

Linux Installation Guide

This installation guide was performed on the Ubuntu VM provided for your use. Other Ubuntu based variants may require some slight modifications to certain steps (as some base dependencies might differ between variants), so keep that in mind. Unfortunately, it is impossible for us to make a guide for every variant out there. Also, some installations steps might have issues depending on your current system setup and configuration. This installation guide was done on a fresh install of Ubuntu 22.04 LTS. To resolve any issues that may arise, Google is your best friend, so look online and you should be able to find a solution that works for you.

A point of note: Ideally, for you to install CaRAS you need to have administrator access to install the required files (so this is if it's your own personal laptop, this is the ideal situation). It is possible to install in shared computing clusters but there are usually restrictions as to what can be installed, so the installation will be trickier unfortunately. You may need to get assistance from the maintaining IT team to help you set it up if in this situation. We have done our best to ensure that all can be installed without outside help, but depending how computer systems are setup, we cannot guarantee this. As a point of note, this tutorial doesn't require administrator access to install anything, but your computer setup might vary. So, attempt to install in your base directory first without administrator help, but if you get stuck with administrator issues, then you will need to contact your institutes IT team for them to help you install the required dependencies.

A reminder again about the installation requirements. In terms of storage space, the installation of CaRAS and all its required generated genome files etc... will take up ~50-60Gb alone – this is NOT including any of your raw and analysis files. So, make sure you set up this on a drive/computer with enough storage space to accommodate these heavy requirements. Off-loading some of the installation directories to an external drive for example is possible (using hard/symbolic links) but is an advanced topic and is beyond the scope of this tutorial.

Prerequisite Installation

1. Before installing CaRAS, you need to make sure that Anaconda 3 is installed. There are thorough guides online but we will cover the basic steps here.
 - 1.1. Firstly, in your web-browser of choice, search for “anaconda3 individual install”

Welcome to Firefox

anaconda3 individual install

anaconda3 individual install — Search with Google

anaconda3 individual install - Google Search

Google

anaconda3 individual install

All Videos Images Shopping News More Settings Tools

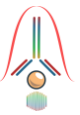
About 39,100 results (0.40 seconds)

www.anaconda.com > products > individual

Individual Edition - Anaconda

With the `conda-install` command, you can start using thousands of open source tools, Python and many other packages. Individual Edition is an open source, flexible solution that provides the utilities to build, distribute, install, update, and manage software in a cross-platform manner.

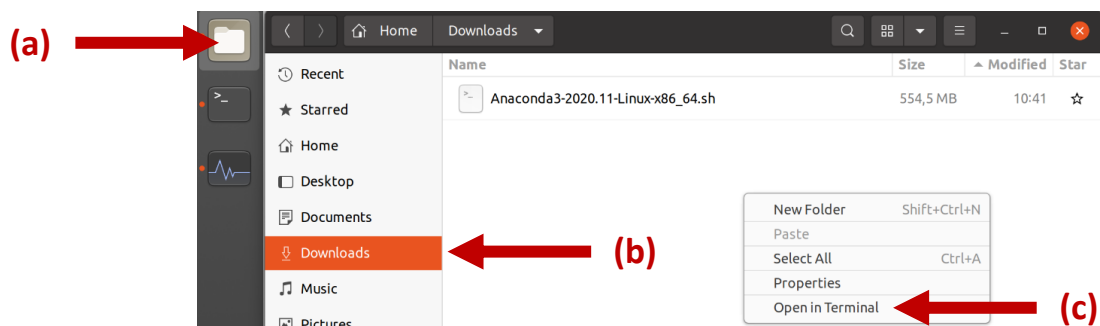
[Open Source](#) · [Help & Training](#) · [Anaconda Fan Testimonials](#) · [Partners](#)



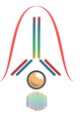
- 1.2. Download the right installer for your system (should be a 64-bit Linux installation) and save it in your Downloads folder (a) → (b) → (c)



- 1.3. Navigate to the folder where the installer “Anaconda-[year]-[OS].sh” file is downloaded, in this example, it was downloaded to the “Downloads” folder.
So, open up the file explorer (a) by clicking the right icon. Navigate to “Downloads” (b), and right mouse click on the white space of the folder. Next select “Open in Terminal” (c).



- 1.4. The following terminal window should now appear:



```
js@FRIDAY:~/Downloads$
```

1.5. Next, type the following command:

```
bash Anaconda3-2020.11-Linux-x86_64.sh
```

Where the filename entered after “bash” is the exact same filename as the 1.3 downloaded from anaconda. For ease of writing, you can write “bash Ana” and then press the TAB key on your keyboard and the terminal should auto complete the rest of the filename for you. Once done, press ENTER.

1.6. Press ENTER to view the license agreement which you must agree to for installation to proceed. To progress through each page of the agreement press the SPACE bar - read it, if you want.

```
js@FRIDAY:~/Downloads$ bash Anaconda3-2020.11-Linux-x86_64.sh
Welcome to Anaconda3 2020.11
In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

Once you get to the end, you must type “yes” in full and press ENTER.

```
Last updated September 28, 2020

Do you accept the license terms? [yes|no]
[no] >>>
```

1.7. Next, you will be asked where to install anaconda3. Unless you are an advanced user and know how to configure anaconda properly, accept defaults and press ENTER again.

```
Do you accept the license terms? [yes|no]
[no] >>> yes

Anaconda3 will now be installed into this location:
/home/chip-ap_user/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/chip-ap_user/anaconda3] >>>
```

Anaconda will then go ahead and install itself. This will take a few minutes.

1.8. When complete, you will be asked to initialize Anaconda3, type “yes” and press ENTER.

```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>>
```

1.9. Great! Anaconda3 is now installed. Now you **must** close and re-start the terminal before the next steps.



Setting up CaRAS

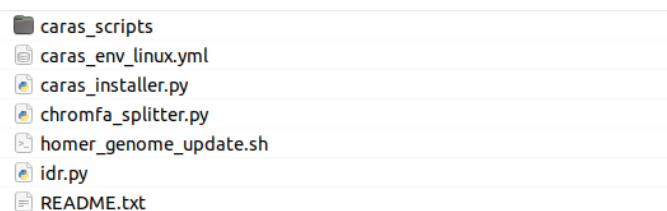
2. Now down to the “meat” of the installation!

- 2.1. Download the latest release of CaRAS (go to release and download the latest version) from our GitHub (<https://github.com/JSuryatenggara/CaRAS>) and place it in its own folder in the location of your choosing. Our recommendation is as follows - in your home directory, create a new folder named “tools” and create a sub-folder named “caras”.

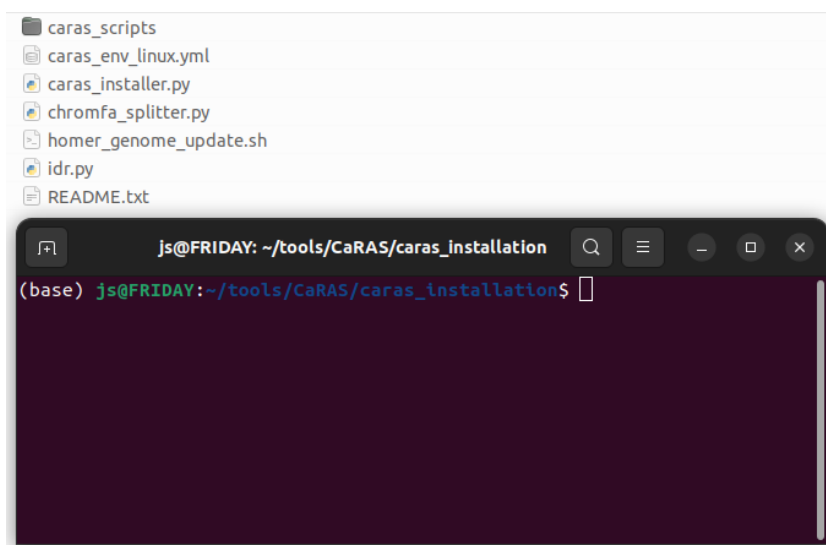
Also, while you’re at it, download the pre-configured genome folders required for CaRAS from (<https://www.dropbox.com/scl/fi/ozrsp9se9t11c1m6t59gv/genomes.zip>). Unzip the file and folders and put them in a “genomes” folder. They are required for running CaRAS successfully.



- 2.2. Make sure you unzip everything and your folder structure looks similar to the following:

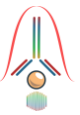


- 2.3. Next, right mouse-click on the white-space of the folder, and pull up the context menu, and click on “Open in Terminal” to bring up a new terminal window:



- 2.4. Now to install CaRAS and its dependencies, from within the extracted folder, type these following commands and press ENTER:

```
chmod +x ./caras_installer.py
```



```
(base) js@FRIDAY:~/tools/CaRAS/caras_installation$ chmod +x ./caras_installer.py ☐  
  
./caras_installer.py
```

```
(base) js@FRIDAY:~/tools/CaRAS/caras_installation$ ./caras_installer.py ☐
```

2.5. As part of the installation process, CaRAS will ask a couple of questions.

The first question asked is whether you want to install CaRAS in its own environment. This is so CaRAS will sit in its own contained “capsule” and won’t be affected by other installed programs and so this will ensure that all the dependencies of CaRAS will be correct and met all the time. So, we highly recommend this option.

```
(base) js@FRIDAY:~/tools/CaRAS/caras_installation$ ./caras_installer.py  
Anaconda directory is /home/js/anaconda3  
Do you want to install CaRAS in a specific environment? (Y/N) ☐
```

If you answer “N”, then CaRAS will be installed in the base environment along with its dependencies.

```
(base) js@FRIDAY:~/tools/CaRAS/caras_installation$ ./caras_installer.py  
Anaconda directory is /home/js/anaconda3  
Do you want to install CaRAS in a specific environment? (Y/N) N  
CaRAS will be installed in base environment. Press ENTER to continue ☐
```

If you answer “Y”, you will be asked to name the environment and then press ENTER. A confirmation statement will be printed for you to confirm with an additional ENTER press.

```
(base) js@FRIDAY:~/tools/CaRAS/caras_installation$ ./caras_installer.py  
Anaconda directory is /home/js/anaconda3  
Do you want to install CaRAS in a specific environment? (Y/N) Y  
Please type in the name of your CaRAS environment: ☐
```

Take note of the environment name you set (we will refer to it in step 2.6 as xxxxx).

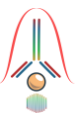
```
(base) js@FRIDAY:~/tools/CaRAS/caras_installation$ ./caras_installer.py  
Anaconda directory is /home/js/anaconda3  
Do you want to install CaRAS in a specific environment? (Y/N) Y  
Please type in the name of your CaRAS environment: caras  
CaRAS will be installed in caras environment. Press ENTER to continue ☐
```

The installation will take a while to run as a lot of packages will need to be installed, so this is highly dependent on your internet and CPU speeds as to how long it will take.

2.6. Once the installation completes, open up the terminal. If you setup CaRAS in an environment in step 2.5, you will need to type the following command (below) where xxxxx is the name of the environment as you defined in step 2.5 before running CaRAS EVERYTIME. Otherwise, this command isn’t necessary.

```
conda activate xxxxx
```

```
(base) js@FRIDAY:~$ conda activate caras ☐
```



If the anaconda environment you assigned in step 2.5 is successfully installed, the terminal will now enter the said environment (shown in the terminal screen below). Any command you run in this terminal screen will make use of everything installed within this environment.

```
(base) js@FRIDAY:~$ conda activate caras
(caras) js@FRIDAY:~$
```

- 2.7. Perform a test to see if CaRAS is properly installed by only typing the script name and press ENTER. If you see a display like below, then CaRAS is ready to go (disclaimer: this does not confirm that every single constituent programs involved in CaRAS run is properly installed. Such test is performed when CaRAS normal run with all the required arguments provided).

```
(caras) js@FRIDAY:~$ caras.py
Importing required modules
Defining functions
Setting argument parser
usage: caras.py [-h] --analysis {bulk,single_cell} --mode {single,paired}
                --genome GENOME --output OUTPUT --setname SETNAME
                [--chipR1 CHIPR1 [CHIPR1 ...]] [--chipR2 CHIPR2 [CHIPR2 ...]]
                [--ctrlR1 CTRLR1 [CTRLR1 ...]] [--ctrlR2 CTRLR2 [CTRLR2 ...]]
                [--sample_table SAMPLE_TABLE]
                [--custom_setting_table CUSTOM_SETTING_TABLE] [--fcmerge]
                [--clustnum CLUSTNUM] [--motif MOTIF]
                [--ref {hg19,hg38,mm9,mm10,mm39,dm6,sacCer3}]
                [--norm_ref {none,sacCer3,eColiK12}] [--goann] [--pathann]
                [--homer_motif {consensus,union,1,2,3,4,5,6} [{consensus,union,1,2,3,4,5,6} ...]]
                [--meme_motif {consensus,union,1,2,3,4,5,6} [{consensus,union,1,2,3,4,5,6} ...]]
                [--thread [1-32]] [--deltemp] [--run]
caras.py: error: the following arguments are required: --analysis, --mode, --genome, --output, --setname
(caras) js@FRIDAY:~$
```

- 2.8. To use CaRAS using the command line, please refer to the guides (included with CaRAS download from GitHub) for full details on how to setup a run with appropriate flags/parameters (for proficient users).
- 2.9. Once the run is complete, please refer to the guides (included with CaRAS download from GitHub) for the tutorial on navigating and interpreting the results.

Subsequent Running of CaRAS

Ok so now that everything is set up to run CaRAS each time is relatively simple. Open a new terminal window.

If you setup CaRAS in its own environment then you need to write the following command first:

```
conda activate xxxxx
```

```
(base) js@FRIDAY:~$ conda activate caras
```

```
(base) js@FRIDAY:~$ conda activate caras
(caras) js@FRIDAY:~$
```

where xxxxx is the name of the environment you created in the installation.

The next command you need to write at the command line will start caras depending on what mode you want:

```
caras.py
```

```
(caras) js@FRIDAY:~$ caras.py
```