fna.gz /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001545.4_P_pygmaeus_2.0.2_genomic.fna.gz    /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001635.25_GRCm38.p5_genomic.fna.gz /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001735.3_TAIR10_genomic.fna.gz      /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001765.3_Dpse_3.0_genomic.fna.gz    /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001895.5_Rnor_6.0_genomic.fna.gz    /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001905.1_Loxafr3.0_genomic.fna.gz   /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000001985.1_JCVI-PMFA1-2.0_genomic.fna.gz      /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000002035.5_GRCz10_genomic.fna.gz      /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000002075.1_AplCal3.0_genomic.fna.gz   /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000002195.4_Amel_4.5_genomic.fna.gz    /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
GCF_000002235.4_Spur_4.2_genomic.fna.gz    /opt/data/nCoV-2019.reference.fasta       1       1       0/1000
```

2. Screen: *tab-separated file*

Note:

Columns:

1. identity
2. shared-hashes
3. median-multiplicity
4. p-value
5. query-ID
6. query-comment

See more information here

mash.screen.tsv

```
0.867054 50/1000  1       1.97626e-323    GCF_000864885.1_ViralProj15500_genomic.fna.gz     NC_004718.3 SARS coronavirus, complete genome
0.825882 18/1000  1       2.24222e-109    GCF_000887595.1_ViralProj51751_genomic.fna.gz     NC_014470.1 Bat coronavirus BM48-31/BGR/2008, complete genome
0.719686 1/1000   1       7.01082e-06     GCF_000899495.1_ViralProj177902_genomic.fna.gz    NC_018871.1 Rousettus bat coronavirus HKU10, complete genome
0.719686 1/1000   1       7.01082e-06     GCF_000894435.1_ViralProj109275_genomic.fna.gz    NC_016993.1 Magpie-robin coronavirus HKU18, complete genome
0.719686 1/1000   1       7.01082e-06     GCF_000889395.1_ViralProj59783_genomic.fna.gz     NC_014637.1 Cafeteria roenbergensis virus BV-PW1, complete genome
0.719686 1/1000   1       7.01082e-06     GCF_000868045.1_ViralProj18867_genomic.fna.gz     NC_009021.1 Bat coronavirus HKU9-1, complete genome
0.719686 1/1000   1       7.01082e-06     GCF_000312225.1_ASM31222v1_genomic.fna.gz [267 seqs] NZ_HE973313.1 Microcystis aeruginosa PCC 9807, whole genome
shotgun sequence [...]
```

VADR

Viral Annotation DefineR: classification and annotation of viral sequences based on RefSeq annotation

VADR

VADR is a suite of tools for classifying and analyzing sequences homologous to a set of reference models of viral genomes or gene families. It has been mainly tested for analysis of Norovirus, Dengue, and SARS-CoV-2 virus sequences in preparation for submission to the GenBank database.

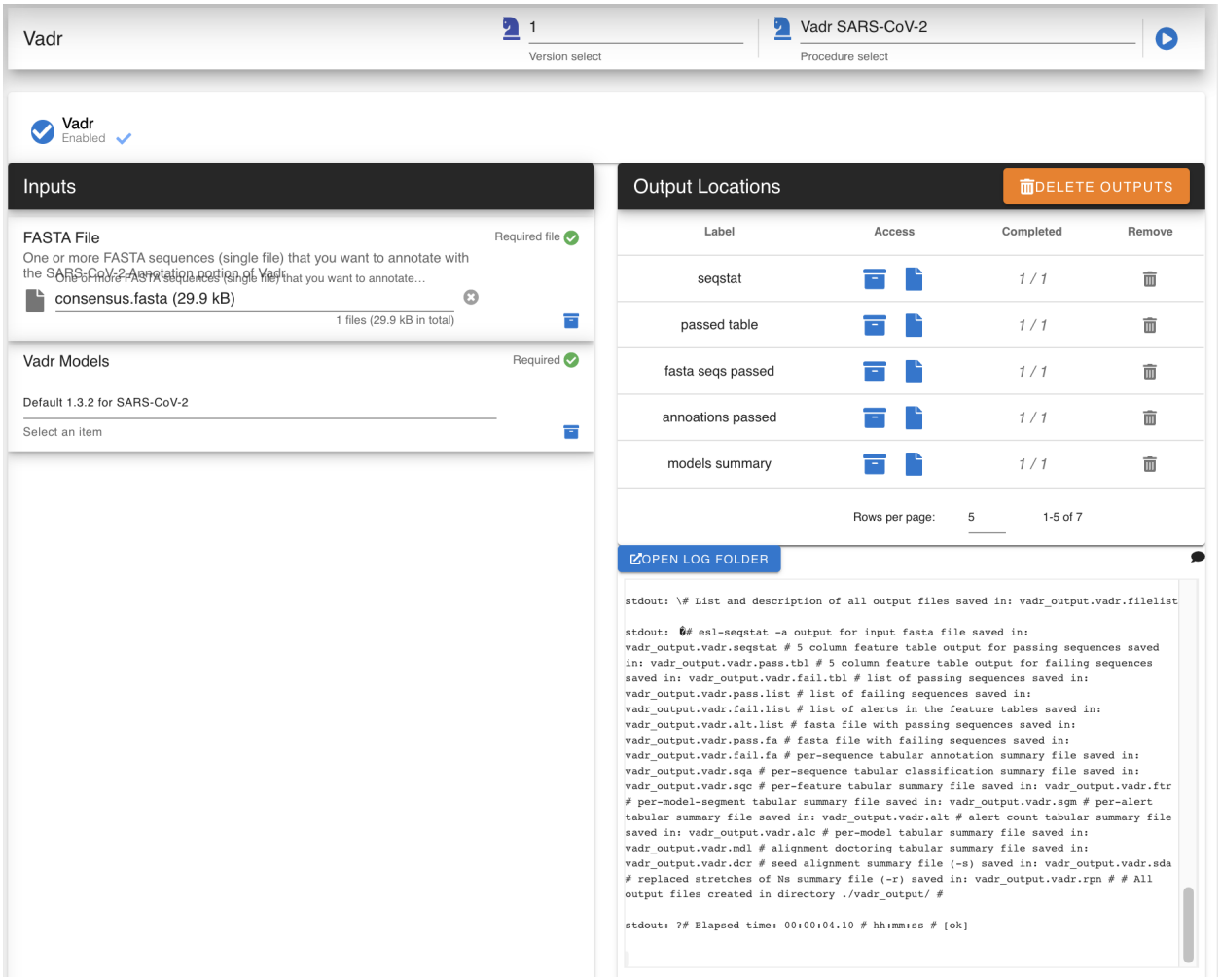
Parameters

- FASTA consensus file: *file*

  Sequencing file you want to get viral annotations from in the VADR pipeline

- **Reference Genome:** ***option or File***
  Can be Sketch format `.msh` or FASTA file.

Returns



1. Table format: *Tab-separated annotation*

   5 column feature table output for passing sequences

```
>Feature MN908947.3
266     21555   gene
                        gene    ORF1ab
266     13468   CDS
13468   21555
                        product ORF1ab polyprotein
                        exception       ribosomal slippage
                        protein_id      MN908947.3_1
266     13483   CDS
                        product ORF1a polyprotein
                        protein_id      MN908947.3_2
266     805     mat_peptide
                        product leader protein
                        protein_id      MN908947.3_1
266     805     mat_peptide
                        product leader protein
                        protein_id      MN908947.3_2
806     2719    mat_peptide
                        product nsp2
                        protein_id      MN908947.3_1
806     2719    mat_peptide
                        product nsp2
                        protein_id      MN908947.3_2
```

2. sgm format: *Tab-separated annotation*

   per-model-segment tabular summary file

vadr_output.vadr.sgm

#	seq	seq			ftr	ftr	ftr	num	sgm	seq	seq	mdl	mdl	sgm			5'	3'	5'	3'
#idx	name	len	p/f	model	type	name	idx	sgm	idx	from	to	from	to	len	str	trc	pp	pp	gap	gap
1.1.1	MN908947.3	29903	PASS	NC_045512	gene	ORF1ab	1	1	1	266	21555	266	21555	21290	+	no	?	?	no	no
1.2.1	MN908947.3	29903	PASS	NC_045512	CDS	ORF1ab_polyprotein	2	2	1	266	13468	266	13468	13203	+	no	?	?	no	no
1.2.2	MN908947.3	29903	PASS	NC_045512	CDS	ORF1ab_polyprotein	2	2	2	13468	21555	13468	21555	8088	+	no	?	?	no	no
1.3.1	MN908947.3	29903	PASS	NC_045512	CDS	ORF1a_polyprotein	3	1	1	266	13483	266	13483	13218	+	no	?	?	no	no
1.4.1	MN908947.3	29903	PASS	NC_045512	gene	S	4	1	1	21563	25384	21563	25384	3822	+	no	?	?	no	no
1.5.1	MN908947.3	29903	PASS	NC_045512	CDS	surface_glycoprotein	5	1	1	21563	25384	21563	25384	3822	+	no	?	?	no	no
1.6.1	MN908947.3	29903	PASS	NC_045512	gene	ORF3a	6	1	1	25393	26220	25393	26220	828	+	no	?	?	no	no
1.7.1	MN908947.3	29903	PASS	NC_045512	CDS	ORF3a_protein	7	1	1	25393	26220	25393	26220	828	+	no	?	?	no	no
1.8.1	MN908947.3	29903	PASS	NC_045512	gene	E	8	1	1	26245	26472	26245	26472	228	+	no	?	?	no	no
1.9.1	MN908947.3	29903	PASS	NC_045512	CDS	envelope_protein	9	1	1	26245	26472	26245	26472	228	+	no	?	?	no	no
1.10.1	MN908947.3	29903	PASS	NC_045512	gene	M	10	1	1	26523	27191	26523	27191	669	+	no	?	?	no	no
1.11.1	MN908947.3	29903	PASS	NC_045512	CDS	membrane_glycoprotein	11	1	1	26523	27191	26523	27191	669	+	no	?	?	no	no
1.12.1	MN908947.3	29903	PASS	NC_045512	gene	ORF6	12	1	1	27202	27387	27202	27387	186	+	no	?	?	no	no

3. sqa format: *Tab-separated file*

   per-sequence tabular annotation summary file

vadr_output.vadr.sqa

#seq	seq	seq			best		sub						seq
#idx	name	len	p/f	ant	model	grp	grp	nfa	nfn	nf5	nf3	nfalt	alerts
1	MN908947.3	29903	PASS	yes	NC_045512	Sarbecovirus	SARS-CoV-2	54	0	0	0	0	-

All output files created in directory your directory containing your input FASTA file under `vadr_output`

Note: See more information **`here <https://github.com/ncbi/vadr/wiki/Coronavirus-annotation#many-alertserrors-in-orf3a-orf6-orf7a-orf7b-orf8-and-orf10-do-not-cause-a-sequence-to-fail-1`_**

1. Output printed to screen saved in: my4.vadr.log

2. List of executed commands saved in: my4.vadr.cmd
3. List and description of all output files saved in: my4.vadr.filelist
4. esl-seqstat -a output for input fasta file saved in: my4.vadr.seqstat
5. 5 column feature table output for passing sequences saved in: my4.vadr.pass.tbl
6. 5 column feature table output for failing sequences saved in: my4.vadr.fail.tbl
7. list of passing sequences saved in: my4.vadr.pass.list
8. list of failing sequences saved in: my4.vadr.fail.list
9. list of alerts in the feature tables saved in: my4.vadr.alt.list
10. fasta file with passing sequences saved in: my4.vadr.pass.fa
11. fasta file with failing sequences saved in: my4.vadr.fail.fa
12. per-sequence tabular annotation summary file saved in: my4.vadr.sqa
13. per-sequence tabular classification summary file saved in: my4.vadr.sqc
14. per-feature tabular summary file saved in: my4.vadr.ftr
15. per-model-segment tabular summary file saved in: my4.vadr.sgm
16. per-alert tabular summary file saved in: my4.vadr.alt
17. alert count tabular summary file saved in: my4.vadr.alc
18. per-model tabular summary file saved in: my4.vadr.mdl
19. alignment doctoring tabular summary file saved in: my4.vadr.dcr
20. ungapped seed alignment summary file (-s) saved in: my4.vadr.sda
21. replaced stretches of Ns summary file (-r) saved in: my4.vadr.rpn

Samtools (Under Construction)

Warning: Under Construction

Unicycler

A hybrid assembly pipeline for bacterial genomes

Unicycler

Note: This pipeline uses SPAdes

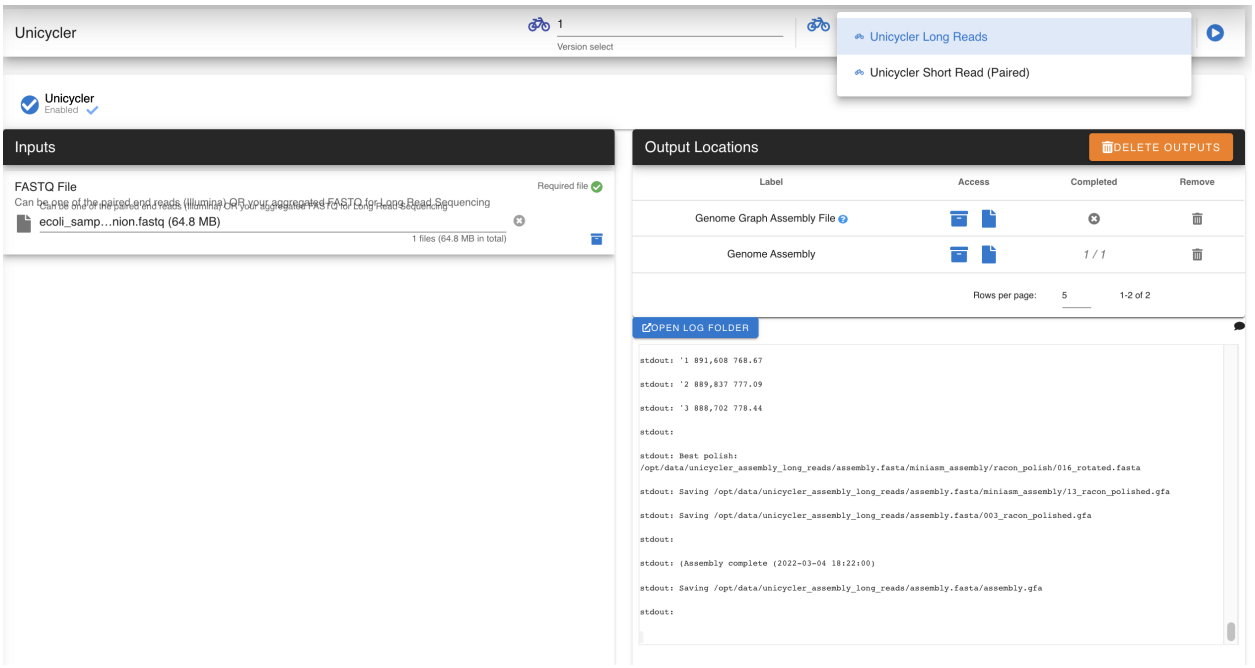
Unicycler Long Reads

Parameters

- FASTQ file: *file*

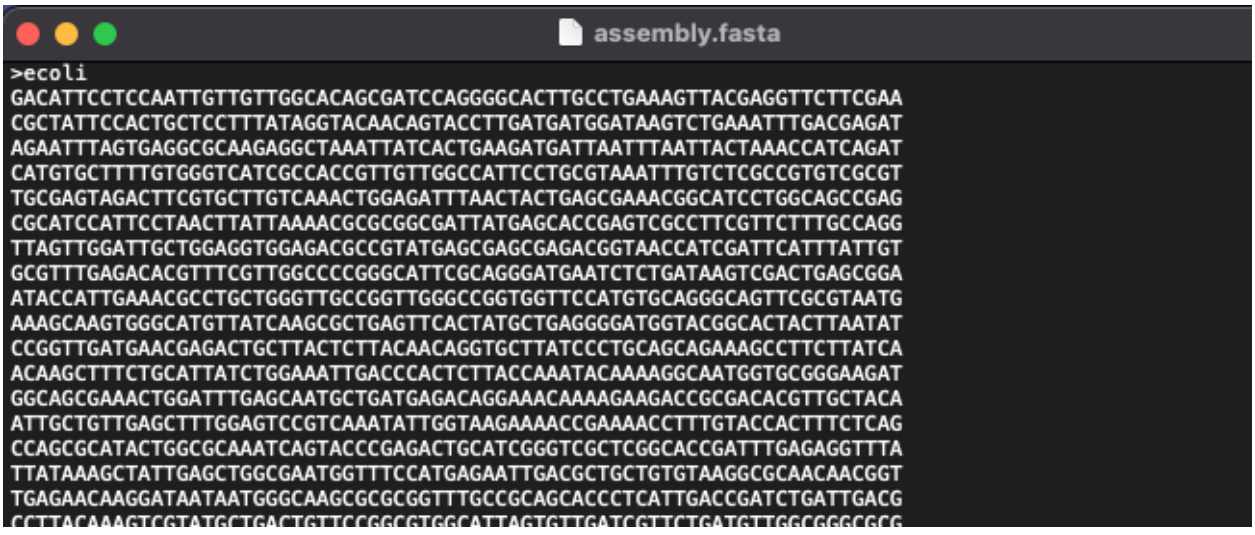
  Sequencing raw read file. Can be short or long read

Returns



1. Genome Assembly: *Tab-separated annotation*

   Assembly FASTA File



2. GFA File: *File*

Can be used in 3rd party tools like Bandage

Unicycler Short Reads

Parameters

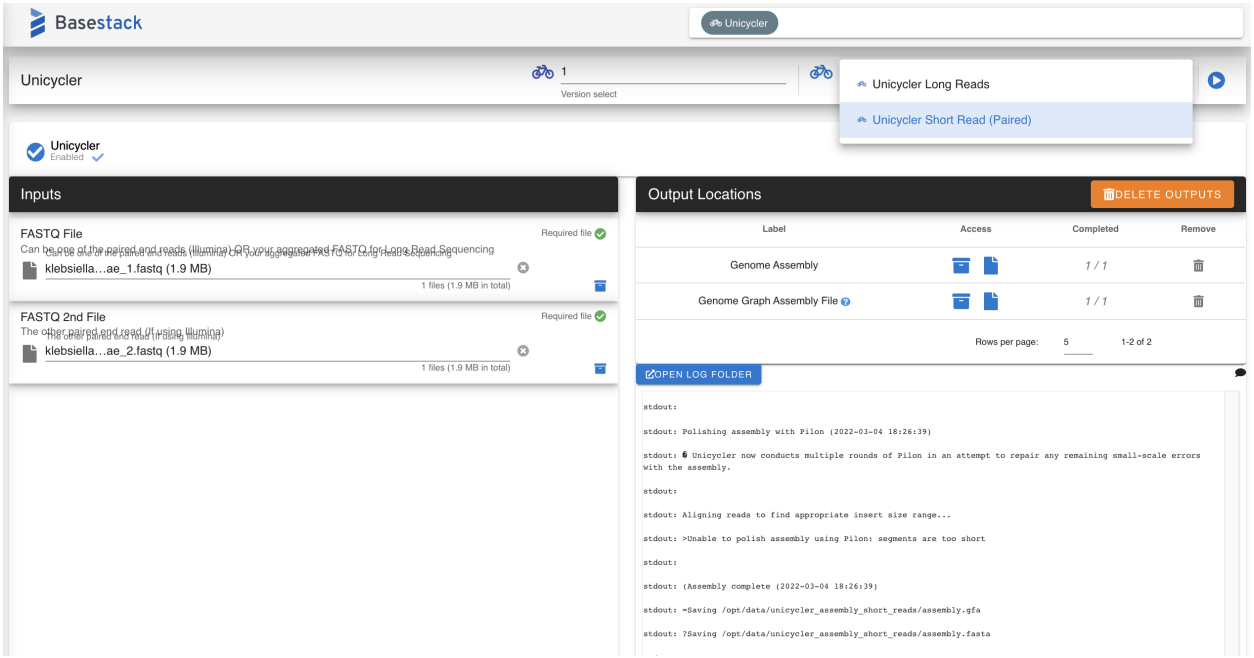
- FASTQ first file: *file*

  First Read file (paired or single-end)

- FASTQ second file: *file*

  Second read file (paired only)

Returns



1. Genome Assembly: *Tab-separated annotation*

   Assembly FASTA File

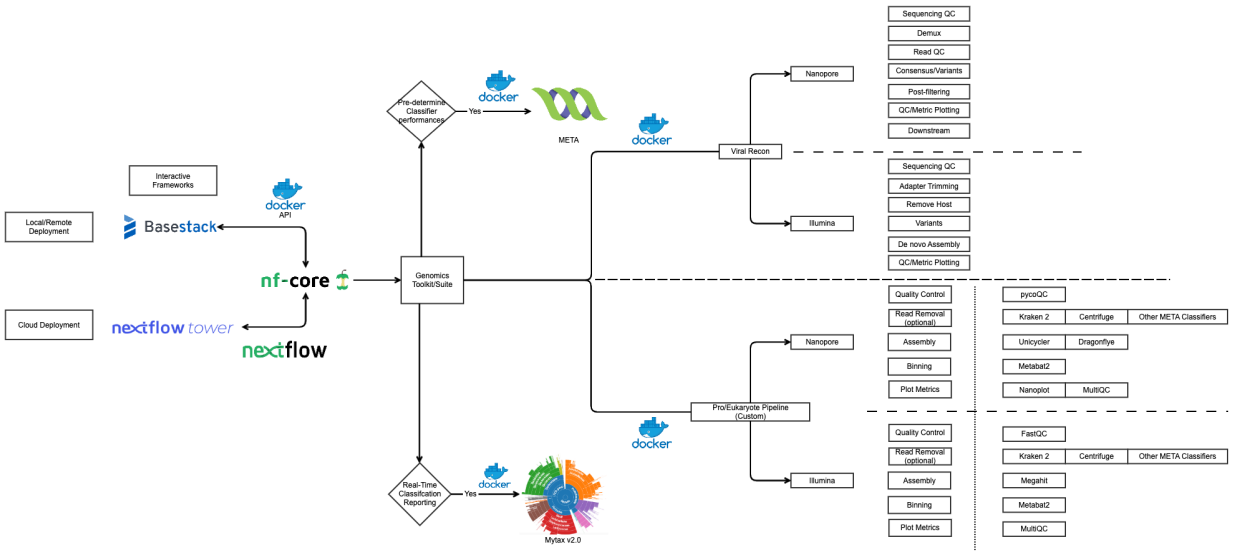
2. GFA File: *File*

   Can be used in 3rd party tools like Bandage

TaxTriage (Metagenomics) (Under Construction)

Warning: This module is under construction and is in alpha-release. Scheduled full release of v1.0 in Oct. 2022

Standard diagram for deployment and pipeline development



The pipeline consists of a variety of alignment/classification steps as well as QC and pre-filtering processes. It is designed to be serve as the initial triage step for identifying unknown organisms present in one or more sample types and supports both Illumina or Oxford Nanopore-generated NGS data.

The pipeline is packaged to include basic quality control to making a (potential) de-novo assembly for each organism that is detected in the sample from a filtering a hierarchical perspective. That is, the most prevalent taxonomic IDs at various ranks in the hierarchical chain are reported, binned, and run through a variety of alignment and assembly steps (for lower levels like species). Finally, a set of flags are generated for each taxonomic map that is the most prevalent per sample.

A list of tools used are listed below for each step

1. Demultiplex and Gather **OPTIONAL, Oxford Nanopore Only**

- Artic Guppyplex - Aggregate Nanopore reads for downstream analysis

2. Quality Control **OPTIONAL**

- PycoQC - Computes metrics and generates interactive QC plots for Oxford Nanopore technologies sequencing data

3. Trimming

- Illumina: Trimgalore
- Oxford: Porechop

4. Filtering

- Kraken2

5. QC Plotting

- Illumina: FastQC

- Oxford: Nanoplot

6. Classification (K-mer approach)

- Kraken2

7. Alignent Stats

- Illumina: BWAMEM2
- Oxford: Minimap2

8. Report Generation

- MultiQC

Please see relevant links in the listed modules for more information on the underlying mechanisms and corresponding papers (if existent)

Parameters

- Samplesheet (.csv): *file*

  Contains a mapping of metadata and a single sample per row. Explanations of the possible columns for Basestack are seen below:

Table 3: Samplesheet Description

Column Name	Description
sample	Custom sample name. This entry will be identical for multiple sequencing libraries/runs from the same sample. Spaces in sample names are automatically converted to underscores (_).
single_end	Is the data single or paired end
fastq_1	Full path to FastQ file for Illumina short reads 1 OR OXFORD reads. File has to be gzipped and have the extension ".fastq.gz" or ".fq.
fastq_2	Full path to FastQ file for Illumina short reads 2. File has to be gzipped and have the extension ".fastq.gz" or ".fq.
barcode	TRUE/FALSE, is the row attributed to a demultiplexed barcode folder of 1 or more fastq files or is it a single file that is .
from	Directory path of the barcode, only used with the column being set as TRUE in the barcode column
trim	TRUE/FALSE, do you want to run trimming on the sample?
platform	Platform used, [ILLUMINA, OXFORD]
sequencing_summary	If detected, output plots based on the the sequencing summary file for that sample

Table 4: Example Samplesheet

sample	fastq_1	fastq_2	platform	from	trim	sequencing_summary	single_end	barcode
Sample_1	AEG588A1_S1_L001_R1_001.fastq.gz	AEG588A1_S1_L001_R2_001.fastq.gz	ILLUMINA	NULL (or leave blank)	FALSE	NULL (or leave blank)	FALSE	FALSE
Sample_2	ecoli_reads.fastq	NULL	OXFORD	NULL	FALSE	sequencing_summary.txt	TRUE	FALSE
Sample_3	NULL	NULL	OXFORD	barcode01	TRUE	FALSE	TRUE	TRUE

For the samples shown above:

1. A paired-end run of Illumina data where we DON'T trim anything (no Trimgalore)
2. A single-end Oxford Nanopore run where all reads are concatenated to a single fastq file. No barcode. There is a sequencing summary file we want to plot for run statistics/plots
3. A single-end Oxford Nanopore run where reads have NOT been demultiplexed and/or aggregated to a single fastq file (like row 2). This will run *artic guppyplex* as well to concatenate all to one fastq file

Returns

1. MultiQC report HTML file
2. Variety of intermediate and output results files for the MultiQC report

- Examples: - SAM/BAM alignment - Filtered FASTQ Files (for downstream use) - Assembly (de novo) - WIP and is not ready just yet - Kraken2 Report(s)

3.2.2 Supplemental

Guppy Minknow

MinKNOW

In order to run the MinION sequencer, you first need to download/install the necessary software from Oxford Nanopore's mirror(s).

```
wget -O- https://mirror.oxfordnanoportal.com/apt/ont-repo.pub | sudo apt-key add -

echo "deb http://mirror.oxfordnanoportal.com/apt $(lsb_release -c | awk '{print $2}')-
↪stable non-free" | sudo tee /etc/apt/sources.list.d/nanoporetech.sources.list

sudo apt-get -y update

sudo apt-get install -y minion-nc
```

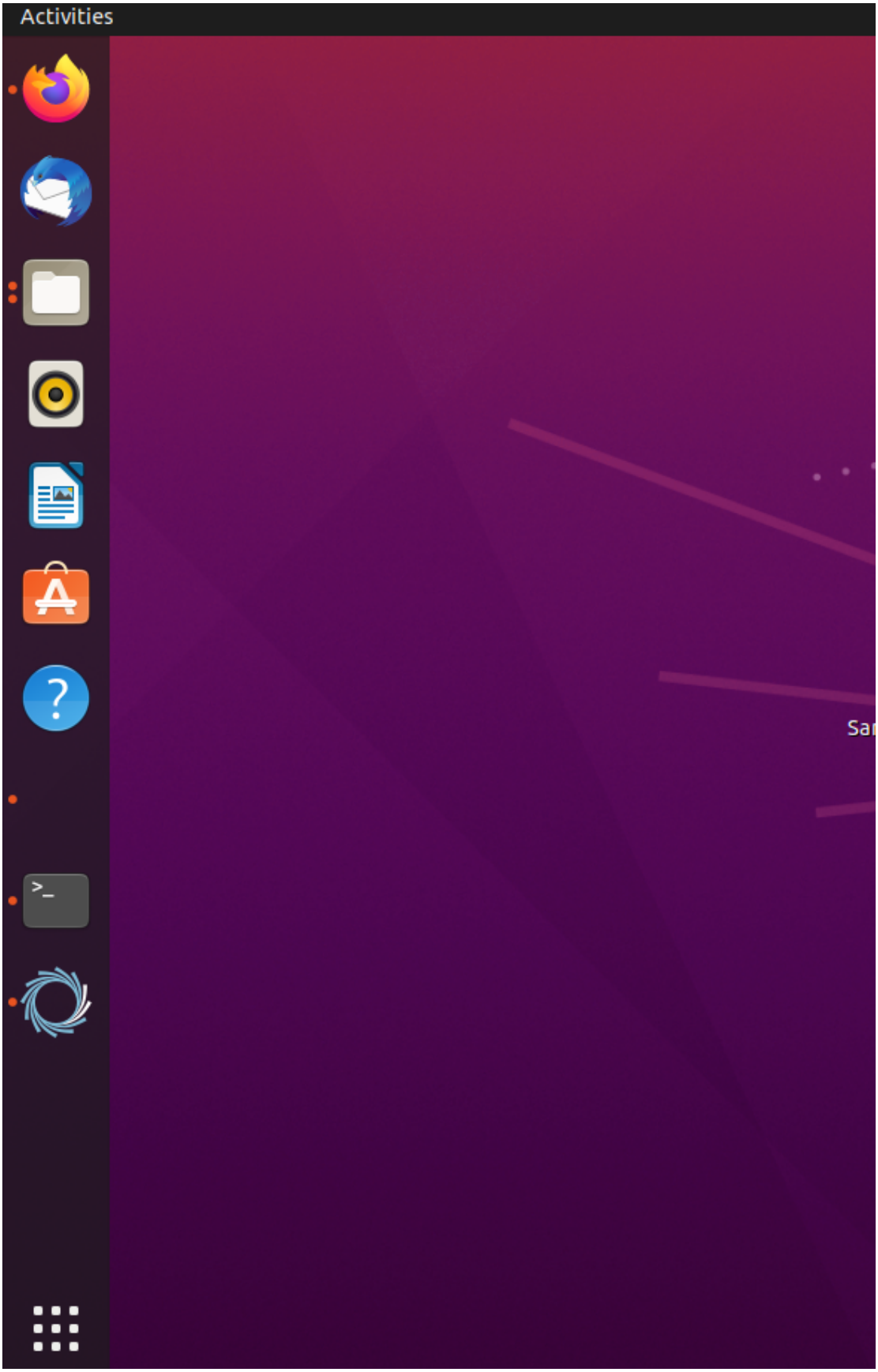
Note: See here

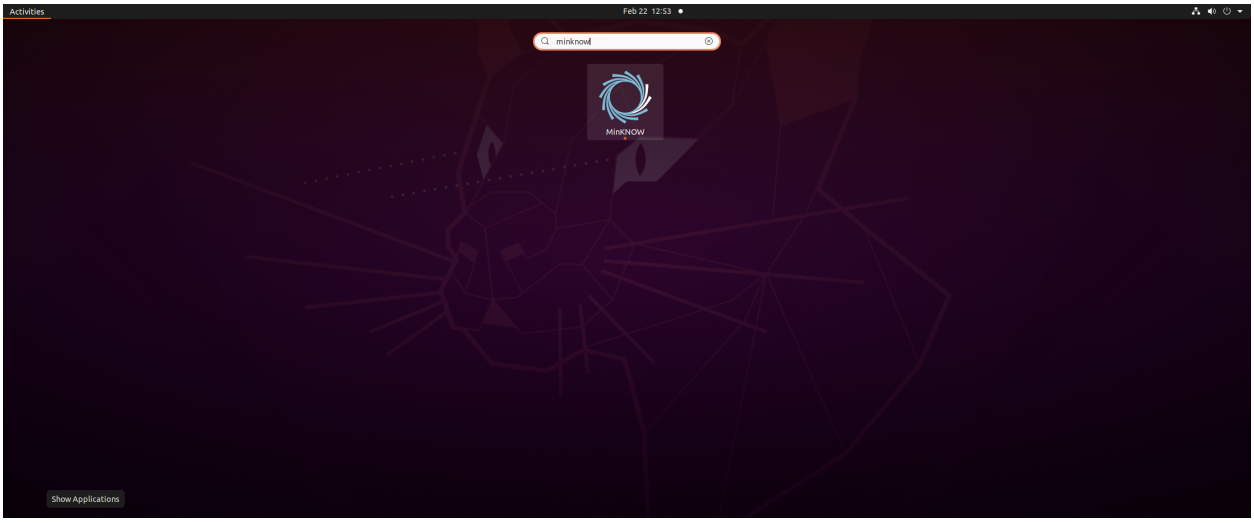
Next, we need to install guppy on your system. Skip this step if you are not using a GPU in your system.

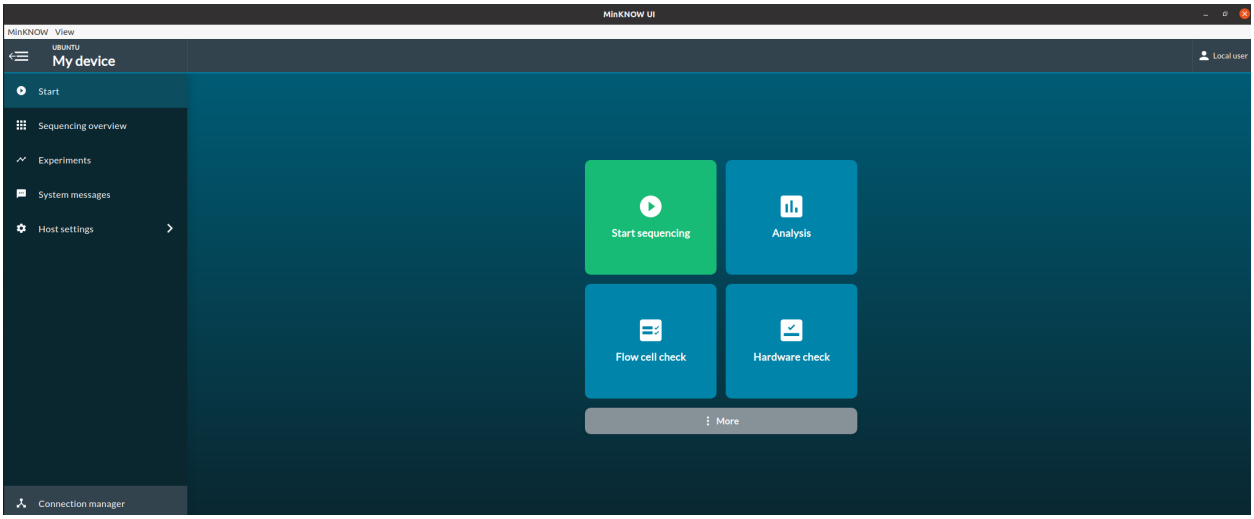
PLEASE NOTE: this option is available only for Linux-based distributions. You have to use CPU-mode for Windows (Fast config basecalling mode)

Running the app

Select Minknow from the Quick-launch location in Ubuntu. Usually, this is in the bottom-left section of the screen. You should see the MinKNOW icon


Activities

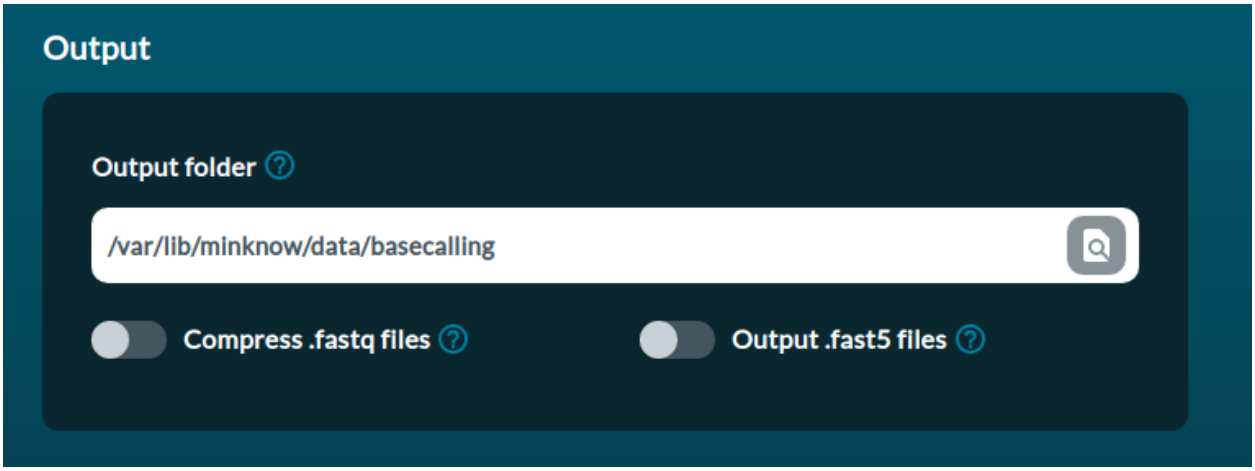




Note: If you dont set your analysis to run offline, you will see a window that asks you to login. Please follow instructions below to disable online-mode

Testing Guppy Basecaller from MinKNOW

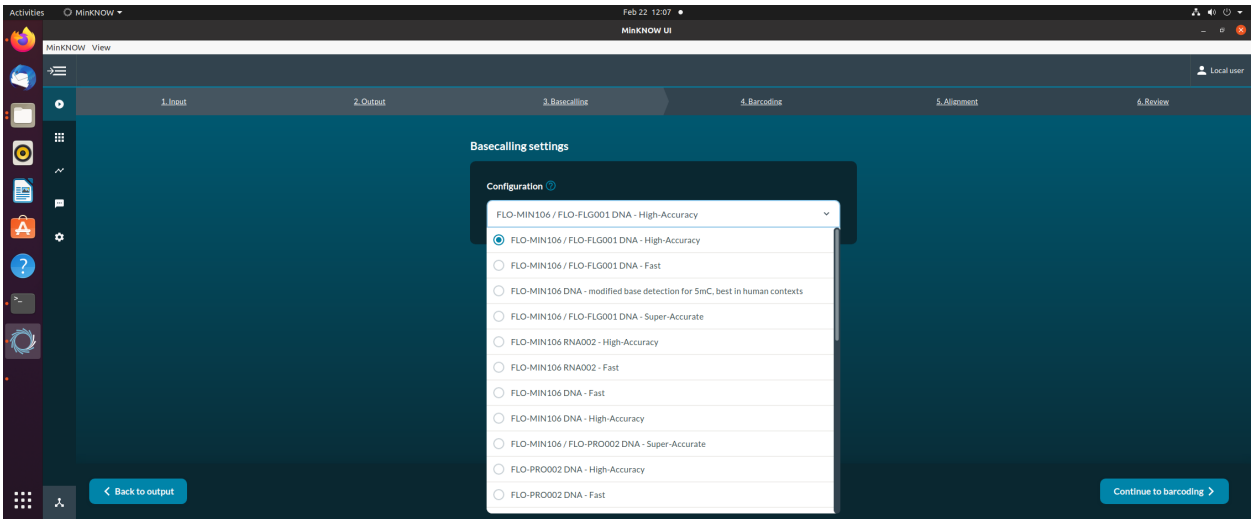
First, select the `Start` at the top-left. Then, select `Basecalling`



Note: By default, all sequencing runs will output to /var/lib/minknow/data on Linux machines.

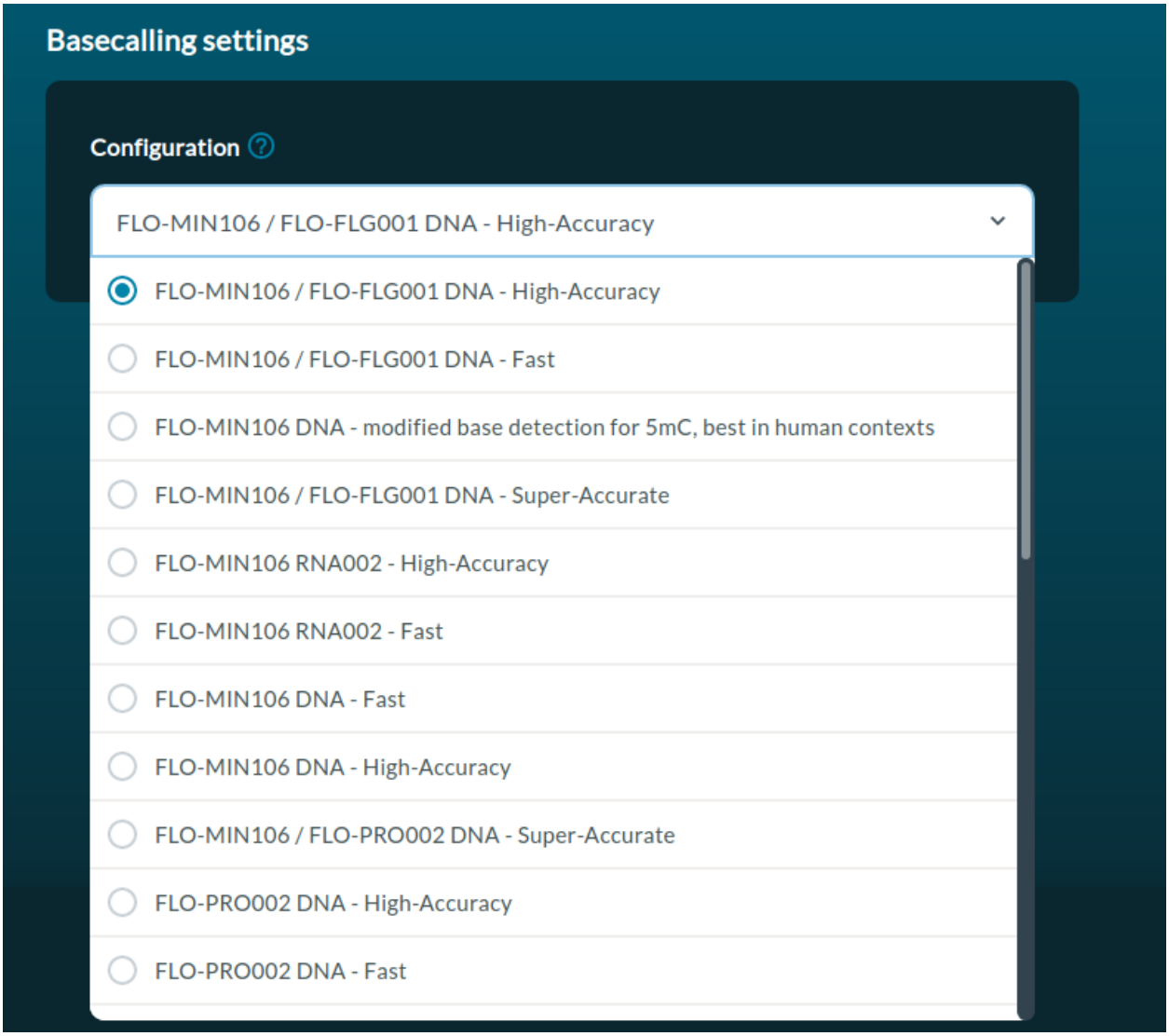
Warning: To run basecalling from the UI, you MUST ensure that permissions allow reading + writing for the `minknow` group on Linux machines

Here, you must select the basecalling configuration you'd like to use



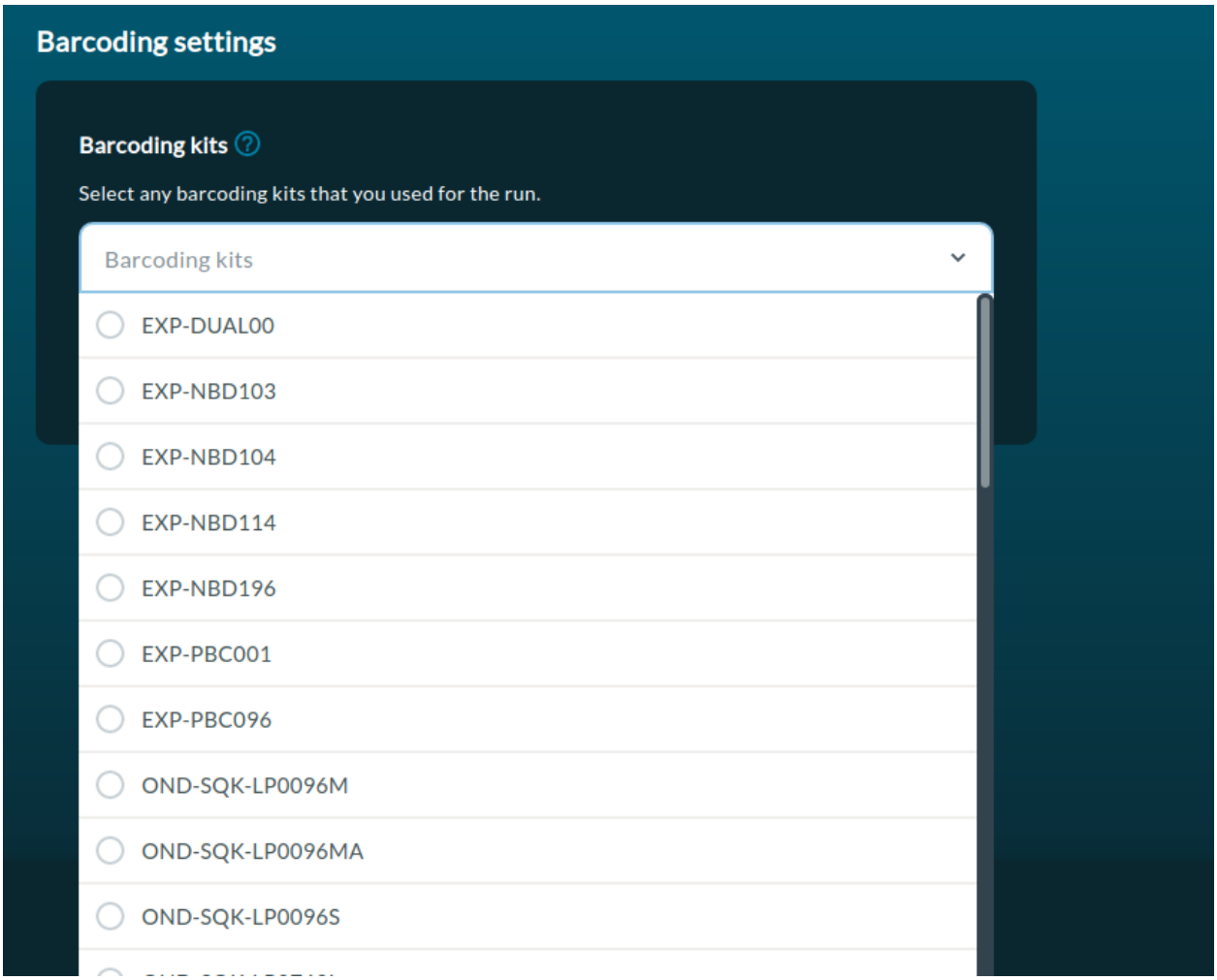
Most, when running the MinION or MK1C, will use the `FLO-MIN106 / FLO-FLG001 DNA` options, it is the default option.

Next, pick whether you want the High-Accuracy or Fast configurations (in the names)



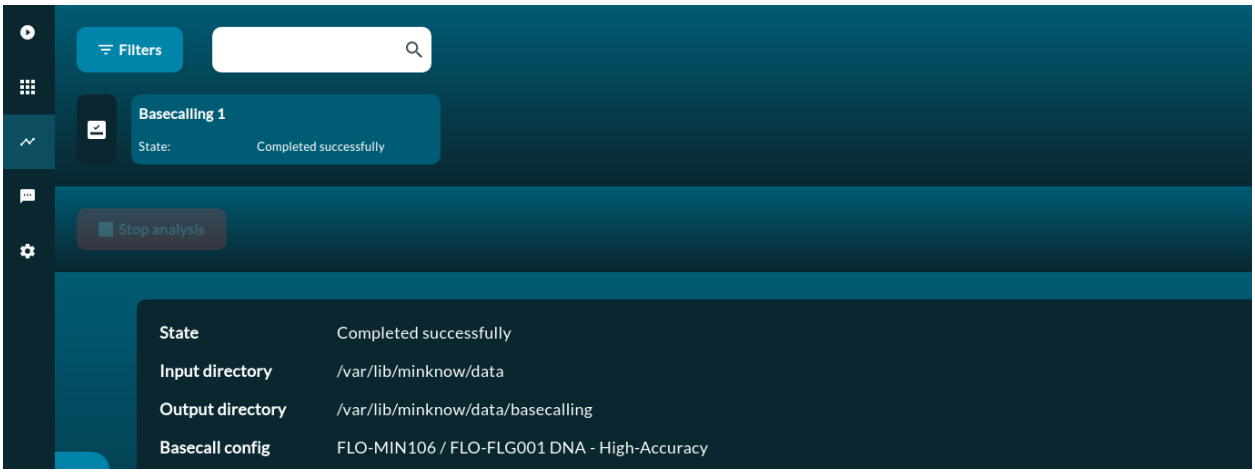
Note: High-Accuracy is RECOMMENDED if using a GPU-capable laptop with a CUDA-compatible device. Make sure you set up GPU basecalling as described here *Guppy GPU Basecaller* Fast accuracy is optional but is RECOMMENDED for non GPU-capable laptops/devices. It will run everything using a CPU.

Warning: If sequencing, it is generally not recommended to run basecalling at the same time if using `Fast`, CPU-only. This is because the system can get overloaded and crash the application



Note: If you don't see any barcoding kits, go here *Barcoding Kits Missing* to fix

Move through the rest of the configurations until you finally can hit `Start`, the green button at the bottom-right of the page



Note: The progress should begin to update as the process goes forward.

..warning::

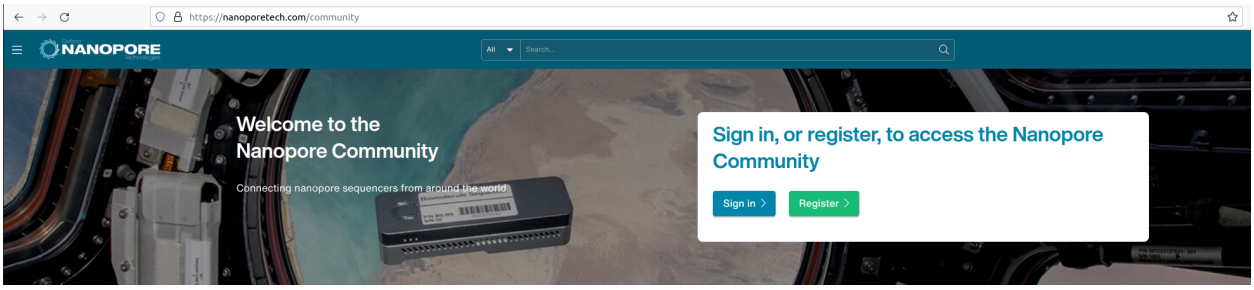
If it errors out, check logs by selecting the right arrow or by looking at all files (recently made) at `/var/log/minknow`. Oftentimes it is an issue with permissions

Oxford Community Forums

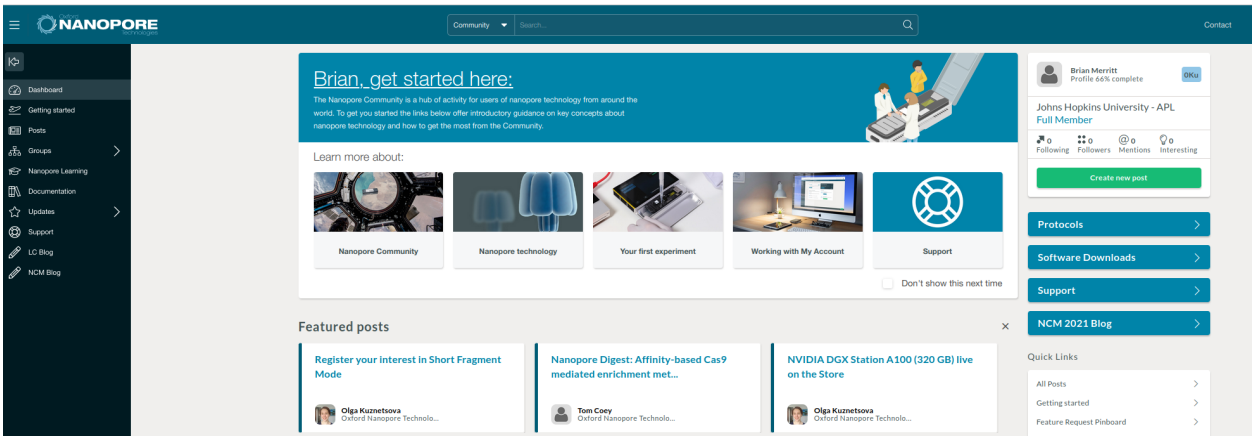
Oxford provides a very active forum for users of ONT software and hardware to interact and post questions or solutions to issues.

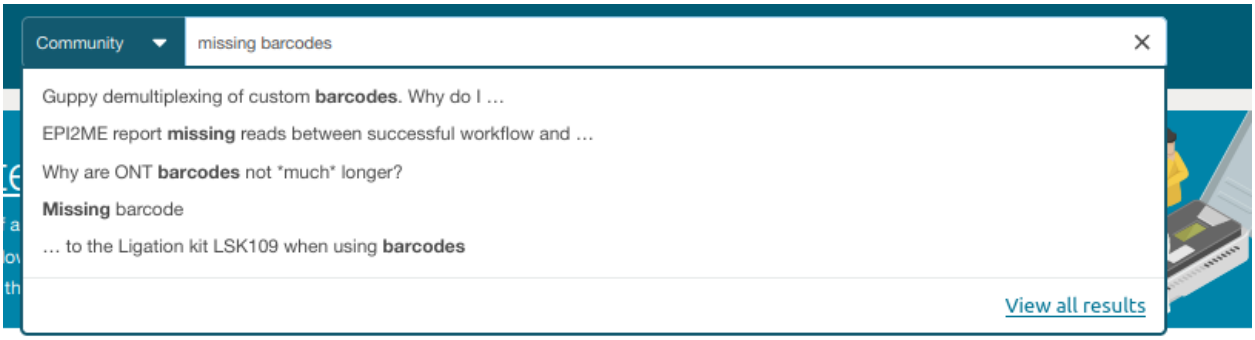
Additionally, the Oxford team will oftentimes provide helpful information on their devices or software

First, head to https://community.nanoporetech.com/



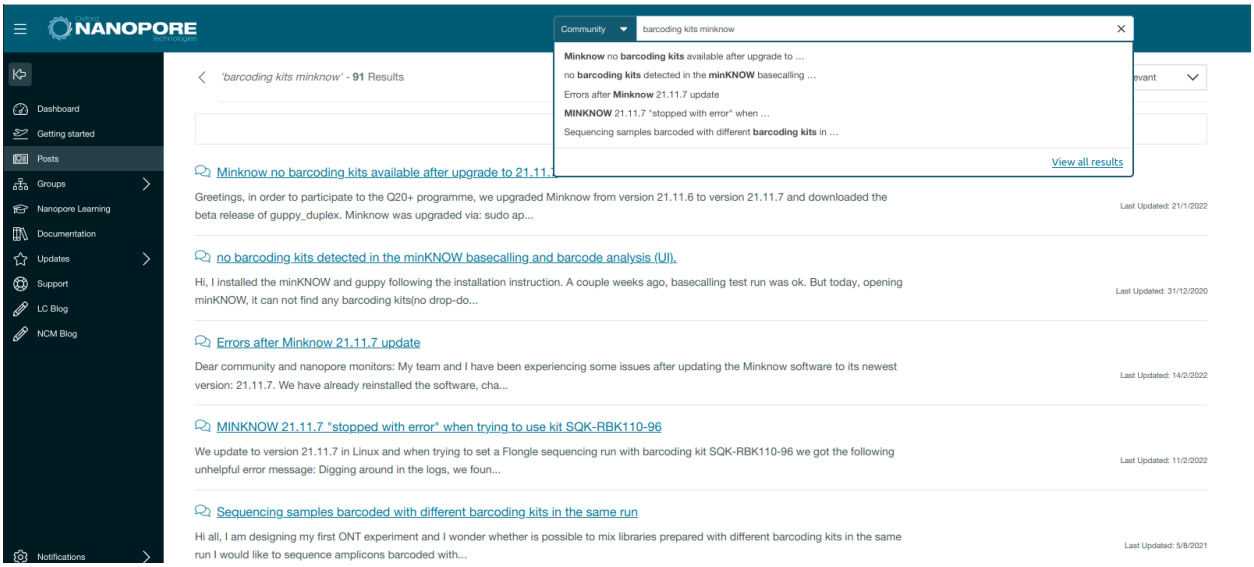
You will need to make an Oxford account to do so. Please register for an account as all Oxford hardware users should have the ability to login and view the community pages available on the site





Now, lets run through an example. We discussed earlier (as a warning) that sometimes you may not see the barcoding kits when basecalling your runs This is a common bug that was found with one of the patches for the MinKNOW software, caught by some community members

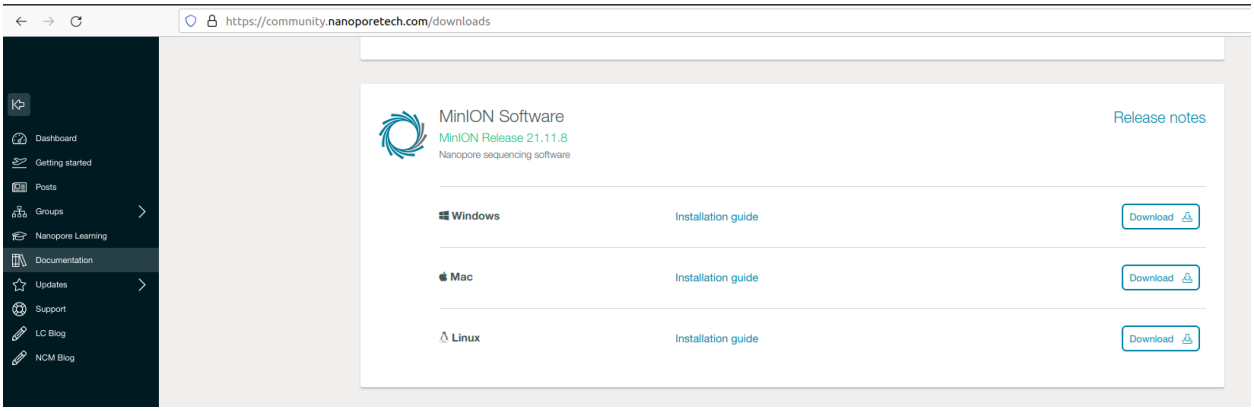
Simply look at the top of the page and enter your query. In this example, lets type "missing barcodes kits" where you will get some suggested items. Either hit `Enter` to view all items or select one of the quick-launch items



This site is an invaludable tool if you're experiencing issues with your software or hardware. Oftentimes, responses to updates or issues are quick, so it is highly recommended to follow these forums from time-to-time

Viewing downloads

Oxford also provides a set of software links you can download applications from. These are available at https://community.nanoporetech.com/downloads



Setting Up Guppy for GPU

See here for more advanced details

If you have a CUDA-capable GPU in your laptop, follow *CUDA* and the above link to set it up for MinKNOW

Note: Running GPU basecalling is not required if you prefer to run it from the command line, detailed here *Guppy GPU Basecaller*

.

CUDA

Ensure that your GPU is CUDA-capable first by typing

```
lspci | grep VGA
```

If you see your GPU model, for example: *NVIDIA Corporation TU102 [GeForce RTX 2080 Ti] (rev A1)* then you have a GPU available on your machine. IF you don't see that AND you know there is a GPU in the machine try to install the drivers first.

Once the drivers are installed go to: https://developer.nvidia.com/cuda-downloads.

Select the appropriate distribution values and copy+paste the commands that populate into your terminal, one-by-one.

On my Ubuntu 20.04 (Focal) machine I head to [here](https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&=Ubuntu&target_version=20.04&target_type=deb_local)

then copy + paste

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-
↪ubuntu2004.pin
sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
wget https://developer.download.nvidia.com/compute/cuda/11.6.1/local_installers/cuda-
↪repo-ubuntu2004-11-6-local_11.6.1-510.47.03-1_amd64.deb
sudo dpkg -i cuda-repo-ubuntu2004-11-6-local_11.6.1-510.47.03-1_amd64.deb
```

(continues on next page)

(continued from previous page)

```
sudo apt-key add /var/cuda-repo-ubuntu2004-11-6-local/7fa2af80.pub
sudo apt-get update
sudo apt-get -y install cuda
```

You should then reboot your machine for everything to take full effect

Once installed you can confirm that it is working by writing:

```
nvidia-smi

and

nvcc --version
```

If both commands return a healthy output, you are all set on CUDA.

Barcoding Kits Missing

Newer installs of Minknow will not render barcoding kits appropriately on starting an analysis or sequencing run. To remedy, you must fix 2-3 files

1. `sudo nano /opt/ont/minknow/conf/app_conf` and replace `use_tcp` to `true`
2. `sudo systemctl edit guppyd.service` and replace `--port` with `5555` and add `--use_tcp`
3. `sudo mv /etc/systemd/system/guppyd.service.d/override.conf /etc/systemd/system/guppyd.service.d/override.conf.old` (if it exists)
4. Finally, run `sudo systemctl daemon-reload` to make changes.

Note: Latest Installs of MinKNOW break GPU-basecalling. There is no fix (we've) discovered that allows it to perform within MinKNOW directly

---

Note: `/etc/systemd/system/guppyd.service.d/override.conf` may not exist on your system and won't be needed to be changed

---

Guppy GPU Basecaller

Finally, you need to configure MinKNOW to use a GPU-capable version of guppy and that the guppy basecaller plays nice with the installed MinKNOW you've pulled.

```
/opt/ont/minknow/guppy/bin/guppy_basecaller --version
```

You should see a version, for example for 5.0.13. In MinKNOW `21.11` it should be around `5.1.15` You MUST download the same version by running:

`wget https://mirror.oxfordnanoportal.com/software/analysis/ont-guppy_<version>_linux64.tar.gz`

Make sure to replace the installed version with the values after `ont-guppy_` e.g. `wget https://mirror.oxfordnanoportal.com/software/analysis/ont-guppy_5.1.15_linux64.tar.gz`

Then, we need to replace the guppy version. Let's first save the cpu-only one before replacing as well.

```
sudo mv /opt/ont/guppy/bin /opt/ont/guppy/bin.sav  &&    sudo mv /opt/ont/guppy/data /
↪opt/ont/guppy/data.sav      # Save the old guppy just in case
tar -xvzf ont-guppy_5.1.15_linux64.tar.gz #Decompress guppy. Replace the version number␣
↪with your own
sudo cp -r ont-guppy/bin /opt/ont/guppy/bin && sudo cp -r ont-guppy/data /opt/ont/guppy/
↪data # Move the newly downloaded guppy
#Disable online need for minknow to ping external servers
sudo /opt/ont/minknow/bin/config_editor --filename /opt/ont/minknow/conf/sys_conf --conf␣
↪system --set on_acquisition_ping_failure=ignore
sudo service minknow restart # Restart minknow
```

Then, add these two lines to your *$HOME/.bashrc*

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64\
                       ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
export PATH=/usr/local/cuda/bin:$PATH
```

Note: Add this to your bashrc for the user if you want to run guppy gpu from the command line

Warning: If you perform the above steps and you still can't access/see the GPU with `nvidia-settings` or `nvidia-smi`, you might need to disable secure boot Here or here are examples of how to do that. You need to enter BIOS to perform this operation

1. Boot and press [F2] to enter BIOS.
2. Go to [Security] tab > [Default Secure boot on] and set as [Disabled].
3. Go to [Save & Exit] tab > [Save Changes] and select [Yes].
4. Go to [Security] tab and enter [Delete All Secure Boot Variables] and select [Yes] to proceed.
5. Then, select [OK] to restart.

OR

1. Open the PC BIOS menu:

   You can often access this menu by pressing a key while your PC is booting, such as F1, F2, F12, or Esc.

   Or

   From Windows, hold the Shift key while selecting Restart. Go to Troubleshoot > Advanced Options: UEFI Firmware Settings.

2. Find the Secure Boot setting in your BIOS menu. If possible, set it to Disabled. This option is usually in either the Security tab, the Boot tab, or the Authentication tab.
3. Save changes and exit. The PC reboots.

Note: As of 21.06, MinKNOW requires an additional step to add CUDA capability (GPU processing) to basecalling and can be found here

See here

In short, the fix quoted at the link states that it requires about 10 steps:

1. Use systemctl to edit the existing guppyd service (this will open a text editor with a copy of the existing service file):

```
sudo systemctl edit guppyd.service --full
```

Ensure that, if it exists, the override conf doesn't override our changes

```
sudo mv /etc/systemd/system/guppyd.service.d/override.conf /etc/systemd/system/guppyd.
service.d/override.conf.old
```

2. Edit that new service file to point to your GPU version of guppy, and add the appropriate device flag. You can change any other server arguments at the same time.

For example, change this line in the service file:

```
ExecStart=/opt/ont/guppy/bin/guppy_basecall_server <things>
```

…to this (make sure you retain the `--port` argument exactly as it used to be – this is how MinKNOW communicates with the basecall server):

```
ExecStart=/home/myuser/ont-guppy/bin/guppy_basecall_server <things> -x cuda:all
```

3. Save the file and exit the text editor (the filename may look odd, but don't worry – systemctl should change it to the correct name later).
4. Do the same for `/etc/systemd/system/guppyd.service.d/override.conf` (edit with the addition of *-x cuda:all*)
5. Stop the MinKNOW service, as described in the documentation.
6. Stop the guppyd service.

```
sudo service guppyd stop
```

7. Check that guppy is no longer running, as described in the documentation, killing any existing basecall servers as required.
8. Start the guppyd service.

```
sudo service guppyd start
```

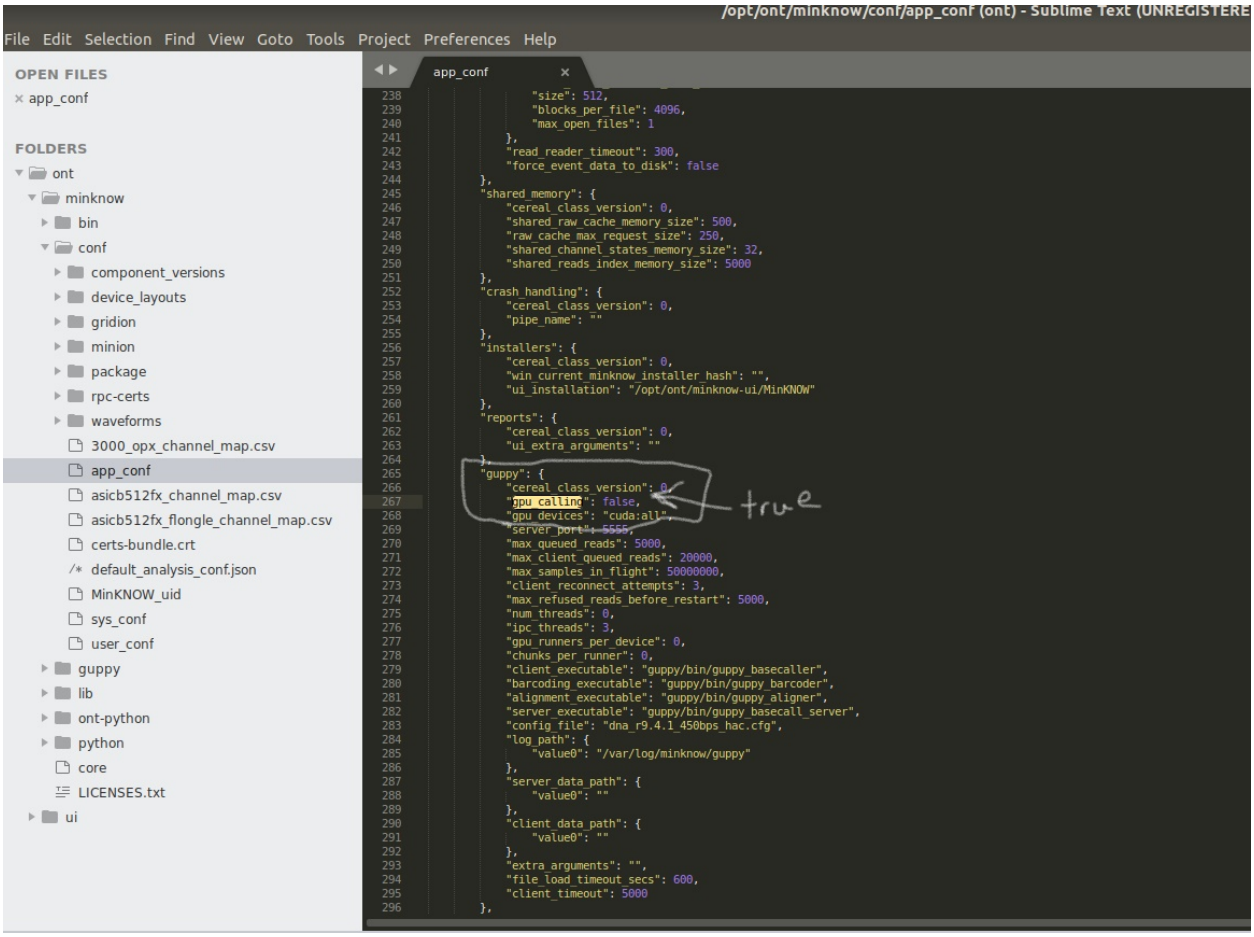
9. Check that the correct version of guppy is running, as described in the documentation. If the guppy basecall server isn't launching correctly, check its log output using journalctl ("-n 100" shows the last 100 entries in the journal) to see what's going wrong:

```
sudo journalctl -u guppyd.service -n 100
```

10. Start the MinKNOW service.

```
sudo service minknow restart
```

You will also need to adjust the configuration file for guppy by modifying `/opt/ont/minknow/conf/app_conf`. Adjust the `gpu_calling` field to true in the JSON, being careful not to modify/delete any commas or quotations.



From there you are all set to run basecalling directly within the MinKNOW application.

If you ever experience issues where the UI does not show experiments once started (Basecalling or Sequencing), try:

1. Close MinKNOW (UI)
2. *sudo service minknow restart*
3. Make sure that a MinION or other Oxford Nanopore devices is plugged in and running
4. Restart MinKNOW (UI)
5. Re-attempt experiment such as basecalling. Often times experiments will then show up

Reduce Runners if GPU basecalling fails

Note: See here

Occasionally, if you've set up GPU basecalling correctly, but still get errors, this may be due to too many runners being called. You can adjust this easily in the `app_conf` file. Adjust the `"chunks_per_runner"` parameter

- If using HAC, set it to `"chunks_per_runner": 160`
- If using SUP, set it to `"chunks_per_runner": 10`

Running Command-line GPU Basecalling

Note: Ensure you've installed a GPU compatible variant by following *Guppy GPU Basecaller*.

```
guppy_basecaller -x cuda:all -i <fast5_folder> -r -s ./fastq_pass -c dna_r9.4.1_450bps_
↪hac.cfg
```

Note: If you get an error (or it doesn't exist) about `guppy_basecaller` not supporting GPU calling, please place the binaries in your path.

If you set up MinKNOW to use guppy GPU, ensure that it is properly working and setup in `/opt/ont/guppy`. Follow *Guppy GPU Basecaller* to set this up Ultimately, you need to run the `ont-guppy/*` contents into `/opt/ont/guppy/`

To check your status of your GPU (ensure CUDA is installed by following *CUDA*) by running `nvidia-smi`

Reinstalling MinKNOW

If you experience issues with MinKNOW, one potential solution is to do a purge of MinKNOW and Reinstalling

Notes (below) are gathered from the Oxford Community Forums here

1. First purge MinKNOW and remove its dependencies with these two commands:

```
sudo apt purge minion-nc

sudo apt autoremove
```

2. Check that there are no residual config files left over:

```
dpkg --list | grep -e minknow -e minion -e guppy
```

3. If Step 2 returns any results, please manually purge those packages like so:

sudo apt purge package1 package2 package3 etc

4. Delete the minknow installation directory:

```
sudo rm -rf /opt/ont/minknow
```

5. Ensure there are no Guppy files or folders present in the following directory:

```
ls -l /etc/systemd/system/
```

6. If any Guppy files or folders are seen in Step 5, please delete them before continuing.
7. After rebooting your computer, update the package listing and install MinKNOW:

```
sudo apt update

sudo apt install minion-nc
```

8. Once MinKNOW is installed, please open MinKNOW and start a test sequencing run with CPU basecalling to ensure it is working as expected before configuring your GPU. You can start an experiment with a CTC or used flow cell for the purposes of this test.

9. To configure your GPU, please follow Steps 1-15 on the this page of the MinKNOW user guide. Based on user feedback, we've updated it to make a few steps a bit clearer and I've confirmed on my own laptop that these work for this patch.

Supplemental

Note: These tools are optional and not needed to run any Basestack feature They are not bundled within Basestack (supplemental to bioinformatic/genomics analysis) and must be downloaded separately. They are (mainly) discussed in the bioinformatics modules of the workshop provided with Basestack

---

- **BEAST v1.10.4:**
  - https://beast.community/
  - Available in all 3 OS types
- **MEGA X**
  - https://www.megasoftware.net/
  - Available in all 3 OS types
- **AliView v1.26:**
  - https://ormbunkar.se/aliview/downloads/linux/linux-version-1.26/
  - Available in all 3 OS types
- MAFFT v7
- https://mafft.cbrc.jp/alignment/software/
- Available in all 3 OS types
- BEAGLE v3.1.2

build from source (instructions)

- https://github.com/beagle-dev/beagle-lib/
- Available in all 3 OS Types

- **TempEst v1.5.3**
  - http://tree.bio.ed.ac.uk/software/tempest/
  - Available in all 3 OS Types
- **Tracer v1.7.1**
  - https://github.com/beast-dev/tracer/releases/tag/v1.7.1
  - Available in all 3 OS Types
- **FigTree v1.4.4**
  - http://tree.bio.ed.ac.uk/software/figtree/
  - Available in all 3 OS Types

Requires Internet. Also available at https://igv.org/app/

- **IQTree v2.1.1**
  - https://github.com/iqtree/iqtree2/releases

  - You will find the executable in the bin sub-folder. Copy all files in bin folder to your system search path such that you can run IQ-TREE by entering iqtree from the Terminal.
  - Available in all 3 OS Types

- **FastTree v2.1.11 (double precision variant)**
  - http://www.microbesonline.org/fasttree/
  - Available in all 3 OS Types (Less user-friendly)
- **Python3**
  - https://www.python.org/downloads/
  - Available in all 3 OS Types
- **biopython**
  - https://biopython.org/wiki/Download
  - Install after installing/updating Python

NVIDIA Jetson Setup

If you need to set up Basestack, you must install the Nightly build variant of the distribution at: https://github.com/jhuapl-bio/Basestack/releases/latest

---

Note: Be sure to set up the Nano or Xavier (NX) properly before doing this. See more information here: https://developer.nvidia.com/embedded/downloads

---

Note: Depending on your distribution, you may already have docker installed it seems as NVIDIA is more fully supporting the Docker build toolkit.

To check, run *docker –version*. If so, skip that section of the supplemental software install process (see below) You will definitely need to be sure to run through the post-installation steps, though. See here: *Docker Installation*

Simply follow these 2 steps:

1. Download the arm64 AppImage from the above-mentioned releases page
2. OPTIONAL: Run the install script. This can be found here

Note: The install script described above will ask several questions for setting up your environment based on your needs. Make sure to select the `r` option for any question asking for `arm64` or `amd64`

Also, try to follow steps in this link for minknow Or, try this link if you have troubles with CUDA and Guppy

---

Warning: The later versions of MinKNOW can lead to a failure to load the barcoding kits in the UI. If you experience this, follow this procedure *Barcoding Kits Missing* to remedy

---

Note: JetPack (Nviida Jetson only) does not ship with `nvidia-smi` as a way to monitor gpu usages. You can instead run `sudo -H pip install -U jetson-stats` to get the `jtop` command to monitor resources on your jetson device

---

Special Mentions

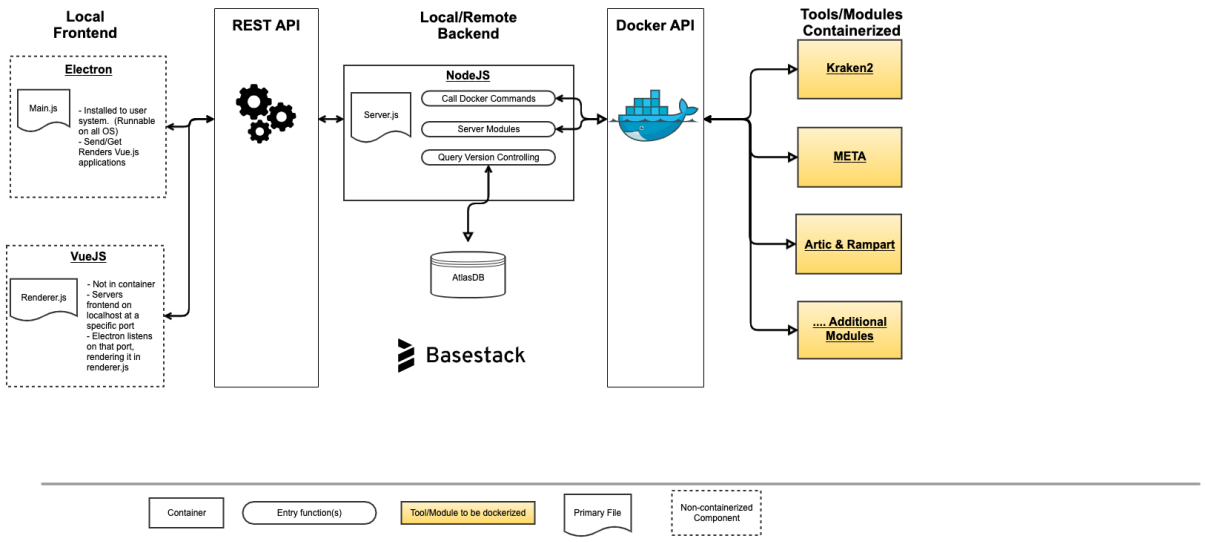
Mentions ###

- Andrew Rambaut, James Hadfield and Team @ artic-network
- Trevor Bedford and Team @ Nextstrain
- UC San Diego and Broad Inst. @ IGV

Authors of Basestack ##

- Brian Merritt (brian.merritt@jhuapl.edu)

3.3 API



3.3.1 Development Setup of Server and App

```
conda activate basestack
npm run dev:server
npm run dev:app (if running the app in parallel)
```

3.3.2 Components

Modules

Basestack uses these calls for anything involving modules specifically in the app ecosystem:

ping POST

Note: To get the status of the server at the specified port, you can use the `curl` or Postman to test your calls.

```
curl localhost:5003/server/ping
```

getAllCatalog GET

```
curl localhost:5003/catalog/all/get
```

Gets all available catalogs, both remote and/or installed

Parameters

Returns

`Array` - List of all catalog entries available either remotely situated or locally available

Example


```
{
  "icon": "dna",
  "title": "Minimap2",
  "tags": [
    "minimap2",
    "alignment",
    "genomics"
  ],
  "status": {
    "installed": true,
    "latest": null,
    "building": true,
    "version": null,
    "running": false,
    "error": null
  },
  "name": "minimap2",
  "remotes": [],
  "modules": [
    {
      "status": {
        "fully_installed": true,
        "latest": null,
        "building": true,
        "version": null,
```


(continues on next page)

(continued from previous page)

```
        "partial_install": true,
        "running": false,
        "error": null
      },
      "name": "minimap2",
```

getInstalledCatalog `GET`

```
curl localhost:5003/catalog/installed/get
```

Gets all installed catalogs, both remote and/or locally found in the app

Parameters

Returns

`Array` - List of installed catalog entries available either remotely situated or locally available

Example

```
{
  "icon": "dna",
  "title": "Minimap2",
  "tags": [
    "minimap2",
    "alignment",
    "genomics"
  ],
  "status": {
    "installed": true,
    "latest": null,
    "building": true,
    "version": null,
    "running": false,
    "error": null
  },
  "name": "minimap2",
  "remotes": [],
  "modules": [
    {
      "status": {
        "fully_installed": true,
        "latest": null,
        "building": true,
        "version": null,
        "partial_install": true,
```

(continues on next page)

(continued from previous page)

```
        "running": false,
        "error": null
      },
      "name": "minimap2",
```

---

moduleBuildDependency POST

```
localhost:5003/module/build/dependency
```

Installs a single dependency belonging to a module's procedure (version)

Parameters

1. • `dependency Int` - Dependency index written for a procedure in the configuration of it
2. • `catalog String` - Catalog name
3. • `module Int` - Index of the module version for a catalog entry
4. • `procedure Int` - Index of the procedure in the module

Returns

`Status` - Returns successful kickoff of installation or Error

Example Body

```
{
    "procedure": 1,
    "catalog": "mytax",
    "module": 0,
    "dependency": 0
}
```

---

procedureBuild POST

```
localhost:5003/procedure/build
```

Installs all dependencies belonging to a procedure

Parameters

1. • `catalog String` - Catalog name
2. • `module Int` - Index of the module version for a catalog entry
3. • `procedure Int` - Index of the procedure to completely install

Returns

`Status` - Returns successful kickoff of installation or Error

Example Body

```
{
    "procedure": 0,
    "catalog": "mytax",
    "module": 0
}
```

---

moduleBuild `POST`

```
localhost:5003/module/build
```

Installs all procedures and their corresponding dependencies for a given catalog's module

Parameters

1. • `catalog String` - Catalog name
2. • `module Int` - Index of the module version for a catalog entry

Returns

`Status` - Returns successful kickoff of installation or Error

Example Body

```
{
    "catalog": "mytax",
    "module": 0
}
```

Procedures

Note: To get the status of the server at the specified port, you can use the `curl` or Postman to test your calls.

getProcedures `GET`

```
curl localhost:5003/procedures/get/:catalog/:module/:token
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

Returns

`Array` - List of all procedures that belong to a specific version of a module in the catalog

Example

```
{
  "status": {
    "error": null,
    "stream": null,
    "running": false,
    "fully_installed": true,
    "partial_install": true
  },
  "dependencies": [
    {
      "target": "jhuaplbio/basestack_mytax",
      "type": "docker",
      "version": "latest",
      "format": "docker",
      "status": {
        "downloading": false,
        "decompressing": false,
        "exists": {
          "version":
↪"sha256:4762c42837c0fcffcb36a08398d7aafff57de72584804e797a63bc28ddd9c9b1"
        },
        "error": null,
        "stream": null,
        "fully_installed": false,
        "partial_install": false,
        "version":
↪"sha256:4762c42837c0fcffcb36a08398d7aafff57de72584804e797a63bc28ddd9c9b1"
      }
    },
```

runProcedure POST

```
localhost:5003/procedure/run
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

Parameters

1. `module` `Int` - Index of the version of the module
2. `catalog` `String` - Name of the specific Catalog
3. `variables` `Object` - Object of all unique variable names to assign to 1 or more services
4. `token` `String` - Autogenerated token on startup of Basestack
5. `procedure` - `Int` - Index of the procedure for the module. Comprises 1 or more services

Returns

`Status` - Success or Error on procedure (1 or more service) completion

cancelProcedure POST

```
localhost:5003/procedure/cancel
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

Parameters

1. `module` `Int` - Index of the version of the module
2. `catalog` `String` - Name of the specific Catalog
3. `token` `String` - Autogenerated token on startup of Basestack
4. `procedure` - `Int` - Index of the procedure for the module. Comprises 1 or more services

Returns

`Status` - Success or Error on cancelation of the procedure (1 or more service)

Jobs

Note: To get the status of the server at the specified port, you can use the `curl` or Postman to test your calls.

---

getJob `GET`

```
curl localhost:5003/service/get/:catalog/:module/:procedure/:service/:token
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

---

Returns

`Array` - List of all services that belong to a specific version of a module's procedure in the catalog

Example

```
curl localhost:5003/service/get/mytax/0/1/0/development

{
"status": 200,
"message": "Completed job setting",
"data": {
    "exists": false,
    "fully_installed": true,
    "procedure": {
        "error": null,
        "stream": null,
        "running": false,
        "building": false,
        "fully_installed": true,
        "partial_install": true
    },
    "dependencies": [
        {
            "downloading": false,
            "decompressing": false,
            "dependComplete": true,
            "exists": {
                "version":
```

(continues on next page)

(continued from previous page)

```
↪"sha256:4762c42837c0fcffcb36a08398d7aafff57de72584804e797a63bc28ddd9c9b1"
            },
            "error": null,
            "stream": null,
            "fully_installed": fals
```

startJob POST

```
localhost:5003/job/start
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

Starts a job, sets necessary variables, designates services and beings the Docker API communication processes. Multi-service procedures will run all services sequentially

Parameters

1. `module` `Int` - Index of the version of the module
2. `catalog` `String` - Name of the specific Catalog
3. `variables` `Object` - Object of all unique variable names to assign to 1 or more services
4. `token` `String` - Autogenerated token on startup of Basestack
5. `procedure` - `Int` - Index of the procedure for the module. Comprises 1 or more services
6. `service` - `List of Int` - Index of the services for the selected procedure.

Returns

`Status` - Success or Error on procedure (1 or more service) completion

Note: Test data download from Gdrive location

Example Body

```
{
  "service": 0,
    "catalog": "mytax",
    "module":0,
    "procedure": 0,
    "token": "development",
```

(continues on next page)

(continued from previous page)

```
    "variables": {
        "file": {
            "source": "/Users/merribb1/Desktop/test-data2/metagenome/sample_metagenome.
↪fastq"
        },
        "db": {
            "option": 1
        }
    }
}
```

cancelJob `POST`

```
localhost:5003/job/cancel
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

Stops the running job. Keeps it present in the backend but it no longer runs its services (and cancels any currently running ones)

Parameters

1. `module` `Int` - Index of the version of the module
2. `catalog` `String` - Name of the specific Catalog
3. `token` `String` - Autogenerated token on startup of Basestack
4. `procedure` - `Int` - Index of the procedure for the module. Comprises 1 or more services

Returns

`Status` - Success or Error on cancelation of the service

setJob `POST`

```
localhost:5003/job/set
```

Note: Where token is a uniquely generated token on accessing the app. Default `development` is always generated

Sets the variables and creates a job for a given procedure that is being viewed from the application piece of the app

Parameters

1. `module` `Int` - Index of the version of the module
2. `catalog` `String` - Name of the specific Catalog
3. `variables` `Object` - Object of all unique variable names to assign to 1 or more services
4. `token` `String` - Autogenerated token on startup of Basestack
5. `procedure` - `Int` - Index of the procedure for the module. Comprises 1 or more services
6. `services` - `List of Int` - Index of the service(s) for the procedure.

Returns

`Status` - Success or Error on setting job (1 or more services) completion

Example Body

```
setJob ``POST``

{
    "token": "development",
    "procedure": 0,
    "catalog": "mytax",
    "module": 0,
    "services": [ 0 ],
    "variables": {
        "file": {
            "source": "/Users/merribb1/Desktop/test-data2/metagenome/sample_metagenome.
↪fastq"
        },
        "db": {
            "option": 1
        },
        "nodes": {
            "option": 0
```

System

Note: To get the status of the server at the specified port, you can use the `curl` or Postman to test your calls.

Pinging the Server GET

Note: You can use Postman as your API management toolkit for testing the server

`curl localhost:5003/server/ping`

returns

```
{"status":200,"message":"Server is running at port: 5003"}
```

getServerLogs GET

```
curl localhost:5003/log/system
```

Returns

`Array` - List of all services that belong to a specific version of a module's procedure in the catalog

Example

```
curl localhost:5003/log/system

{
"status": 200,
"message": "Got system log",
"data": [
    "2022-02-14T23:17:12.235Z [info]: stdout: Indel at position 29094: [0, 75, 0, 0, 0,␣
↪13, 0] 75",
    "Indel at position 29130: [0, 0, 1, 77, 0, 4, 15] 78",
    "",
    "2022-02-14T23:17:12.236Z [info]: stdout: Indel at position 29323: [230, 0, 0, 0, 0,␣
↪35, 18] 230",
    "Indel at position 29376: [0, 0, 0, 97, 0, 2, 41] 97",
    "",
    "2022-02-14T23:17:12.237Z [info]: stdout: Indel at position 29385: [0, 125, 0, 0, 0,␣
↪21, 41] 125",
    "",
    "2022-02-14T23:17:12.237Z [info]: stdout: Indel at position 29426: [2, 0, 73, 0, 0,␣
↪13, 0] 75",
    "",
    "2022-02-14T23:17:12.237Z [info]: stdout: Indel at position 29753: [0, 110, 0, 0, 0,␣
↪3, 18] 110",
    "",
    "2022-02-14T23:17:12.238Z [info]: stdout: Indel at position 29775: [92, 0, 0, 0, 0,␣
↪7, 16] 92",
    "",
    "2022-02-14T23:17:12.239Z [info]: stdout: Indel at position 29799: [1, 0, 119, 0, 0,␣
```

(continues on next page)

(continued from previous page)

```
↪3, 35] 120",
    "",
    "2022-02-14T23:17:12.277Z [info]: stdout: -[2022-02-14 23:17:12] Starting Module 4␣
↪Merging and Allele Frequencies on     /opt/data/artic-pipeline/4-draft-consensus/
↪Sample3_NB03.nanopolish.merged.vcf, /opt/data/artic-pipeline/4-draft-consensus/Sample3_
↪NB03.medaka.merged.vcf, /opt/data/artic-pipeline/4-draft-consensus/Sample3_NB03.
↪samtools.vcf",
    "",
    "2022-02-14T23:17:13.124Z [info]: stdout: ^[2022-02-14 23:17:13] SAMPLE Sample3_
↪NB03: Module 4 Samtools and Merging: processing complete",
    "",
```

Note: Above logs were delivered during a run of a *Consensus pipeline* procedure