

Virtual Machine Setup and Usage Guide v1

Sometimes, individuals don't want to corrupt or change their pre-configured systems with new software. For such scenarios, setting up a virtual machine (VM) as a test environment is perfect as it allows the user to setup an environment completely independent of their main operating setup. The down-side to such a setup however, is the additional hardware requirements to run a 2nd VM on top of an existing system. This arrangement is also ideal on a shared lab computer where users can run their ChIP-Seq analysis in the VM and then deactivate it when done, such that only a single high-powered computer is required per lab, rather than using shared resources or providing high-powered computer hardware to all users.

In our other tutorials we have shown users how to setup ChIP-AP in self-contained conda environments (ideally on their own individual laptops/computers) which shouldn't disrupt the user's typical workflow or other configurations. Despite this, some users would still prefer a VM to run ChIP-AP thereby completely separating it from existing setups. In this tutorial, we will walk you through setting up Oracle's VirtualBox (even though tutorials can be found online) and show you how to configure the VM environment to adequately run the pre-configured ChIP-AP VM. The last stage will be setting up the shared folder between the host OS and the guest OS to facilitate accessing data from within the VM that resides on your host OS – this is a critical step to get right to make this entire process worth it.

VirtualBox is only 1 virtualization platform available. There's VMWare, Qemu, Hyper-V or Parallels to name a few. All these platforms have similar setup processes however only VirtualBox will be covered in this tutorial. If using the others, Google is your best friend!

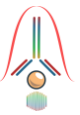
Terminology

Ok before we begin its best to clear the air about certain terminology when discussing VM's. The operating system (OS) that your machine runs natively is referred to as the "host" OS and any OS that you install in a VM is referred to as the "guest" OS. So, if I have a HP/Dell consumer laptop, it will likely be running Windows as a "host" OS and then I can setup the ChIP-AP Linux VM as a "guest" OS on it through virtualization.

System Requirements

By its nature, running a 2nd OS through virtualization will demand very capable hardware. Even through virtualization though, there is a slight performance hit for the guest OS, it will never run as fast as if it were the native host OS. This trade-off is acceptable as you can run different test VM environments for different purposes. In the case of ChIP-AP, the recommended system requirements are as follows

- **Host OS** – Linux (Ubuntu-variants 16.04+), MacOS (10.13+), Windows 10 (v1903+)
- **CPU** – (min) Octa-Core Intel/AMD CPU. Virtualization is still not supported on newer Apple Silicon CPU's
- **RAM** – (min) 16Gb
- **Storage (SSD/HDD)** – The guest OS image will max out at ~150Gb of space. For your analysis, an additional 30-100Gb of storage space is needed.
- **Screen Resolution** – A minimum resolution of 1920*1080 is required for the dashboard interface within the guest OS. If less, only the wizard will be usable.

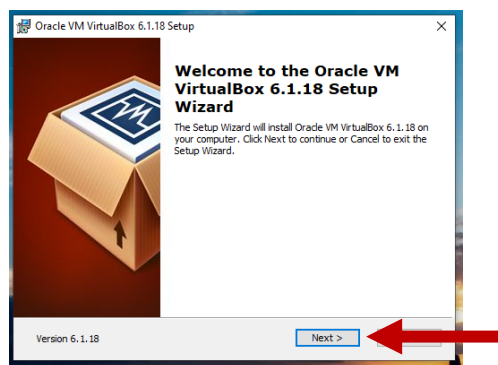


VirtualBox Installation

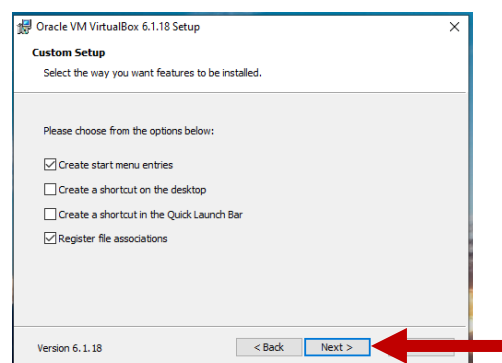
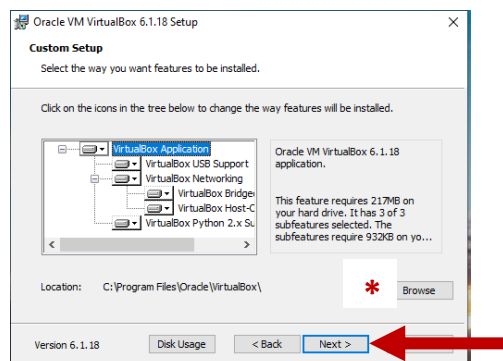
1. Oracle's VirtualBox can be downloaded from <https://www.virtualbox.org/> (a). Make sure you download the correct installer for your host OS (b).

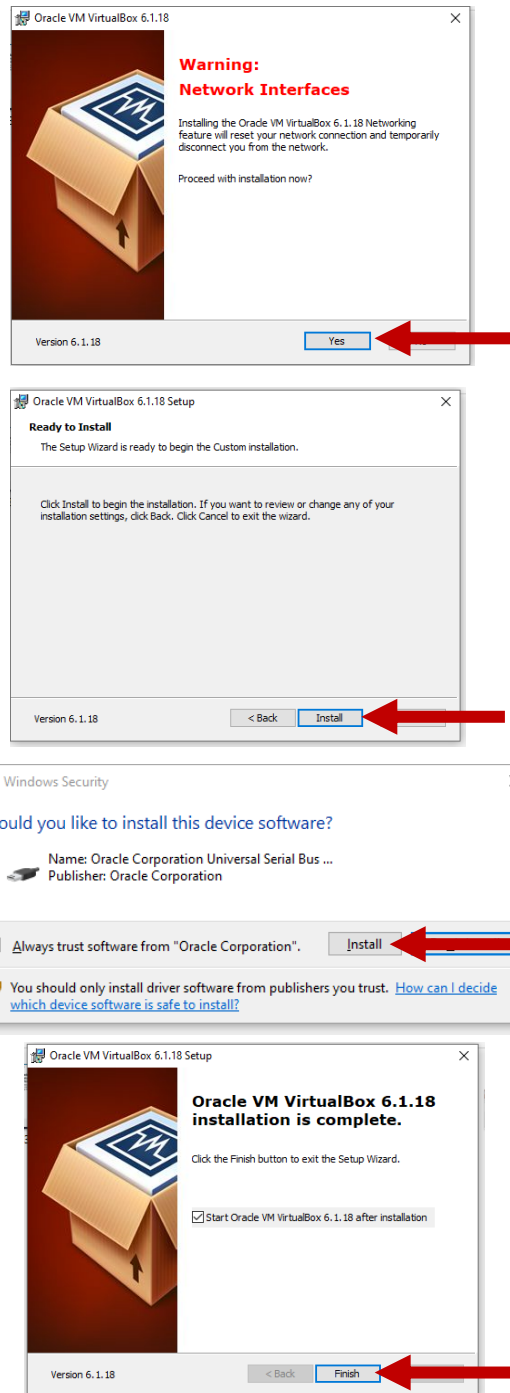
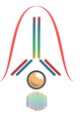


2. Once the installer is downloaded, run it.



3. Proceed through the installation accepting default unless you want to change the installation directory (*).

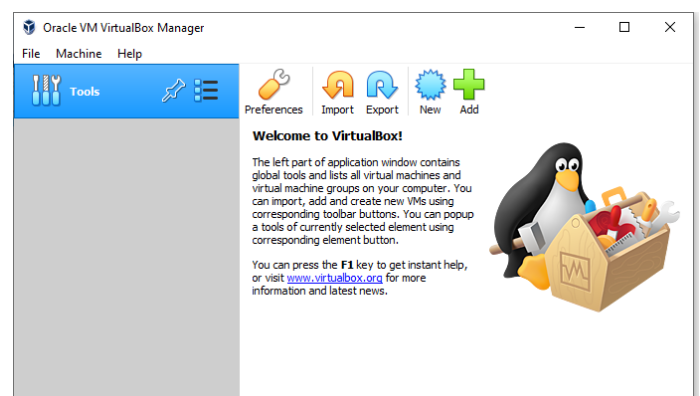


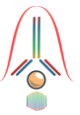


4. VirtualBox is now installed, next is the configuration to facilitate running of the guest OS.

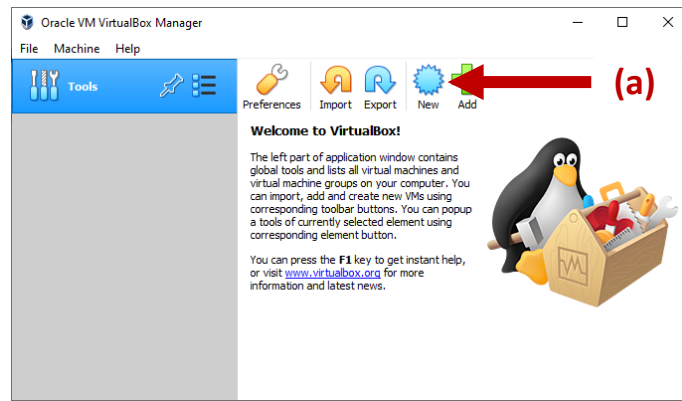
Configuring VirtualBox

1. Once you start VirtualBox, the base screen will appear. At this stage, download the pre-configured Linux guest VM image from our dropbox https://www.dropbox.com/s/5pm0lcf3gl616oy/ChIP-AP_Ubuntu_LTS2004.vhd (100Gb download)





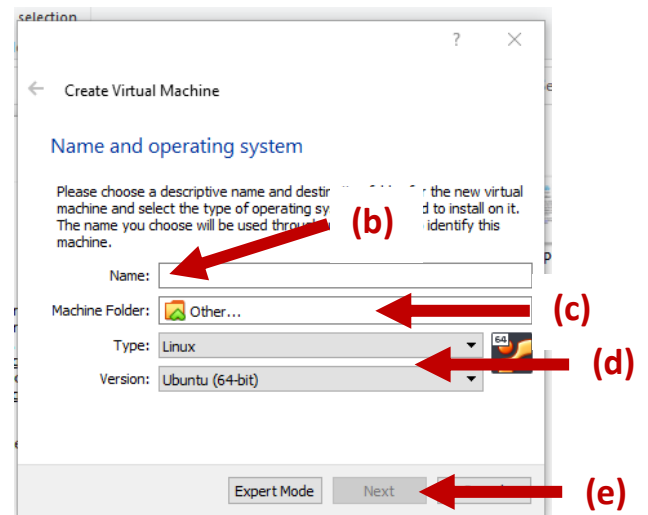
- Next, you need to make a new virtual machine container. To do this click on “New” (a).



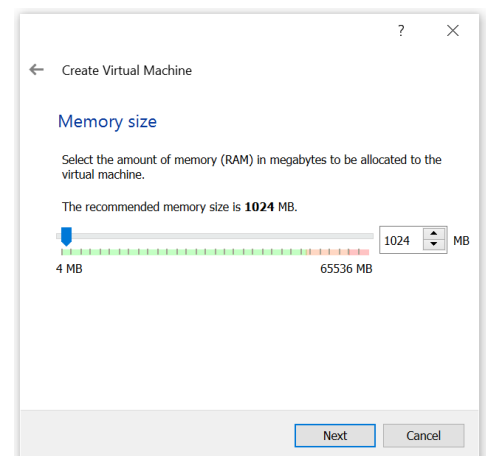
- This will bring up the new configuration menu. Name the image anything you want (b) – in our case we called it (ChIP-AP_Ubuntu_LTS2004).

For “Machine Folder,” this is where you want all the files associated with the image are to be saved (c). So make sure this is on a drive that is large enough to contain all required files (~150Gb to be safe).

For “Type” and “Version,” these need to be set to Linux and Ubuntu (64-bit) respectively (d). When all is filled in, you can click “Next” to continue (e).

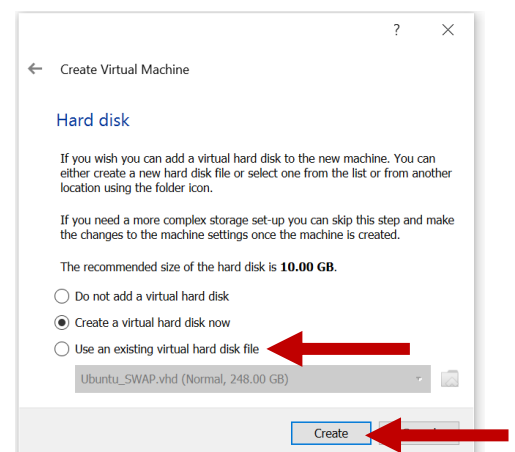


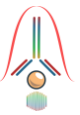
- On the next screen, you will need to set the ram requirements for the guest OS. For this we recommend as much as possible but as a minimum, 10Gb should be ok ($1024 * 10 = 10240$ MB).



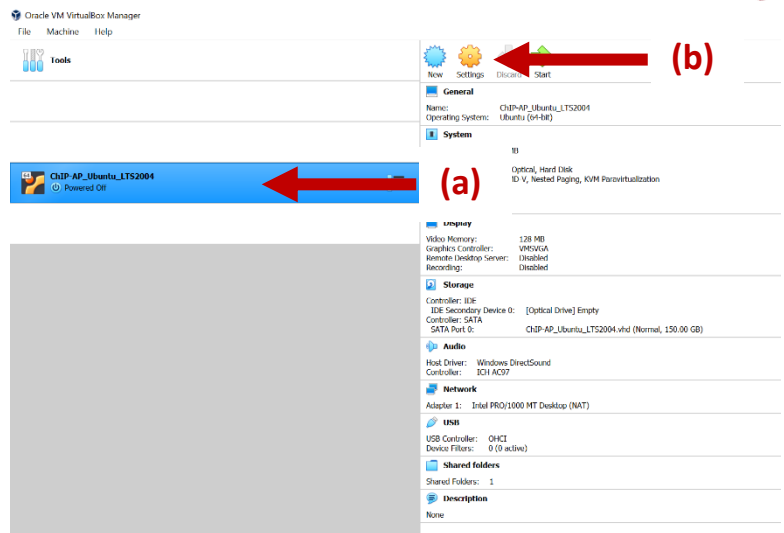
- On the next screen, you need to select what the HDD for the image will be. Select “Use existing virtual hard disk file.” This is the VM image downloaded above from our dropbox, so select that from the appropriate menu. Once selected, click on “Create”

You will be taken to the main VirtualBox screen with the new machine showing up in the far left column.

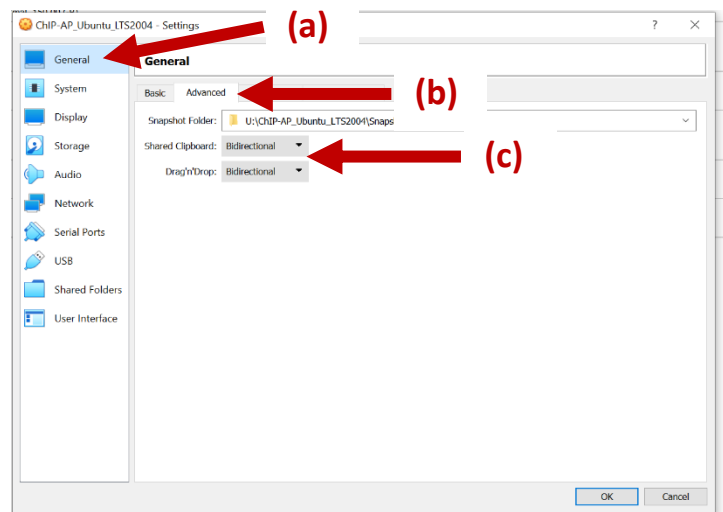




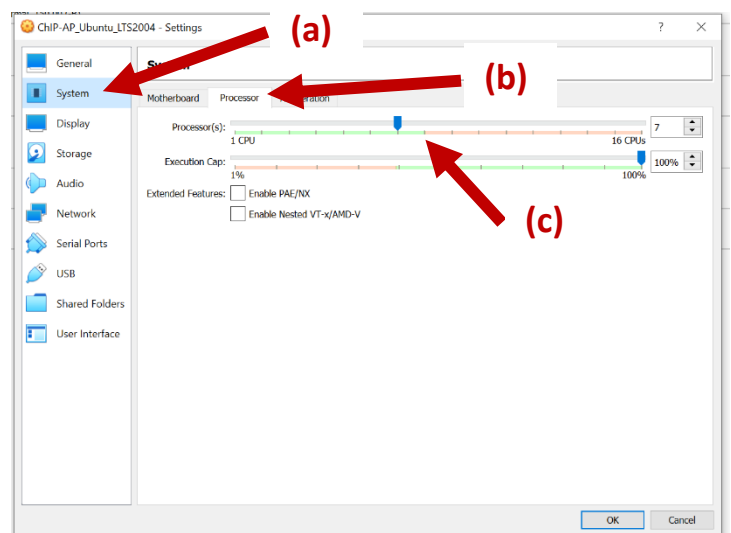
6. Next, we need to configure more regarding the available resources for the VM. So, select the newly created machine on the left (a) and then click “Settings” at the top (b).



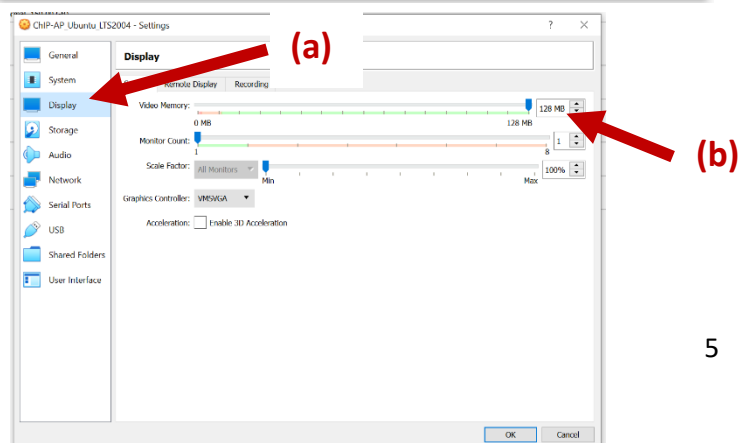
7. In the setting dialog box, click “General” (a) and then click on “Advanced” (b). Make sure that “Shared Clipboard” and “Drag’n’Drop” are both set to “Bidirectional” (c).

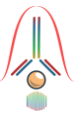


8. Under “System” (a), select the “Processor” tab (b) and then set your processor count to (recommended) half the number of processors available (c). The more cores you give the guest OS the faster you can get ChIP-AP to run, but this comes at a cost of slowing your host OS. **NEVER** set this number to all available cores in your system, the max should always be 1 less than the max available – this is to maintain system stability under load.

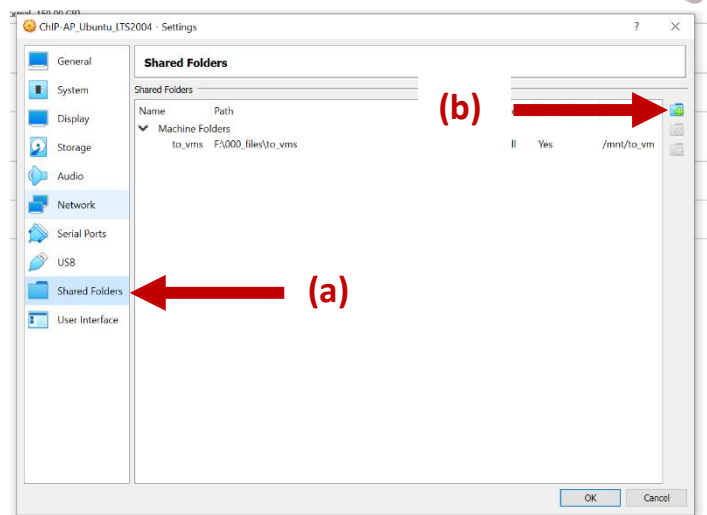


9. Under “Display” (a), set the “Video Memory” to 128 MB (b).





10. This next step is critical to get right otherwise you will not be able to share information between the guest and the host OS. On the “Shared Folder” tab (a), click the “New” icon (b).



11. On the Edit Share window that appears, the “Folder Path” selection (a) is the folder on your HOST OS that you wish to be visible to the guest OS, so select whichever folder you want. For example

`F:\000_files\to_vms`

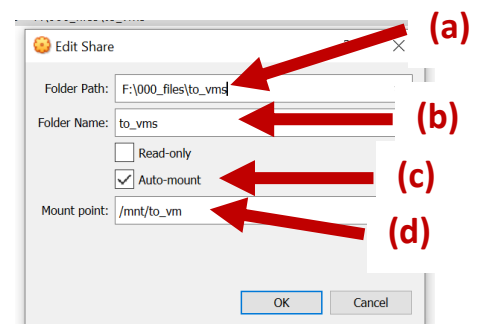
The “Folder Name” (b), needs to be identical to what is the folder name selected in (a). So, in our example

`to_vms`

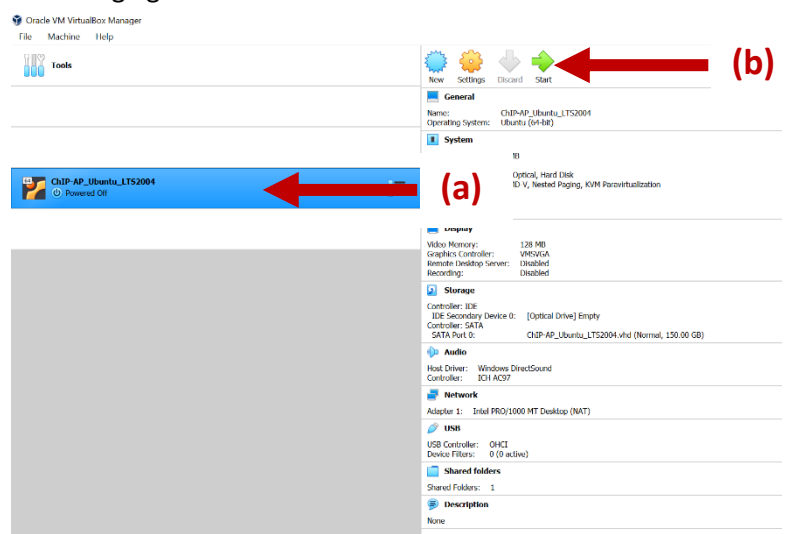
Next select “Auto-mount” (c).

Finally, the “mount point” (d) **MUST MUST MUST** be set to
`/mnt/to_vms`

Click “OK” when configured

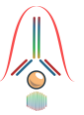


What this step has done is set a link between your host and guest OS as to which folder can be seen and accessed by both. The reason the mount-point must be set to `/mnt/to_vms`, is because this is how we have configured the CHIP-AP VM to look at for the shared folder. If you deviate from these instructions, getting the shared folder to work right can be quite tricky so don't deviate from these instructions unless you know where you're doing or are prepared to sink a couple of hours on google to get it working again!



12. Now that all the VM is configured you can start the VM.

Depending on the speed of your system, this can take a short/long time. On our setup, the VM start up takes ~20-30 seconds.

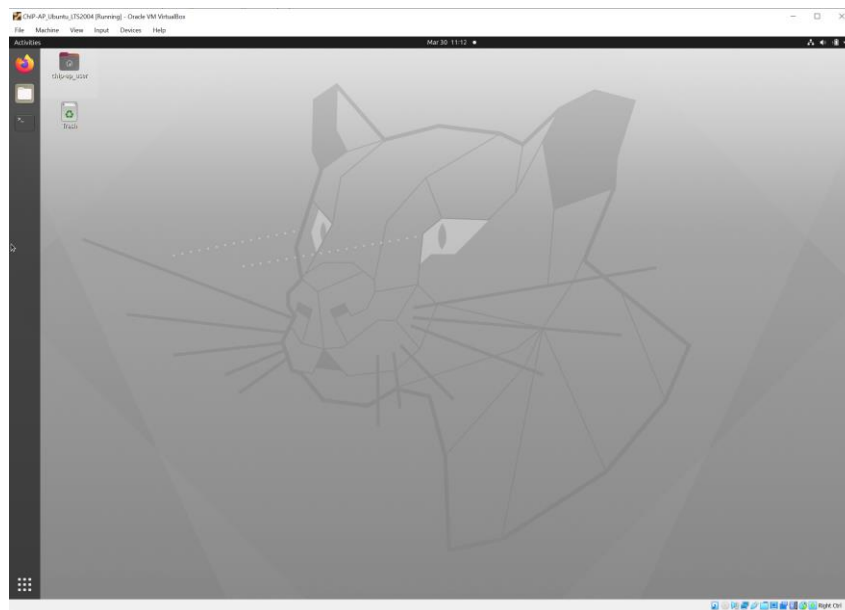


VM Usage

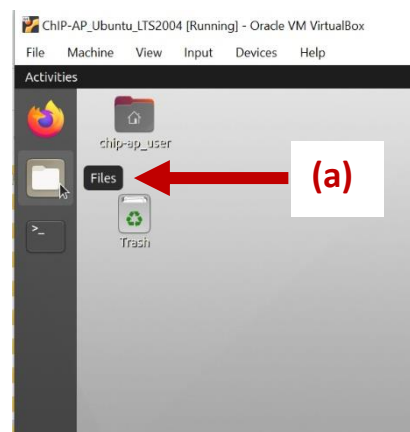
The username for the pre-configured VM is : `chip-ap_user`

The password for the pre-configured VM is : `chip-ap`

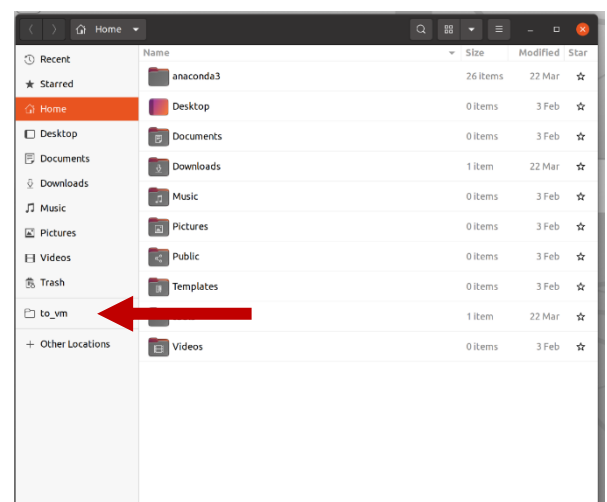
1. Once started, the ChIP-AP VM desktop will appear

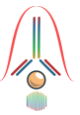


2. As a quick tour, if you click “Files” in the top left (a)

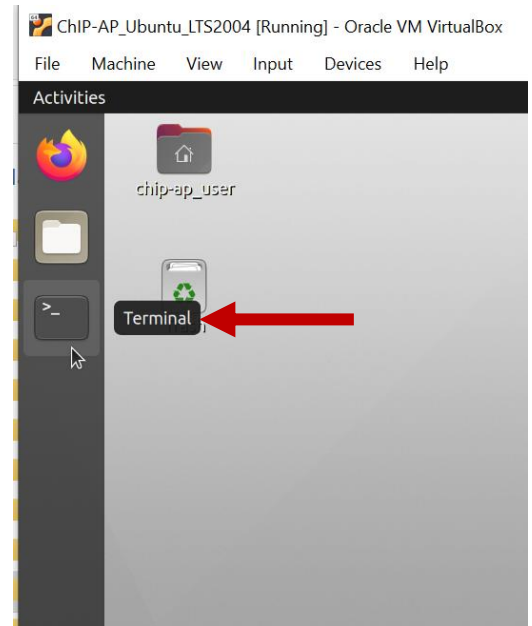


In the Nautilus window that appears (this is the equivalent of Windows Explorer or Finder), you will find the bookmarked “to_vm” folder on the left, this is where you will find your shared folder between the guest and host OS'es.





3. To run ChIP-AP, go to the “Terminal” in the top left of the desktop again.



4. In the terminal window that appears...

```
chip-ap_user@chipap-VM: ~
(base) chip-ap_user@chipap-VM:~$ conda activate python3_chipap
(python3_chipap) chip-ap_user@chipap-VM:~$ chipap_wizard.py
(python3_chipap) chip-ap_user@chipap-VM:~$ chipap_dashboard.py
(python3_chipap) chip-ap_user@chipap-VM:~$ chipap_v4.1.py
Importing required modules
Defining functions
Setting argument parser
usage: chipap_v4.1.py [-h] --mode {single,paired} [--peak {narrow,broad}]
                    --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] [--motif MOTIF]
                    [--ref {hg19,hg38,mm9,mm10,dm6,sacCer3}] [--goann]
                    [--pathann] [--thread [1-7]] [--deltemp] [--run]
chipap_v4.1.py: error: the following arguments are required: --mode, --genome, --output, --setname
(python3_chipap) chip-ap_user@chipap-VM:~$
```

Before running anything, you need to activate the configured conda environment by typing

```
conda activate python3_chipap
```

Next, depending on what you want to do, you can run ChIP-AP in 1 of 3 ways, there is the GUI Wizard

```
chipap_wizard.py
```

Or the GUI dashboard

```
chipap_dashboard.py
```

Or if you want to go all out with the command line, you can use the command option (`chipap_v4.1.py`) providing all the necessary input parameters and run it that way. For full details about all the command line parameters and functions of everything, please refer to the documentation located on our github (<https://github.com/JSuryatenggara/ChIP-AP>).

As a recommendation, we suggest you set the output save directory to the `/mnt/to_vm` folder such that you are able to access the results from the host OS easily.