

A guideline for annotating non-chromosomal DNA (phages and plasmids) of microbial community with PPR-Meta

PROTOCOL

1. Physical host version

1.1 Operating system and related requirements

PPR-Meta has been tested on Ubuntu 16.04. You would better choose the same operating system. PPR-Meta needs Python 2.7.12, numpy 1.13.1, h5py 2.6.0, TensorFlow 1.4.1, Keras 2.0.8, MATLAB R2018a (or MATLAB Component Runtime R2018a). For compatibility, we recommend installing the tools with similar versions as described above. If GPU is available on your machine, we recommend installing a GPU version of the TensorFlow to speed up the program. PPR-Meta can be run with either an executable file or a MATLAB script. If you run PPR-Meta through the executable file, you need to install the MCR while MATLAB is not necessary. If you run PPR-Meta through the MATLAB script, MATLAB is required.

1.2 Installing related requirements

The numpy, h5py, TensorFlow, and Keras are python packages, which can be installed with “pip”. If “pip” is not already installed on your machine, use the command “sudo apt-get install python-pip python-dev” to install “pip”:

```
pip install numpy
pip install h5py
pip install tensorflow==1.4.1      #CPU version
pip install tensorflow-gpu==1.4.1 #GPU version
pip install keras==2.0.8
```

If you are going to install a GPU version of the TensorFlow, specified NVIDIA software should be installed. See <https://www.tensorflow.org/install/gpu> to know whether your machine can install TensorFlow with GPU support.

When running PPR-Meta through the executable file, MCR should be installed. See

<https://www.mathworks.com/help/compiler/install-the-matlab-runtime.html> to install MCR. On the target computer, please append the following to your LD_LIBRARY_PATH environment variable according to the tips of MCR:

```
<MCR_installation_folder>/v94/runtime/glnxa64  
<MCR_installation_folder>/v94/bin/glnxa64  
<MCR_installation_folder>/v94/sys/os/glnxa64  
<MCR_installation_folder>/v94/extern/bin/glnxa64
```

A screenshot of the tips when installing MCR is shown below (Figure 1.):

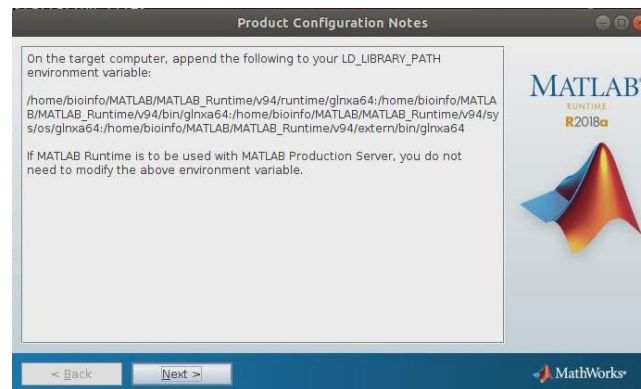


Figure 1. The installation interface of MCR.

When running PPR-Meta through the MATLAB script, you can see <https://www.mathworks.com/support/> to install the MATLAB.

1.3 Usage of PPR-Meta

PPR-Meta has a repository in Github. To run PPR-Meta, you can download PPR-Meta package using git, and then change the directory to PPR-Meta:

```
git clone https://github.com/zhenchengfang/PPR-Meta.git  
cd PPR-Meta
```

Also, you can download PPR-Meta as a zipped file, and then unpack the zipped file and change the directory to PPR-Meta:

```
wget http://cqb.pku.edu.cn/ZhuLab/PPR_Meta/PPR_Meta_v_1_0.zip  
unzip PPR_Meta_v_1_0.zip
```

```
cd PPR_Meta_v_1_0
```

1.3.1. Run by executable file (in the command line)

PPR-Meta could run by an executable file. You can just simply execute the command:

```
./PPR_Meta <input_file_folder>/input_file.fna <output_file_folder>/output_file.csv
```

The input file must be in fasta format containing the sequences to be identified. We can use the file “example.fna” which contains 300 sequences in the program folder to test the PPR-Meta by simply executing the command:

```
./PPR_Meta example.fna result.csv
```

1.3.2. Run by MATLAB script (in MATLAB GUI)

PPR-Meta could also run by MATLAB script. After setting the working path of MATLAB to the program folder before running the program, you can execute the following command directly in the MATLAB command window.

```
PPR_Meta('<input_file_folder>/input_file.fna',  
         '<output_file_folder>/output_file.csv')
```

To test the sequences in the example.fna:

```
PPR_Meta('example.fna', 'result.csv')
```

1.3.3. Run with a specified threshold

For each input sequence, PPR-Meta will output three scores (between 0 to 1), representing the probability that the sequence belongs to a phage, chromosome, or plasmid. By default, the prediction of PPR-Meta is the category with the highest score. You can also specify a threshold. In this way, a sequence with the highest score lower than the threshold will be labelled as "uncertain". In general, with a higher threshold, the percentage of uncertain predictions will be higher while the remaining predictions will be more reliable. For example, if you want to get reliable phage and plasmid sequences in the file “example.fna”, you can take 0.7 as the threshold. Please execute:

```
./PPR_Meta example.fna result.csv 0.7 (Run by executable file)
```

or

PPR_Meta('example.fna', 'result.csv', '0.7') (Run by MATLAB script)

When running PPR-Meta, you can ignore the warning about the information of the CPU/GPU, as shown in the screenshot below (Figure 2.).

```
2018-08-02 07:59:08.760305: I tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
2018-08-02 07:59:09.081744: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:892] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2018-08-02 07:59:09.082159: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Found device 0 with properties:
  name: GeForce GTX 1060 6GB major: 6 minor: 1 memoryClockRate(GHz): 1.835
  pciBusID: 0000:01:00.0
totalMemory: 5.92GiB freeMemory: 5.62GiB
2018-08-02 07:59:09.082179: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1120] Creating TensorFlow device (/device:GPU:0) -> (device: 0, name: GeForce GTX 1060 6GB, pci bus id: 0000:01:00.0, compute capability: 6.1)
```

Figure 2. Warning message when running PPR-Meta.

1.3.5 Output

The current version of PPR-Meta uses “comma-separated values (CSV)” as the format of the output file. You can use “.csv” as the extension of the output file. PPR-Meta will automatically add the “.csv” extension to the file name if the output file does not take “.csv” as its extension”. The output of PPR-Meta consists of six columns, representing “sequence header” (the same as the corresponding header in the fasta file), “sequence length”, “the probability that the sequence belongs to the phage”, “the probability that the sequence belongs to the chromosome”, “the probability that the sequence belongs to the plasmid” and “the possible source of the sequence”, respectively. A screenshot of the output file is shown in Figure 3:

	A	B	C	D	E	F
1	Header	Length	phage_score	chromosome_score	plasmid_score	Possible_source
2	phage1_source="Arthrobacter phage Mudcat"	7484	0.941588231	0.002345994	0.056065768	phage
3	phage2_source="Bacillus phage AR9"	8912	0.831541201	0.035068852	0.13338995	phage
4	phage3_source="Bacillus phage Aurora"	6983	0.730311051	0.181169275	0.088519651	phage
5	phage4_source="Bacillus phage Belinda"	1885	0.999458821	7.78E-06	0.000533381	phage
6	phage5_source="Bacillus phage DIGNKC"	4363	0.951073721	0.005356195	0.043570097	phage
7	phage6_source="Bacillus phage DirtyBetty"	3608	0.991743643	0.000850173	0.007406195	phage
8	phage7_source="Bacillus phage Eldridge"	5233	0.894087191	0.020961364	0.08495145	phage
9	phage8_source="Bacillus phage Nemo"	186	0.97037226	0.009645794	0.019981954	phage
10	phage9_source="Bacillus phage Nigalana"	3508	0.98406637	0.000495979	0.015437657	phage

Figure 3. The output file of PPR-Meta.

2. Virtual machine version

Running PPR-Meta in the virtual machine is much easier for the user who is not familiar with the command line, and the virtual machine can be installed on any PC.

The running time of the virtual machine version may be longer because the virtual machine could not speed up with GPU. The code is also modified to separate the input sequences into more batches to reduce memory requirements, which may increase the

running time.

The following is the step by step guide to run PPR-Meta on a virtual machine. The version of the VirtualBox used was 6.0.4.

2.1 Install the virtual machine and run PPR-Meta on a virtual machine.

Step 1: download the “VM_Bioinfo.vdi.7z” file from the PPR-Meta homepage (http://cqb.pku.edu.cn/ZhuLab/PPR_Meta/VM_Bioinfo.vdi.7z). The “7z” file can easily be decompressed using current compressing software, such as “WinRAR”, “WinZip”, and “7-Zip”.

Step 2: download the VirtualBox software from <https://www.virtualbox.org> and install the VirtualBox. The VirtualBox is easy to install, you just need to select an installation folder and click the “next” button in each step.

Step 3: Open VirtualBox, click the “New” button to create a virtual machine, as shown in Figure 4.

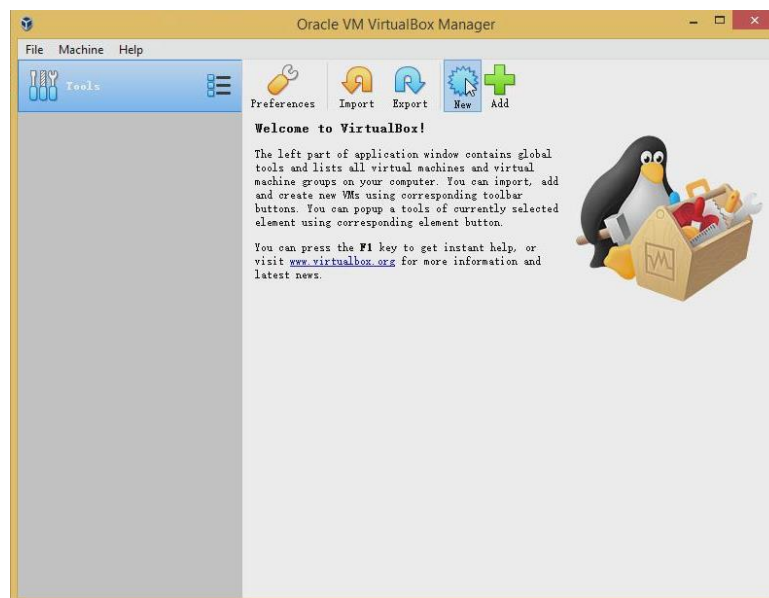


Figure 4. Correction interface of VirutualBox.

Step 4: Specify a name, select “Linux” as the operating system and select “Ubuntu” as the version of the operating system. Then, click “Next”, as shown in Figure 5.

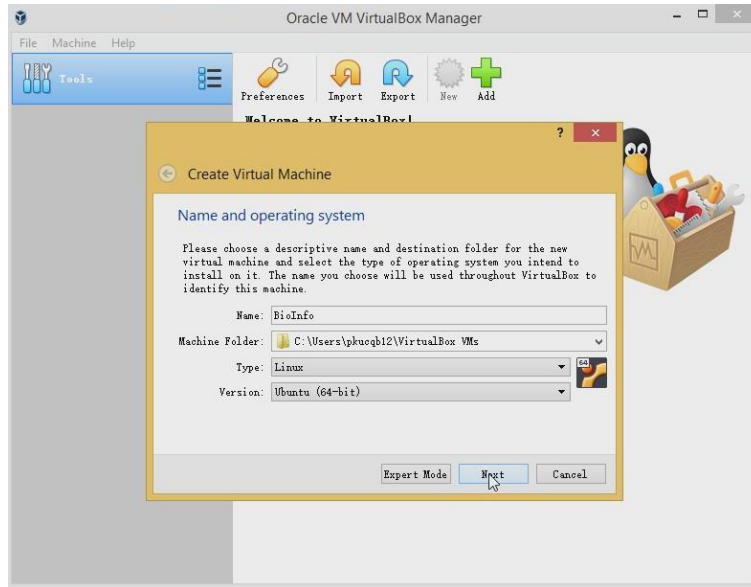


Figure 5. The interface to choose the operating system.

Step 5: If possible, allocate a larger amount of memory to the virtual machine. Click “next” (see Figure 6).

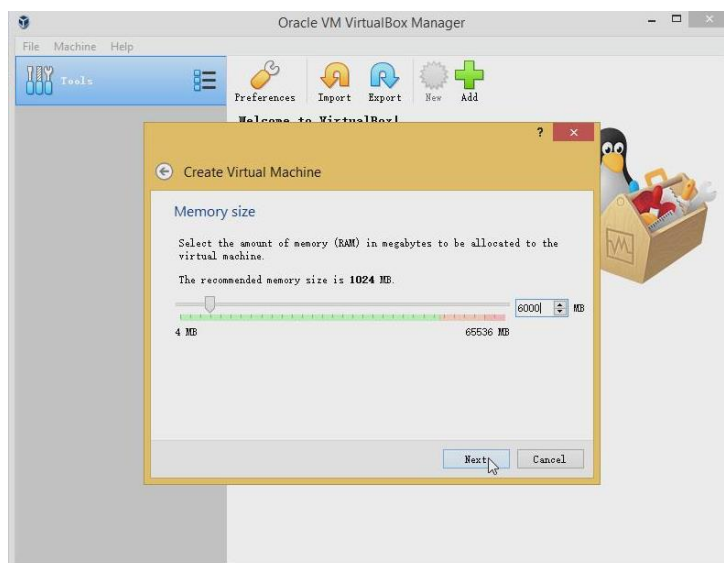


Figure 6. The interface to choose the memory of a virtual machine.

Step 6: Select “Use an existing virtual hard disk file”, and specify the “VM_Bioinfo.vdi” file downloaded from PPR-Meta homepage. Click “Create” (see Figure 7).

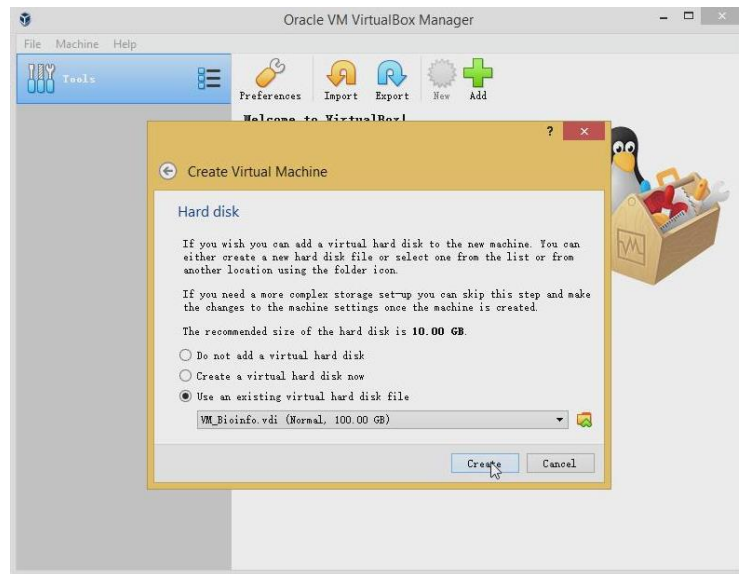


Figure 7. The interface to choose a hard disk.

Step 7: Click “start” to open the machine (see Figure 8).

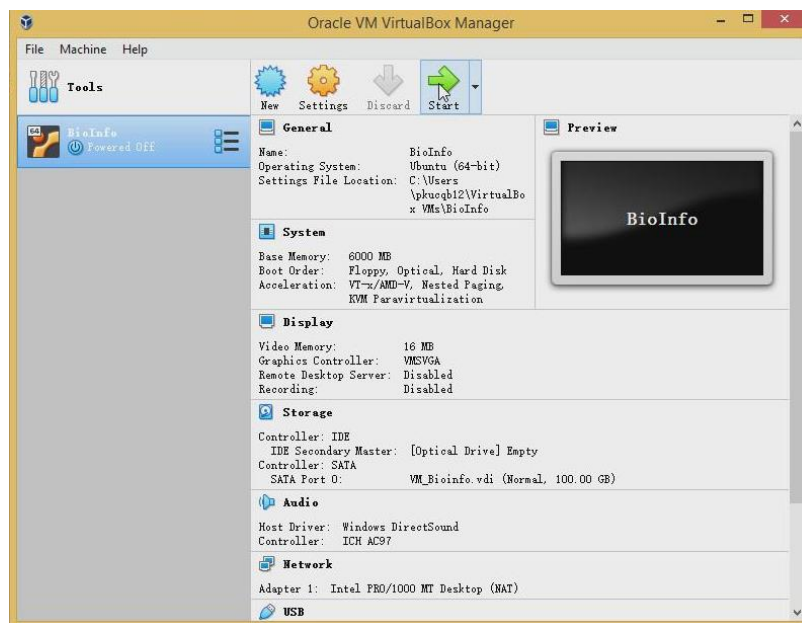


Figure 8. The interface to start the virtual machine.

Step 8: The PPR-Meta is on the desktop (see Figure 9).



Figure 9. The desktop of the virtual machine.

Step 9: Go into the “PPR_Meta_v_1_0” folder, click the right click and open the terminal (see Figure 10).

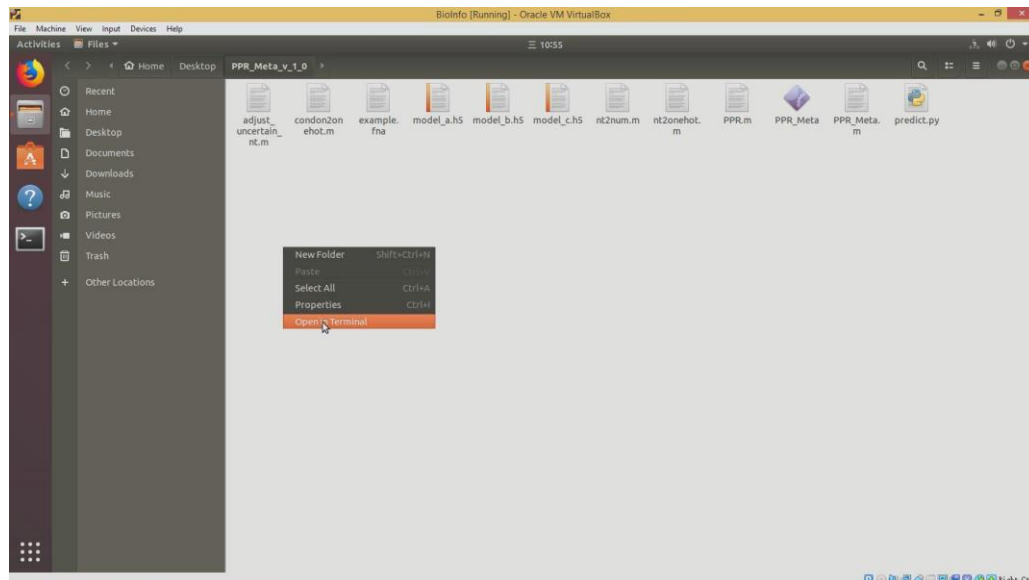


Figure 10. The folder of PPR-Meta.

Step 10: Now you can run PPR-Meta (see also 1.3.1 and 1.3.3). You can ignore the “FutureWarning” and the information about the CPU (see Figure 11).

```

bioinfo@bioinfo: ~/Desktop/PPR_Meta_v_1_0
bioinfo@bioinfo:~/Desktop/PPR_Meta_v_1_0$ ./PPR_Meta example.fna result.csv

/usr/lib/python2.7/dist-packages/h5py/_init_.py:36: FutureWarning: Conversion of the second argument of 'issubdtype' from 'float' to 'np.floating' is deprecated. In future, it will be treated as 'np.float64 == np.dtype(float).type'.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
2019-02-06 10:56:31.653024: I tensorflow/core/platform/cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary was not compiled to use
: SSE4.1 SSE4.2 AVX AVX2

```


Figure 11. The interface to run PPR-Meta.

2.2 Exchange file between the physical host and virtual machine.

Step 1: Create shared folders in both the physical host (shared_host, see Figure 12) and virtual machine (shared_VM, see Figure 13).

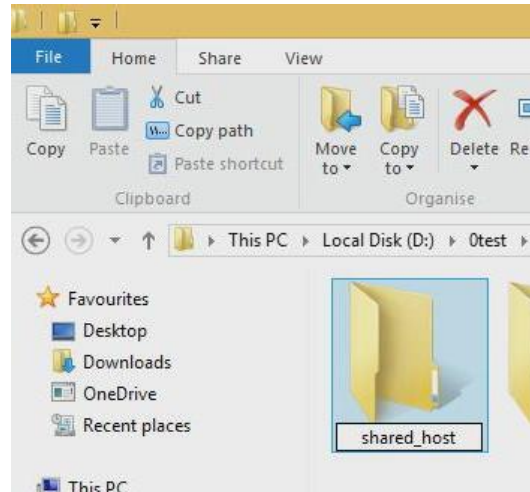


Figure 12. The shared folder on the physical host.

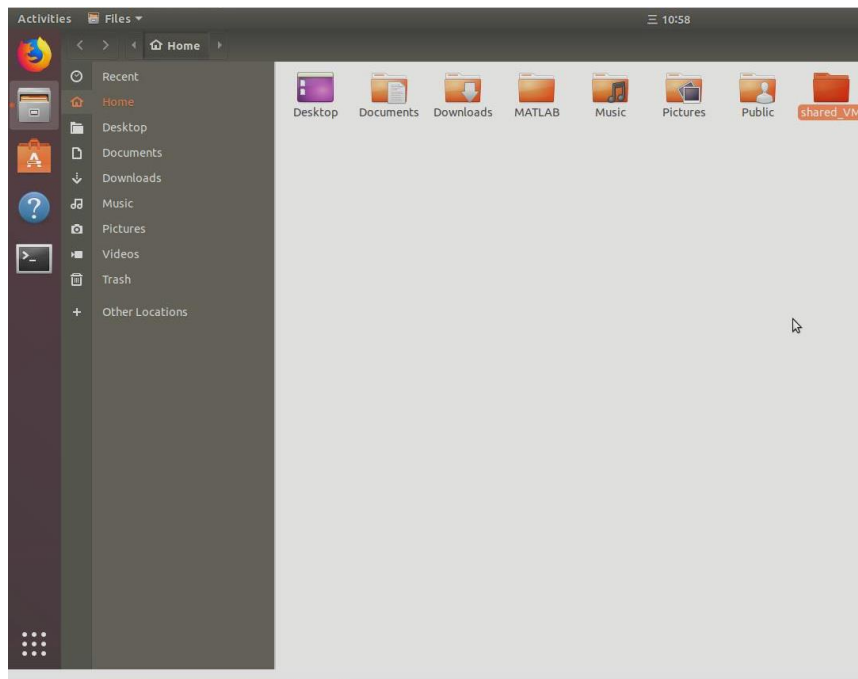


Figure 13. The shared folder on the virtual machine.

Step 2: In the window of VirtualBox, click “Devices”, “Shared Folder”, “Shared Folders Settings” (see Figure 14).

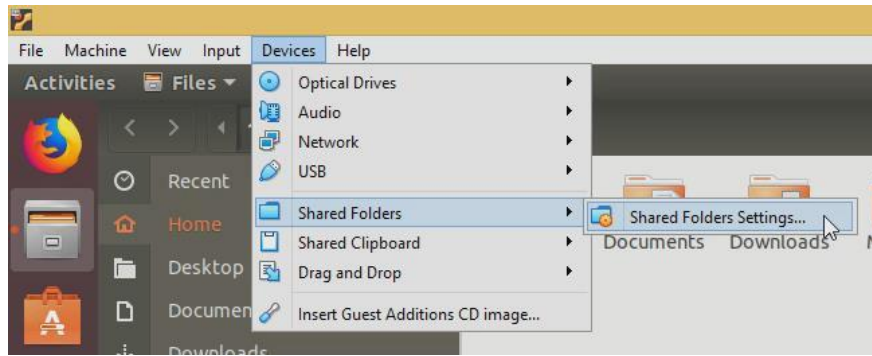


Figure 14. The interface to enter the setting manu.

Step 3: Add shared folder of the physical host (click ‘+’ symbol, see Figure 15) and select “Auto-mount” (see Figure 16).

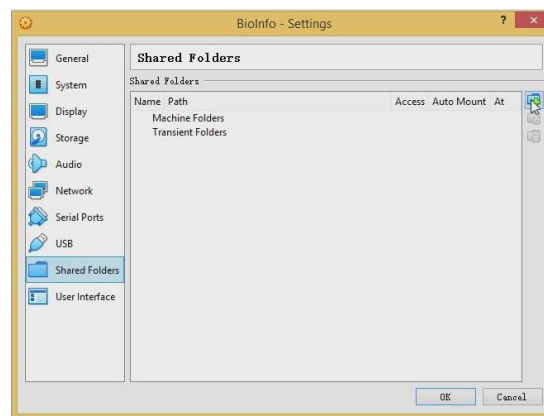


Figure 15. The interface to click ‘+’ symbol.

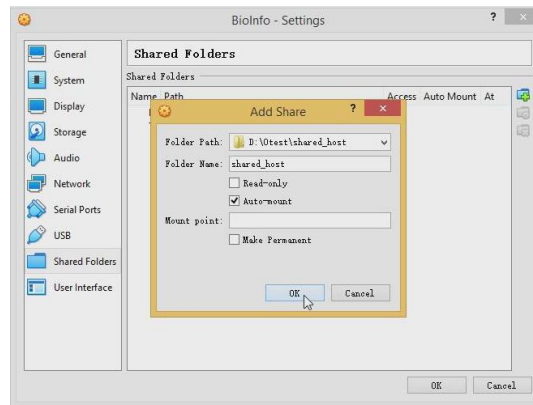


Figure 16. The interface to choose ‘Auto-mount’.

Step 4: Restart the virtual machine.

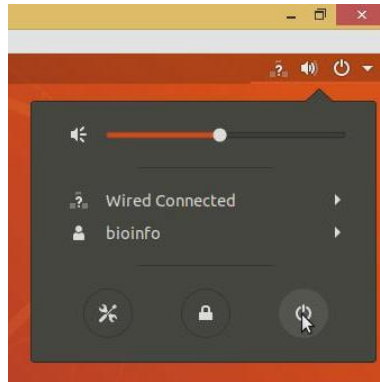


Figure 17. The interface to restart the virtual machine.

Step 5: Go to the parent folder of the shared folder in the virtual machine, click the right click and open the terminal. Copy the command in the “command.txt” file to the terminal:

```
sudo mount -t vboxsf shared_host shared_VM
```

You should replace “shared_host” and “shared_VM” to the shared folders’ name that you specify. The password of the virtual machine is 1. It can be seen in Figure 18.

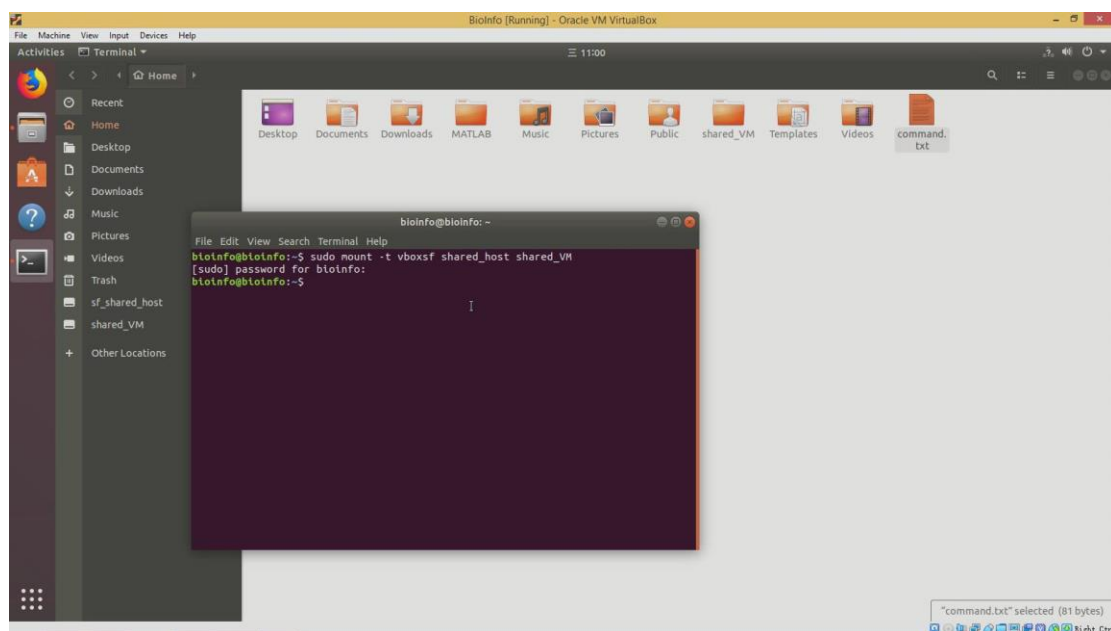


Figure 18. The interface to mount the shared folder.

Step 6: Then you can exchange files between the virtual machine and the physical host. For example, you can copy the file in the virtual machine to the “shared_VM” folder, and this file will also exist in the “shared_host” folder in the physical host, vice versa. If you want to know more about the file exchange, click “Help” -> “Contents” -> “Guest

Additions” -> “Shared folders” in the VirtualBox window for more details.

3. Docker version

Although the virtual machine version needs no requirement installing, it is a little complicated to install a virtual machine, load a virtual machine and transfer files between the virtual machine and the physical host. Here we release a new docker version of PPR-Meta, which is more convenient and widely used for open-source tools. It can run on any operating system if you have installed docker. Next, we will show s step-by-step usage.

3.1 Installing Docker on multiple platforms

You can download and install Docker on Windows, Mac and Linux platforms from <https://docs.docker.com/get-docker/> (see Figure 19). You can see the following options and click a corresponding platform for you. After switching to detailed installing instructions, you can install a docker. Especially, if you are a non-root user on Linux platform, you should add yourself to a docker group so that you can have access to running docker (see <https://docs.docker.com/engine/install/linux-postinstall/>). To test whether Docker has installed correctly, you could open your terminal and run:

```
docker run hello-world
```

If no error messages, the docker has been successfully installed.

Docker Desktop

The fastest way to containerize applications on your desktop

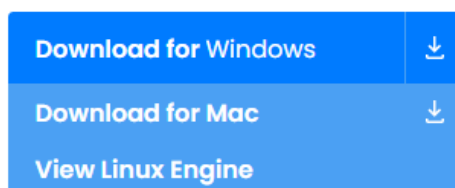


Figure 19. The interface to download and install Docker on different platforms.

3.2 Using docker repository of PPR-Meta

When you install docker successfully, you can start to use docker version of PPR-Meta in a terminal.

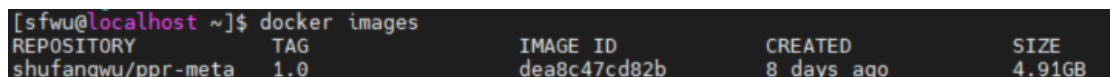
Step 1: Open a new terminal and change the path to the location that you want.

Step 2: Pull down the PPR-Meta image from Docker Hub by running the command:

```
docker pull shufangwu/ppr-meta:1.0
```

Step 3: Obtain all the local docker images that you have by running the command (see Figure 20):

docker images



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
shufangwu/ppr-meta	1.0	dea8c47cd82b	8 days ago	4.91GB

Figure 20. The interface to obtain all the local docker images.

Step 4: Start a PPR-Meta docker container based on downloaded PPR-Meta images by running the command:

```
docker run -it shufangwu/ppr-meta:1.0 bash.
```

Then you can enter a PPR-Meta docker container and begin to use PPR-Meta as shown in Figure 21.

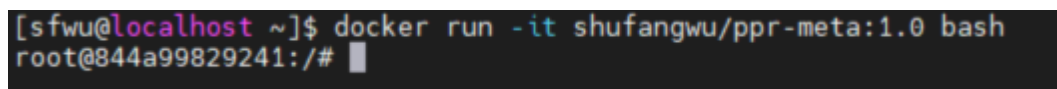


Figure 21. The interface to enter a PPR-Meta container.

Step 5: Change the folder to the /home/PPR-Meta and test 'example.fna' file as the same as the procedures in 1.3.1 and 1.3.3 by running the command (see Figure 22):

```
cd /home/PPR-Meta
```

```
./PPR_Meta example.fna result.csv
```

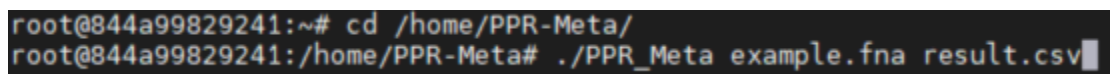
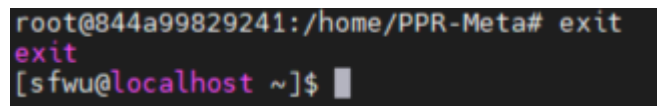


Figure 22. The interface to run PPR-Meta in a docker container.

3.3 Reenter the container and transfer files between your host and the PPR-Meta docker container that was used before.

Step 1: If you want to exit from a current using container and stop this container, you can just type ‘exit’ in the command line and then click ‘Enter’ on the keyboard (see Figure 23). While you just want to exit and keep the container running, you can press Ctrl+P+Q. Then you can come back to your host.



```

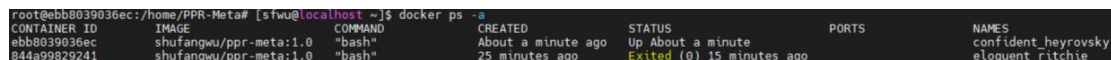
root@844a99829241:/home/PPR-Meta# exit
exit
[sfwu@localhost ~]$

```

Figure 23. The interface to exit the container.

Step 2: Obtain the information about all the available containers:

docker ps -a



```

root@ebb8039036ec:/home/PPR-Meta# [sfwu@localhost ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
ebb8039036ec       shufangwu/ppr-meta:1.0  "bash"             About a minute ago  Up About a minute  0.0.0.0:22->22      confident_heyrovsky
844a99829241       shufangwu/ppr-meta:1.0  "bash"             25 minutes ago     Exited (0) 15 minutes ago                  eloquent_fitchie

```

Figure 24. The interface to obtain all the available containers.

You need to pay attention to ‘CONTAINER ID’ and ‘STATUS’ columns. For example, the container (CONTAINER ID: 844a99829241)’s STATUS is ‘Exited’, which means this container is stopped when using the ‘exit’ command to exit. However, the container (CONTAINER ID: ebb8039036ec)’s STATUS is ‘Up’, which means this container is running when using ‘Ctrl+P+Q’ to exit. The running containers could transfer files between itself and the host while the stopped containers could not. You can also rerun a stopped container by:

docker start 844a99829241

(‘844a99829241’ is the CONTAINER ID of the stopped container)

Step 3: Transfer input_file.fna file from your host to PPR-Meta container by running the command:

docker cp <Folder_path_on_host>/input_file.fna

CONTAINER_ID:<Destination_path_on_container>

For example, you can use the command in your host:

docker cp ./input_file.fna ebb8039036ec:/home/PPR-Meta

(‘ebb8039036ec’ must be a running CONTAINER ID that you want to use)

Transfer output.csv file from the PPR-Meta container to your host by running the command:

```
docker cp CONTAINER_ID:<Folder_path_on_container>/ output.csv  
      <Folder_path_on_host>
```

For example, you can use the command in your host:

```
docker cp ebb8039036ec:/home/PPR-Meta/output.csv ./
```

(‘ebb8039036ec’ must be a running CONTAINER ID that you want to use).

Step 4: After transfer an input file to the container, you can reenter this container and then use PPR-Meta to make a prediction. A running container can be reentered. If the container’s statute is ‘Exited’, you should first start it by using the command:

```
docker start CONTAINER_ID
```

Then you can enter this running container by command:

```
docker attach CONTAINER_ID
```

```
(or docker exec -it CONTAINER_ID bash)
```

You will enter the container and start to make a prediction as the same as the procedure in 3.2 step 5.