

SNPcaster

Installation &

Operation

Manual

Feb. 25, 2026

Version Supported:

v.0.9.6 or later

Disclaimer

The information in this manual has been carefully prepared to ensure accuracy.

However, we do not guarantee that it is completely free of errors.

The authors accept no responsibility for any damage or losses that may result from the use of information contained in this manual.

Table of Contents

Table of Contents	2
0. Quick Start.....	6
1. System Requirements and Basic Usage of the Analysis Machine	7
1.1. System Requirements	7
1.2. Command-Line Applications.....	8
1.2.1. Windows	8
1.2.2. Mac	11
1.2.3. Linux.....	11
1.3. Command-Line Operations	12
1.4. Directory Overview and Navigation	12
1.4.1. What Are Directories and Paths?.....	12
1.4.2. Opening a Command-Line Application in a Specific Directory.....	13
1.4.3. Changing the Current Directory	15
2. Installing Docker	18
2.1. Installing Rancher Desktop	19
2.1.1. Windows	19
2.1.2. Mac	25
2.1.3. Rancher Desktop Startup Configuration	29
2.1.4. Verifying Rancher Desktop Operation	30
2.2. Installing Docker Engine	31
2.2.1. Windows	31
2.2.2. Mac	33
2.2.3. Linux.....	38
2.2.4. Docker Engine Operation Verification.....	42
3. Setting Up the SNPcaster Environment	43
3.1. Downloading SNPcaster.....	43
3.1.1. Download via Web Browser (e.g., Microsoft Edge, Google Chrome)..	43

3.1.2. Download via Command Line (Git).....	44
3.2. (Reference) Folder and File Structure	45
3.3. Installing SNPcaster on the Analysis Terminal.....	48
3.3.1. Create and edit the .env file	48
3.3.2. Running the Build	49
3.3.3. Troubleshooting for Build Failures	51
3.4. Upgrading SNPcaster	57
3.4.1. For installations via a web browser (Google Chrome, etc.)	57
3.4.2. For installations via the Command Line (Git)	59
4. How to Start and Stop the System	61
4.1. How to Start	61
4.1.1. Execute the Start Command	61
4.1.2. Port Forwarding Configuration (For Remote Access Only)	62
4.1.3. Accessing Jupyter Lab	63
4.2. How to Stop SNPcaster Docker Container	63
5. Analysis Execution Procedure.....	65
5.1. Folder Structure	65
5.2. Learn How to Use Jupyter Lab	66
5.2.1. Jupyter Lab Tutorial	66
5.2.2. Basic Usage of Jupyter Lab	67
5.3. Creating a Project.....	69
5.3.1. Creating a New Project.....	69
5.3.2. Executing the Creation of a New Project	71
5.3.3. Running the Analysis.....	72
5.3.4. Basic Usage	74
5.3.5. Execution of SNP Reanalysis.....	78
6. Command Line Operations	80
6.1. Launching the Command Input Application	80
6.1.1. Windows	80
6.1.2. Mac	80
6.1.3. Linux.....	80
6.2. Creating Docker Images and Containers	81
6.3. Run from the Docker Container	81
6.3.1. Data Placement	81
6.3.2. Access the Container via Bash	81

6.4.	Run from the Docker Image.....	82
6.4.1.	Data Placement	82
6.4.2.	Check the Image ID	82
6.4.3.	Execute the Command	83
7.	Program Overview	85
7.1.	SNPcaster.....	85
7.1.1.	Advantages of the Program.....	85
7.1.2.	Analysis Workflow.....	87
7.1.3.	Output Files	90
7.2.	grape_qc_assembly	95
8.	Troubleshooting and Q&A.....	97
9.	reference materials	100
9.1.	About Docker	100
9.2.	How to Install Docker Desktop	102
9.2.1.	Windows	102
9.2.2.	Mac	105
9.3.	How to Uninstall Docker Desktop	109
9.3.1.	Backup of Existing Containers and Images	109
9.3.2.	Uninstalling Docker Desktop.....	111
9.3.3.	Restoring the Backup	111
9.4.	How to Uninstall Rancher Desktop	112
9.4.1.	Windows	112
9.4.2.	Mac	115
10.	Authors and Citation	117
10.1.	Authors	117
10.2.	Citation	117

Revision History

Ver.	Date	Summary of Revisions
1.3	2026-02-25	Modified to be compatible with v 0.9.6
1.2	2025-12-22	Modified to be compatible with v 0.9.5 Added detailed explanations based on user feedback
1.1	2025-05-28	Updated based on changes to the distribution method. Now compatible with version v.0.9.1
1.0	2024-06-07	Initial release

0. Quick Start

First, follow the steps below to install the necessary software and perform a basic analysis.

In [1.System Requirements and Basic Usage of the Analysis Machine](#), you can check the required specifications for the analysis terminal and the basics of command-line input.

If you are already familiar with command-line operations, skip this section.

Follow [2.Installing Docker](#) to install Docker.

Then, proceed with [3.Setting Up the SNPcaster Environment](#) to install SNPcaster.

Follow [4.How to Start and Stop the System](#) to launch the Docker environment for SNPcaster.

Finally, refer to [5.Analysis Execution Procedure](#) to execute the analysis. Start by running the analysis using SNPcaster_quickstart.ipynb.

Note: This manual mainly focuses on the installation of SNPcaster.

Detailed analysis procedures and output file explanations are included within the Jupyter Notebook itself, so please refer to it.

(For basic instructions on using Jupyter Notebooks, refer to [5.Analysis Execution Procedure](#) section.)

1. System Requirements and Basic Usage of the Analysis Machine

1.1. System Requirements

We recommend the following specifications:

(These are not mandatory. Please refer to them only if the program fails to run.)

- OS: Windows 10 or later, macOS, or Linux
Note: For Linux, functionality has been confirmed only on Ubuntu.
- Memory: 40 GB or more recommended (10 GB may also be sufficient)
- CPU: Performance equivalent to Intel Core series or higher, or Apple chips (M1, M2, etc.)
- Available disk space on the drive where Docker is installed: 50 GB or more
(Installing SNPCaster will consume approximately 20 GB)

1.2. Command-Line Applications

To use this program, some command-line operation is required.

Here, we explain the basics of entering commands.

Various command-line applications can be used depending on the OS.

In this manual, we refer to all of them collectively as **Command-Line Applications**.

1.2.1. Windows

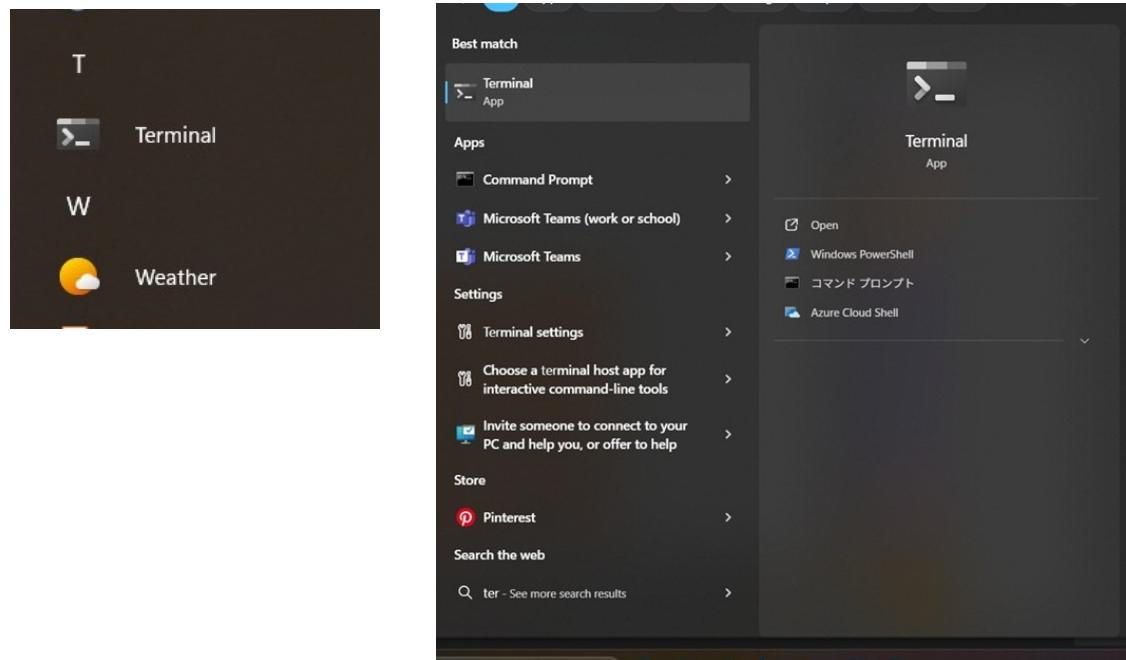
1.2.1.1. Command-Line Applications

Available options include Windows PowerShell and Windows Terminal (often shown as “Terminal”).

Note that Windows Terminal may not be installed by default. In that case, install it from the Microsoft Store.

Windows Terminal is recommended because it allows the use of the “Open in Terminal” function, which is described later.

(It comes pre-installed in Windows 11.)

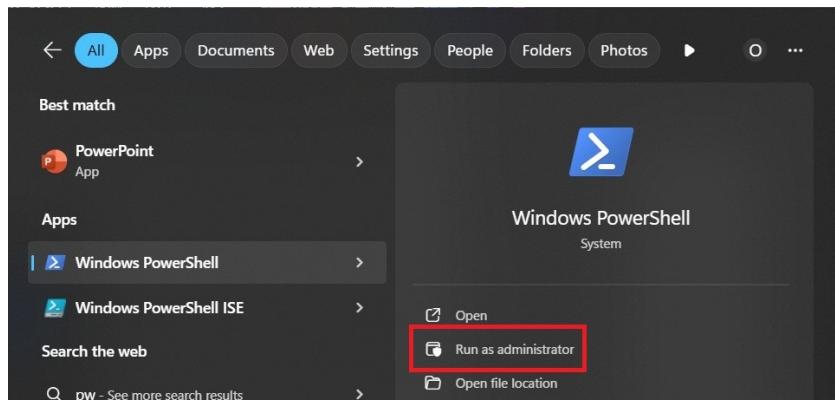


1.2.1.2. Installing Windows Subsystem for Linux (WSL)

On Windows, Docker runs on WSL, so in this section, we will enable WSL2.

Note: This section includes memory settings, so even if you have already installed WSL, please read through it.

First, run Windows PowerShell or Windows Terminal as an administrator.



If you have never launched WSL before, execute the following command to initialize it:

```
wsl --install -d Ubuntu-24.04
```

(The term "Ubuntu" refers to a Linux distribution. While it is expected to work with other distributions as well, we have only tested it on Ubuntu.)

On the first launch, you will be prompted to enter a username as shown below.

Please enter your desired username and press Enter.

A few seconds later, you will be prompted to enter a password (**Characters will not be displayed as you type for security**).

Enter the password and press Enter.

Note: Please make sure to remember this password, as it will be required during the installation process.

```
PS C:\Users\t-yamagishi> wsl --install -d Ubuntu
Ubuntu は既にインストールされています。
Ubuntu を起動しています...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: test
New password: [REDACTED] → パスワードは入力した文字が表示されません
```

You will be prompted to re-enter the password for confirmation.
Please enter the password again and press Enter.
If the password is correct, the installation will be completed.

```
Retype new password: [REDACTED]
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

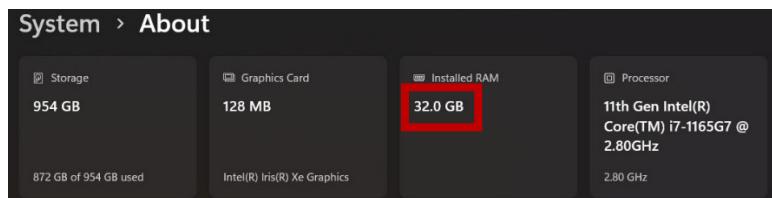
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Tue Dec 10 15:27:37 JST 2024
```

Next, you will configure memory allocation for WSL.
By default, WSL is limited to a maximum of 8GB of memory, so we need to change this setting.
Open File Explorer and navigate to C:\Users\[YourUsername] (replace [YourUsername] with your actual Windows username). Create a file named **.wslconfig** (if the file already exists, open it for editing).
Open **.wslconfig** in a text editor (such as Notepad) and enter the following content:

```
[wsl2]
memory={DesiredMemorySize}GB
```

In **{DesiredMemorySize}**, enter the amount of memory (in numeric form) that you want to allocate to Docker.
This value should be at least 1GB less than your PC's total memory capacity to prevent Docker from consuming all available memory.
Note: It is recommended to allocate at least 16GB of memory.
To check your system's total memory, right-click the Windows logo and select System.
On the right side of the screen, look for Installed RAM—this is your total available memory.



To apply the changes made to the `.wslconfig` file, open a terminal and run the following command to restart WSL:

```
wsl --shutdown  
wsl -d Ubuntu-24.04
```

WSL is now fully configured and ready for use.

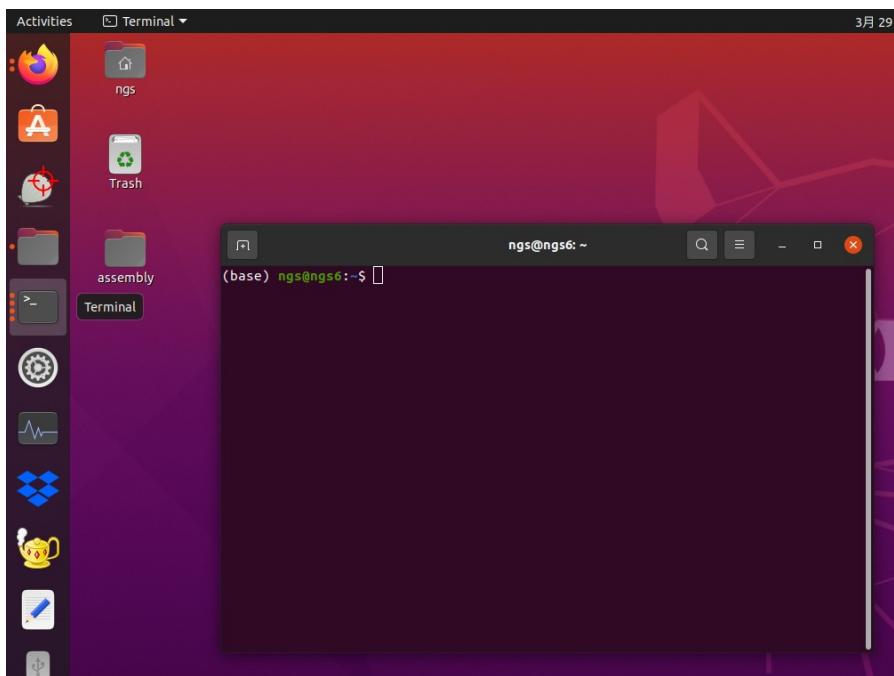
1.2.2. Mac

The **Terminal** is available by default in the initial setup.



1.2.3. Linux

The Terminal is available by default in the initial state.



1.3. Command-Line Operations

In this manual, command-line input (the part entered into a terminal or command prompt) is displayed using gray shading, as shown below.

```
$ echo hello world
```

The ‘\$’ symbol indicates the beginning of a command and should not be typed. In the example below, only echo hello world should be entered.

The output resulting from the execution of a command is shown within a black-bordered box like the one below.

Please note that the actual output may vary depending on your system environment.

```
hello world
```

1.4. Directory Overview and Navigation

1.4.1. What Are Directories and Paths?

A directory is what Windows users commonly refer to as a “folder” — a place where files are stored.

A path refers to the location of a file or directory within the file system.

When working in the command line, it’s important to be aware of your current working directory (also known as the current directory).

For example, if a file is located in the current directory, you can refer to it simply by its file name (this is called a relative path).

However, if the file is located in a different directory, you cannot access it using just the file name — you’ll need to specify its absolute path or navigate to its location first.

If you’re unsure, using the absolute path is generally the safest option.

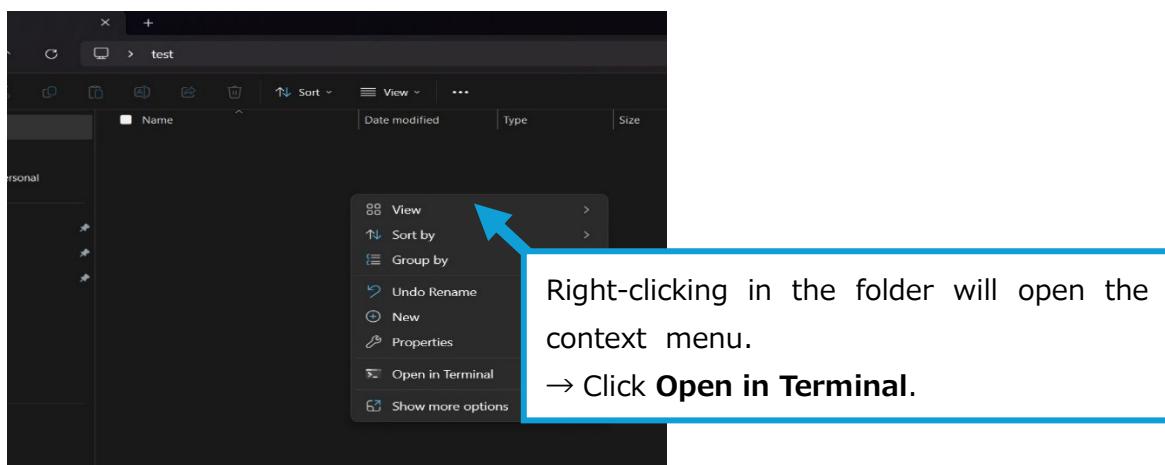
1.4.2. Opening a Command-Line Application in a Specific Directory

To begin working, open a command-line application with the target folder set as the current directory by following the steps below.

1.4.2.1. Windows

Navigate to the folder where you want to work.

Right-click inside the folder and select Open in Terminal from the context menu.

A screenshot of a Windows PowerShell window titled "Windows PowerShell". The terminal output shows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\share\Desktop\test>
```

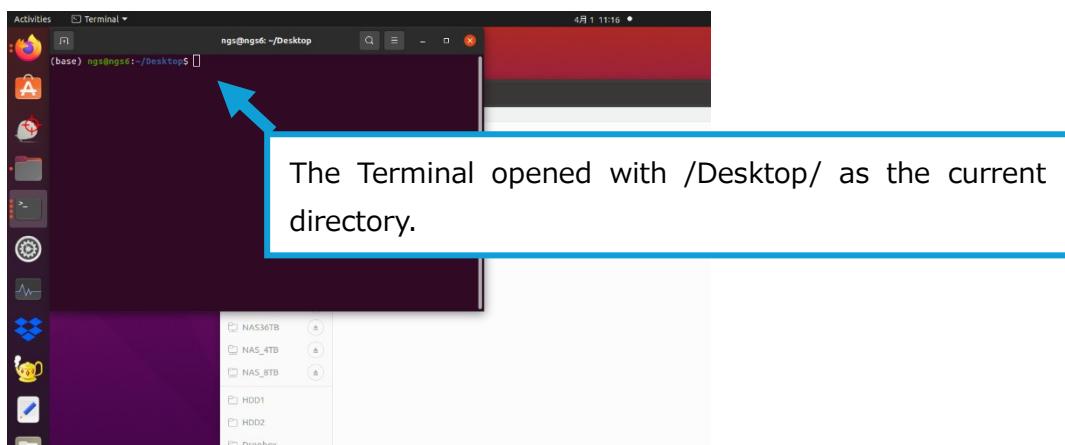
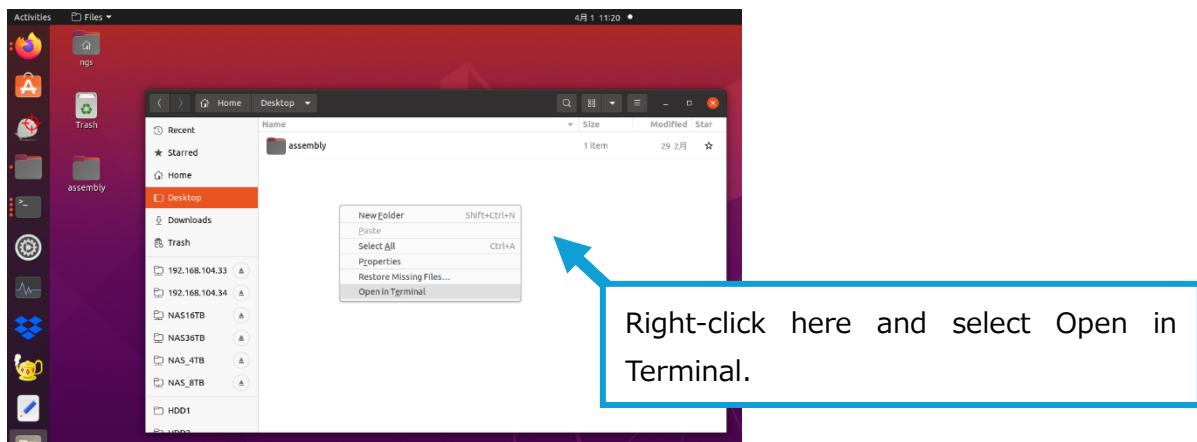
The terminal opened with the selected folder as the current directory.

1.4.2.2. Mac

Similar to Windows, you can open a terminal with the displayed folder set as the current directory by right-clicking the folder in Finder and selecting New Terminal at Folder from the context menu.

1.4.2.3. Linux

Similar to Windows, you can open a terminal with the current folder as the working directory by selecting 'Open in Terminal' from the context menu on the folder.



1.4.3. Changing the Current Directory

Change the current directory as needed. Directory navigation can be done using the [cd] command on Windows, Mac, and Linux alike.

1.4.3.1. Windows

You can change directories using the [cd] command. Absolute paths can be checked from the address bar in File Explorer. However, when using the command-line application, please note the following points:

- Replace backslashes (\) in the address bar path with forward slashes (/).
- To switch to a different drive, you need to enter the path starting with mnt/ followed by the drive letter.

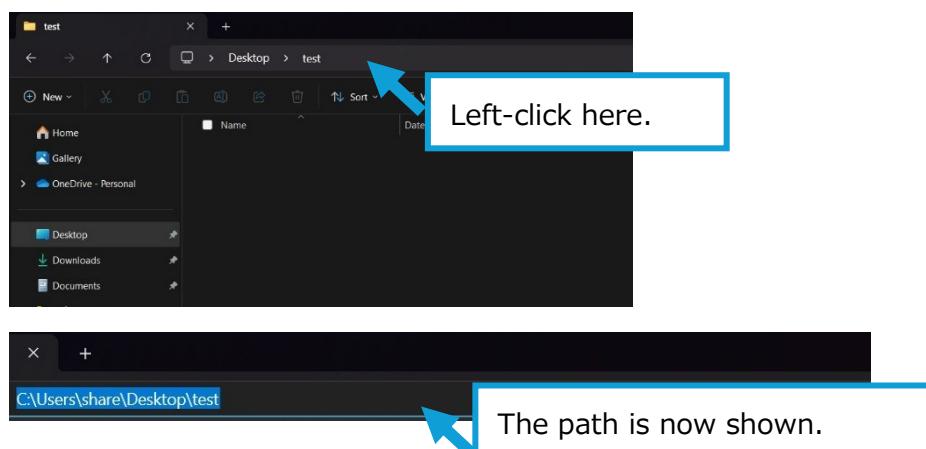
For example, to change the current directory from

C:\Users\share on the C drive to

D:\Genomedata\2024-04-01_analysis on the D drive, enter the following command:

```
$ cd mnt/d/Genomedata/2024-04-01_analysis
```

Note: Please note that the drive paths may vary slightly depending on your environment. Adjust the commands accordingly to fit your setup.

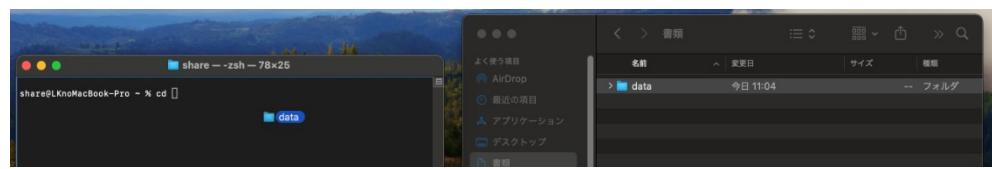


1.4.3.2. Mac

On Mac and Linux, you can change the current directory in almost the same way. Here are two common methods:

- Drag and Drop

By dragging and dropping the target folder onto the terminal window, the absolute path will be entered automatically.



Drag and drop the folder you want into the terminal window.



The absolute path of the folder has been entered.

1.4.3.3. Linux

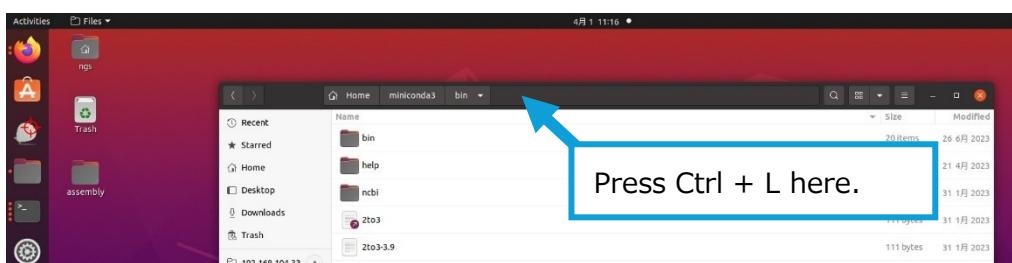
As with Mac, there are two main methods:

- Drag and Drop

Similar to Mac, you can drag and drop the target folder into the terminal to automatically enter its absolute path.

- How to Display the Path

In Ubuntu's default file manager (equivalent to File Explorer on Windows), you can display the path using the following method:



Note: With advanced file managers (such as Nemo, which requires additional installation), you can set the address bar to always display the path.

2. Installing Docker

SNPcaster is designed to run on Docker.

This chapter explains how to install Docker.

The most well-known tool for installing Docker is Docker Desktop, but depending on your organization's size, it may require a paid license.

Therefore, we recommend using Rancher Desktop, which is available free of charge.

Additionally, while installation via command line is required, the core Docker functionality — Docker Engine — is also available for free.

(All information is current as of December 2024.)

Based on the above, please choose the Docker installation method according to the following criteria.

Installing Docker using any one of these methods will enable you to use SNPcaster.

- If you want to install Docker easily and free of charge on Windows or Mac:
→ Proceed to [Installing Rancher Desktop](#).
- If you are a Linux user, comfortable with command-line operations, or had trouble installing Rancher Desktop:
→ Proceed to [Installing Docker Engine](#).
- If you want to use Docker Desktop (either you are exempt from the paid license or already have a license agreement):
→ Instructions for [How to Install Docker Desktop](#) are provided in the respective section.

2.1. Installing Rancher Desktop

Rancher Desktop is an application that provides Docker command-line functionalities along with a graphical interface to manage Docker images and containers. This section describes how to install Rancher Desktop.

* Please refer to Section [9.1. About Docker](#) for details on how Docker works.

* Although there is an installer available for Linux, SNPcaster's operation on Linux with Rancher Desktop has not been verified. Linux users are recommended to use [Installing Docker Engine](#) instead.

**** Important Notice ****

Rancher Desktop does not work properly if it is running simultaneously with Docker Desktop.

If you plan to use Rancher Desktop, please uninstall Docker Desktop by following the instructions in the [9.3. How to Uninstall Docker Desktop](#) section before installing Rancher Desktop.

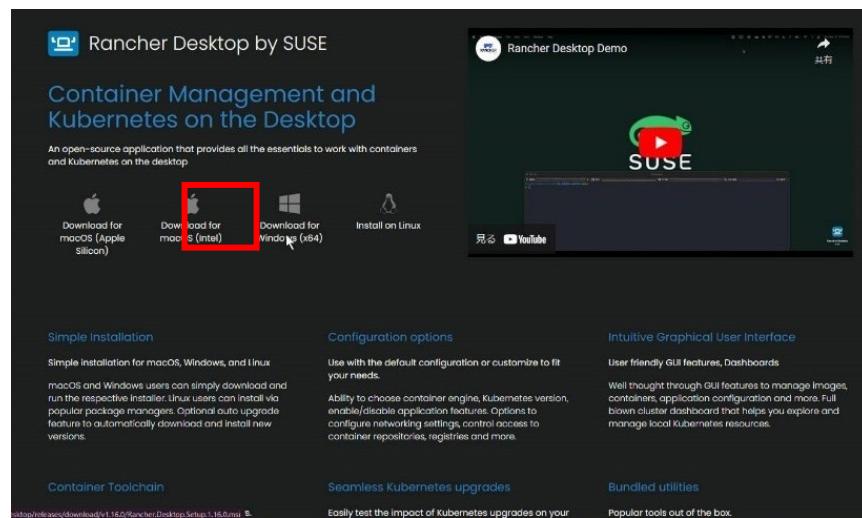
2.1.1. Windows

- 2.1.1.1. If the Windows Subsystem for Linux (WSL) is not yet set up, please complete the WSL installation by following the instructions in the WSL setup section.

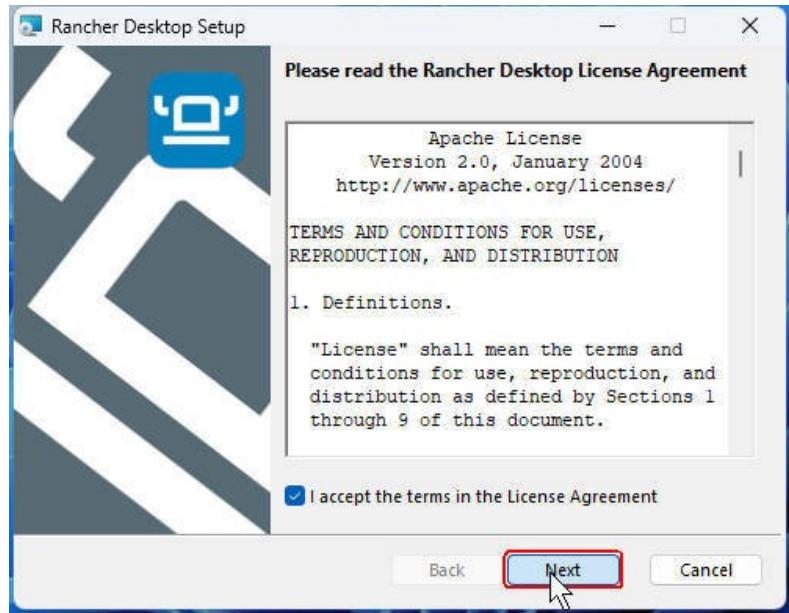
Note: Rancher Desktop is supposed to configure WSL automatically during installation. However, in our testing, installation often failed when WSL was not already configured. Therefore, this step has been added.

- 2.1.1.2. Download Rancher Desktop from the following website. Clicking the Windows link on the site will automatically start downloading the installer.

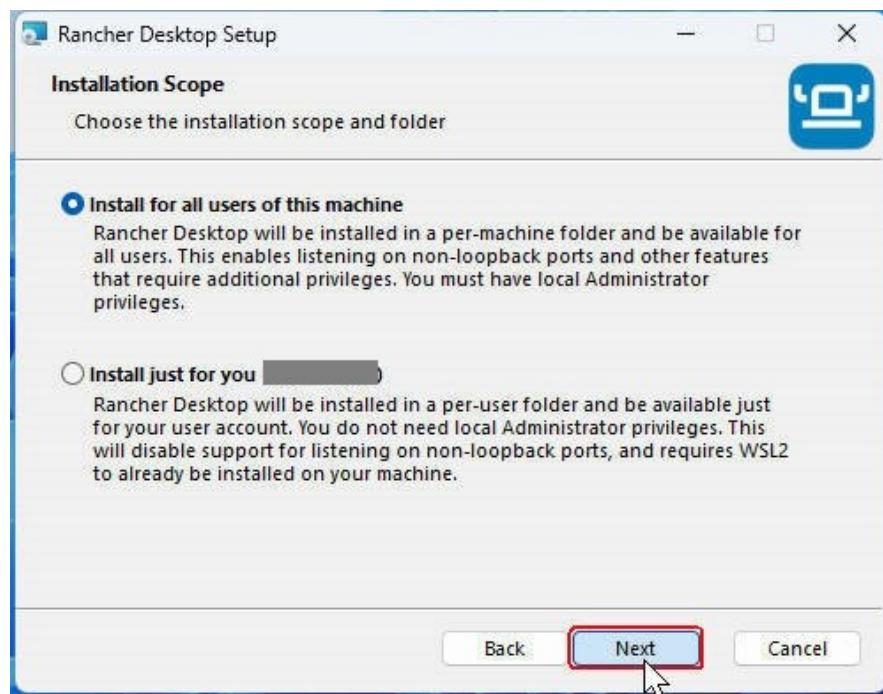
URL : <https://rancherdesktop.io/>



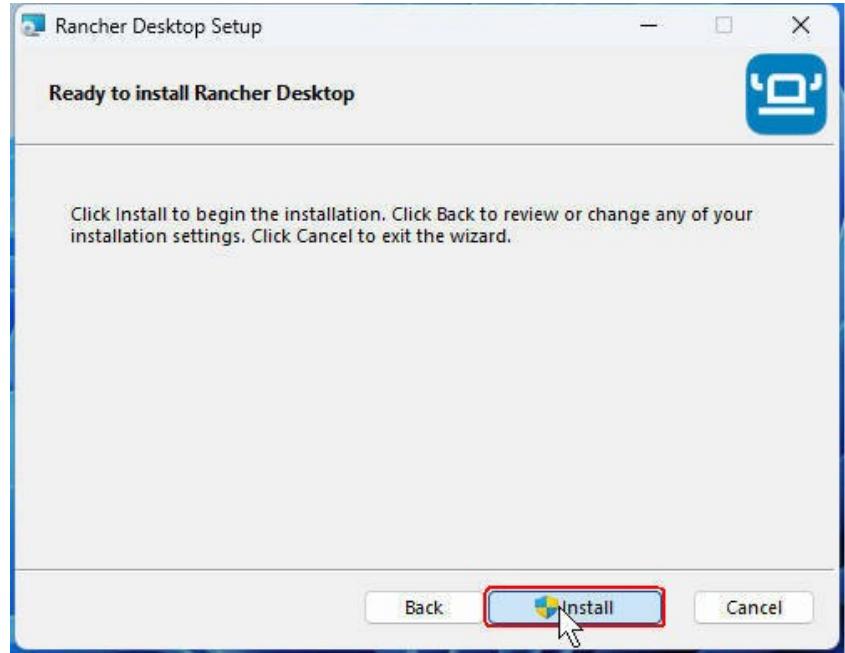
- 2.1.1.3. Double-click the downloaded installer to open the license agreement screen. After reviewing the contents, check the box for I accept… and click Next.



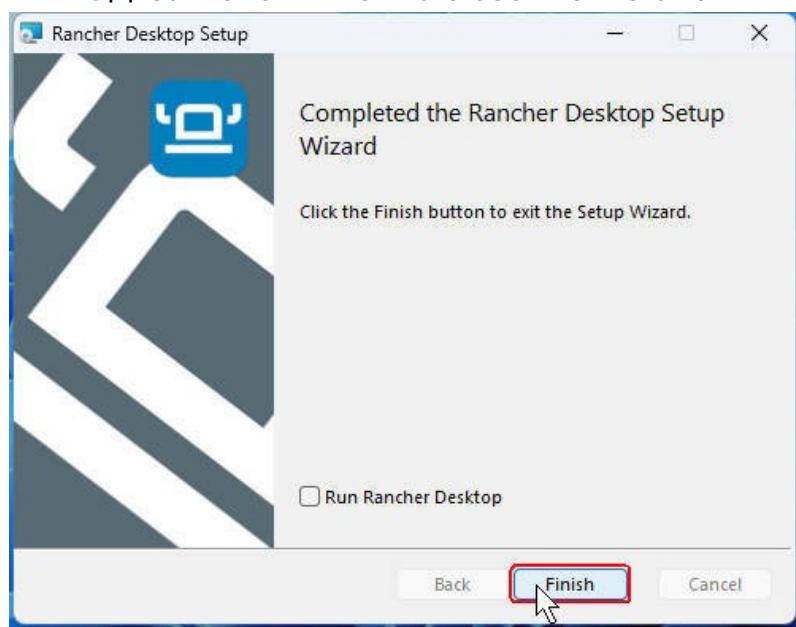
- 2.1.1.4. Select whether to install Rancher Desktop for all users or just yourself, then click Next.



- 2.1.1.5. A confirmation screen for starting the installation will appear. Click Install. If a prompt appears saying, "Do you want to allow this app to make changes to your device?", select Yes.



Once the installation is completed successfully, a completion screen will appear. Click Finish to close the installer.



- 2.1.1.6. An icon for Rancher Desktop will be created on your desktop.
Double-click the icon to launch the application.

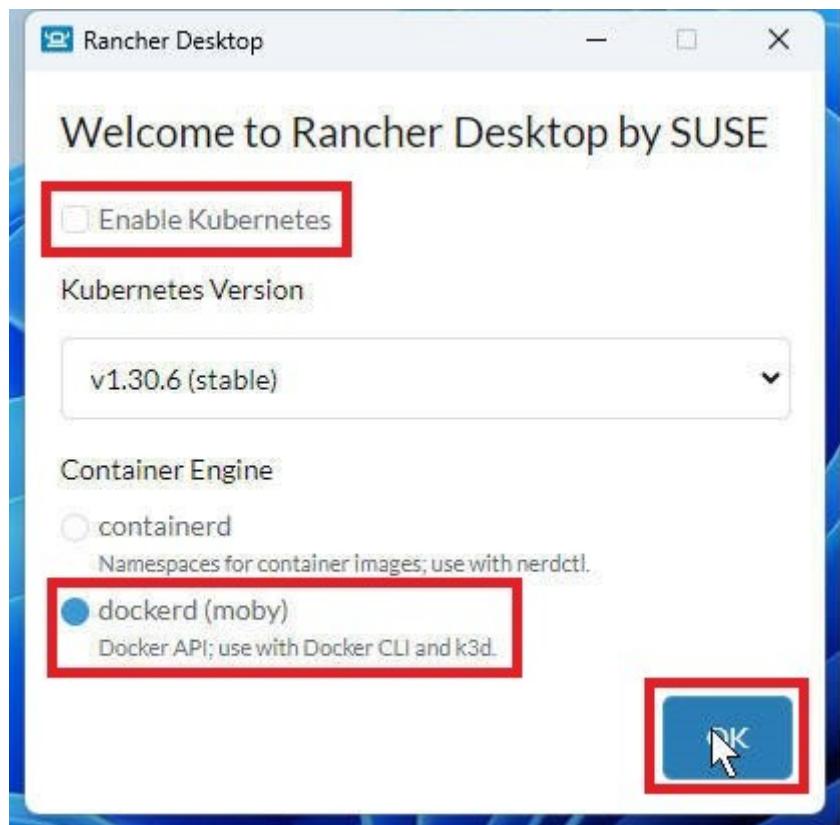


- 2.1.1.7. The initial setup screen will appear.

Uncheck Enable Kubernetes.

Select dockerd (moby) as the Container Engine, then click OK in the bottom right corner.

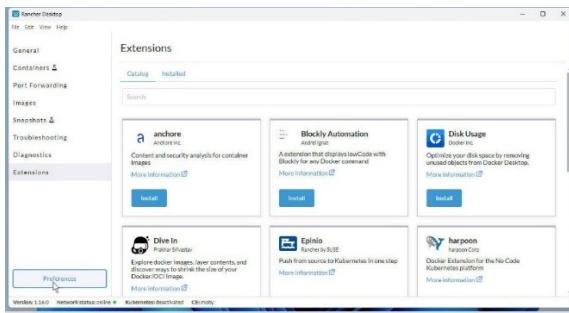
* Note: Although SNPcaster can run with Kubernetes enabled, we recommend turning it off. In our testing, enabling Kubernetes often caused Rancher Desktop to fail to start properly.



2.1.1.8. (For users who require proxy settings only) Configure the proxy settings as needed.

**** If you do not require proxy settings, please proceed to [Rancher Desktop Startup Configuration](#). ****

Click Preferences in the bottom-left corner of the screen.



Select WSL from the left panel, then click the Proxy tab in the center of the screen to display the proxy settings.

Check the box for “Enable the proxy used by rancher-desktop”.

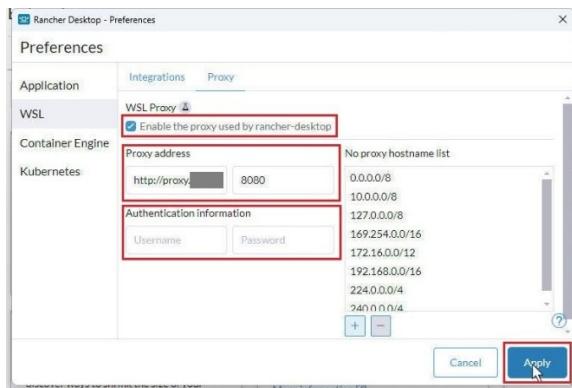
In the text box to the left of Proxy address, enter the URL of the proxy server, and enter the port number in the box to the right.

Under Authentication Information, enter the authentication credentials (username and password) required for using the proxy server.

If no authentication is required, leave these fields blank.

After clicking Apply at the bottom right, the proxy settings are complete.

Please proceed to [Rancher Desktop Startup Configuration](#).



2.1.2. Mac

- 2.1.2.1. Click the Apple logo at the top left of the desktop screen, then select About This Mac. Check whether your chip is Intel or Apple Silicon (M1, M2, etc.).



- 2.1.2.2. Download Rancher Desktop from the following site.

Depending on the chip you confirmed in the previous step, click either “Download for macOS (Apple Silicon)” or “Download for macOS (Intel)” on the website to automatically download the installer.

Note: In the image, the box shown is for the Intel chip. For Apple Silicon chips, select the option to the left.

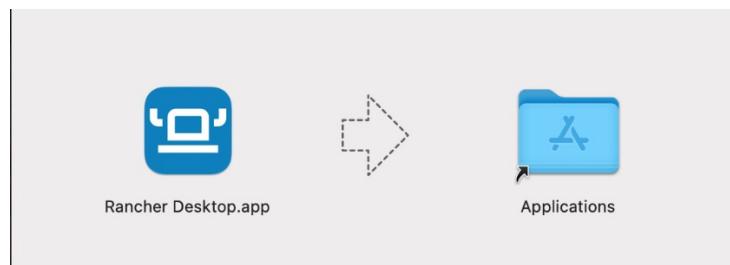
URL:<https://rancherdesktop.io/>

The image displays two screenshots of the Rancher Desktop by SUSE website. The left screenshot shows the main landing page with the title 'Container Management and Kubernetes on the Desktop'. It features download links for macOS (Apple Silicon) and macOS (Intel), where the macOS (Intel) link is highlighted with a red box. Below these are links for Windows (x64) and Install on Linux. The right screenshot shows a demo interface with a green arrow icon and the SUSE logo.

- 2.1.2.3. Double-click the downloaded installer and follow the prompts to install.

Drag and drop the Rancher Desktop icon into the Applications folder.

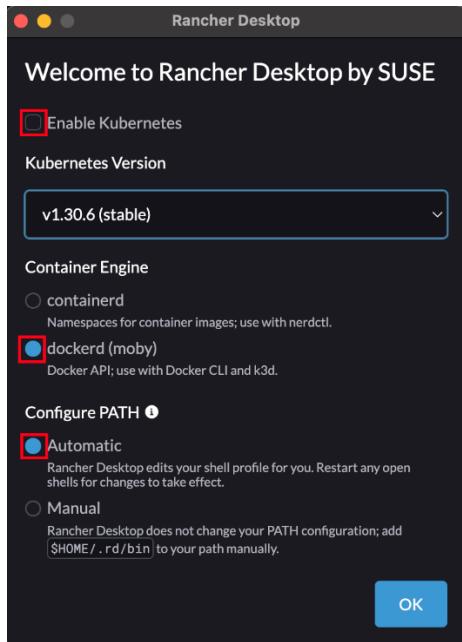
If a message appears stating that privileged access is required, click OK to grant privileged access.



- 2.1.2.4. Follow the instructions in the displayed messages to complete the installation. After installation, when you start Rancher Desktop, the initial setup screen will appear. **Uncheck Enable Kubernetes**

Select Container Engine as **dockerd(moby)**, choose **Automatic** for Configure PATH, and click OK at the bottom right.

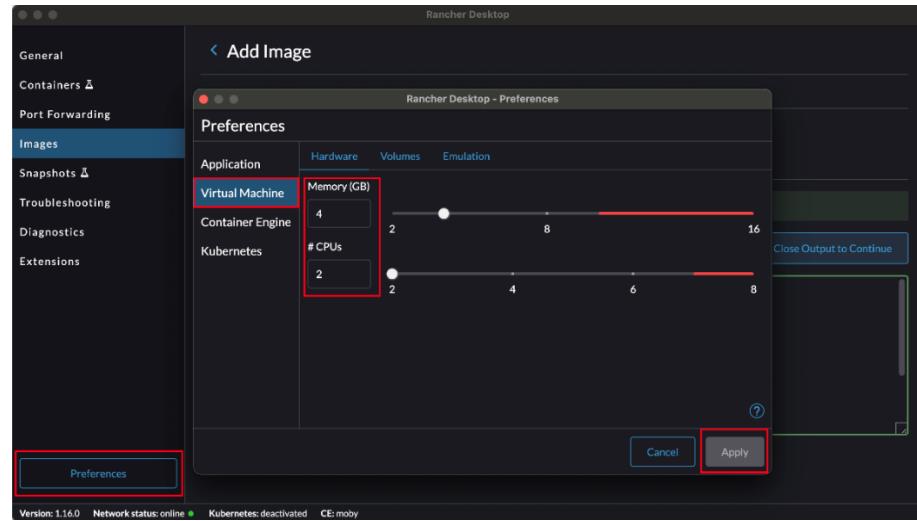
Note: Although SNPcaster can run with Enable Kubernetes turned ON, testing has shown that Rancher Desktop often fails to start with it enabled. Therefore, it is recommended to keep it OFF.



2.1.2.5. Configure the amount of memory and CPUs available to Rancher Desktop.

Go to Preferences > Virtual Machine > Hardware, adjust the values for Memory and CPUs, then click **Apply**.

Please refer to the section [System Requirements](#) for the necessary specs.



2.1.2.6. (For those who need proxy settings only) Configure the proxy settings.

If you do not need this, please proceed to [Rancher Desktop](#)

[Startup Configuration.](#)

Open the Terminal (refer to the [Mac](#) section for how to open it), then enter and execute the following command:

```
LIMA_HOME="$HOME/Library/Application Support/rancher-
desktop/lima" "/Applications/Rancher
Desktop.app/Contents/Resources/resources/darwin/lima/bi
n/limactl" shell 0
```

You will be logged into the lima-rancher-desktop (the Rancher Desktop virtual environment), so execute the following commands:

```
sudo vi /etc/init.d/docker
```

The Vi editor will open. Add the following proxy settings at the end of the file.

```
export http_proxy=<proxy serverURL>:<port number >/
export https_proxy=<proxy serverURL>:<port number >/
```

The proxy configuration is complete after restarting Docker with the following command.

```
sudo service docker restart
```

Please proceed to [Rancher Desktop Startup Configuration.](#)

2.1.3. Rancher Desktop Startup Configuration

This section explains how to configure Rancher Desktop to start automatically.

By default, Rancher Desktop does not launch automatically after rebooting your PC.

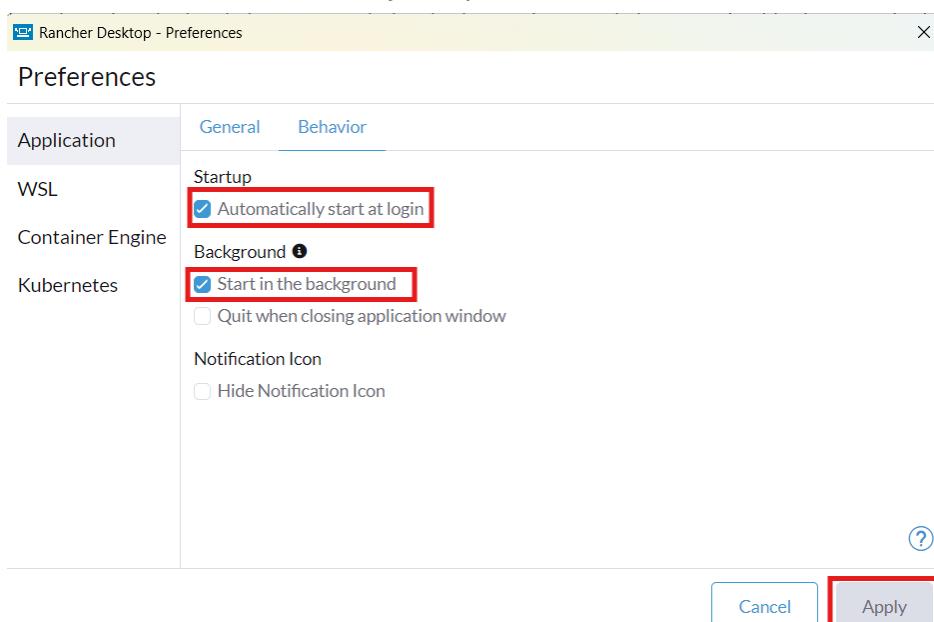
Following the steps in this section will allow Rancher Desktop to start automatically when your PC boots up, making Docker available as well. Although the screenshots used in this section are from Windows, the configuration steps apply to both Windows and Mac.

2.1.3.1. Modify Rancher Desktop Settings

Open Preferences > Application > Behavior from the bottom left of the main Rancher Desktop window.

Check both “Automatically start at login” and “Start in the background”, then click Apply to save the changes.

Note: The Start in the background option prevents the Rancher Desktop window from appearing when it launches automatically. This setting is optional and can be enabled or disabled based on your preference.



That completes the setup. Rancher Desktop will now automatically launch when your PC starts.

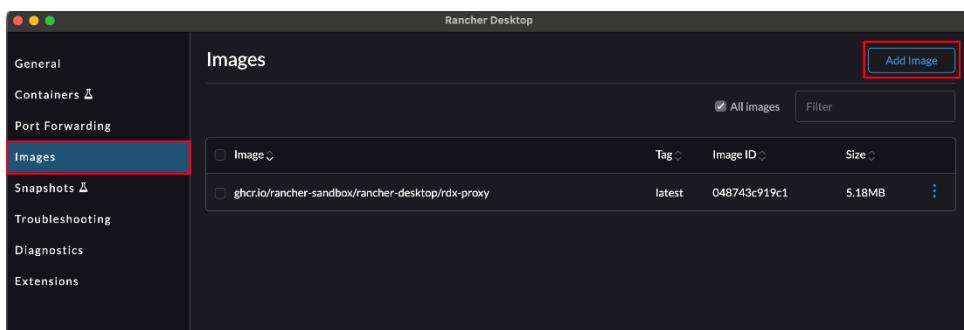
2.1.4. Verifying Rancher Desktop Operation

Let's confirm that Rancher Desktop is functioning correctly.

The screenshots shown are from macOS, but the steps are the same for Windows.

2.1.4.1. Perform an Operation Check for Rancher Desktop

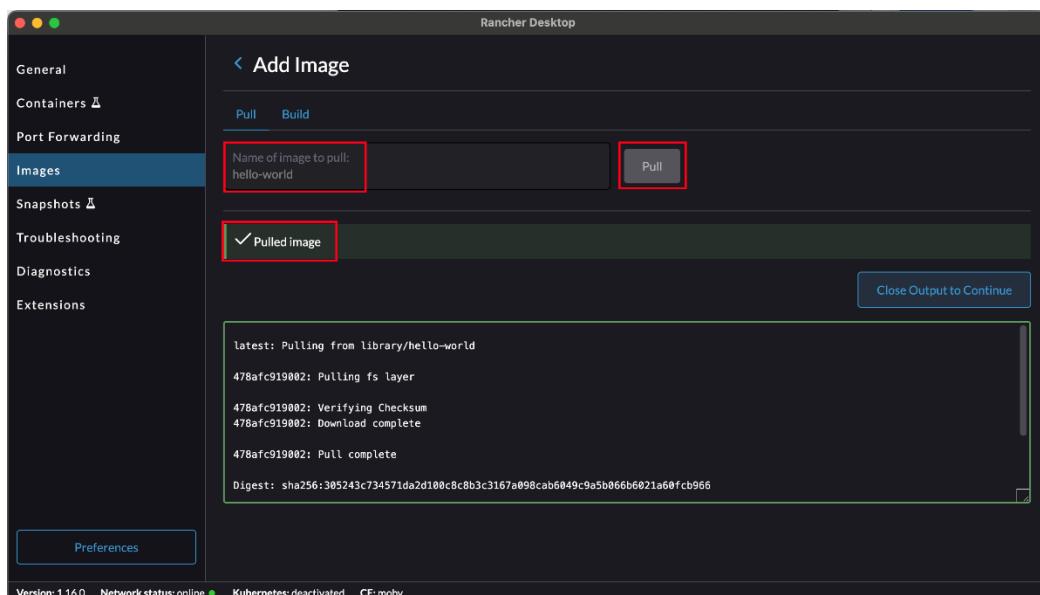
From the main screen, go to Images > Add Image.



In the Name of image to pull field, enter **hello-world** and click Pull.

If **Pulled image** is displayed, Rancher Desktop is functioning correctly.

Note: If an error occurs, it is often due to a proxy configuration issue. Please review the proxy settings from the previous steps.



This completes the setup of Docker using Rancher Desktop.

Please proceed to [Setting Up the SNPcaster Environment](#).

2.2. Installing Docker Engine

Docker Engine can be installed using command-line operations.

If you are using a Mac, please proceed to Mac section.

If you are using a Mac, please proceed to Linux section.

2.2.1. Windows

This section outlines the steps to prepare for installing Docker Engine on a Windows system.

- 2.2.1.1. If WSL is not yet set up, please complete the setup by referring to the section [Installing Windows Subsystem for Linux \(WSL\)](#).

- 2.2.1.2. Run the following command in PowerShell to start WSL:

```
wsl -d Ubuntu-24.04
```

```
PS C:\Users\██████████> wsl -d Ubuntu-24.04
test@████:/mnt/c/Users/███████$ |
```

Once WSL (Ubuntu) starts, the display on the left side will change.

- 2.2.1.3. Navigate to the folder where the SNPcaster folder is located.

Use the following command to change directories.

If you are unsure of the path in WSL, please refer to [Windows current directory section](#).

```
cd /mnt/{The folder where SNPcaster was downloaded on Windows}
```

- 2.2.1.4. (For those who need proxy settings only) If you do not require proxy settings, please proceed to the next step.

Create a .env file containing your proxy information.

Please perform the steps in [Create and edit the .env file](#) section.

After creating the .env file, run the following command in the same folder as the .env file to set the proxy settings as

environment variables.

```
bash ./preparation/docker-engine/set_proxy.sh
```

If prompted to enter a password during execution, please enter it. Once "Done!" is displayed, the process is complete. Since the newly added environment variables are not loaded immediately after executing set_proxy.sh, run the following command to update them.

```
source ~/.bashrc
```

The Docker Engine proxy setup is now complete.

- 2.2.1.5. Install the Docker engine. Please execute the following command.

```
bash ./preparation/docker-engine/install_docker_engine.sh
```

You will be prompted to enter your password (as shown ; the password will not be displayed).

```
[sudo] password for test: 
```

Once "Done!" is displayed, the Docker Engine installation is complete.

- 2.2.1.6. Close the terminal once, then reopen it.

Use the following command to log out from the terminal.

```
exit
```

Then, log in again. The procedure is the same as the one performed in step [2.2.1.2](#)

This completes the Docker Engine installation.

From now on, if you want to use SNPcaster, simply run the following command to enable the use of the docker command.

```
$ wsl.exe -d Ubuntu-24.04
```

Proceed to [Docker Engine Operation Verification](#) and confirm that Docker is working correctly.

2.2.2. Mac

- 2.2.2.1. (For those who need proxy settings only) If you do not require proxy settings, please proceed to the next step.

Configure the proxy settings.

[Open the terminal](#) and run the following command.

Note: Replace <proxy-server-URL>:<port> with the values appropriate for your environment.

e.g.) `http://proxy.example.com:8080`

```
echo 'export http_proxy=<proxy serverURL>:<port number >' >> ~/.zshrc
echo 'export https_proxy=<proxy serverURL>:< port number >' >> ~/.zshrc
echo 'export HTTP_PROXY=<proxy serverURL>: port number >' >> ~/.zshrc
echo 'export HTTPS_PROXY=<proxy serverURL>:< port number >' >> ~/.zshrc
```

Run the following command to apply the settings.

```
source ~/.zshrc
```

This completes the proxy configuration for the terminal.

- 2.2.2.2. Install Homebrew, the package manager for macOS.

Run the following command.

```
cd ~
mkdir .homebrew
curl -L https://github.com/Homebrew/brew/tarball/master | tar xz --strip 1 -
C .homebrew
```

Run the following command to add Homebrew to your PATH.

```
echo 'export PATH=$HOME/.homebrew/bin:$PATH' >> ~/.zshrc
echo 'export HOMEBREW_CACHE=$HOME/.homebrew/caches' >> ~/.zshrc
source ~/.zshrc
```

2.2.2.3. Install the Docker CLI client.

Run the following command.

```
brew install docker docker-compose
```

You will see output similar to the following.

Please make a note of it, as it will be used in later configuration steps.

```
Compose is a Docker plugin. For Docker to find the plugin, add  
"cliPluginsExtraDirs" to ~/docker/config.json:
```

```
"cliPluginsExtraDirs": [  
    "/opt/homebrew/lib/docker/cli-plugins"  
]
```

* Note: The `/opt/homebrew` part may appear as `/Users/{Mac username}/.homebrew` depending on your environment.

Next, configure the Docker CLI.

Edit the `/Users/{Mac username}/.docker/config.json` file using your preferred text editor.

Perform the following two steps:

- **Delete** the line that starts with `credsStore`.
- Add the `cliPluginsExtraDirs` entry that was shown during the Docker Compose installation to the end of the file.

When doing so, make sure to add a comma (,) at the end of the previous line—unless it already ends with one

```
Users > share > .docker > {} config.json > [ ] cliPluginsExtraDirs
1   {
2     "auths": {},
3     "credsStore": "osxkeychain", ← delete the line
4     "currentContext": "lima-snpcaster-vm", ← add
5     "cliPluginsExtraDirs": [ ← add
6       "/Users/share/.homebrew/lib/docker/cli-plugins"
7     ]
8 }
```

After completing the above edits, save the file.

2.2.2.4. Install Lima as the environment to run the Docker Server.

```
brew install lima
```

If an error message like the one below appears, follow the instructions in the message to install the required additional components.

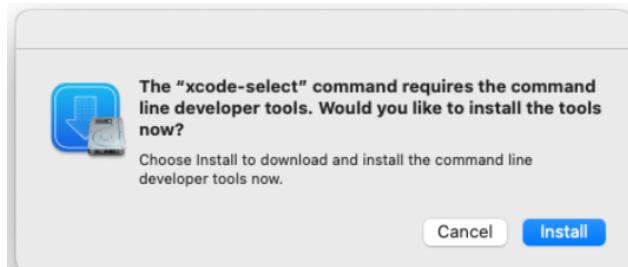
```
[share@wf-toyama242-10 ~ % brew install lima
Error: No developer tools installed.
Install the Command Line Tools:
  xcode-select --install
```

Run the following command.

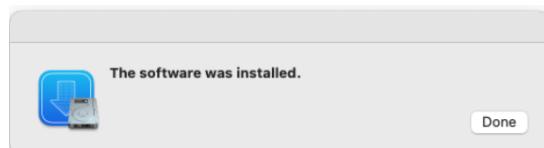
```
xcode-select --install
```

Additionally, if you see the error xcode-select: note: install requested for command line developer, an installation window like the one below will appear.

Proceed with the installation.



Once the installation is complete, the following screen will appear. Click Done to finish.



Run the following command again to install lima.

```
brew install lima
```

2.2.2.5. Create a virtual machine on Lima to run the Docker Server.

Run the following command:

* Note: `--cpus=` specifies the number of CPU cores to allocate, and `--memory=` specifies the amount of memory (GB) to allocate. Please adjust these values according to the performance of your machine.

Refer to [System Requirements](#) for the necessary specifications.

```
limactl create --name=snpcaster-vm \
    --cpus=6 --memory=14 \
    --vm-type=vz --mount-type=virtiofs --rosetta \
    template://docker
```

A confirmation message like the one below will appear.

With "Proceed with the current configuration" selected, press Enter.

```
[share@wf-toyama242-10 ~ % limactl create --name=snpcaster-vm \
    --cpus=6 --memory=14 \
    --vm-type=vz --mount-type=virtiofs --rosetta \
    template://docker
? Creating an instance "snpcaster-vm" [Use arrows to move, type to filter]
> Proceed with the current configuration
Open an editor to review or modify the current configuration
Choose another template (docker, podman, archlinux, fedora, ...)
Exit
```

Run the following command, and if a row with `snpcaster-vm` appears under the NAME column, the creation is complete.

```
limactl list
```

Next, configure Lima. Open `/Users/{Mac username}/.docker/config.json` in your preferred text editor. Add `writable: true` below `- location: "~"` which appears right after `mounts:`. Make sure to indent the line with two half-width spaces at the beginning.

```
25  mounts:
26  - location: "~"
27  |   writable: true
28  - location: "/tmp/lima"
29  |   writable: true
```

Once saved, the configuration is complete.

Note: The above configuration allows Docker to have write access to the home directory (`/Users/{Mac username}/`). Therefore, make sure to place and run the **SNPcaster folder within `/Users/{Mac username}/`.**

If you run it outside of this directory, SNPcaster will not be able to output results and an error will occur

- 2.2.2.6. Start the virtual machine you created. Run the following command.

```
limactl start snpcaster-vm
```

Note: After restarting your PC, you will need to run the above command each time to start the virtual machine (VM).

- 2.2.2.7. Connect the Docker CLI to the Docker Server started in the previous step.

Run the following command.

Note: This command is displayed on the screen when you start the virtual machine in the previous step. Please copy and use it from there.

```
docker context create lima-snpcaster-vm --docker  
"host=unix:///Users/{Mac username}/.lima/snpcaster-  
vm/sock/docker.sock"  
docker context use lima-snpcaster-vm
```

This completes the installation of Docker Engine.

Now, you can use the docker command from the [terminal](#).

Proceed to [Docker Engine Operation Verification](#) to ensure it is functioning correctly.

2.2.3. Linux

Install Docker CLI and Docker Compose.

Please execute the commands as the **root user** or a user with **sudo** privileges.

2.2.3.1. Open a terminal and run the following command to install the packages required for Docker.

Note: If you are using the root user, **sudo** is not necessary.

- Ubuntu

```
sudo apt install -y \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

- CentOS

```
sudo yum install -y \ yum-utils \ device-mapper- \
    persistent-data \ lvm2
```

2.2.3.2. Use the following command to download and run the official Docker installation script, get-docker.sh, from the Docker website.

- Ubuntu
- CentOS

```
curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh
```

Note: If curl does not work, setting up a proxy may resolve the issue.

```
nano ~/.curlrc
```

Edit the content as shown below. (Press i to enter edit mode.)

The following is an example where the proxy IP address is proxy.go.jp and the port number is 8080.

```
proxy=http://proxy.go.jp:8080
```

After entering the above settings, save the file.

(Press Ctrl + X to exit. When prompted to save, type Y to save.)

2.2.3.3. Run the following command to configure Docker to start automatically when the server or PC is rebooted.

Note: If you are using the root user, sudo is not required.

- Ubuntu・CentOS

```
sudo systemctl enable docker  
sudo systemctl start docker
```

2.2.3.4. To run Docker commands without sudo, add your user to the docker group using the following command.

- Ubuntu・CentOS

```
sudo gpasswd -a $(whoami) docker
```

If you want to use Docker with a different user than the currently logged-in user, replace \$(whoami) with the appropriate username.

```
sudo gpasswd -a $(whoami) docker
```

2.2.3.5. To reload the settings, close the terminal and then reopen it.

Run the following command. If the version number is displayed, everything is set up correctly. Proceed to the next step.

```
docker compose version
```

```
(base) [ ] :~$ docker compose version  
Docker Compose version v2.35.1
```

2.2.3.6. (For users who require proxy setting only) Configure the proxy server using the steps below.

**** If you do not need proxy settings, please proceed to [Docker Engine Operation Verification](#). ****

* If the following configuration does not resolve the issue, please contact your organization's IT department.

Edit the configuration file by using the following command:

- Ubuntu·CentOS

```
sudo systemctl edit docker
```

An editor will open in the terminal. Add the following content (or modify it if it already exists).

Please make sure to add the content between the lines:

Anything between here and the comment below...contents of the drop-in file

and

Edits below this comment will be discarded.

Do NOT place your edits outside of these lines, as they will not be applied.

Replace the proxy IP address and port number.

```
### Editing /etc/systemd/system/docker.service.d/override.conf
### Anything between here and the comment below will become the
contents of the drop-in file
[Service]
Environment="HTTP_PROXY=http://proxy.go.jp:8080"
Environment="HTTPS_PROXY=http://proxy.go.jp:8080"

### Edits below this comment will be discarded
```

Once completed, save the file and close the editor.

(Press Ctrl + X to exit. When prompted to save, type Y to confirm and save.)

After that, restart Docker using the following command:

- Ubuntu
- CentOS

```
sudo systemctl daemon-reload  
sudo systemctl restart docker
```

Next, configure the Docker client to allow communication from within containers.

The configuration file is located at:

`~/.docker/config.json` under the home directory of the user.

Use the following commands to create the `~/.docker` directory and edit the config.json file.

```
mkdir -p ~/.docker  
nano ~/.docker/config.json
```

Edit the content using the format below.

Replace the server IP address (proxy.go.jp) and port number (8080) with the appropriate values for your environment.

```
{  
  "proxies": {  
    "default": {  
      "httpProxy": "http://proxy.go.jp:8080",  
      "httpsProxy": "http://proxy.go.jp:8080"  
    }  
  }  
}
```

After entering the above settings, save the file and close the editor.

(Press Ctrl + X to exit. When prompted to save, type Y to confirm and save.)

This completes the proxy configuration. Please proceed to [Docker Engine Operation Verification](#).

2.2.4. Docker Engine Operation Verification

In this section, you will verify that Docker Engine is working correctly. Please make sure Docker Engine is installed before proceeding.

2.2.4.1. Launch a container to verify Docker Engine functionality.

Open a terminal where the “docker” command is available, and run the following command:

```
docker run hello-world
```

If you see the message “Hello from Docker!” as shown below, Docker is working correctly.

```
test@NGSW:/mnt/d/test/SNPcaster_docker_engine$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fc966
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
```

The container and Docker image used for the verification are no longer needed, so delete them using the following commands:

```
docker image rm -f hello-world:latest
```

This concludes the operation verification.

Please proceed to Setting Up the SNPcaster Environment.

3. Setting Up the SNPcaster Environment

In this section, you will set up the SNPCaster environment running on Docker.

3.1. Downloading SNPcaster

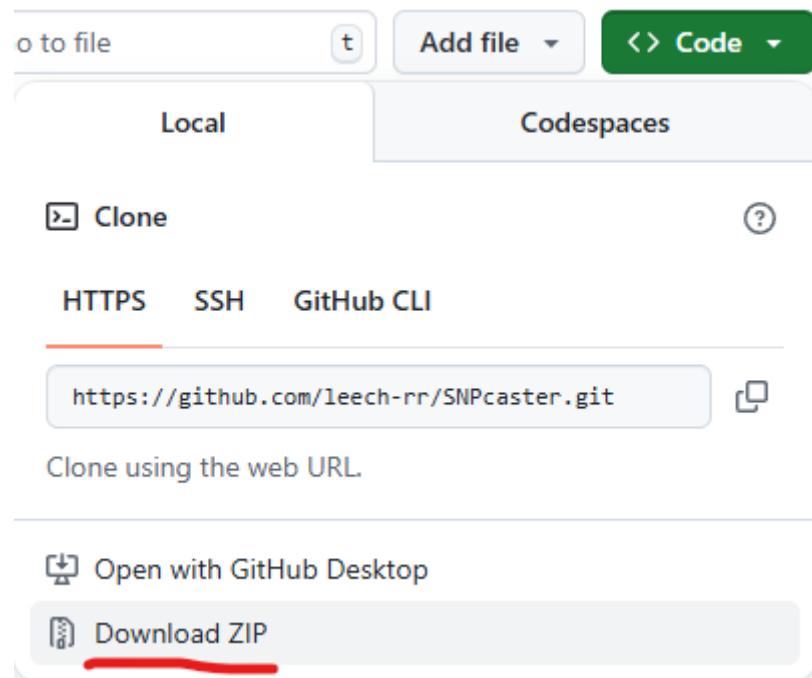
First, download the SNPcaster folder available on GitHub.

You can download it either via a web browser (by clicking) or from the command line.

3.1.1. Download via Web Browser (e.g., Microsoft Edge, Google Chrome)

Open <https://github.com/leech-rr/SNPcaster/tree/main>

Click the Code button in the upper right corner, then click Download ZIP to start the download.



Once the download is complete, move the ZIP file to your desired working directory and extract it.

3.1.2. Download via Command Line (Git)

When using Git to download, it's convenient because you can easily update to the latest version later by simply running git pull. However, since Git operations may require some familiarity, if you're short on time, we recommend to [Download via Web Browser \(e.g., Microsoft Edge, Google Chrome\)](#) instead.

To download using Git, you need to have Git installed beforehand. If it is not already installed, please refer to the installation instructions at the following URL and install Git:

<https://git-scm.com/book/ja/v2/%E4%BD%BF%E3%81%84%E5%A7%8B%E3%82%81%E3%82%8B-Git%E3%81%AE%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB>

Launch a command-line tool such as Windows PowerShell (Windows), Terminal (Mac), or Terminal (Linux).

(If you're not familiar with these tools, please refer to [Opening a Command-Line Application in a Specific Directory](#).)

Please run the following commands before downloading.

Note: On Windows, skipping these settings is known to cause installation failures.

```
git config core.autocrlf false
```

You can download the files from Git by executing the following command.

Please replace {your_target_folder} with the path to the folder where you want to download the files.

```
cd {path_to_your_target_folder}
git clone https://github.com/leech-rr/SNPcaster.git
```

3.2. (Reference) Folder and File Structure

The structure of SNPcaster is as follows (most files are omitted for brevity).

```
snpcaster
|   .env.example
|   .gitignore
|   COPYING
|   docker-compose.yml
|   NOTICE.md
|   README.md
|   SNPcaster_bugs.md
├── app
|   |   Dockerfile
|   |   notebook
|   |   :
|   |   setup
|   |   :
|   └── src
|       |   grape_qc_assembly
|       |       :
|       └── snpcaster
|           |       :
└── doc
    |   :
└── preparation
    |   :
└── project
    └── .gitignore
```

The structure is as follows.

Folder/File Name	Contents
.env.example	Template for the environment .env file.
.gitignore	Git management file. Do not edit.
COPYING	This file indicates that SNPcaster is licensed under GPLv3.0.
docker-compose.yml	This is a configuration file for launching the Docker container.
NOTICE.md	Used by SNPcaster
README.md	This is the project's README.
SNPcaster_bugs.md	This is a list of reported bugs.
app	This folder contains the necessary files and directories for building the SNPcaster environment.
app/ Dockerfile	This file contains the instructions for building the Docker image.
app/ notebook	This folder contains programs and other resources used for analysis with Jupyter Notebook.
app/ setup	This folder contains scripts used for creating and launching the Docker environment.
app/ src	This folder contains the program files and other resources required for executing the analysis.
app/src/ grape_qc_assembly	This folder contains the programs and resources required for executing Grape.
app/src/ snpcaster	This folder contains the programs and resources required to run SNPcaster.
doc	This folder contains documentation files.
preparation	This folder includes programs and scripts for preparing the environment to use Docker, such as installing Docker Engine.

project	This is the working directory for analysis. Output results will be saved here. After starting the container, a notebook folder will be created here to store Jupyter Notebook outputs.
project / .gitignore	This is a Git management file. Do not edit this file.

3.3. Installing SNPcaster on the Analysis Terminal

To install SNPcaster on the analysis terminal, build the Docker image and use it to launch the container.

3.3.1. Create and edit the .env file

**** Only perform this step if you are using a proxy environment. ****

To set the environment variables used during the build process, you need to create a .env file.

Open the downloaded SNPcaster folder using File Explorer (Windows), Finder (Mac), or a similar tool.

Then, copy “.env.example” to “.env”.

📁 .git	2024/11/20 16:21
📁 jupyter	2024/11/20 15:49
📄 .env	2024/11/20 16:14
📄 .env.example	2024/11/20 16:14
📄 .gitignore	2024/11/20 15:35
📄 docker-compose.yml	2024/11/20 16:20
📄 SNPcaster_bugs.md	2024/11/20 15:22

Edit the environment variables.

Open the “.env” file you created with your preferred text editor.

Delete ‘#’ at the beginning of lines 10 and 11.

Replace the server URL and port number according to your environment.

```
1 LOCAL_UID=1000
2 LOCAL_GID=1000
3
4 #If you want to configure a proxy, please set it below.
5 #Specify the HTTP proxy URL in HTTP_PROXY (remove the leading #).
6 #Specify the HTTPS proxy URL in HTTPS_PROXY (remove the leading #).
7 #Example format: HTTP_PROXY=http://proxy.example.com:8080
8 #Please replace proxy.example.com:8080 with the appropriate value for your environment.
9 #If authentication is required, use the following format:
10 #http://{username}:{password}@{URL}:{port}
11 HTTP_PROXY=http://proxy.example.com:8080
12 HTTPS_PROXY=https://proxy.example.com:8080
13
```

Remove the # at the beginning of each line.

HTTP_PROXY: For HTTP settings, specify the same content in the format {URL}:{port}

HTTPS_PROXY: For HTTPS settings, specify the same content in the format {URL}:{port}

* If authentication is required, use the format:

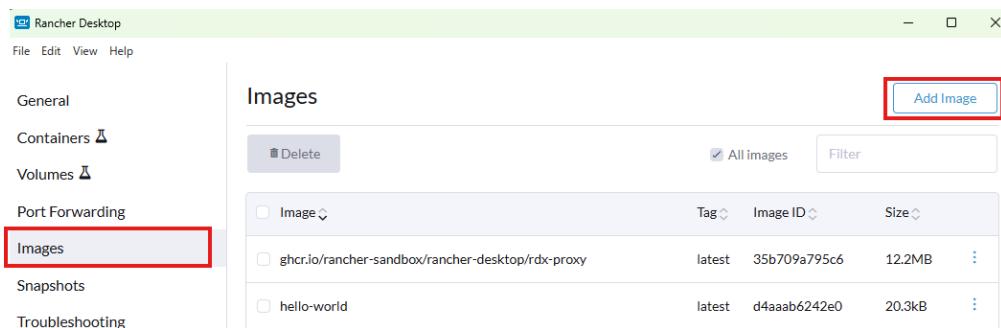
http://{username}:{password}@{URL}:{port}

3.3.2. Running the Build

Please perform the following steps only if you are using Rancher Desktop on Windows and require proxy settings. (If this does not apply to you, please proceed to the next page.)

(Only for: Windows + Rancher Desktop + Proxy Environment)

Launch Rancher Desktop and click Images > Add Image.

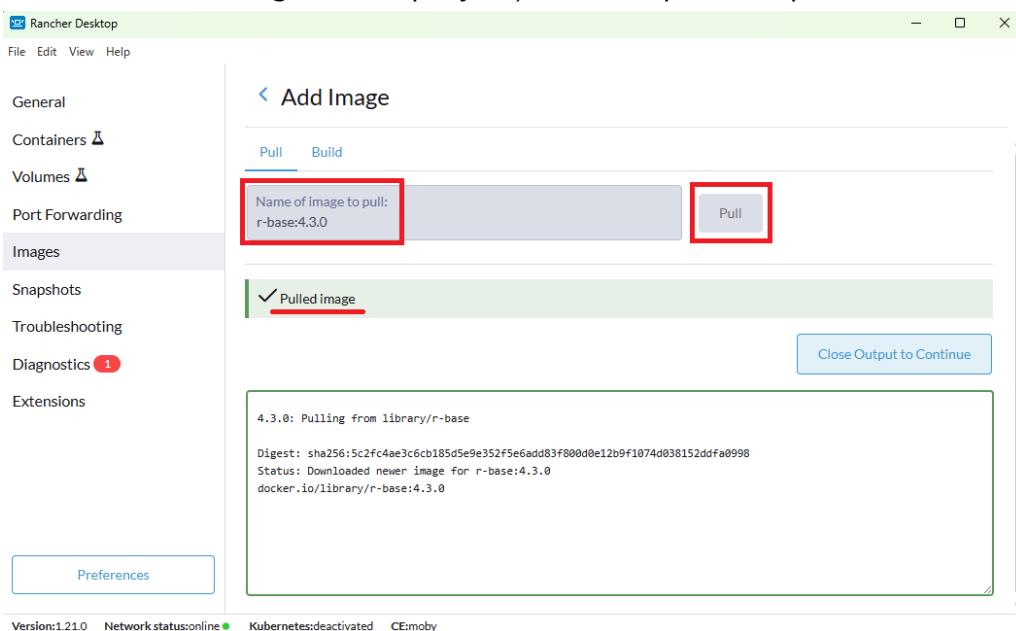


On the Pull tab, enter

r-base:4.3.0

in the "Name of Image to pull:" field and click Pull.

Once "Pulled Image" is displayed, the setup is complete.

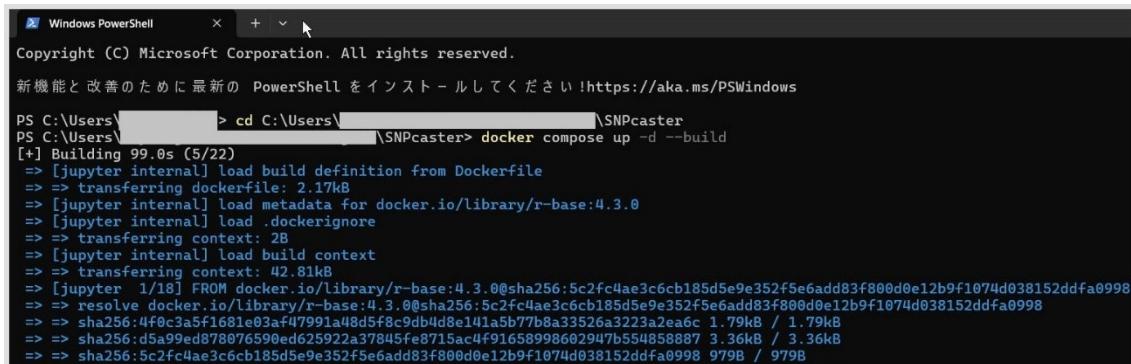


[For all environments]

Open a command-line application in the SNPcaster folder and enter the following command.

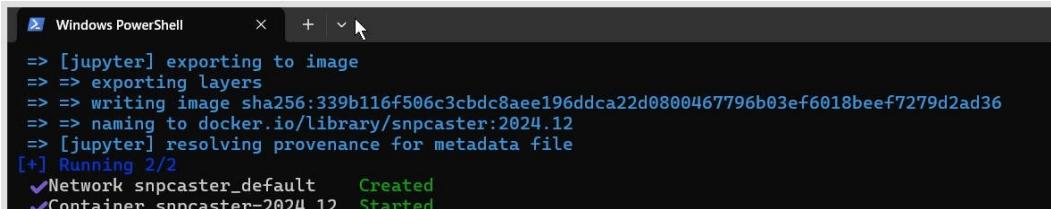
```
docker compose up -d --build
```

The Docker image build will begin, and the progress will be displayed as shown below.



```
PS C:\Users\<username>\SNPcaster> docker compose up -d --build
[+] Building 99.0s (5/22)
=> [jupyter] load build definition from Dockerfile
=> => transferring dockerfile: 2.17kB
=> [jupyter] load metadata for docker.io/library/r-base:4.3.0
=> [jupyter] load .dockignore
=> => transferring context: 2B
=> [jupyter] load build context
=> => transferring context: 42.81kB
=> [jupyter] FROM docker.io/library/r-base:4.3.0@sha256:5c2fc4ae3c6cb185d5e9e352f5e6add83f800d0e12b9f1074d038152ddfa0998
=> => resolve docker.io/library/r-base:4.3.0@sha256:5c2fc4ae3c6cb185d5e9e352f5e6add83f800d0e12b9f1074d038152ddfa0998
=> => sha256:4f0c3a5f1681e03af47991a48d5f8c9db4d8e141a5b77b8a33526a3223a2ea6c 1.79kB / 1.79kB
=> => sha256:d5a99ed878076590ed625922a37845fe8715ac4f91658998602947b554858887 3.36kB / 3.36kB
=> => sha256:5c2fc4ae3c6cb185d5e9e352f5e6add83f800d0e12b9f1074d038152ddfa0998 979B / 979B
```

Once the build is complete, “Container snpcaster-<version number> Started” will be displayed.



```
=> [jupyter] exporting to image
=> => exporting layers
=> => writing image sha256:339b116f506c3cbdc8aee196ddca22d0800467796b03ef6018beef7279d2ad36
=> => naming to docker.io/library/snpcaster:2024.12
[+] Running 2/2
  ✓ Network snpcaster_default   Created
  ✓ Container snpcaster-2024.12 Started
```

Enter docker ps to check the created container.

```
docker ps
```

If an entry with snpcaster-<version> appears under the “Names” column, the container has been successfully confirmed.

Please proceed to [4. How to Start and Stop the System](#).

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c44d0bb24b88	snpcaster:0.9.2	"/usr/local/bin/entr..."	45 minutes ago	Up 45 minutes	0.0.0.0:59829->8888/tcp, :::59829->8888/tcp	snpcaster-0.9.2

A red arrow points from the text "snpcaster-0.9.2" in the previous paragraph to the "NAMES" column of the table, highlighting the successful creation of the container.

3.3.3. Troubleshooting for Build Failures

If the build fails in a proxy environment, it is likely due to internet connectivity issues. In such cases, building in a non-proxy environment (e.g., at home) may resolve the issue. Once the image is successfully built, it is stored locally on your PC, allowing you to launch SNPcaster offline from then on.

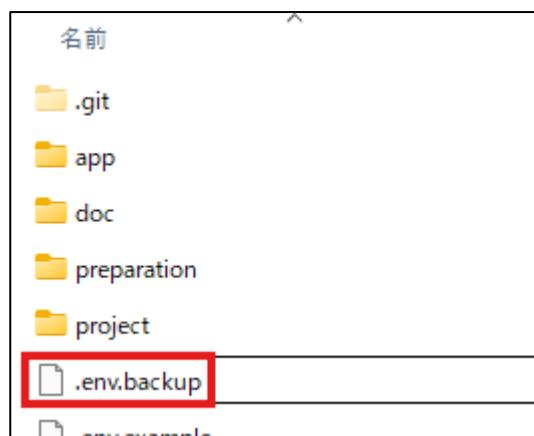
If you wish to try building without a proxy, you must first disable all the proxy settings configured so far. Please follow the steps below to turn off the proxy settings.

3.3.3.1. Common Steps (for all OS)

[Deactivating .env]

Rename .env in the SNPcaster folder to .env.backup.

The next steps differ by OS.



The following steps vary depending on your OS.

3.3.3.2. Windows

The instructions differ depending on whether you are using Rancher Desktop or Docker Engine.

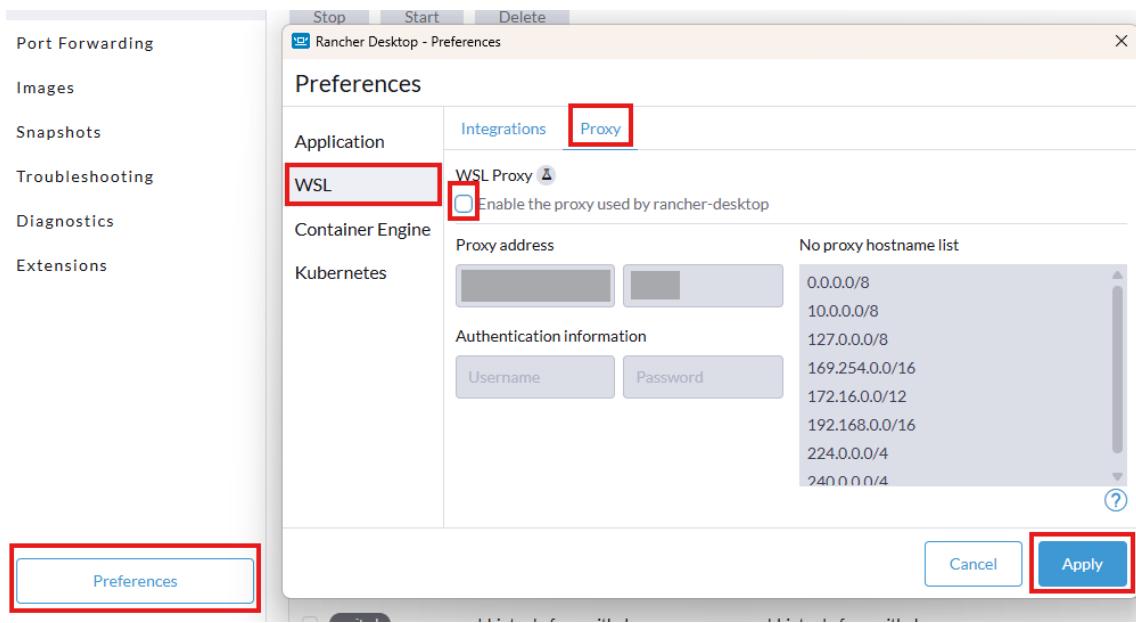
[Rancher Desktop]

Launch Rancher Desktop.

Click Preferences in the bottom-left corner.

Select WSL from the left-hand menu, then go to the Proxy tab.

Uncheck "Enable the proxy used by rancher-desktop" and click Apply.



Please proceed to build the Docker image in a non-proxy environment.

[Docker Engine]

Open the terminal and log in to WSL using the command below:

```
wsl -d Ubuntu-24.04
```

After logging in, run:

```
nano ~/.bashrc
```

In the editor, scroll to the bottom. Comment out the lines in the following section by adding a “#” at the start, but leave the export PATH... line as is.

```
# Porxy setteing by set_proxy.sh
#export http_proxy=""
#export HTTP_PROXY=""
#export https_proxy=
#export HTTPS_PROXY=
#export no_proxy="localhost, 127.0.0.
#export NO_PROXY="$no_proxy"
export PATH="/opt/conda/bin:${PATH}"
```

Press Ctrl+X followed by Y to save. Then, run the following command to apply the changes:

```
source ~/.bashrc
```

Proceed to build the Docker image in a non-proxy environment.

3.3.3.3. Mac

The instructions differ depending on whether you are using Rancher Desktop or Docker Engine.

[Rancher Desktop]

Open the terminal (refer to [Mac](#) section for instructions on how to open it) and run the following command:

```
sudo vi /etc/init.d/docker
```

The vi editor will launch. Delete the following proxy settings that were added to the last lines of the file.

```
export http_proxy=<proxy URL>:<port>/
export https_proxy=<proxy URL>:<port>/
```

After restarting Docker with the command below, the proxy settings will be disabled:

```
sudo service docker restart
```

Now, proceed to build the Docker image in a non-proxy environment.

[Docker Engine]

Open the terminal (refer to the [Mac](#) section for instructions on how to open it) and run the following command:

```
sudo vi ~/.zshrc
```

The vi editor will launch. Scroll down to the section containing the following four lines and "comment them out" by adding a # at the beginning of each export line, then save the changes.

```
#export http_proxy=""  
#export HTTP_PROXY=""  
#export https_proxy="  
#export HTTPS_PROXY=
```

Run the following command to apply the updated settings:

```
source ~/.zshrc
```

The proxy settings for the terminal are now turned off.

Now, proceed to build the Docker image in a non-proxy environment.

3.3.3.4. Linux

Disable the proxy server settings in the systemd configuration file.

Run the command below to edit the configuration file:

Ubuntu / CentOS

```
sudo systemctl edit docker
```

The editor will open in the terminal.

Comment out the three lines (highlighted in yellow) starting from [Service] by adding a "#" at the beginning of each line.

```
### Editing /etc/systemd/system/docker.service.d/override.conf  
### Anything between here and the comment below will become the  
contents of the drop-in file  
# [Service]  
# Environment="HTTP_PROXY=http://proxy.go.jp:8080"  
# Environment="HTTPS_PROXY=http://proxy.go.jp:8080"  
  
### Edits below this comment will be discarded
```

Once finished, save and exit the editor.
(Press Ctrl+X, then type Y when prompted to save.)

Next, disable the Docker client settings. Rename the `~/.docker/config.json` file using the following command:

```
mv ~/.docker/config.json ~/.docker/config.json.backup
```

Finally, restart Docker with the following commands:

Ubuntu / CentOS

```
sudo systemctl daemon-reload  
sudo systemctl restart docker
```

The proxy settings have been successfully disabled.
Please proceed to build the Docker image in a non-proxy environment.

3.3.3.5. Building the Docker Image in a Non-proxy Environment

Connect to an internet environment without a proxy and follow the steps in [3.3.2.Running the Build](#). Once the build is complete, SNPcaster will function correctly without any proxy settings.

Note: However, features requiring internet access, such as downloading sample data, will be unavailable.

If you plan to perform the analysis on the same machine where the build was executed: Proceed to [4.How to Start and Stop the System](#).

If you wish to transfer the built Docker image to another machine:
Proceed to the next step.

3.3.3.6. Transferring the Docker Image to Another Machine

If you wish to move the built Docker image to another machine, export the image file by running the following command in your terminal:

```
docker save -o snpcaster.tar snpcaster:{version number}
```

Note: Please replace {version_number} with the actual version of SNPcaster you are using (e.g., 0.9.5).

A file named snpcaster.tar will be generated in the current folder. Copy this file to the destination machine using external storage, such as a USB flash drive.

Open the terminal in the folder where you copied the file on the destination machine, and run the following command to load the image:

```
docker load -i snpcaster.tar
```

After the image is successfully loaded, proceed to [4.How to Start and Stop the System](#) to launch the application.

3.4. Upgrading SNPcaster

To use a newer version of SNPcaster, please follow the update procedure that matches the method you used to download the software.

[3.4.1. For installations via a web browser \(Google Chrome, etc.\)](#)

[3.4.2. For installations via the Command Line \(Git\)](#)

3.4.1. For installations via a web browser (Google Chrome, etc.)

If you installed SNPcaster using the browser-based method described in Section [3.1.1](#), please follow the steps below to update.

3.4.1.1. Deleting Old SNPcaster Containers

First, delete the Docker containers from the previous version by following these steps:

1. Open your terminal or command-line app and navigate to the downloaded snpcaster folder.
2. Run the command below to delete the Docker containers:

Note: Deleting the containers will not affect your analysis data stored in the project folder.

Note: If you have saved any data outside the project folder (e.g., via manual command execution), **please back up that data before proceeding with the deletion.**

```
docker compose down -v
```

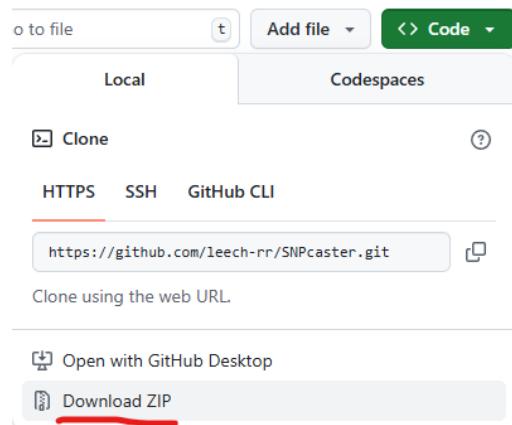
```
[+] Running 2/2
✓ Container snpcaster    Removed  0.7s
✓ Network k2077_snpcaster_notebook_default      Removed  0.3s
```

You may safely close the web browser used to access Jupyter Lab.

3.4.1.2. Downloading the New Version

Open the following URL in your preferred browser:
<https://github.com/leech-rr/SNPcaster/tree/main>

Click the Code button at the top right, then select Download ZIP.

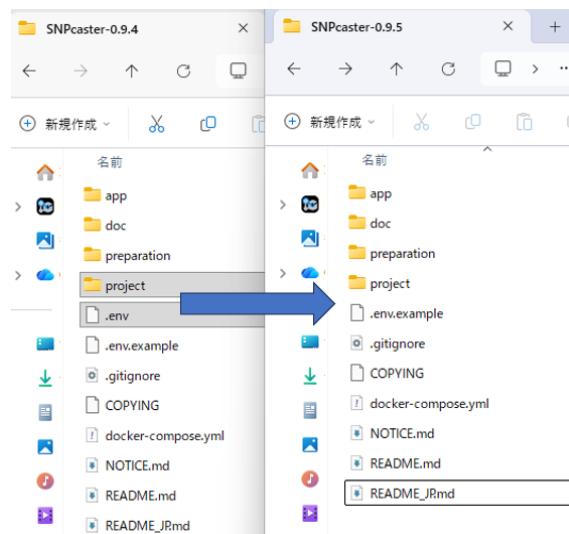


Once the download is complete, move the file to your desired workspace and extract (unzip) it.

3.4.1.3. Data Migration

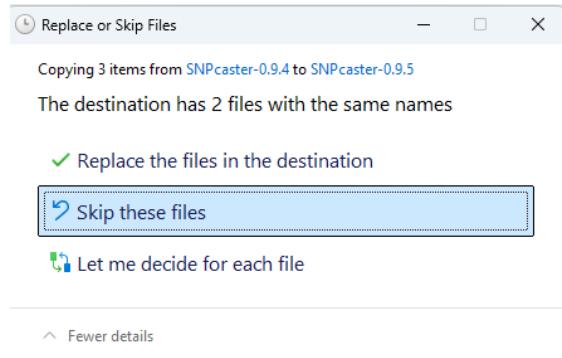
Move the following files/folders from the old version's folder to the new one:

- **project** folder
- **.env** file (if you created one for proxy settings)



When moving the project folder, a confirmation prompt may appear. Please select "Skip these files" (do not replace).

Note: Even if you accidentally choose "Replace," it will not cause any issues with the application.



3.4.1.4. Building and Starting SNPcaster

The remaining steps are the same as the initial launch. Please proceed to [4.1. How to Start](#).

3.4.2. For installations via the Command Line (Git)

If you installed SNPcaster using the Git-based method described in Section [3.1.2](#), you can update using just three commands.

3.4.2.1. Deleting Old SNPcaster Containers

First, delete the Docker containers from the previous version by following these steps:

1. Open your terminal or command-line app and navigate to the existing snpcaster folder.
2. Run the command below to delete the Docker containers:

Note: Deleting the containers will not affect your analysis data stored in the project folder.

Note: If you have saved any data outside the project folder, **please back up that data before proceeding with the deletion.**

```
[+] Running 2/2
✓ Container snpcaster    Removed 0.7s
✓ Network k2077_snpcaster_notebook_default      Removed 0.3s
```

```
docker compose down -v
```

You may safely close the web browser used to access Jupyter Lab.

3.4.2.2. Fetching the Latest Version of SNPcaster

Run the following command to download the latest version of SNPcaster from Git.

```
cd {download_folder}  
git pull
```

Note: Please replace {download_folder} with the actual path to the folder where you initially downloaded SNPcaster.

3.4.2.3. Building and Launching SNPcaster

The remaining steps are the same as the initial startup. Please proceed to [4.1.How to Start](#).

4. How to Start and Stop the System

This chapter explains how to start and stop the SNPcaster Docker container.

If you have just completed Step [3.3.2.Running the Build](#), the container should already be running.

In that case, please proceed to [5.Analysis Execution Procedure](#) to begin your analysis.

4.1. How to Start

4.1.1. Execute the Start Command

Open a command-line application and navigate to the downloaded “snpcaster” folder.

Once there, start the Docker container by executing the following command:

```
docker compose up -d --build
```

```
[+] Running 1/0  
✓ Container snpcaster Running
```

If the message ✓ Container snpcaster-<version> Running is displayed, the container has started successfully.

[Tips] When Startup Fails

An error may occur, and the startup may fail with a message like the one below:

Error when the port is already in use

```
[+] Running 0/1  
- Container snpcaster Starting  
          0.4s  
Error response from daemon: driver failed programming external connectivity on  
endpoint snpcaster (f2ce4c3e2fd6a520c04ea148c23dc863e84d3d9019d68e30c46d038d1  
02fd59c): Bind for 0.0.0.0:59829 failed: port is already allocated
```

Error when a container with the same name already exists

```
[+] Running 0/1
- Container snpcaster-0.9.0 Creating

    2.8s
Error response from daemon: Conflict. The container name "/snpcaster-0.9.0" is
already in use by container "76d4ceaaffc34de24a0fe8b6b795027703469b858d808ed0
6b243342bea20265". You have to remove (or rename) that container to be able to
reuse that name.
```

These errors occur when a container from an older version of SNPcaster has already been created or is currently running. Please stop the old version of the container using the steps in [4.2. How to Stop SNPcaster Docker Container](#), and then try running the startup command again.

4.1.2. Port Forwarding Configuration (For Remote Access Only)

Note: If you are running SNPcaster on your own PC, this configuration is **NOT** necessary.

When using a remote server, it is useful to configure SSH port forwarding so that accessing port 59829 on your local machine forwards the connection to port 59829 on the remote server.

Run the following command on your local machine:

Note: Replace the username and server IP address with the appropriate values for your server environment.

Note: If you are running the Docker container on your local machine, this step is not necessary.

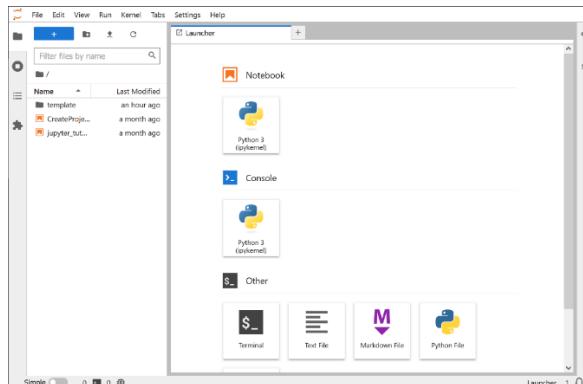
```
# ssh -L 59829:localhost:59829 <user name>@<remote server
IP>
```

```
ssh -L 59829:localhost:59829 ubuntu@192.168.32.55
```

4.1.3. Accessing Jupyter Lab

Launch a web browser such as Google Chrome, enter <http://localhost:59829/> in the address bar, and press Enter.

After a short while, Jupyter Lab will open. If the following screen appears, the setup is complete.



You are now ready to begin analysis. Please proceed to [5.Analysis Execution Procedure](#).

4.2. How to Stop SNPcaster Docker Container

Note: Stopping is not mandatory, but JupyterLab may not start properly after restarting your PC. In such cases, please stop the container using the method below and then restart SNPcaster.

Open your command-line application and navigate to the downloaded “snpcaster” folder. Then, run the following command to stop (actually remove) the Docker container:

Note: Even if you delete the container, the analysis data stored in the **project folder** will remain intact.

```
docker compose down -v
```

```
[+] Running 2/2
✓ Container snpcaster    Removed 0.7s
✓ Network k2077 snpcaster notebook default      Removed 0.3s
```

You can safely close the web browser that you used to access Jupyter Lab.

[Tips] Identifying the Active SNPcaster Folder

Search for the container to stop from the list of running containers.
Open a command-line application and run the following command.

```
docker ps -a
```

A list will be displayed as shown in the image below.

Find the container with a name that starts with snpcaster-<version number> in the "NAMES" column (on the right side of the image),
then copy the corresponding "CONTAINER ID" from the left side of the image.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
76d4ceaaaffc3	snpcaster:0.9.0	/usr/local/bin/entr_	21 hours ago	Up 11 minutes	0.0.0.0:59829->8888/tcp, [::]:59829->8888/tcp	snpcaster-0.9.0

By executing the following command, you can stop and remove the container.

Note: Replace {CONTAINER ID} with the CONTAINER ID you copied earlier.

```
docker rm -f {CONTAINER ID}
```

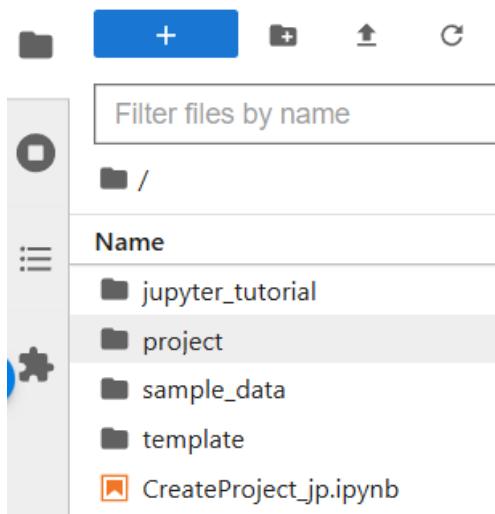
This completes the container shutdown.

5. Analysis Execution Procedure

After [Accessing Jupyter Lab](#) steps, perform the analysis by following the steps below.

5.1. Folder Structure

The initial folder structure is as follows:



- jupyter_tutorial
 - A folder that contains notebooks and other materials for learning how to use
- project
 - A folder for placing analysis projects.
 - Shared with the project folder inside the SNPcaster directory.
- sample_data
 - Folder containing sample reference sequences for STEC.
- template
 - Folder containing template notebooks for analysis.
- CreateProject.ipynb
 - Notebook for creating a new project

5.2. Learn How to Use Jupyter Lab

This section provides basic instructions for beginners on how to use Jupyter Lab.

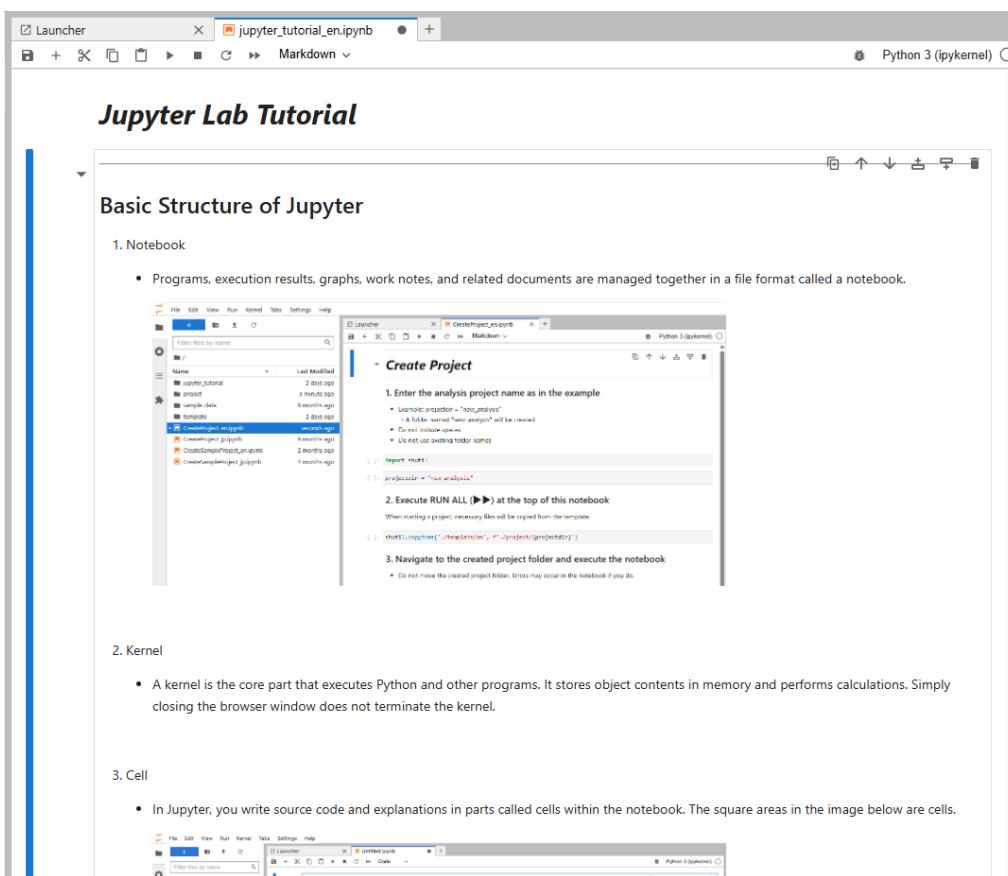
If you are already familiar with Jupyter Lab, feel free to skip ahead to [Creating a Project](#).

5.2.1. Jupyter Lab Tutorial

Double-click the "jupyter_tutorial" folder from the left sidebar, then double-click "jupyter_tutorial.ipynb" to open the notebook on the right pane.

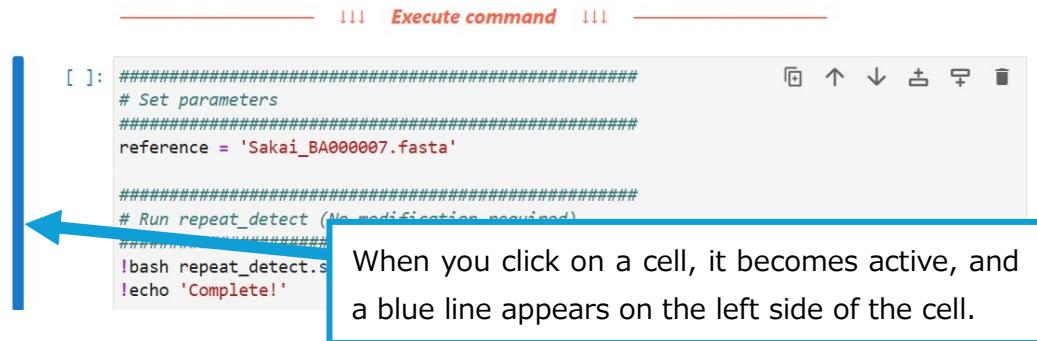
This notebook provides instructions on how to use Jupyter Lab, including how to edit and run Jupyter Notebooks.

If you are new to Jupyter, it is recommended that you read through this notebook first.



5.2.2. Basic Usage of Jupyter Lab

Click on the cell you want to execute.



The screenshot shows a Jupyter Lab interface with a code cell selected. A blue vertical bar is positioned on the far left of the cell, indicating it is active. The cell contains the following Python code:

```
[ ]: #####  
# Set parameters  
#####  
reference = 'Sakai_BA000007.fasta'  
  
#####  
# Run repeat_detect (No modification required)  
#####  
!bash repeat_detect.sh  
!echo 'Complete!'
```

A blue arrow points from the text "When you click on a cell, it becomes active, and a blue line appears on the left side of the cell." to the blue vertical bar.

When you click on a cell, it becomes active, and a blue line appears on the left side of the cell.

The brackets [] on the left are blank until the cell is run.



The screenshot shows a Jupyter Lab interface with a code cell selected. A blue vertical bar is positioned on the far left of the cell, indicating it is active. The cell contains the same Python code as the previous screenshot. Below the cell, there is a blank output area where no results have been displayed yet.

Nothing is displayed inside the square brackets [].

When a program is executed in each cell, the brackets on the left side of the cell change to [*], indicating that the cell is currently running.

```
[*]: #####  
      Set parameters  
#####  
# Run repeat  
#####  
!bash repeat
```

When you select this cell and press ► (Run), the program will be executed.
While the program is running, [*] will be displayed to indicate it is in progress.

When the program finishes running, a number will appear inside the brackets [].

```
[3]: #####  
      # Set parameters  
#####  
reference =  
#####  
# Run repeat  
#####  
!bash repeat  
echo 'Complete'
```

When the program finishes running, a number will appear inside the brackets [].
However, even if the program ends with an error, a number will still be displayed.
Please also check any generated files and log output for more details.

5.3. Creating a Project

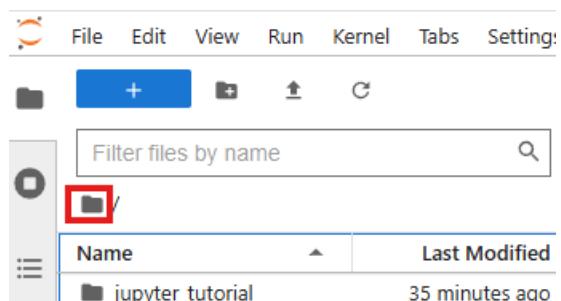
SNPcaster/grape is designed to process a group of strains to be compared within a single folder. This single folder corresponds to one project.

This section explains how to create a new project.

5.3.1. Creating a New Project

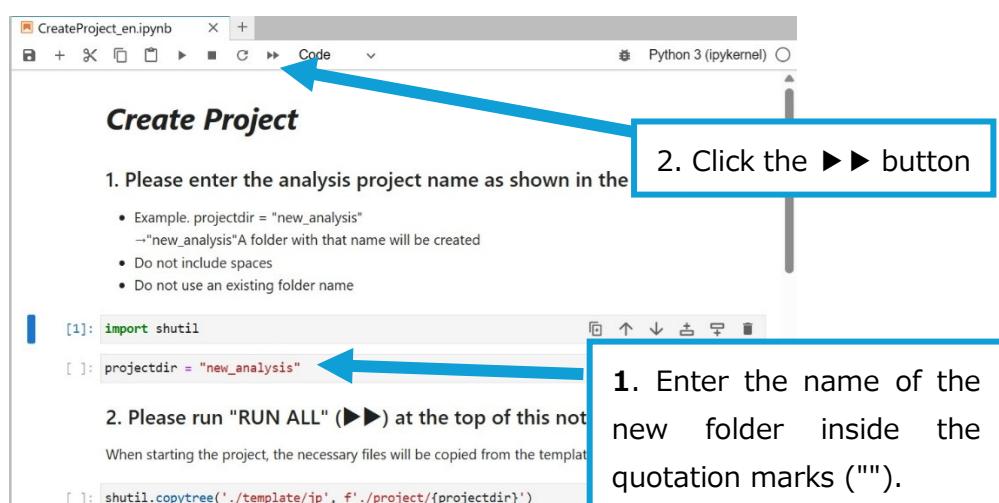
Click the  icon (highlighted by  in the image below) located under the search box in the left-hand sidebar.

This will navigate you to the root folder, and “/” will be displayed.



Double-click on “CreateProject_jp.ipynb” in the list on the menu bar. The Japanese project creation notebook will open on the right side.

Follow the instructions described in the notebook to create a project for analysis.



Create Project

1. Please enter the analysis project name as shown in the

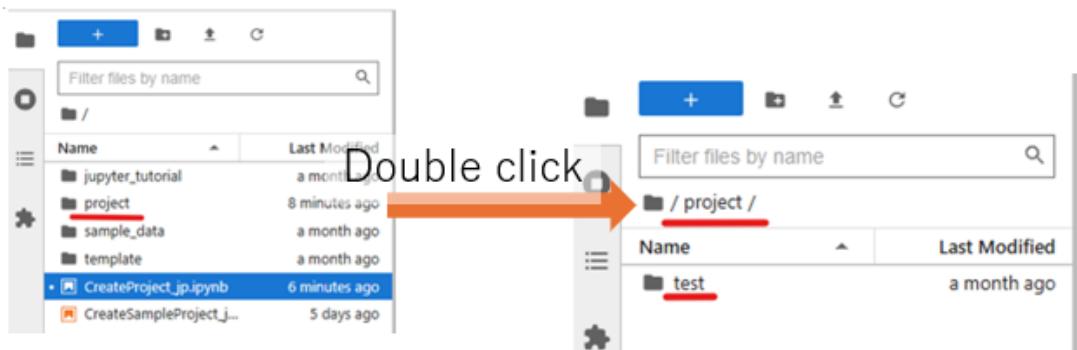
- Example: `projectdir = "new_analysis"`
→ "new_analysis" A folder with that name will be created
- Do not include spaces
- Do not use an existing folder name

2. Click the ►► button

1. Enter the name of the new folder inside the quotation marks ("").

```
[1]: import shutil  
[ ]: projectdir = "new_analysis"  
2. Please run "RUN ALL" (►►) at the top of this notebook  
When starting the project, the necessary files will be copied from the template  
[ ]: shutil.copytree('./template/jp', f'{projectdir}')
```

Once the execution of CreateProject.ipynb is complete, a new analysis project folder (referred to as the "test" folder below) will be created **within the project folder**.

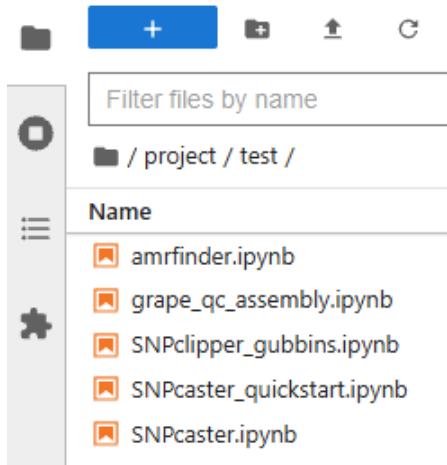


Important: Only use the project folder for your files and folders. Locations outside this folder are not synced with your host machine. Any files stored elsewhere will be lost when the Docker container is deleted or stopped via docker compose down.

5.3.2. Executing the Creation of a New Project

Double-click the newly created analysis project folder.

By default, the following files are included:



- amrfinder.ipynb
 - Analysis Notebook for AMR Genes, etc.
- grape_qc_assembly.ipynb
 - Analysis notebook for grape
- SNPcaster_quickstart.ipynb
 - Simplified analysis notebook for SNPcaster
- SNPcaster.ipynb
 - Analysis notebook for SNPcaster
- SNPclipper_gubbins.ipynb
 - Execution notebook for SNPclipper/gubbins

5.3.3. Running the Analysis

Double-click to open the notebook you wish to use for analysis from the list below.

Detailed instructions for the analysis are provided within each notebook, so please refer to the explanations inside.

For basic usage instructions, please refer to [Basic Usage](#).

- amrfinder.ipynb
 - This program detects antimicrobial resistance (AMR) and virulence genes.
- grape_qc_assembly.ipynb
 - Performs assembly and quality check of short-read data.
- SNPcaster_quickstart.ipynb
 - A simplified SNP analysis program for Shiga toxin-producing Escherichia coli (STEC).
 - All necessary reference sequences are pre-prepared, so you can start the analysis simply by preparing the short-read data and the strain list.
- SNPcaster.ipynb
 - This program includes all the necessary preparations for running SNPcaster, such as generating masked regions.
 - This program allows for detailed configurations, such as detecting repeat regions from the reference genome and performing analysis using a draft genome as the reference. It also includes detailed explanations of the output files.
- SNPclipper_gubbins.ipynb
 - This is a program that allows you to perform masking or clustered SNP analysis under different conditions after running SNPcaster, or to additionally execute Gubbins.

Note:

- Due to memory limitations, running multiple analyses simultaneously may cause the process to be interrupted. If this occurs, please execute only one analysis at a time.
- When analyzing a large number of strains at once, SNPcaster may terminate the analysis prematurely. The number of strains

that can be processed depends on the genome size of the test strains and the memory capacity of your system (approximately 300 strains with 192 GB of memory). If this issue occurs, please reduce the number of strains analyzed at one time.

- It has been confirmed that phylogenetic analysis does not function on low-end CPUs without AVX support (such as Intel Atom and Celeron) or on ARM-based processors (such as Apple Silicon). In these cases, please use the web-based version for phylogenetic analysis.

➤ <https://www.hiv.lanl.gov/content/sequence/IQ TREE/iqtreetree.html>

Usage Guide by Purpose

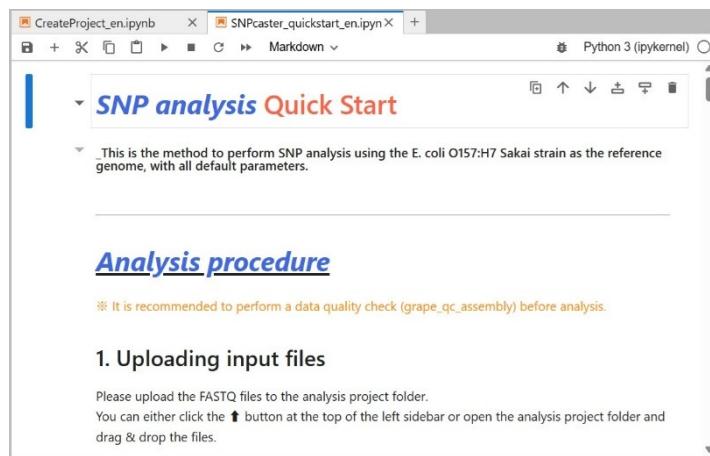
- If you want to check if everything is working correctly
 - Please use SNPcaster_quickstart.ipynb to perform the operation check.
- If you want to perform SNP analysis
 - It is recommended to perform data checking using grape_qc_assembly.ipynb before running the analysis with SNPcaster.ipynb.
For Enterohemorrhagic Escherichia coli (EHEC), you can also use SNPcaster_quickstart.ipynb.
- If you want to run Gubbins after SNP analysis, or consider different conditions for clustered SNPs.
- → Please use SNPclipper_gubbins.ipynb for this purpose.
- If you want to reanalyze samples that have already been analyzed.
 - Please refer to [5.3.5.Execution of SNP Reanalysis](#).
- If you want to use it via the command line.
 - Please refer to [6.Command Line Operations](#)(For Advanced Users).

5.3.4. Basic Usage

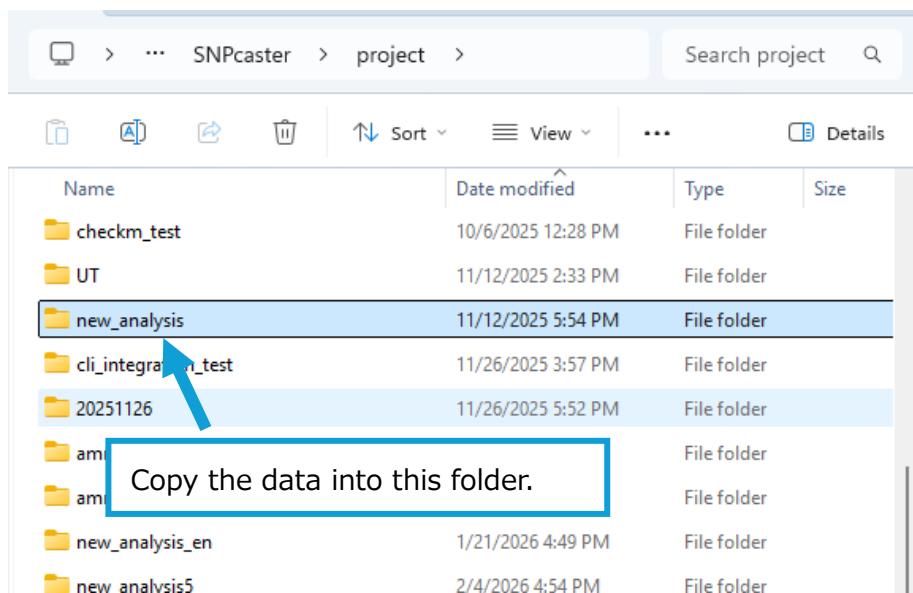
Here is an explanation of the basic usage using “SNPcaster_quickstart.ipynb” as an example.

5.3.4.1. Uploading Input Files

Please upload (copy) your data using either the (Upload) button in the JupyterLab sidebar or by dragging and dropping the files into the appropriate folder as shown below.



If you started with a project named "new analysis", folders like the ones below will be created. Please copy your files into the appropriate folder.



Name	Date modified	Type	Size
checkm_test	10/6/2025 12:28 PM	File folder	
UT	11/12/2025 2:33 PM	File folder	
new_analysis	11/12/2025 5:54 PM	File folder	
cli_integration_test	11/26/2025 3:57 PM	File folder	
20251126	11/26/2025 5:52 PM	File folder	
ami		File folder	
ami		File folder	
new_analysis_en	1/21/2026 4:49 PM	File folder	
new analysis5	2/4/2026 4:54 PM	File folder	

5.3.4.2. Creating the Strain List

Please create the strain list using the method described in JupyterLab.

If you use the method described in section 2.2, creating the list on **Windows may cause errors**.

So please use the method described in section 2.1 or 2.3.

The screenshot shows a JupyterLab interface with two open notebooks: 'CreateProject_en.ipynb' and 'SNPcaster_quickstart_en.ipynb'. The current notebook is in 'Markdown' mode, indicated by the tab at the top. The cell content is as follows:

2. Creating the strain list

The strain list will be created using one of the following methods.

2.1. Create using Jupyter Lab

- You can create a new file by clicking the '+' button at the top of the left sidebar and selecting 'Other' → 'Text file'.
- Once editing is complete, press Ctrl + S (Mac: Command + S), set the file name, and save it.

2.2. Create the file using Notepad, Text Editor, Vim, or any other editor, and then copy it to the analysis folder.

- There is a possibility that the file may not work correctly if created on Windows (due to differences in newline characters).
- Please create the file on Linux or Mac.

2.3. Create using the following command:

- After `file_name =`, enclose the list name in double quotes (e.g., "`file_name`").
- After `user_input = f""` input the strain list (one strain name per line)

5.3.4.3. Creating a FASTQ List

Please create the FASTQ list using the methods described in sections 3.1 and 3.2 of the JupyterLab notebook.

A FASTQ list is a file where each line contains:
the strain name,
followed by the names of the two paired-end short-read FASTQ files associated with that strain.

This list serves as the input for BactSNP, which is run internally by SNPcaster.

The screenshot shows a JupyterLab interface with two open notebooks: 'CreateProject_en.ipynb' and 'SNPcaster_quickstart_en.ipynb'. The current notebook is 'SNPcaster_quickstart_en.ipynb'. The visible content is as follows:

3. Creating the FASTQ list

The following steps will be used to create a list of FASTQ file names (short-read data file names) associated with each strain in the strain list created in step 2.

3.1. Run the `find_strain_pairs.py` script. [¶](#)

- After `file_name =`, enclose the strain list name created in the previous step in double quotes ("").
- If you want to modify any parameters as described in Set parameters, make the necessary changes and execute the command.
 - It is also possible to run the command without any modifications.

Execute command

```
[ ]: import subprocess
#####
# Set Parameters
#####
# Enter the file name of the strain List created in step 2
#> "strains.txt"
```

5.3.4.4. Running the SNPcaster Program

You can run SNPcaster by following the steps in the order shown in the image below.

The screenshot shows a Jupyter Notebook interface with two tabs: "CreateProject_en.ipynb" and "SNPcaster_quickstart_en.ipynb". The "SNPcaster_quickstart_en.ipynb" tab is active, displaying a Python code cell:

```
[ ]: #####  
# Parameter Configuration  
#####  
# Reference Genome File  
reference = '../sample_data/ehec/Sakai_BA00007.fasta'  
# Strain List File  
list = "list.txt"  
# The threshold for allele frequency  
allele_freq = 0.9  
# Cluster SNP Removal Interval  
cluster = 0  
# 0 means don't run Gubbins, 1 means  
gubbins = 0  
# Number of BactSNP Jobs  
jobs = 4  
# Number of Threads  
threads = 8  
# The file name containing a list  
fastq_list = "list_fastq.tsv"  
# If you want to apply masking (e.g.,  
#mask = "../sample_data/ehec/mask.fasta"  
#####
```

Annotations explain the steps:

1. Clicking this cell will activate it (a blue line will appear on the left side).
2. Only edit this section.
For EHEC analysis, modify only the list name (the filename created in step 2 above) and the number of threads (adjust according to the specifications of your analysis system).
3. Press the ► button.

5.3.4.5. Constructing a Phylogenetic Tree

The results files from SNPcaster are organized into a single folder. The folder name follows the format: "snpcaster_[date][time][list name]", and it differs for each analysis.

To construct a phylogenetic tree, you need to specify files within the result folder, so please change the input file name accordingly.

The screenshot shows a Jupyter Notebook interface with two tabs: "CreateProject_en.ipynb" and "SNPcaster_quickstart_en.ipynb". The "SNPcaster_quickstart_en.ipynb" tab is active, displaying a configuration section and a command cell:

If necessary, change or remove the name of the reference sequence (default is "Ref").

- command
 - bash snpcaster_20240115_094704.list_test*.log.sh [input] [threads] [bootstrap]
- Options

Parameter	Required	Description
input	•	Input file (e.g., xx.fasta). Input alignment in PHYLIP/FASTA/NEXUS/CLUSTAL/MSF format
threads	•	Number of threads for modestest-ng (e.g., 12). Threads for raxml-ng are selected automatically
bootstrap	-	Bootstrap (default is 100)

Please check the log file raxml-ng.log in the execution directory.

```
[ ]: #####  
# Set parameters  
#####  
input = "snpcaster_20240115_094704.list_test/3.list_test.log.sh"  
# Remove the # at the beginning of the line  
# Delete the above input line + remove the # at the beginning of the line  
# input = "snpcaster_20240115_094704.list_test/5.results.log.sh"  
threads = 8  
bootstrap = 1000  
#####  
# Run raxml-ng  
#####
```

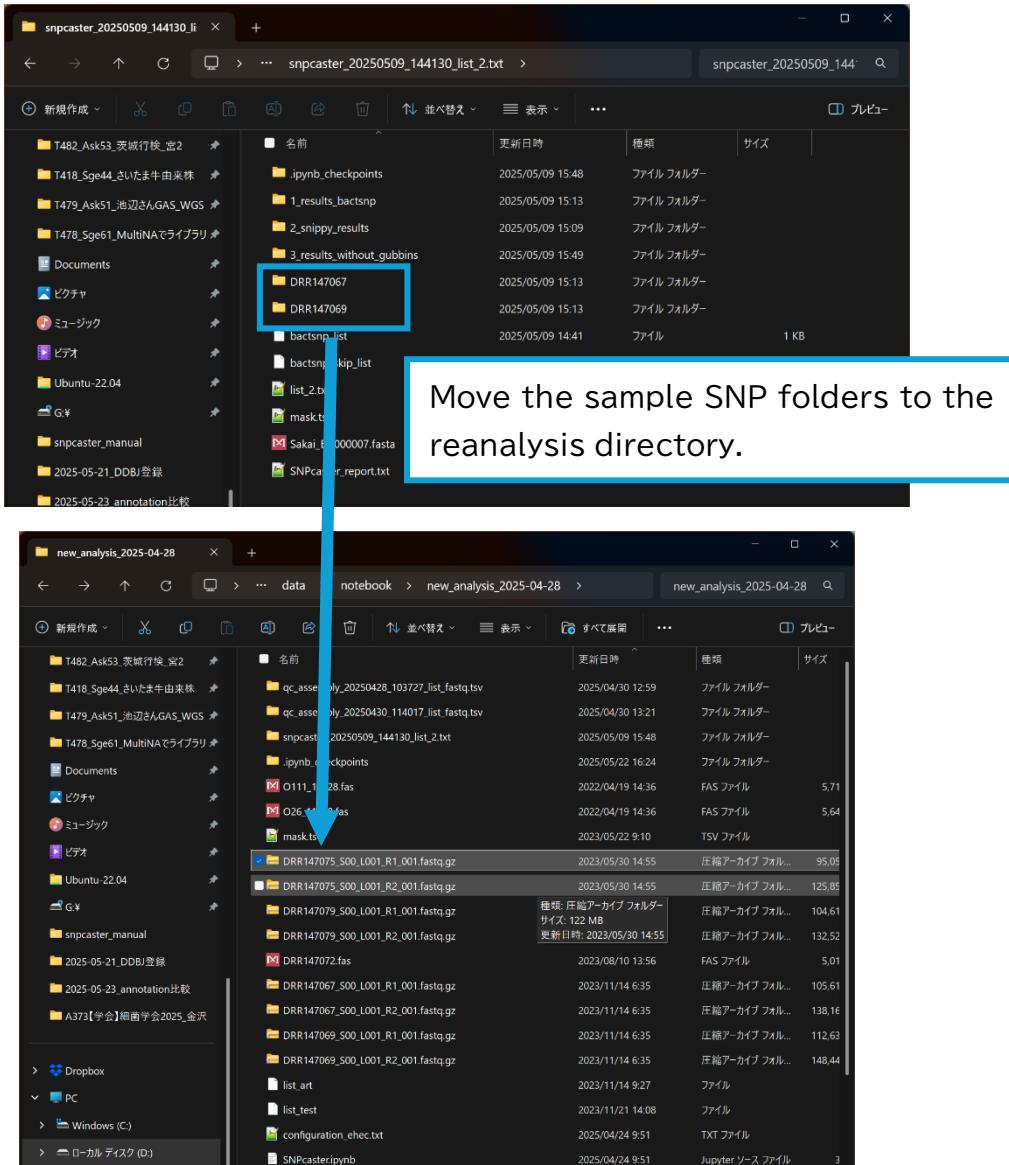
Annotations explain the configuration:

- Only this section should be edited.
- Please replace the analysis folder name with the actual folder name.
- Also, adjust the number of threads according to the specifications of your analysis machine.

5.3.5. Execution of SNP Reanalysis

By using the functionality of snippy-core for extracting core genome SNPs, rapid reanalysis becomes possible. For example, suppose that three strains—A, B, and C—were analyzed in the initial run. When performing SNP analysis that includes a new strain D (i.e., strains A, B, C, and D), it is possible to skip the BactSNP analysis for strains A, B, and C by utilizing the sample SNP folder obtained from the initial analysis, and run BactSNP only for strain D. Since the majority of analysis time is consumed by BactSNP, this leads to a significant reduction in processing time. Conversely, if reanalysis is performed only with strains A and B, BactSNP is not executed at all. Even in this case, a substantial time saving can be expected.

5.3.5.1. Move the sample SNP folder into the analysis directory.



5.3.5.2. Execute snpcaster.sh.

At this point, the FASTQ list does not need to include information for reanalysis strains, i.e., those for which sample SNP folders already exist.

6. Command Line Operations

All scripts that can be executed in Jupyter Notebook are already included in the system PATH within the Docker container, so they can be run from any directory inside the container. This section describes several basic methods for executing these scripts via the command line. Please choose the method that best suits your preference. For script filenames and command-line arguments, please refer to the Jupyter Notebook.

6.1. Launching the Command Input Application

Please start the appropriate application for your environment from the options below.

6.1.1. Windows

6.1.1.1. For Rancher Desktop or Docker Desktop user

Proceed to [Creating Docker Images and Containers](#).

6.1.1.2. For Docker Engine user

After launching the Terminal or PowerShell, log into WSL using the following command, then proceed to [Creating Docker Images and Containers](#).

```
wsl.exe -d Ubuntu-24.04
```

6.1.2. Mac

6.1.2.1. For Rancher Desktop or Docker Desktop user

Proceed to [Creating Docker Images and Containers](#).

6.1.2.2. For Docker Engine user

After launching the terminal, use the following command to start the VM for snpcaster, then proceed to [Creating Docker Images and Containers](#).

```
limactl start snpcaster-vm
```

6.1.3. Linux

6.1.3.1. For Docker Engine

Proceed to create [Creating Docker Images and Containers](#).

6.2. Creating Docker Images and Containers

When executing from the command line, it is also necessary to create the Docker image or container using docker-compose beforehand.

If you haven't done so yet, please follow the section on [Execute the Start Command](#). After that, execute the command either [from the Docker container](#) or [from the Docker image](#), depending on the method you prefer.

6.3. Run from the Docker Container

This is the execution method after creating the Docker container.

6.3.1. Data Placement

On the host machine (i.e., your PC's OS environment), place the required folders and read data in the `[SNPcaster download folder]/project` directory.

This folder is mounted to the container (i.e., the virtual environment). Mounting is similar to using a shared folder, so when you place files and folders in this directory on the host machine, they will be treated as if they are located in `/home/snpcaster/notebook/project` inside the container.

6.3.2. Access the Container via Bash

The container is started by [Execute the Start Command](#).

Use the following command to check the container's ID or name (either is fine) that will be used for execution.

```
docker ps
```

As shown below, the leftmost column is the ID, and the rightmost column is the name.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
812c6b62d47e	snpcaster:0.9.0	"/usr/local/bin/entr..."	5 hours ago	Up 5 hours	0.0.0.0:59829->8888/tcp, [::]:59829->8888/tcp	snpcaster-0.9.0

Executing the following command will launch a bash terminal within the container environment.

Note: Replace [Container name or ID] with the one you identified

above.

```
docker exec -it [Container name or ID] bash
```

The screen will switch to the bash terminal, then use the following command to navigate to the mounted folder.

```
cd /home/snpcaster/notebook/project
```

Since the folders and files you placed on the host are now available, you can proceed to execute any command of your choice.

6.4. Run from the Docker Image

This is the method for executing using a Docker image. A temporary container is created from the Docker image, and the specified command can be executed within that container.

6.4.1. Data Placement

Place the required folders and read data for the analysis in any directory of your choice on the host machine (i.e., your PC's OS environment).

6.4.2. Check the Image ID

Execute the following command to display a list of SNPcaster images.

```
docker images -f "reference=snpcaster"
```

The TAG column represents the version number, so find the image of the SNPcaster version you wish to use from the list and copy the IMAGE ID.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
snpcaster	0.9.0	d457f9a73f20	5 hours ago	16.6GB

6.4.3. Execute the Command

You can execute any command of your choice using the following command.

```
docker run -it --rm -v "{The absolute path of the folder to be mounted}: /home/snpcaster/notebook/project" {Image ID} {The command you want to execute + arguments }
```

Note:

In the SNPcaster image, `/home/snpcaster/notebook/project` is set as the default working directory (specified via the WORKDIR instruction in the Dockerfile).

Therefore, by mounting your local directory to `/home/snpcaster/notebook/project` as shown above, your files will be directly accessible within the working directory during execution.

Here is an example of execution.

Due to the need to specify the absolute path of the folder to be mounted, the command for execution slightly differs depending on whether you are using Windows or Mac/Linux (the only difference is whether to use `$($PWD.Path)` or `$(pwd)`).

Example command for running snpcaster.sh on Windows (Terminal/PowerShell):

```
docker run -it --rm -v "$($PWD.Path)/new_analysis:/home/snpcaster/notebook/project"
d457f9a73f20 snpcaster.sh -i list.txt -r Sakai_BA000007.fasta
-f list_fastq.tsv
```

Example command for running snpcaster.sh on Mac/Linux:

```
docker run -it --rm -v  
"$(pwd)/new_analysis:/home/snpcaster/notebook/project"  
d457f9a73f20 snpcaster.sh -i list.txt -r Sakai_BA000007.fasta  
-f list_fastq.tsv
```

Alternatively, by replacing the command with bash, you can start the SNPcaster container with your current directory mounted to /home/snpcaster/notebook/project.

Example:

```
docker run -it --shm-size 8g --rm -v "(pwd):  
/home/snpcaster/notebook/project" {Image ID} bash
```

This allows you to use **TAB completion**, making command entry much smoother. (Note: TAB completion is not available when executing commands directly through the docker run ... [COMMAND] syntax.)

To finish your session, simply type exit.

7. Program Overview

7.1. SNPcaster

7.1.1. Advantages of the Program

SNPcaster is a program that combines BactSNP, Snippy, and Gubbins. Its main advantages are as follows: 1) Accurate SNP extraction, 2) Rapid re-analysis, and 3) Automatic removal of recombination regions, etc.

First, let's list the advantages and disadvantages of the existing SNP extraction software, BactSNP and Snippy.

Table 7.1. Advantages and Disadvantages of BactSNP and Snippy

	BactSNP	Snippy
Advantages	Accurate SNP Extraction	Rapid Re-analysis Gubbins-specific files are automatically generated
Disadvantages	<ul style="list-style-type: none">• Re-execution requires starting from scratch.• Creating Gubbins-specific files is time-consuming.• Processing a large number of strains may lead to program crashes due to memory shortages.	Depending on the data, SNPs may be excessively extracted.

As shown in the table, BactSNP is an SNP extraction pipeline with high accuracy, but it also has a high rate of detecting false positive SNPs (1, 2). On the other hand, re-analysis requires starting from scratch for all strains, which takes time. Additionally, to use other programs like Gubbins, it is necessary to edit the output files appropriately. Snippy is a widely used program around the world (<https://github.com/tseemann/snippy>). For re-analysis, by using the folders generated for each strain, SNP analysis can be skipped for strains that have already been processed, allowing for faster re-analysis. However, it is known to generate false positive SNPs (1, 2).

In SNPcaster, the results from BactSNP are edited and made compatible for analysis with Snippy (using the snippy-core function). By analyzing them with snippy-core, the program combines the advantages of both tools. A disadvantage is that the analysis time is slightly longer than that of BactSNP. However, since the entire analysis process is automated, considering the preparation steps before running the program and organizing the output files, the overall analysis time is expected to be shorter.

The advantages of SNPcaster are as follows:

SNPs with the same high accuracy as BactSNP can be obtained.

- By using the strain-specific sample SNP folders (which contain pseudogenomes and vcf files), BactSNP can be skipped during re-analysis, allowing for faster re-analysis.
- The automated pipeline allows for the automatic removal of recombination regions using tools like Gubbins.
- The above analyses can be performed automatically by simply providing a list of strains.
- SNPs with high accuracy, comparable to BactSNP, can be obtained.
- BactSNP と同等の高い正確性の SNP が得られる。

Table 7.2. Comparison of BactSNP, Snippy, and SNPcaster

	BactSNP	Snippy	SNPcaster
SNP Accuracy	○	△	○ (Comparable to BactSNP)
Rapid Re-execution	×	○	○
Creation of Gubbins data	△	○	○
Automation of Analysis	×	×	○
Handling numerous strains	△	○	○

7.1.2. Analysis Workflow

This program primarily uses BactSNP, Snippy, and Gubbins, and integrates them into a workflow using scripts (Figure 7.1).

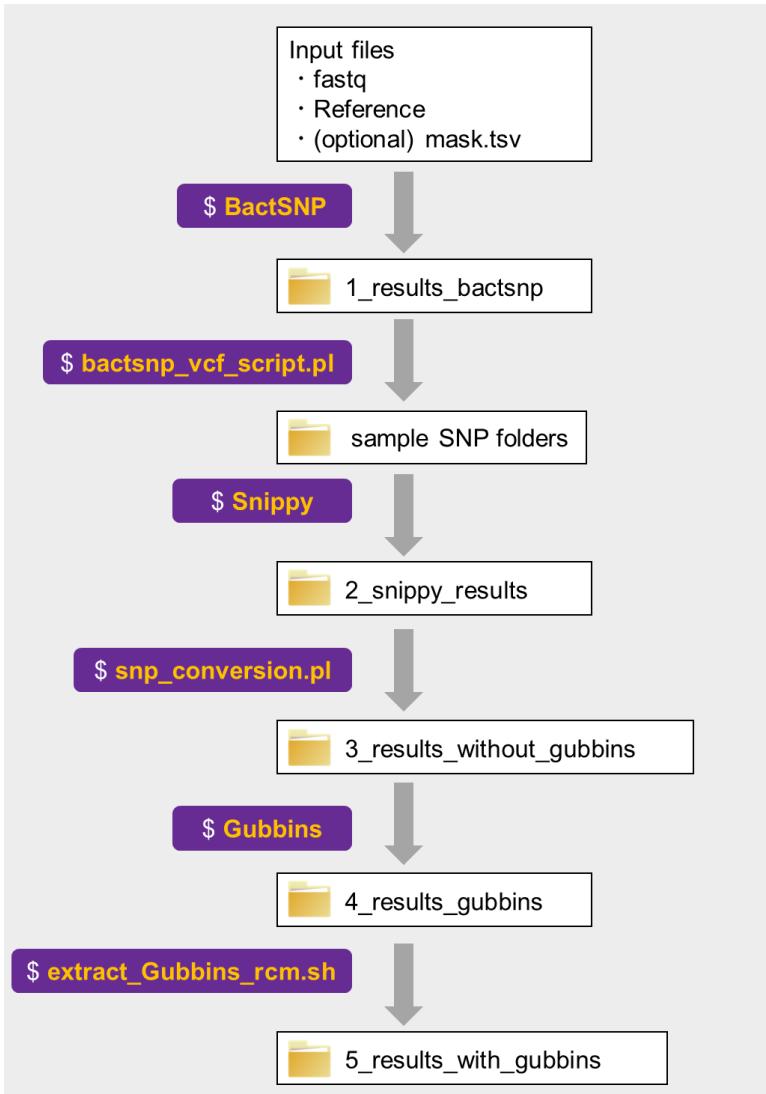
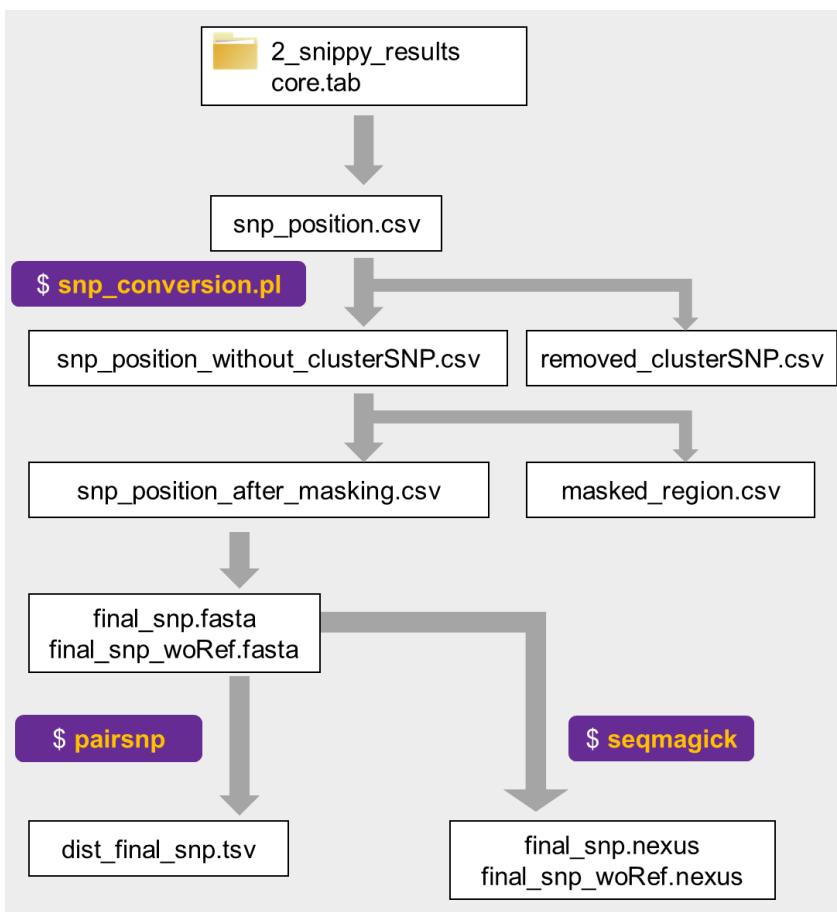


Figure 7.1.
SNPcaster Workflow

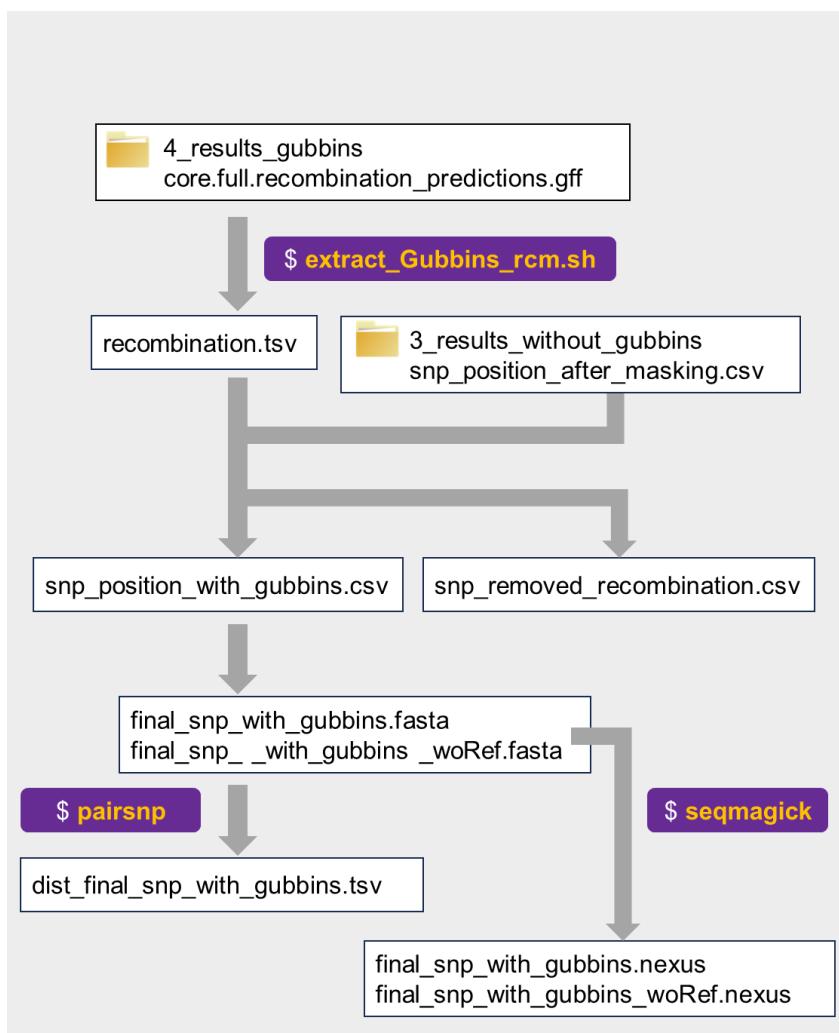
First, BactSNP and Snippy (snippy-core) are run to extract SNPs. The output files from each program are stored in the directories 1_bactsnp_results and 2_snippy_results, respectively. From these files, clustered SNPs and masked regions are removed (if specified as options), and result files are generated for the case where Gubbins is not used (Figure 7.2). These files are stored in the 3_results_without_gubbins folder.

Figure 7.2. File Processing Flow After Snippy



If recombination region detection by Gubbins is to be performed, Gubbins will be executed based on the core.full.aln file obtained from Snippy. The result files from Gubbins are stored in the 4_results_gubbins folder. Using the recombination region information obtained from this analysis and the SNP information in the 3_results_without_gubbins folder, recombination regions detected by Gubbins are removed (Figure 7.3). As a result, the SNP information excluding the recombination regions detected by Gubbins is stored in the 5_results_with_gubbins folder.

Figure 7.3. File Processing Flow After Gubbins



Many files in the 3_results_without_gubbins and 5_results_with_gubbins folders are similar. For example, final_snp_without_gubbins.fasta and final_snp_with_gubbins.fasta. By comparing these files, it is easy to assess the impact of recombination region removal by Gubbins.

7.1.3. Output Files

When SNPcaster is executed, a folder named `snpcaster_[YYYYMMDD][HHMMSS][list_name]` is created. For example, if the analysis is started at 8:25:15 AM on March 9, 2025, using a list file named "list," the folder `snpcaster_20250309_082515_list` will be created.

Inside the folder, the following files will be stored (the contents of the folder are explained in the next section).

- **SNPcaster report.txt:** Contains the parameters used and information after SNP extraction (described later).
- **Reference genome file:** A copy of the file used.
- **List file:** A copy of the list file used.
- **Masked region file:** A copy of the masked region file used.
- **bactsnp_skip_list** (if any strains were skipped during BactSNP = if sample SNP folders existed): A list of strains that were skipped.
- **missing_list** (if the fastq files for any strains are missing): A list of strains for which the specified fastq files are not found.

The SNPcaster report.txt file contains the following information:

Version	Version number of the SNPcaster program
Strain list	Name of the strain list file used
Fastq list	Name of the fastq file used
Mask file	Name of the file containing masked (excluded from analysis) regions
Reference file	Name of the reference genome file used
Clustered SNP removal (bp)	Interval of SNPs removed as clustered SNPs (value of the -c option)
Threads	Value for parallel processing (-t option value)
BactSNP allele-freq	Allele frequency threshold used in BactSNP. For example, with a value of 0.9, the SNP will not be detected unless more than 90% of the short reads support the SNP (-a option value).
BactSNP jobs	Parallel processing value for BactSNP (-j option value)
Recombinogenic region detection (Gubbins)	Whether Gubbins was performed (ON/OFF)
Core genome size (bp)	
Before Gubbins	Core genome size before performing Gubbins. If specified, masked regions are also excluded.
After Gubbins	Core genome size after performing Gubbins
Number of informative SNP sites (with reference)	
No masking	Number of positions where SNPs are located in the reference genome (i.e., length of the sequence in final.snp.fasta)
After clustered SNP removal	(Only if clustered SNPs are specified) Number of SNPs remaining after removing clustered SNPs
After masking	(Only if masked regions are specified) Number of SNPs remaining after removing masked regions
After Gubbins	(Only if Gubbins is performed) Number of SNPs remaining after removing recombinant regions identified by Gubbins

Number of informative SNP sites (without reference)	
No masking	Number of positions with differences between samples from the above SNP locations
After clustered SNP removal	The same as above
After masking	The same as above
After Gubbins	The same as above
Removed SNPs	
Clustered SNP	Number of SNPs removed as clustered SNPs
Masked region	Number of SNPs removed as masked regions
Gubbins	Number of SNPs removed as recombinant regions by Gubbins

The folders stored in the results folder are as follows:

- **Sample name folder:** Contains SNP information for each sample. Details are provided later.
- **1_results_bactsnp:** BactSNP result files. These will not be created unless BactSNP is executed.
For more details, visit <https://github.com/IekAdN/BactSNP>
- **2_snippy_results:** Snippy result files.
For more details, visit <https://github.com/tseemann/snippy>
- **3_results_without_gubbins:** Contains results when Gubbins recombination removal was not performed. Details are provided later.
- **4_results_gubbins:** Gubbins result files.
For more details, visit <https://github.com/nickjcroucher/gubbins>
- **5_results_with_gubbins:** Contains results when Gubbins recombination removal was performed. Details are provided later.

Sample name folder

This folder contains the pseudogenome file (snps.aligned.fa), which was converted from the BactSNP results for use with snippy-core, and the file (snps.vcf) that shows the SNP locations. During re-analysis, placing this folder in the directory to be analyzed will skip the BactSNP step, thereby reducing the processing time.

3_results_without_gubbins

This folder contains the following files. The file with the prefix final.snp is used for phylogenetic analysis. Other files can be checked as needed.

SNP sequences: These are the sequences that contain only the differing parts of the core genome. These are used for phylogenetic analysis and other purposes.	
final.snp.fasta	FASTA format: Includes reference sequences.
final.snp.nex	NEXUS format: Includes reference sequences.
final.snp_woRef.fasta	FASTA format: No reference sequences included.
final.snp_woRef.nex	NEXUS format: No reference sequences included.
The following files contain information on SNP positions and the number of SNPs between strains. Please review them as needed.	
core.full.fasta	The sequence of the core genome region (core.region.tsv).
core_region.tsv	The core genome region (the region common to all strains).
core_region_after_masking.tsv	Core genome region after masking. This file is generated by removing the masked regions (from mask.tsv) from the core_region.tsv.
core_region_final.tsv	Core genome region after various processes. If masking is enabled, the contents will be the same as core_region_after_masking.tsv; otherwise, it will be the same as core_region.tsv.
dist_final.snp.tsv	Pairwise SNP distance (number of SNPs between strains).
dist_final.snp_matrix.tsv	The matrix format of dist_final.snp.tsv
snp_position.csv	The CG SNPs generated by Snippy. The information in this file is the same as the "core.tab" file generated by Snippy.
snp_position_sample_only.csv	A list of SNPs with different bases between samples, extracted from the snp_position.csv file.
snp_position_without_clustered SNP.csv	CG SNPs excluding clustered SNPs. This file will not be created if the clustered SNP distance is set to 0.
snp_position_without_clustered SNP_sample_only.csv	A list of SNPs with different bases between samples, extracted from the snp_position_without_clusteredSNP.csv file.
removed_clusterSNP.csv	Clustered SNPs that have been removed. This file will not be created if the cluster SNP distance is set to 0.
snp_position_after_masking.csv	CG SNPs excluding clustered SNPs and masked regions. The SNPs in this file are the same as those in the final.snp.fasta.
snp_position_after_masking_sample_only.csv	A list of SNPs with different bases between samples, extracted from the snp_position_after_masking.csv file.
masked_region.csv	CG SNPs removed in the masked regions.
snp_position_final.csv	CG SNPs after the removal of clustered SNPs and the application of masking.

5_results_with_gubbins

This folder contains the following files. The files with "final.snp" are those used for phylogenetic analysis. Other files can be checked as needed. Many of these files are identical to those in the 3_results_without_gubbins folder.

SNP 配列	
final.snp_after_gubbins.fasta	FASTA format: Includes reference sequences.
final.snp_after_gubbins.nex	NEXUS format: Includes reference sequences.
final.snp_after_gubbins_woRef.fasta	FASTA format: No reference sequences included.
final.snp_after_gubbins_woRef.nex	NEXUS format: No reference sequences included.
Other files.	
core_summary_with_gubbins.tsv	Core genome regions.
dist_final.snp_with_gubbins.tsv	Pairwise SNP distances.
dist_final.snp_matrix_without_recombination.tsv	Matrix format of the dist_final.snp_with_gubbins.tsv file.
recombination_region.csv	CG SNPs removed as recombinant regions by Gubbins.
snp_position_after_gubbins.csv	SNP positions excluding clustered SNPs, masked regions, and recombinant regions detected by Gubbins. The SNPs in this file are the same as those in final.snp_with_gubbins.fasta.
snp_position_after_gubbins_sample_only.csv	A list of SNPs with different bases between samples, extracted from the.snp_position_with_gubbins.csv file.

引用文献

- 7.1. Yoshimura D., Kajitani R., Gotoh Y., Katahira K., Okuno M., Ogura Y., Hayashi T., and Itoh T. Evaluation of SNP calling methods for closely related bacterial isolates and a novel high-accuracy pipeline: BactSNP. 2019. Microbial genomics 5.
- 7.2. Derelle R., von Wachsmann J., Maklin T., Hellewell J., Russell T., Lalvani A., Chindelevitch L., Croucher N. J., Harris S. R., and Lees J. A. Seamless, rapid, and accurate analyses of outbreak genomic data using split k-mer analysis. 2024. Genome Res. 34:1661-1673.

7.2. grape_qc_assembly

This program trims short-read sequences (optional), then calculates data metrics (such as coverage). At the same time, it performs de novo assembly using SPAdes or SKESA to construct a draft genome. Based on the resulting draft genome, checkM and QUAST are used to assess contamination and other factors. When a configuration file is used, the assembly is performed only with data that meets the specified coverage threshold, and the results of the quality check (QC) are output.

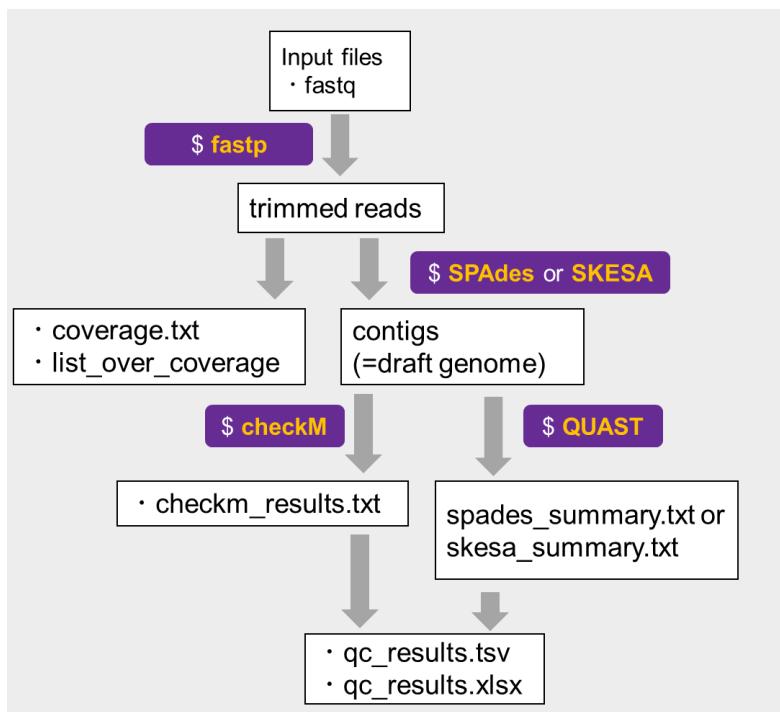


Figure 7.4.
Processing Flow of
grape_qc_assembly

The files created are as follows:

[Strain].fasta	Draft genome
assembly_summary.tsv	A summary of statistics such as contig length, calculated using QUAST.
coverage.txt	Coverage summary.
list_assembly.tsv	A list of FASTQ files for strains included in list_over_coverage.
list_over_coverage	A list of strains that exceed the coverage specified in the configuration file. Only the strains in this list will be assembled.
qc_results.tsv	The results of QUAST and CheckM (such as genome size, contamination, etc.) will be compared to the ranges specified in the configuration file. If the values fall within the specified range, "pass" will be indicated; if they fall outside, "fail" will be noted. Only when all values are "pass," the "qc_results" (overall assessment) column will be marked as "PASS."
qc_results.xlsx	The Excel version of qc_results.tsv (with the same content).
qc_results.xlsx	The Excel version of qc_results.tsv (with the same content).

When cleanup=0 is specified, the following folders, including intermediate files, will remain and not be deleted:

- fastp folder: Contains the following FASTQ files trimmed by fastp:
 - [strain_name]_1(2).fastq.gz: Read1 or Read2 files with paired reads found.
 - [strain_name]_u1(u2).fastq.gz: Read1 or Read2 files without a pair found.
- Contains the results of the QUAST execution.
 - For details of the output, please refer below.
<https://github.com/ablab/quast#output>
- checkkm folder: Contains the results of the CheckM execution.
 - For details of the output, please refer below.
<https://github.com/Ecogenomics/CheckM>

8. Troubleshooting and Q&A

Note: The methods of analysis and interpretation of results may vary depending on the species. Even within a species, the interpretation might differ by strain. The examples provided below have been validated for *Escherichia coli* (particularly enterohemorrhagic *E. coli*), but please note that these may differ for other species.

Assembly & QC Program

Q1. Are there any standard QC criteria?

A1. There is no unified standard. However, the CDC's PulseNet has published SOPs that provide certain guidelines for quality control.

<https://www.aphl.org/programs/global-health/Pages/PulseNet-International-SOPs.aspx>

Q. Which assembler is better, SPAdes or SKESA?

A. There is no significant difference between the two. SPAdes tends to produce longer contigs, while SKESA is often considered to have better contig accuracy. For more details, please refer to the next page.

<https://github.com/rrwick/Trycycler/wiki/Guide-to-bacterial-genome-assembly>

SNP Analysis

Q. What is the difference between SNP and SNV?

A. SNP stands for single nucleotide polymorphism, while SNV stands for single nucleotide variation. An SNP refers to a variation that is present at a certain frequency (usually a few percent) within a population, whereas SNV is a more general term that refers to any single nucleotide variation. Although the precise definitions differ, the terms are often used synonymously in microbial genomics.

Q. What is a clustered SNP?

A. A clustered SNP refers to SNPs that are located close to each other. For example, it refers to a situation where another SNP is found within 10 bp of a given SNP. When the distance between SNPs is small, it suggests

that the mutations may not have occurred independently, but rather, they might have arisen together due to recombination. While there is no specific threshold for this distance, clustered SNPs may be excluded from analysis in some cases.



CTACCGTGAGCTGGGGGGATCTCTCTACTCCAAACG
CTACCGTGAGCTGGGGGATCTCTCTACTCCAAACG

Whether or not to exclude clustered SNPs, and the specific distance (in bp) within which to exclude them, should be considered on a species-by-species basis.

In the case of *Escherichia coli* studies involving outbreak investigations, clustered SNPs occur very rarely (in only a few percent of cases). Therefore, it is advisable to initially conduct the analysis without excluding clustered SNPs. If a large number of SNPs are observed between strains that are epidemiologically close, then excluding clustered SNPs can be considered as a next step.

Q. Is Gubbins necessary?

A. Gubbins is used to remove recombinant regions. Whether to use it or not, like with clustered SNPs, should be considered on a species-by-species basis.

In outbreak investigations of *Escherichia coli*, the use of Gubbins typically does not show significant differences (less than 1%). Therefore, similar to the approach with clustered SNPs, it is recommended to first conduct the analysis without using Gubbins. If many SNPs are observed between strains that are epidemiologically close, then Gubbins can be applied as a next step.

Q. How do I choose the reference genome?

A. Using a closely related strain (e.g., one strain from an outbreak cluster) will result in the detection of a larger number of SNPs. This is because the core genome size (the region where SNPs can be detected) increases.

When selecting a reference genome, the core genome size output from

the SNPcaster report can be helpful. For example, in *Escherichia coli* with a genome size of 5.5Mb, if the core genome size is 2Mb, using a closely related strain as the reference may result in a core genome size of over 4Mb. In such cases, the number of SNPs theoretically doubles or more.

Q. `raxml-ng` is not working.

A. The program may not function properly if the path name is too long. Try moving the files or renaming the folder containing the SNP files you are analyzing.

Q. It's not working for some reason.

A. If files created on Windows are mixed in, they might have different line-ending codes, causing the program to malfunction (especially for strain lists). In this case, recreate the file on Linux or Mac, or use the following command to change the line-ending codes:

```
nkf -d --overwrite [filename]
```

Additionally, if folder names contain full-width (double-byte) characters, the program may fail to run.

Q. The fastq file cannot be loaded.

A. SNPcaster only accepts paired-end data obtained from Illumina sequencers as input files. Additionally, the file names must follow one of the following formats:

- Illumina format (e.g., `strain_S00_L001_R1_001.fastq.gz`)
- Simple format (e.g., `strain_R1.fastq.gz`)

Q. The analysis in JupyterLab is stuck with [*] indefinitely.

A. Try stopping the execution and then running the program again. If you were analyzing a large number of strains, try reducing the number of strains for the analysis and running it again.

9. reference materials

9.1. About Docker

Docker is a platform used to create, distribute, and run virtual environments within a computer. A virtual environment is a setup that makes it appear as if a separate computer is running inside a single physical machine. Traditionally, tools such as VMware and VirtualBox have been used to create virtual environments. However, these tools start up an entire independent operating system within each virtual machine, which results in slow startup times and large file sizes due to the need for a full OS image.

In contrast, Docker runs virtual environments by borrowing the necessary OS functionalities from the host operating system through a program called the Docker Engine. Because of this, Docker containers do not need to independently run most of the OS processes themselves. As a result, Docker environments are lightweight and start up quickly.

Docker can run multiple pre-configured programs, and the virtual environment containing these programs is called **container**.

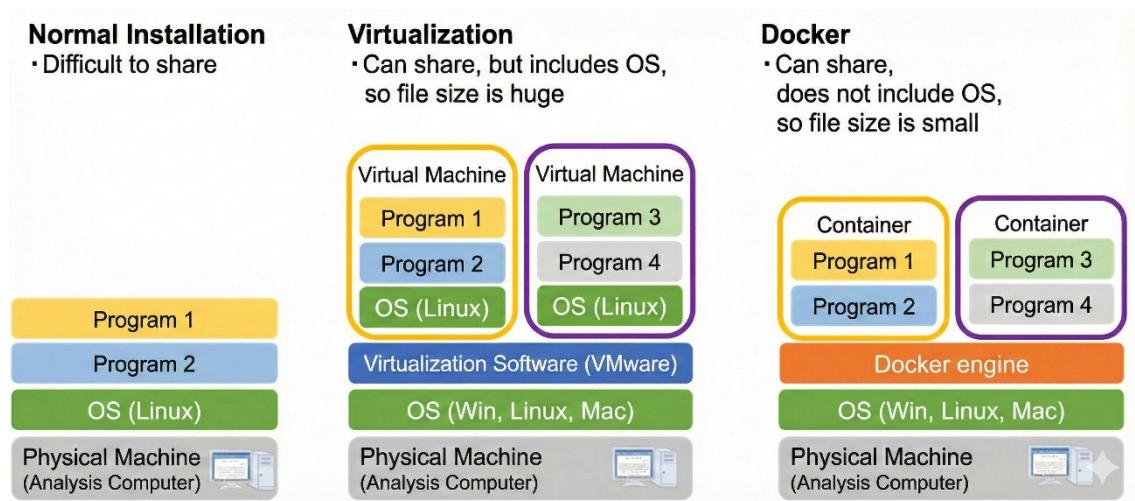


Figure 9.1. Differences between Docker and Virtualization

To create a Docker container, an **image** is required. While the distinction between an image and a container can be confusing, you can think of an image as a blueprint, and the container as the actual running virtual environment on your computer.

Images can be shared globally via a dedicated server called Docker Hub, where many organizations, companies, and individuals have published images pre-installed with various tools such as Python, R, and more.

In SNPcaster, the command `docker compose up` performs the following three steps; image creation, container creation and container startup. During the image creation step, tools like BactSNP and other necessary software are installed. The result is a single image file containing all the required programs. This step can take some time during the first startup, but once the image has been created, subsequent startups are much faster since they only involve creating a container from the existing image.

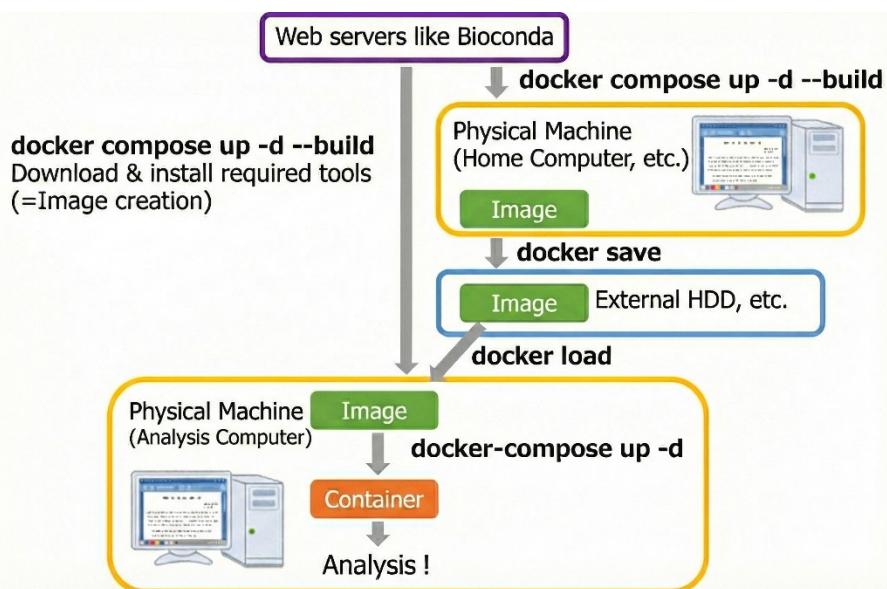


Figure 9.2. Workflow from Image Building to Analysis Execution in SNPcaster

The `docker compose` command installs the necessary software from external servers. In environments with strict security constraints (e.g., behind a proxy), you can build the image on a machine with less restricted access and then import it via an external hard drive or other storage media (as shown on the right side of the diagram).

9.2. How to Install Docker Desktop

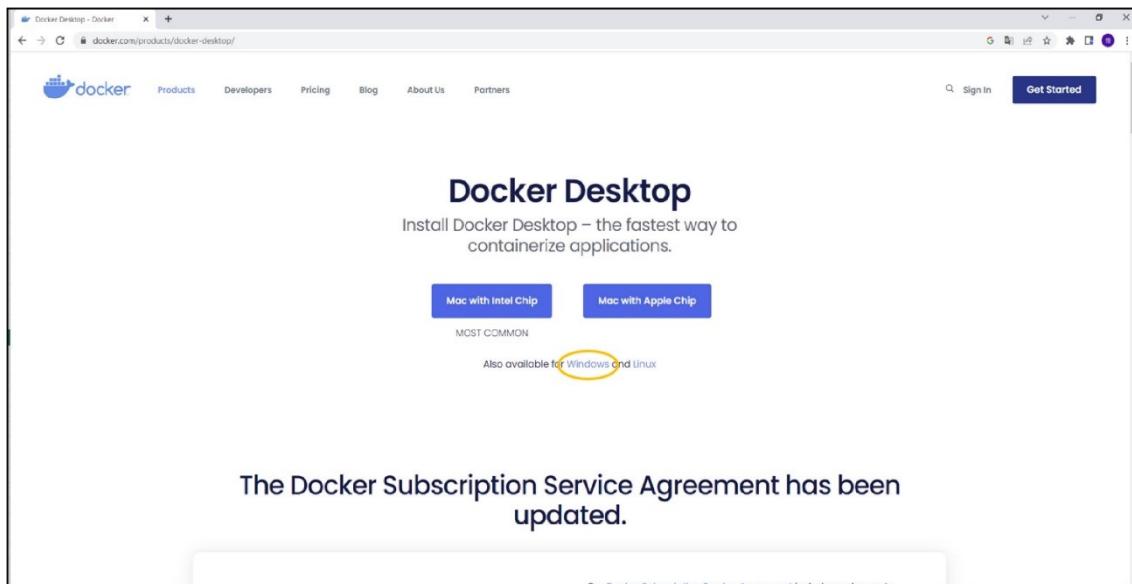
If you are not subject to a paid Docker Desktop license or have already purchased a license, please follow the steps below to install Docker Desktop.

9.2.1. Windows

9.2.1.1. Download Docker Desktop from the website below.

Clicking the Windows link on the site will automatically start downloading the installer.

URL: <https://www.docker.com/products/docker-desktop/>



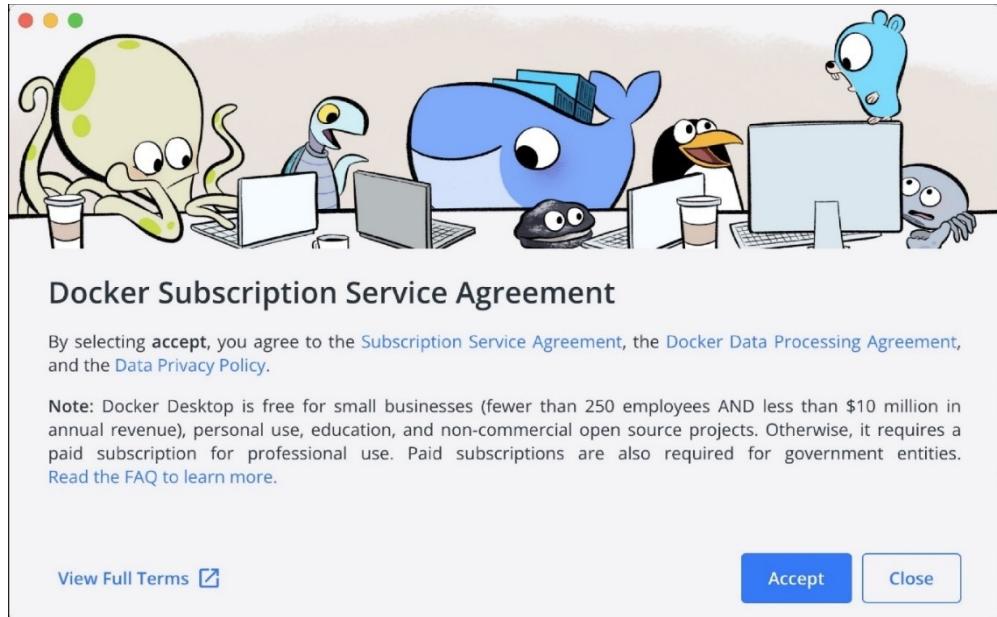
9.2.1.2. Double-click the downloaded installer and follow the on-screen instructions to complete the installation.

If you encounter an error message stating "WSL 2 installation is incomplete" during installation, please copy and paste the following URL into your web browser's address bar, download the file, and run it.

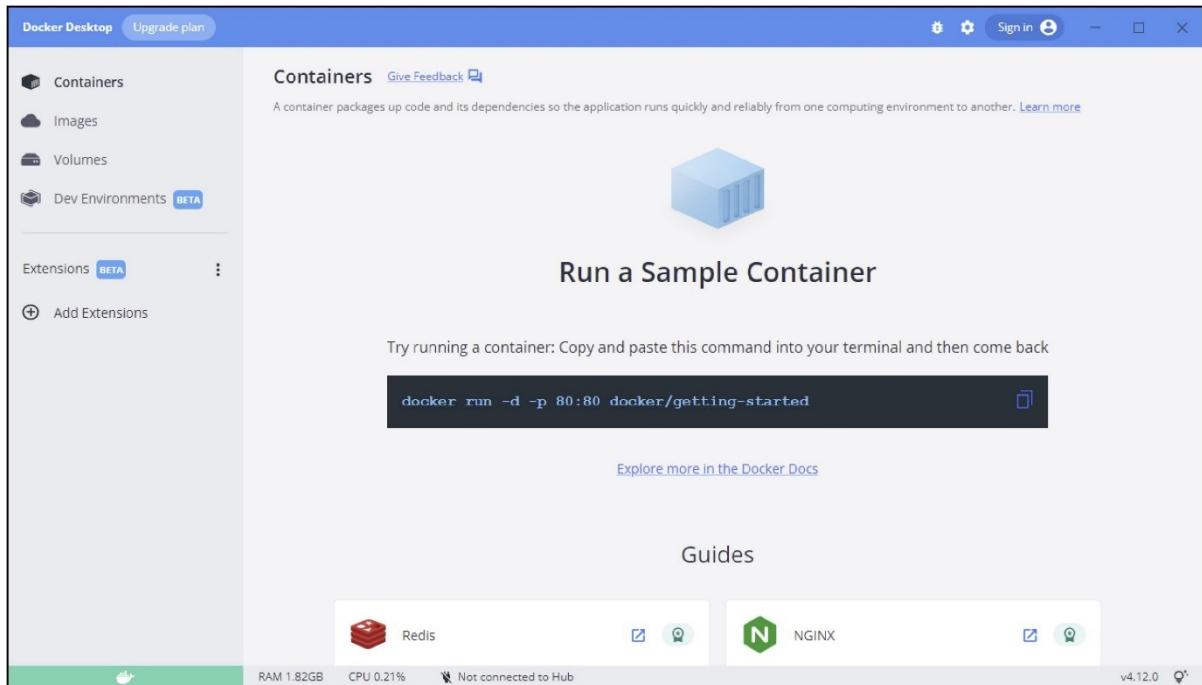
URL:

https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi

9.2.1.3. Once the installation is complete, the following screen will appear. Please review the Docker service terms and click to accept them.



9.2.1.4. Launch the installed Docker Desktop in this state. When the screen shown below appears, the installation of Docker Desktop is complete.



9.2.2. Mac

- 9.2.2.1. Click the Apple logo at the top of the desktop screen, then select “About This Mac” to check whether your chip is Intel or Apple (such as M1, M2, etc.).

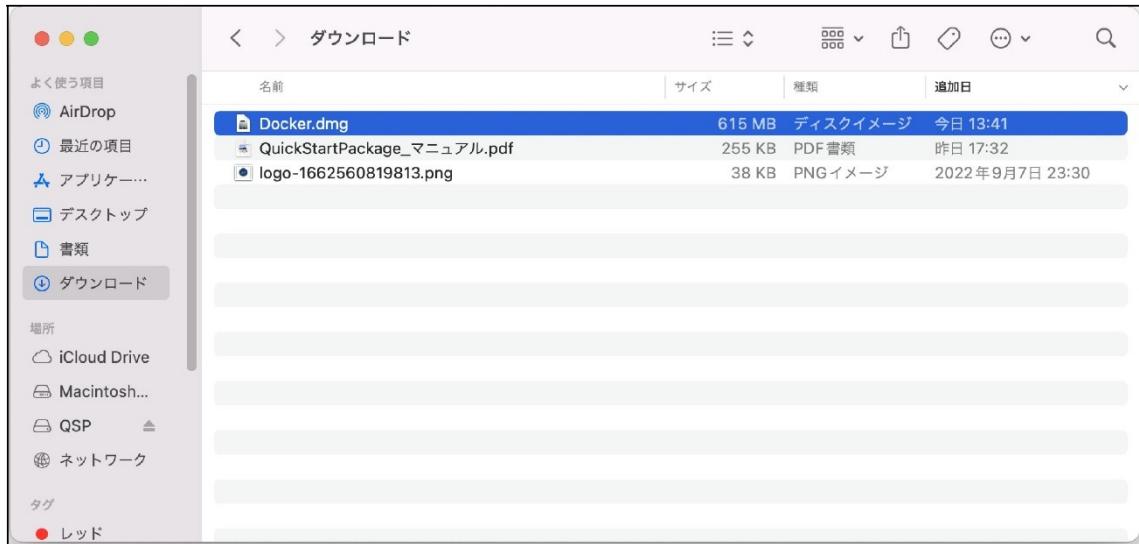


- 9.2.2.2. Download the Docker Desktop for the chip you checked from the website below.

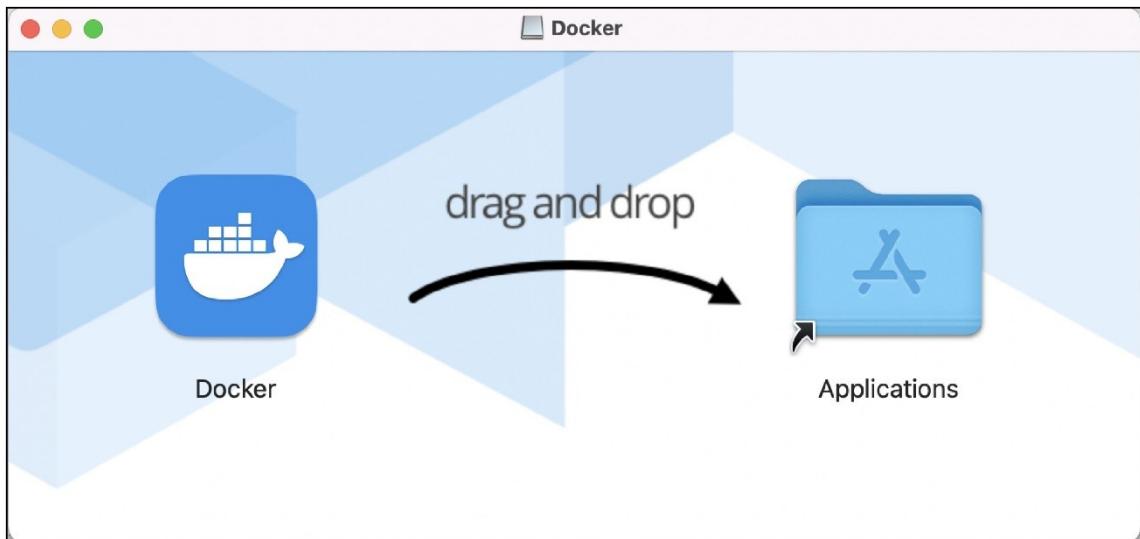
URL:<https://www.docker.com/products/docker-desktop/>

The Docker Desktop landing page. The header includes the Docker logo, navigation links for Products, Developers, Pricing, Blog, About Us, and Partners, a search bar, and a 'Get Started' button. The main section features the heading 'Docker Desktop' and the subtext 'Install Docker Desktop – the fastest way to containerize applications.'. Below this are two blue buttons: 'Mac with Intel Chip' (highlighted in blue) and 'Mac with Apple Chip'. A note at the bottom states 'Also available for Windows and Linux'.

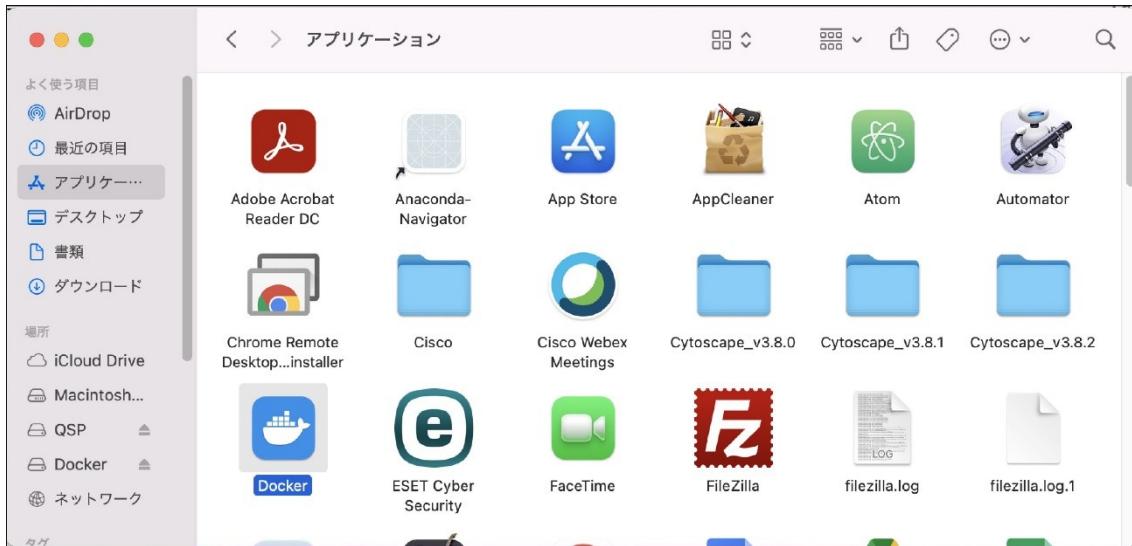
9.2.2.3. Double-click the downloaded installer and follow the on-screen instructions to complete the installation.



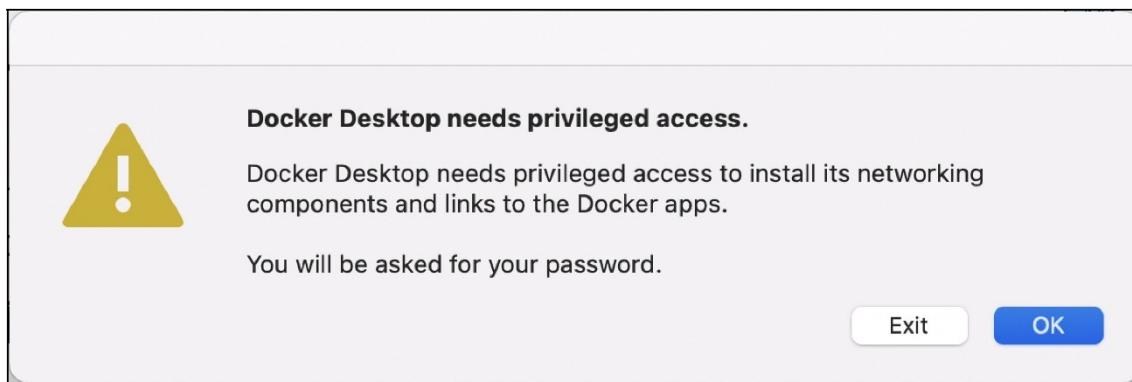
For Mac, drag and drop the Docker icon into the Applications folder.



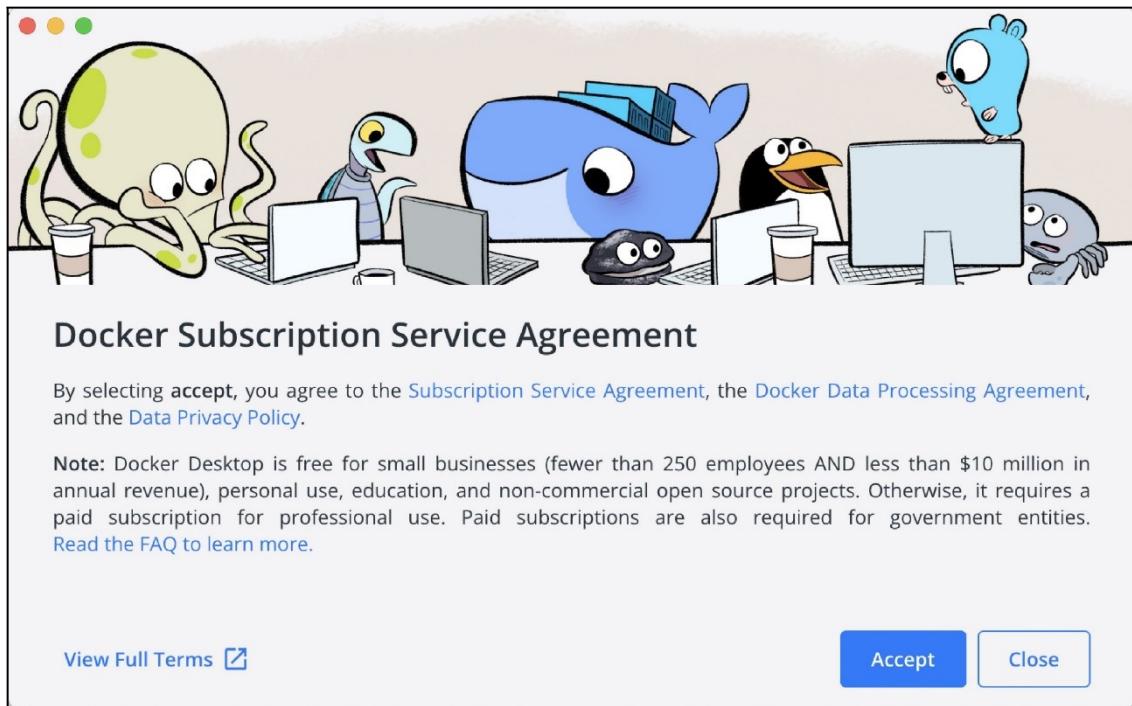
9.2.2.4. Open the Applications folder in Finder and double-click the Docker icon.



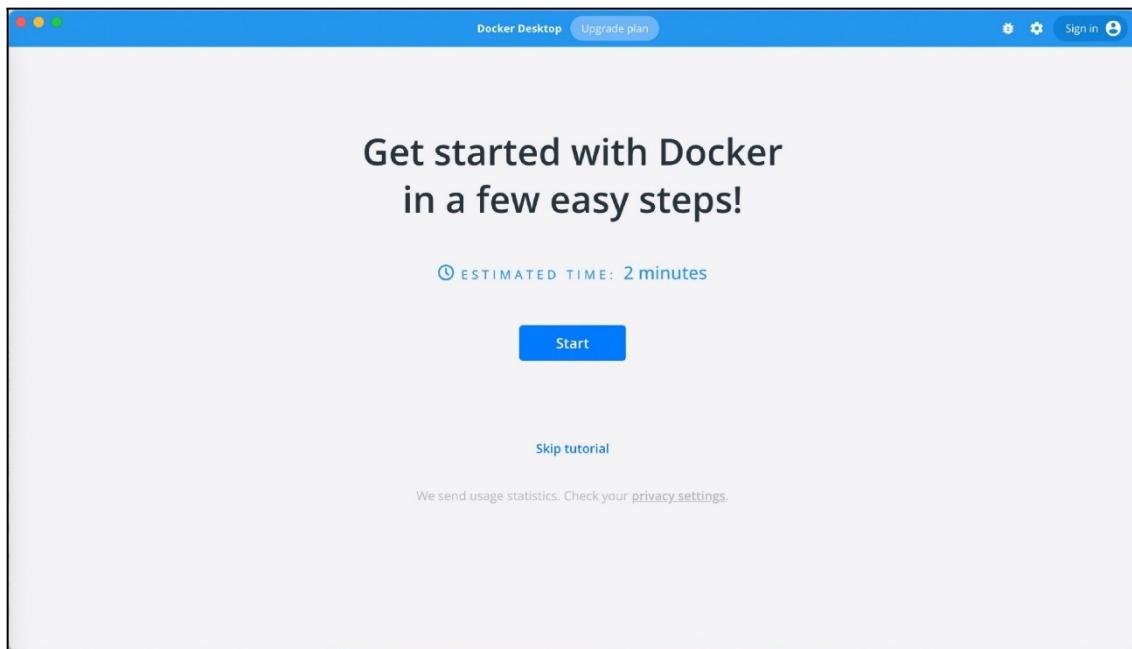
If a message appears stating that privileged access is required, click OK to grant the necessary permissions.



9.2.2.5. Please review the Docker service terms and accept them.



9.2.2.6. When the screen shown below appears, the installation of Docker Desktop is complete.



9.3. How to Uninstall Docker Desktop

This section explains the steps to uninstall Docker Desktop.

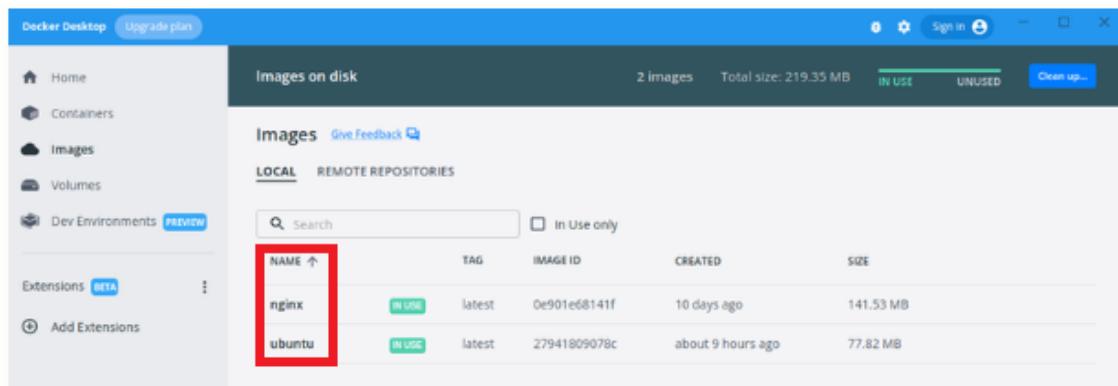
9.3.1. Backup of Existing Containers and Images

If you create a backup of your existing Docker containers and images, you can migrate them to Rancher Desktop or Docker Engine. If you have containers or images to use after migration, please follow the backup steps in this section.

Tips: For information about the relationship between Docker containers and images, please refer to the "About Docker" section. If you have not installed any additional packages after starting the container, it is sufficient to back up only the Docker image named **snpcaster:[version-number]**. For instructions, please refer to the next step, 9.3.1.1.

9.3.1.1. When backing up a Docker image, first check the name of the image you want to back up.

Start Docker Desktop and select "Images" from the left-hand panel to display a list of Docker images. Please check the name of the image you want to back up in the "NAME" column.



Open a terminal and navigate to the folder where you want to save the backup using the following command:

```
cd {the path to the folder where you want to save the backup}
```

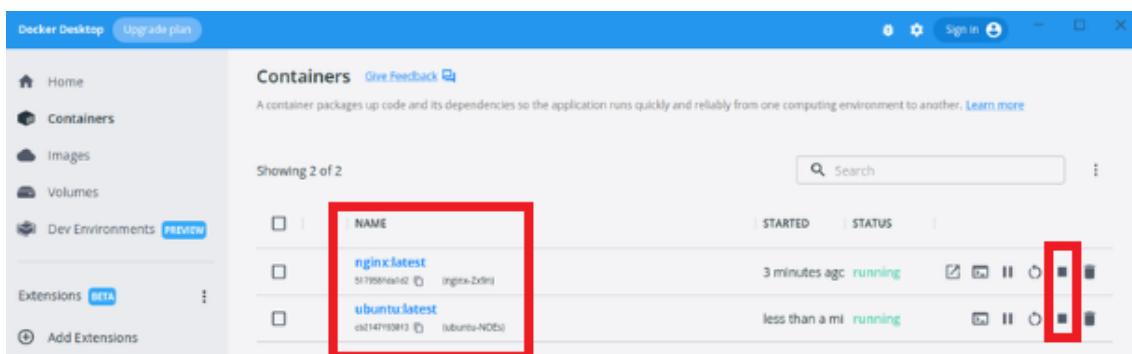
Next, run the following command to create a backup file directly under the folder you navigated to.

Note: This process may take some time.

```
docker save {image name} > {any name}.tar
```

Now, backup for one image is complete. If you have multiple images, please repeat this process for each one.

- 9.3.1.2. If you want to back up a Docker container, first stop the container you want to back up. Start Docker Desktop and select "Containers" from the left-hand panel. The "Name" column shows the container names, so check the name of the container you want to back up. Click the ■ button to stop the container when you back up it.



Next, run the following command to create an image. You can specify any name you like for {desired image name}.

```
docker commit {container name} {desired image name}
```

With this, the container has been converted into an image.

Please create the backup file following the steps in the previous section [9.3.1.1](#).

**** Caution ****

When starting a Docker container, the folders shared between the host (Windows/Mac) and the container (called volume mounts) are not backed up by this procedure.

Although the folders used for volume mounts are not deleted when you uninstall Docker Desktop, please be careful not to accidentally delete any important data manually.

Tips: For more details, please refer to the information below.

URL: <https://docs.docker.jp/desktop/backup-and-restore.html>

9.3.2. Uninstalling Docker Desktop

Once the backup is complete, please uninstall Docker Desktop by following the steps provided on the official website below.

- **Windows:** <https://docs.docker.jp/desktop/install/windows-install.html#id9>
- **Mac:** <https://docs.docker.jp/desktop/install/mac-install.html#docker-desktop>

9.3.3. Restoring the Backup

After installing Docker using one of the methods described in [Installing Docker](#), you can proceed to restore the Docker image backup you created earlier.

9.3.3.1. If you want to back up a Docker container, first stop it.

Open a terminal and navigate to the folder where the backup is saved using the following command.

```
cd {the path to the folder where you want to save the backup}
```

Next, run the following command to restore the backed-up Docker image.

Note: This process may take some time.

```
docker load -i {backup file name}.tar
```

Once the process is complete, you can verify the restored image with the following command.

```
docker image list
```

9.4. How to Uninstall Rancher Desktop

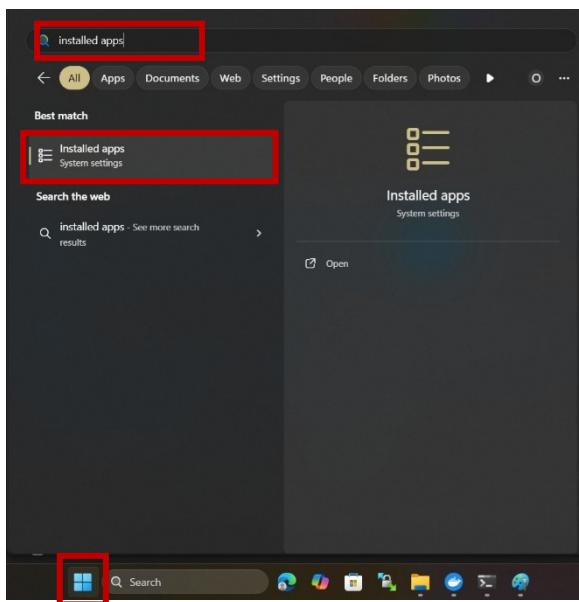
If you want to uninstall Rancher Desktop, please follow the steps below.

9.4.1. Windows

To uninstall on Windows, please follow the steps below.

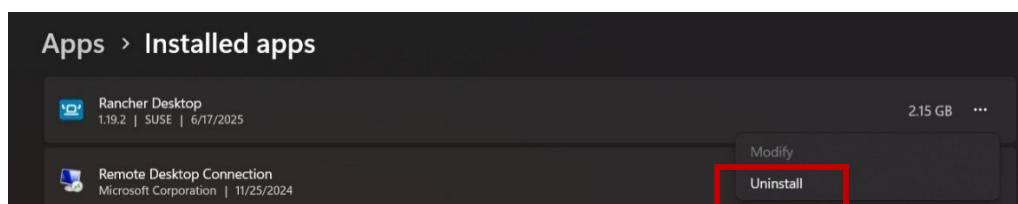
- 9.4.1.1. Click the Windows button and type "Installed apps" into the search box.

The list of installed apps will appear, so click on it.

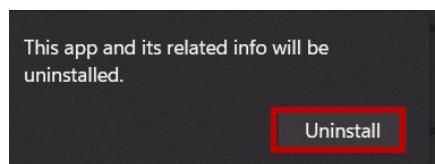


- 9.4.1.2. Type "Rancher" into the search box at the top of the screen.

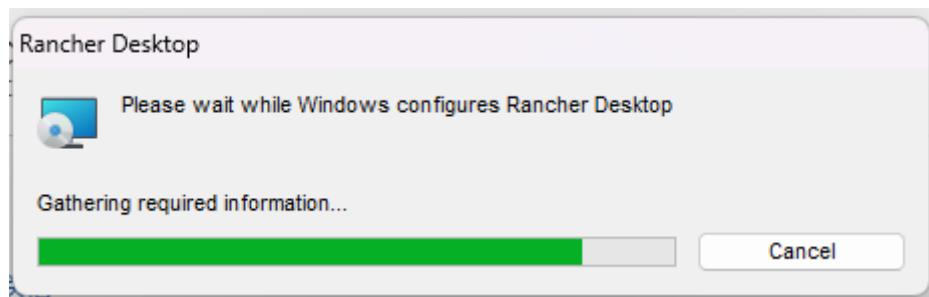
Rancher Desktop will appear. Click the ... button on the right side, then select **Uninstall**.



A confirmation prompt will appear, so click **Uninstall**.



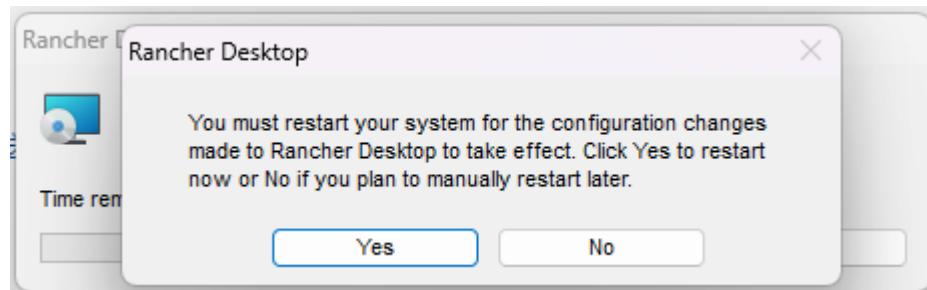
The uninstallation process will begin.



During the process, a prompt saying "Do you want to allow this app to make changes to your device?" will appear. Select **Yes**.

When the uninstallation is complete, a dialog will appear asking you to restart your computer.

If you want to restart immediately, select **Yes**; if you prefer to restart later by yourself, select **No**.



After the restart, the Rancher Desktop application will be completely removed.

However, the distribution that Rancher Desktop automatically creates on WSL during installation will remain, so you need to delete it in the next step.

- 9.4.1.3. Open the terminal and check the distributions created by Rancher Desktop using the following command:

```
wsl.exe -l
```

If **rancher-desktop** and **rancher-desktop-data** remain, please proceed to delete them in the next step.

```
PS C:\Users\XXXXXX> wsl.exe -l
Linux 用 Windows サブシステム ディストリビューション：
Ubuntu (既定)
rancher-desktop
rancher-desktop-data
```

- 9.4.1.4. In the terminal, enter the following command to delete the distributions created by Rancher Desktop:

```
wsl --unregister rancher-desktop-data
wsl --unregister rancher-desktop
wsl.exe -l
```

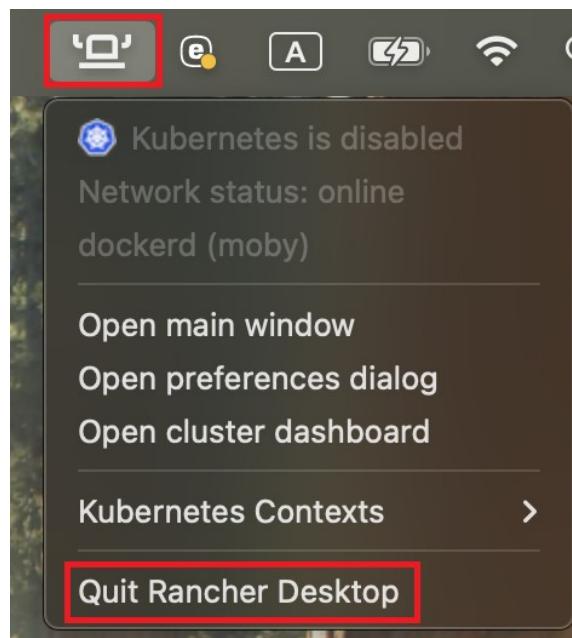
If rancher-desktop and rancher-desktop-data no longer appear after running the above command, the deletion has been completed successfully.

```
PS C:\Users\XXXXXX> wsl --unregister rancher-desktop-data
登録解除。
この操作を正しく終了しました。
PS C:\Users\XXXXXX> wsl --unregister rancher-desktop
登録解除。
この操作を正しく終了しました。
PS C:\Users\XXXXXX> wsl -l
Linux 用 Windows サブシステム ディストリビューション：
Ubuntu (既定)
```

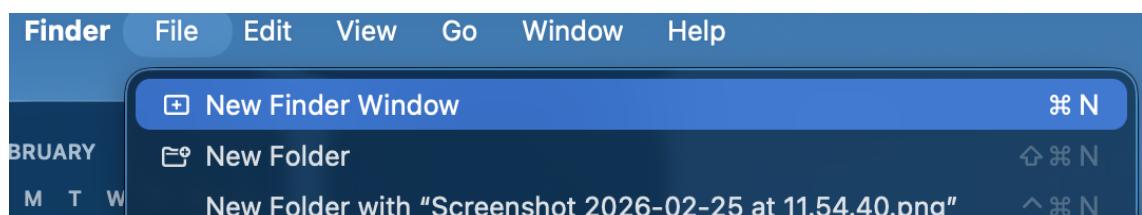
9.4.2. Mac

To uninstall on a Mac, please follow the steps below.

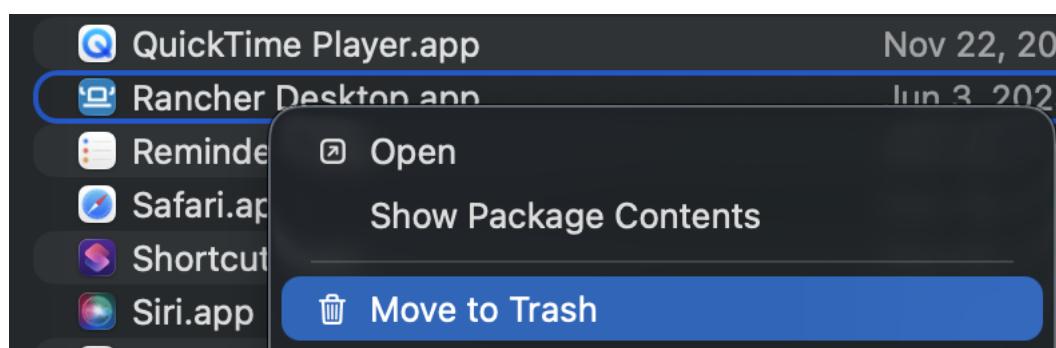
- 9.4.2.1. If Rancher Desktop is running, click the icon at the top right of the screen and select Quit Rancher Desktop.



- 9.4.2.2. Open a new Finder window.



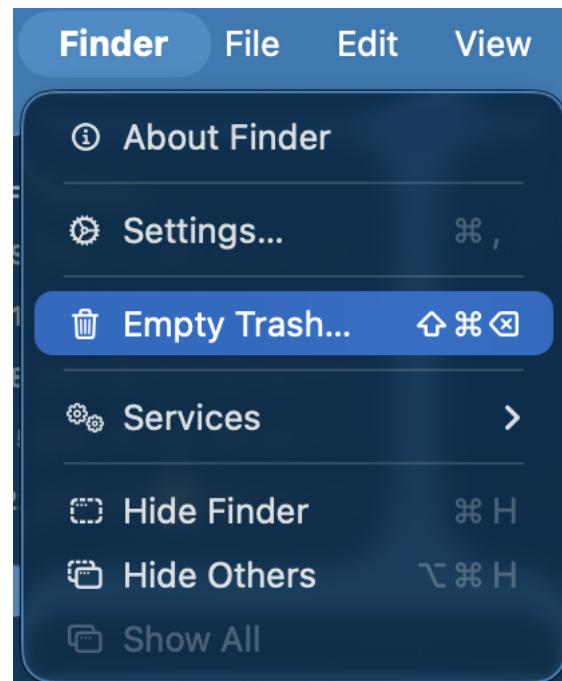
Find Rancher Desktop.app in the Applications folder and right-click it. Then, click Move to Trash.



- 9.4.2.3. If you are not an administrator, enter the administrator account credentials and click OK to move the app to the Trash.



- 9.4.2.4. To permanently delete the app, click Empty Trash from Finder.



This completes the uninstallation process.

10. Authors and Citation

10.1. Authors

Kenichi Lee

Department of Bacteriology I, National Institute of Infectious Diseases.

Toshiaki Yamagishi

Department of Bacteriology I, National Institute of Infectious Diseases.

10.2. Citation

We plan to release this program on GitHub and in academic papers in the future. When describing methods using this program in your papers, please refer to either the GitHub page (<https://github.com/leech-rr/SNPcaster>) or the following publications.

10.1. Lee K, Iguchi A, Uda K, Matsumura S, Miyairi I, Ishikura K, Ohnishi M, Seto J, Ishikawa K, Konishi N, Obata H, Furukawa I, Nagaoka H, Morinushi H, Hama N, Nomoto R, Nakajima H, Kariya H, Hamasaki M, Iyoda S. 2021. Whole-genome sequencing of Shiga toxin-producing *Escherichia coli* OX18 from a fatal hemolytic uremic syndrome case. *Emerg Infect Dis* 27:1509-1512.

10.2. Lee K., Iguchi A., Terano C., Hataya H., Isobe J., Seto K., Ishijima N., Akeda Y., Ohnishi M., and Iyoda S. Combined usage of serodiagnosis and O antigen typing to isolate Shiga toxin-producing *Escherichia coli* O76:H7 from a hemolytic uremic syndrome case and genomic insights from the isolate. 2024. *Microbiology spectrum* 12:e0235523.