Manual of DeePhage Release Version 1.0

Table of Contents

Any question, please do not hesitate to contact me: wu-shufang@pku.edu.cn

DeePhage can run either in the physical host or virtual machine. The physical host version can speed up with GPU. For non-computer professionals, we recommend using the virtual machine version.

Part I. Physical host version

1. Operating system

Linux (DeePhage has been tested on Ubuntu 16.04)

2. Requirements

●Python 3.6.7 (https://www.python.org/)
●Python packages:
  ○numpy 1.16.4(http://www.numpy.org/)
  ○h5py 2.9.0(http://www.h5py.org/)
●TensorFlow 1.4.0 (https://www.tensorflow.org/)
●Keras 2.1.3 (https://keras.io/)
●MATLAB Component Runtime (MCR) R2018a (https://www.mathworks.com/products/compiler/matlab-runtime.html)

or

MATLAB R2018a (https://www.mathworks.com/products/matlab.html)

Note:
1. For compatibility, we recommend installing the tools with the similar version as described above.
2. If GPU is available in your machine, we recommend installing a GPU version of the TensorFlow to speed up the program.
3. DeePhage can be run with either an executable file or a MATLAB script. If you run DeePhage through the executable file, you need to install the MCR while MATLAB is not necessary. If you run DeePhage through the MATLAB script, MATLAB is required.

3. Preparation

Please install numpy, h5py, TensorFlow, Keras, MCR (or MATLAB) according to their manuals.

The numpy, h5py, TensorFlow, and Keras are python packages, which can be installed with “pip”. If “pip” is not already installed in your machine, use the command “sudo apt-get install python-pip python-dev” to install “pip”. Here are example commands of installing the above python packages using “pip”.

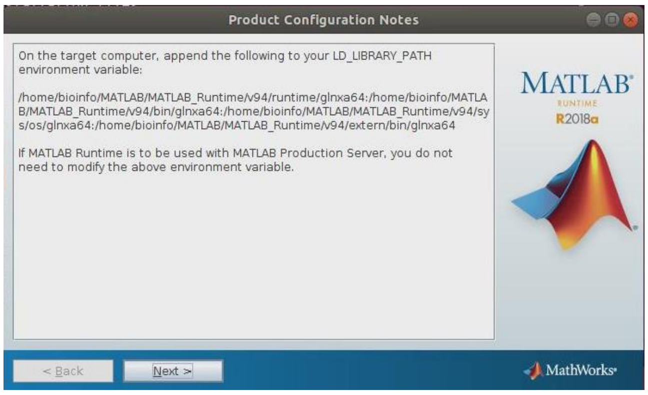
```
pip install numpy
pip install h5py
pip install tensorflow==1.4.1        #CPU version
pip install tensorflow-gpu==1.4.1 #GPU version
pip install keras==2.1.3
```

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 DeePhage 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:



Product Configuration Notes

On the target computer, append the following to your LD_LIBRARY_PATH environment variable:

```
/home/bioinfo/MATLAB/MATLAB_Runtime/v94/runtime/glnxa64:/home/bioinfo/MATLA
B/MATLAB_Runtime/v94/bin/glnxa64:/home/bioinfo/MATLAB/MATLAB_Runtime/v94/sy
s/os/glnxa64:/home/bioinfo/MATLAB/MATLAB_Runtime/v94/extern/bin/glnxa64
```

If MATLAB Runtime is to be used with MATLAB Production Server, you do not need to modify the above environment variable.

When running DeePhage through the MATLAB script, please see https://www.mathworks.com/support/ to install the MATLAB.

4. Usage

To run DeePhage, please download DeePhage package using git, and then change directory to DeePhage:

```
git clone https://github.com/shufangwu/DeePhage.git
cd DeePhage
```

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

```
wget http://cqb.pku.edu.cn/ZhuLab/DeePhage/DeePhage_v_1_0.zip
unzip DeePhage_v_1_0.zip
cd DeePhage_v_1_0
```

4.1. Run by executable file (in command line)

In this form, please simply executes the command:

./DeePhage <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. The users can use the file “example.fna” which contains 20 sequences in the program folder to test the DeePhage by simply executing the command:

./DeePhage example.fna result.csv

4.2. Run by MATLAB script (in MATLAB GUI)

In this form, please execute the following command directly in the MATLAB command window.

DeePhage(‘<input_file_folder>/input_file.fna’, ‘<output_file_folder>/output_file.csv’)

For example, if you want to identify the sequences in example.fna, please execute:

DeePhage(‘example.fna’, ‘result.csv’)

Remember to set the working path of MATLAB to the program folder before running the program.

4.3. Run with specified cutoff

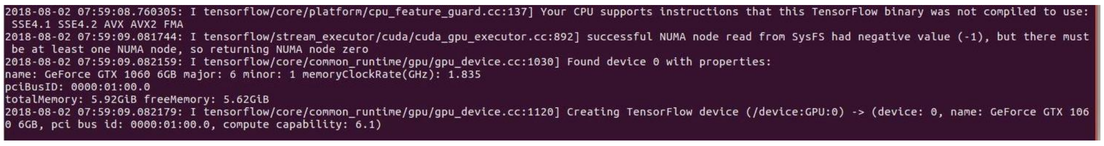
For each input sequence, DeePhage will output one scores (between 0 to 1), representing the lifestyle prediction score. The sequence with a score higher than 0.5 would be regarded as a virulent phage-derived fragment and the sequence with a score lower than 0.5 would be regarded as a temperate phage-derived fragment. Users can also specify a cutoff. In this way, a sequence with a score between (0.5-cutoff/2, 0.5+cutoff/2) will be labelled as "uncertain". In general, with a higher cutoff, the percentage of uncertain predictions will be higher while the remaining predictions will be more reliable. For example, if you want to get relatively reliable predictions in the file “example.fna”, you can take 0.6 as the cutoff. Please execute:

./DeePhage example.fna result.csv 0.6 (Run by executable file)

or

DeePhage(‘example.fna’, ‘result.csv’,‘0.6’) (Run by MATLAB script)

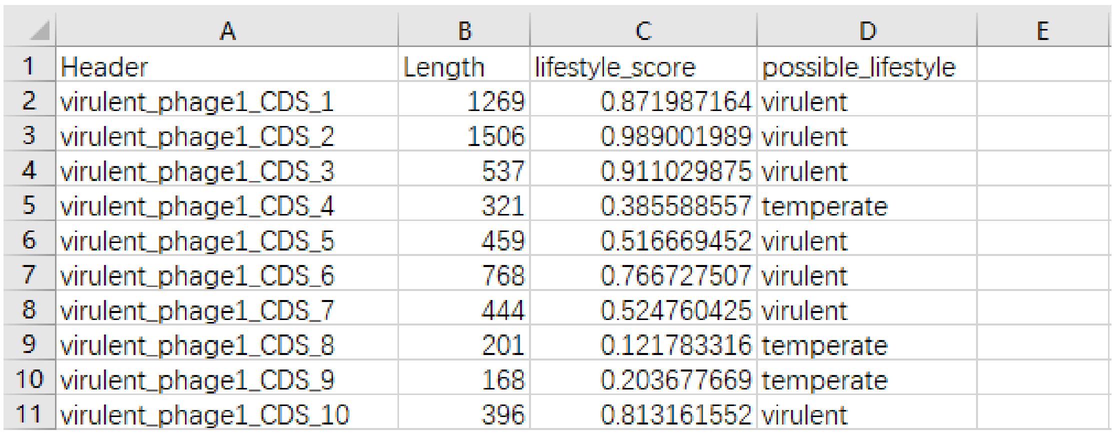
Note: When running DeePhage, you can ignore the warning about the information of the CPU/GPU, similarly as shown in the screenshot below.



```
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 106
0 6GB, pci bus id: 0000:01:00.0, compute capability: 6.1)
```

5. Output

The output of DeePhage consists of four columns, representing “sequence header” (the same with the corresponding header in the fasta file), “sequence length”, “the lifestyle prediction score”, “the possible lifestyle of the sequence”, respectively. If DeePhage is executed under specified cutoff, the sequence with the score between (0.5-cutoff/2, 0.5+cutoff/2) will be label as uncertain in ‘possible_lifestyle’ columns. Here is a screenshot of the output file:



	A	B	C	D	E
1	Header	Length	lifestyle_score	possible_lifestyle	
2	virulent_phage1_CDS_1	1269	0.871987164	virulent	
3	virulent_phage1_CDS_2	1506	0.989001989	virulent	
4	virulent_phage1_CDS_3	537	0.911029875	virulent	
5	virulent_phage1_CDS_4	321	0.385588557	temperate	
6	virulent_phage1_CDS_5	459	0.516669452	virulent	
7	virulent_phage1_CDS_6	768	0.766727507	virulent	
8	virulent_phage1_CDS_7	444	0.524760425	virulent	
9	virulent_phage1_CDS_8	201	0.121783316	temperate	
10	virulent_phage1_CDS_9	168	0.203677669	temperate	
11	virulent_phage1_CDS_10	396	0.813161552	virulent	

Note:
The current version of DeePhage uses "comma-separated values (CSV)" as the format of the output file. Please use ".csv" as the extension of the output file. DeePhage will automatically add the ".csv" extension to the file name if the output file does not take ".csv" as its extension"

Part II. Virtual machine version

Running DeePhage in the virtual machine is much easier for user who is not familiar with the command line, and the virtual machine can be installed in any PC. Note that the running time of the virtual machine version may be longer because the virtual machine could not speed up with GPU. We also modified the code to separate the input sequences into more batches to reduce memory requirements, which may increase the running time.

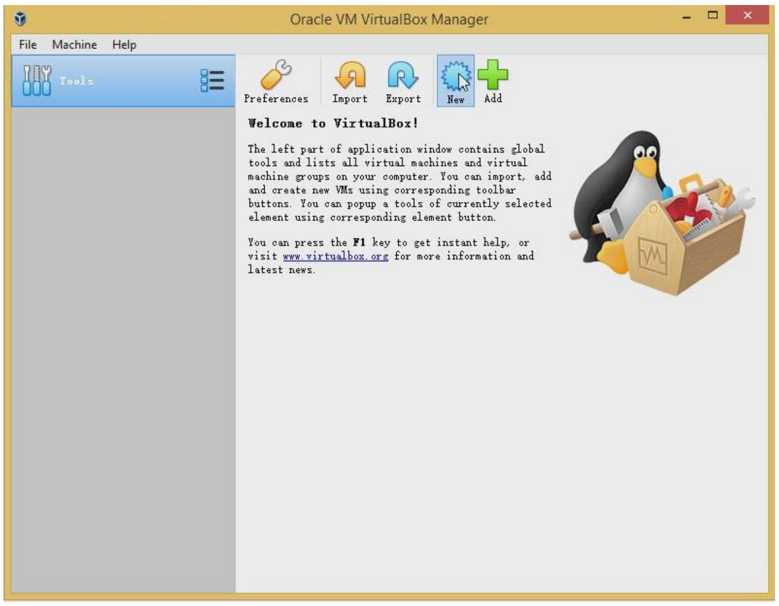
The following is the step by step guide to run DeePhage in virtual machine. The version of the VirtualBox we used was 6.0.8.

1. Install the virtual machine and run DeePhage in virtual machine.

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

Step 2: download the VirtualBox software form 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 virtual machine.

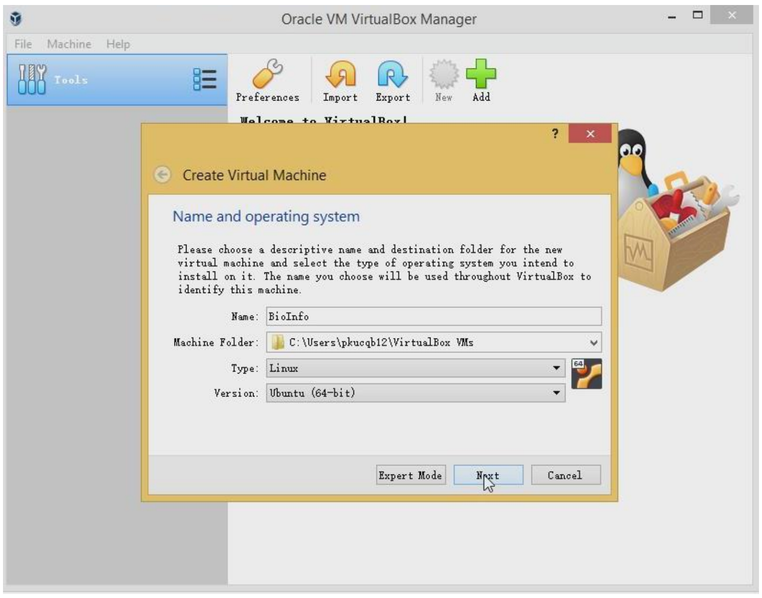


Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

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



Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

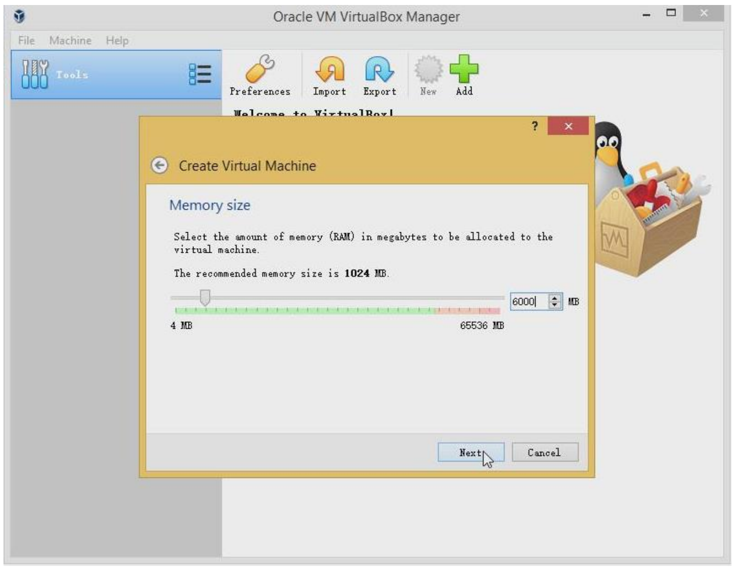
Name: BioInfo

Machine Folder: C:\Users\pkucqb12\VirtualBox VMs

Type: Linux

Version: Ubuntu (64-bit)

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


Oracle VM VirtualBox Manager
File Machine Help
Tools
Preferences Import Export New Add
Welcome to VirtualBox!

Create Virtual Machine

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

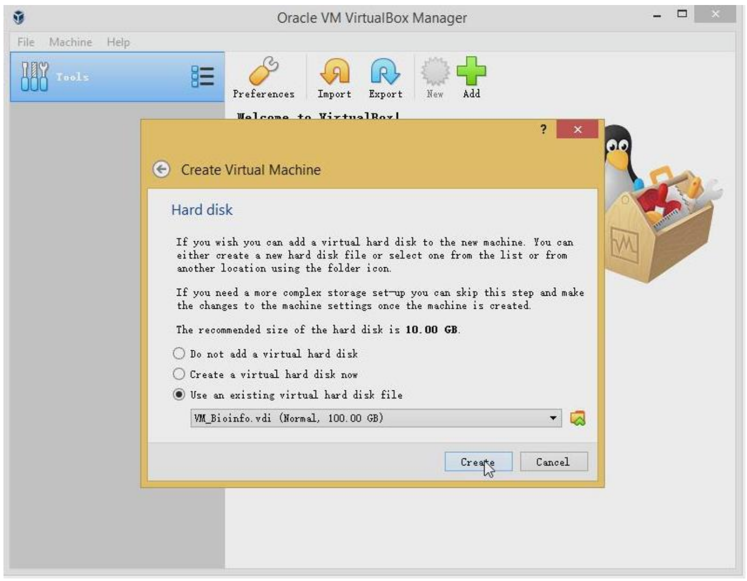
The recommended memory size is **1024** MB.

6000 MB

4 MB 65536 MB

Next Cancel

Step 6: Select "Use an existing virtual hard disk file", and specify the "VM_Bioinfo.vdi" file downloaded from DeePhage homepage. Click "Create".


Oracle VM VirtualBox Manager
File Machine Help
Tools
Preferences Import Export New Add
Welcome to VirtualBox!

Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

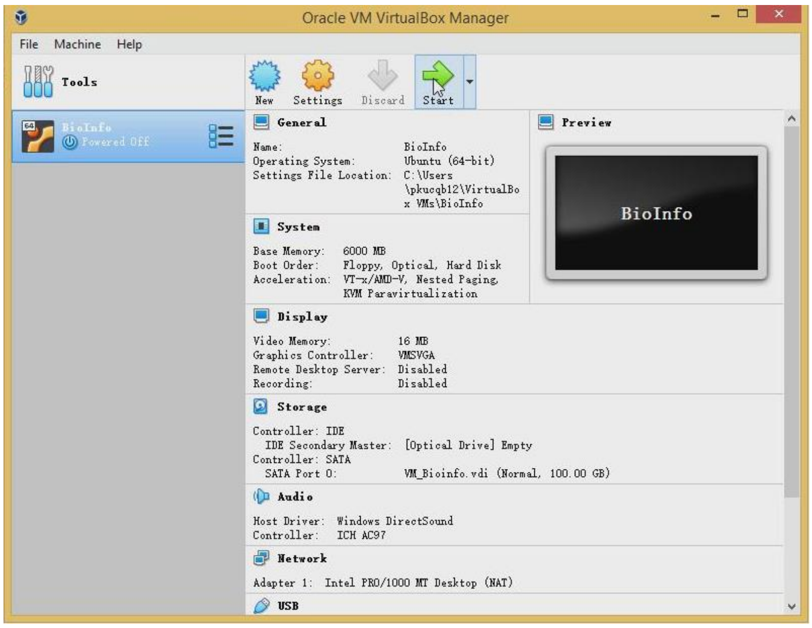
The recommended size of the hard disk is **10.00 GB**.

○ Do not add a virtual hard disk
○ Create a virtual hard disk now
● Use an existing virtual hard disk file

VM_Bioinfo.vdi (Normal, 100.00 GB)

Create Cancel

Step 7: Click "start" to open the machine.



General

```
Name:                    BioInfo
Operating System:        Ubuntu (64-bit)
Settings File Location:  C:\Users
                         \pkucqb12\VirtualBo
                         x VMs\BioInfo
```

System

```
Base Memory:   6000 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  VT-x/AMD-V, Nested Paging,
               KVM Paravirtualization
```

Display

```
Video Memory:           16 MB
Graphics Controller:    VMSVGA
Remote Desktop Server:  Disabled
Recording:              Disabled
```

Storage

```
Controller: IDE
  IDE Secondary Master:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:           VM_Bioinfo.vdi (Normal, 100.00 GB)
```

Audio

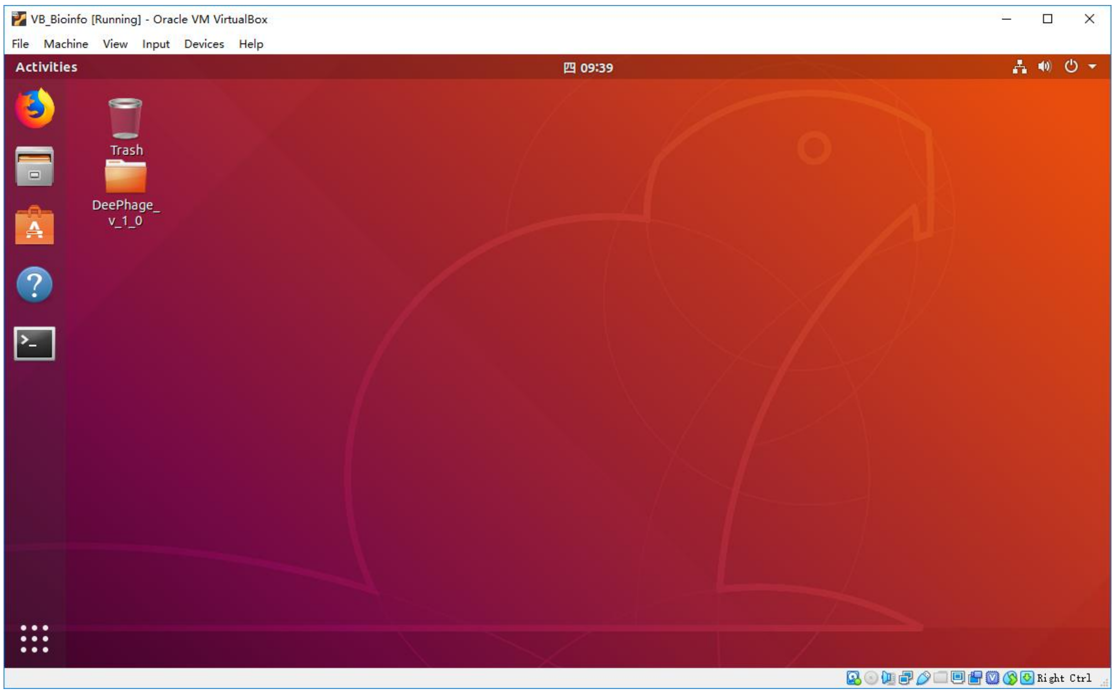
```
Host Driver:  Windows DirectSound
Controller:   ICH AC97
```

Network

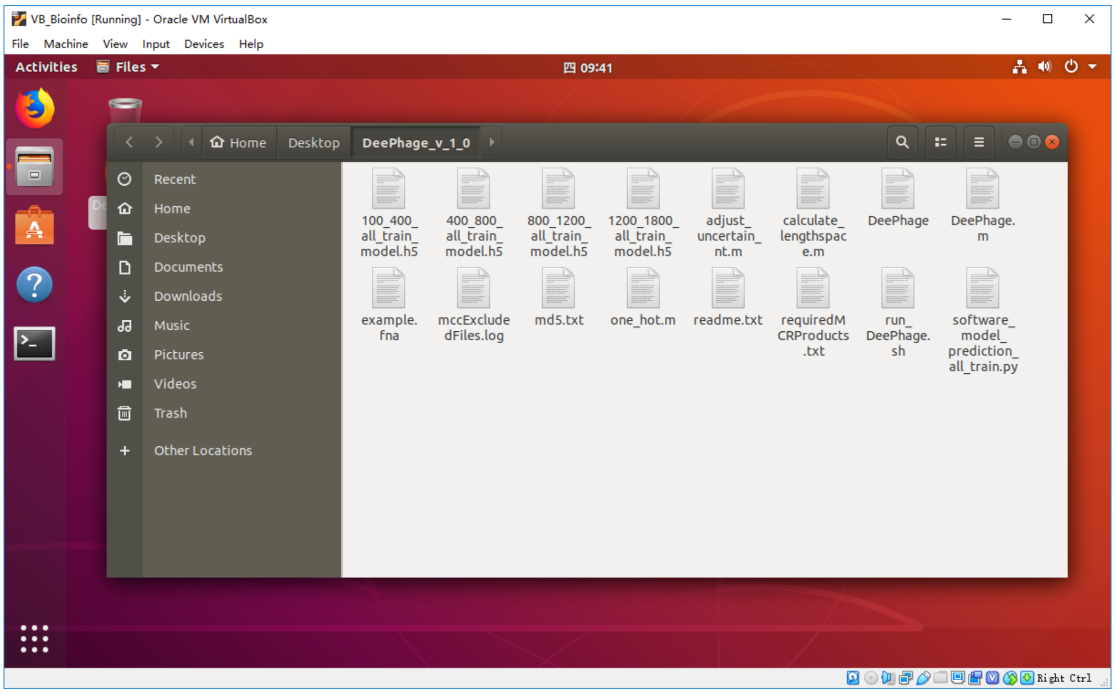
```
Adapter 1:  Intel PRO/1000 MT Desktop (NAT)
```

USB

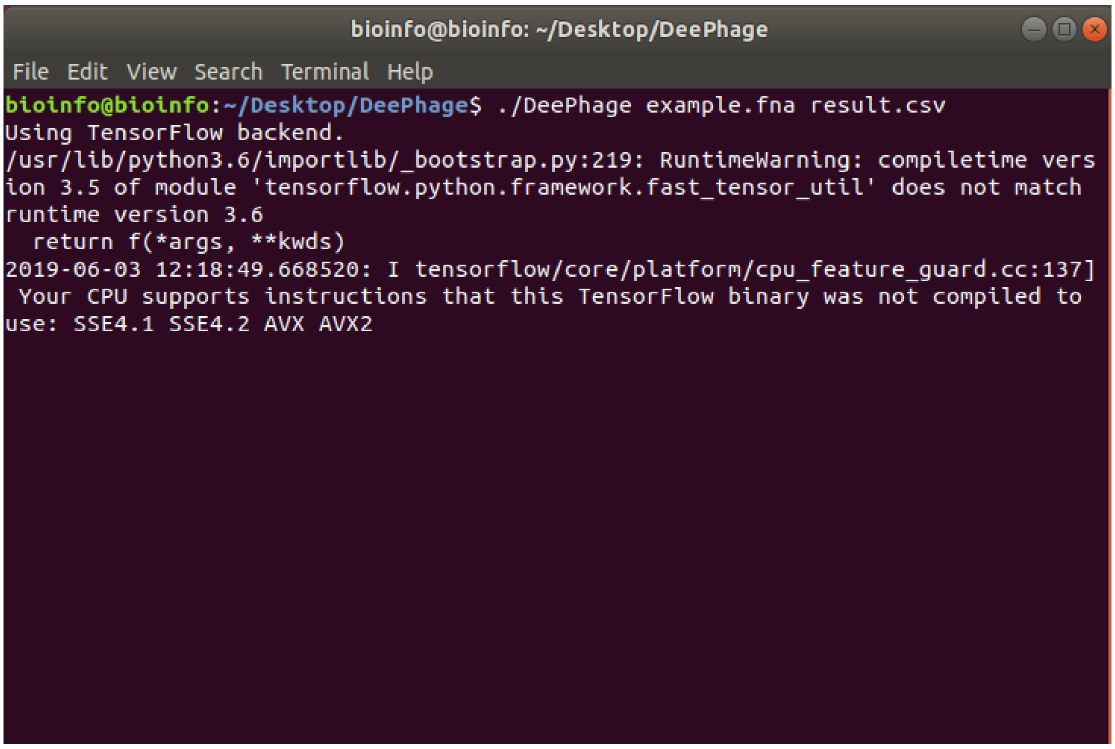
Step 8: The DeePhage is on the desktop.


VB_Bioinfo [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
09:39
Trash
DeePhage_
v_1_0
Right Ctrl

Step 9: Go into the “DeePhage_v_1_0” folder, click the right click and open the terminal.



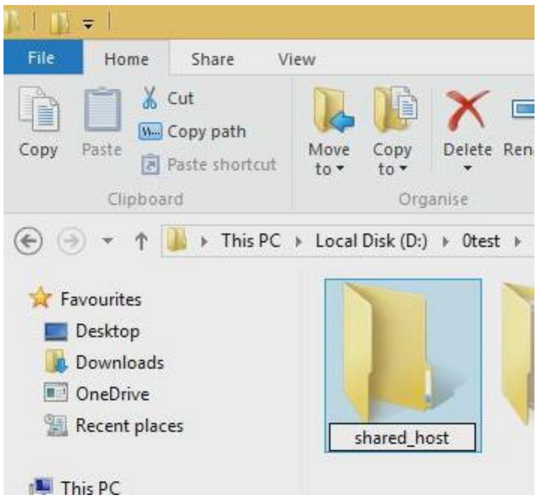
Step 10: Now you can run DeePhage (see also part Ⅰ, section 4.1 and 4.3). You can ignore the "FutureWarning" and the information about the CPU.

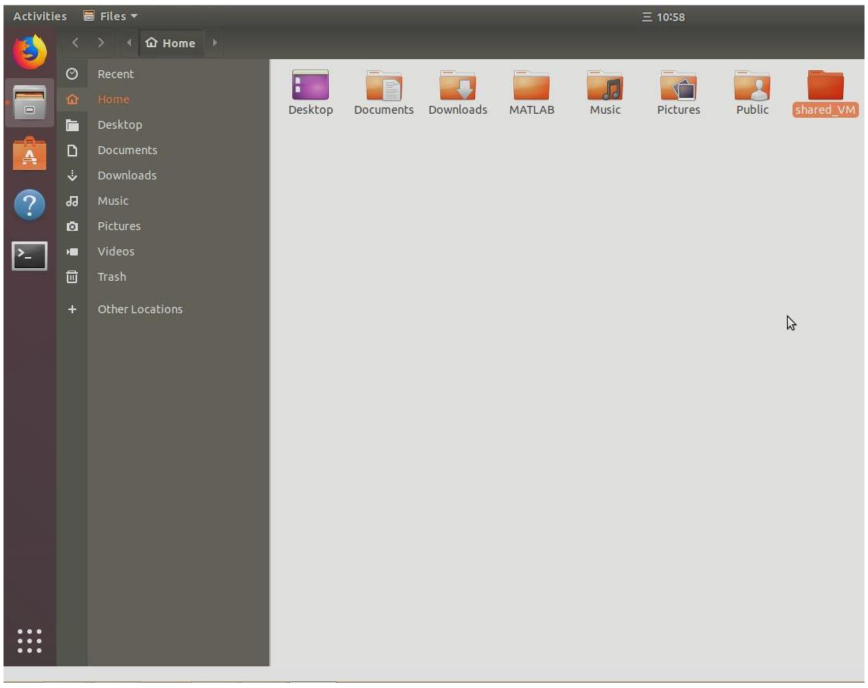

bioinfo@bioinfo: ~/Desktop/DeePhage
File Edit View Search Terminal Help

```
bioinfo@bioinfo:~/Desktop/DeePhage$ ./DeePhage example.fna result.csv
Using TensorFlow backend.
/usr/lib/python3.6/importlib/_bootstrap.py:219: RuntimeWarning: compiletime vers
ion 3.5 of module 'tensorflow.python.framework.fast_tensor_util' does not match
runtime version 3.6
  return f(*args, **kwds)
2019-06-03 12:18:49.668520: 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
```

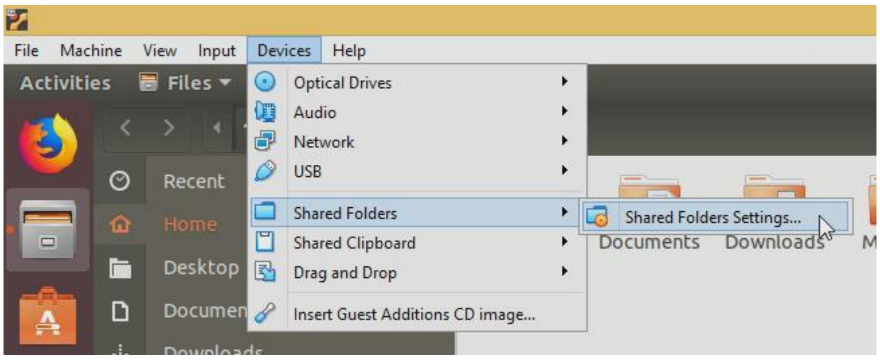
2. Exchange file between physical host and virtual machine.

Step 1: Create shared folders in both physical host (shared_host) and virtual machine (shared_VM).

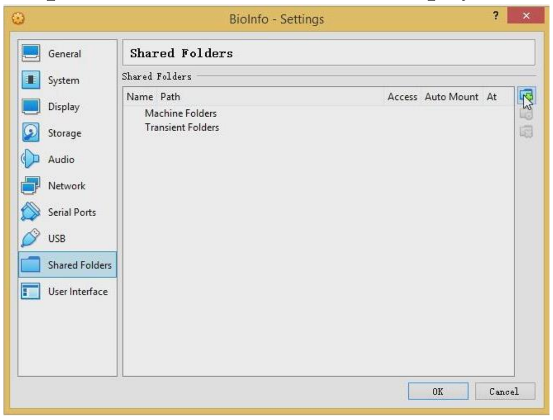


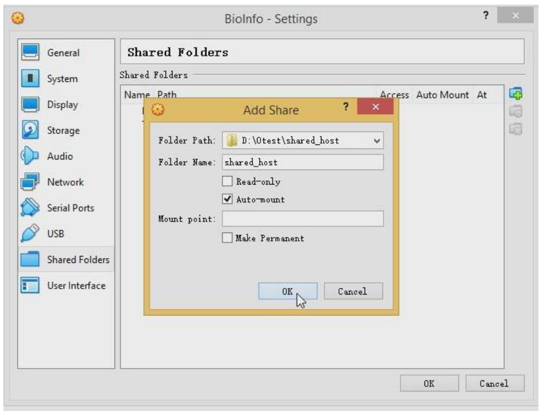


Step 2: In the window of VirtualBox, click "Devices", "Shared Folder", "Shared Folders Settings".



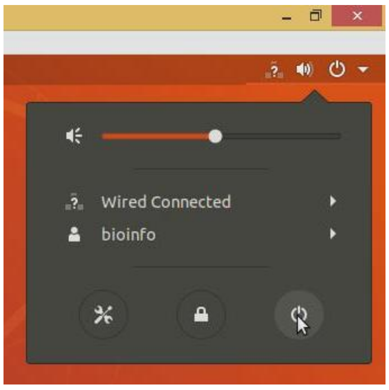
Step 3: Add shared folder of the physical host and select "Auto-mount".





Folder Path: D:\0test\shared_host
Folder Name: shared_host
Read-only
✓ Auto-mount
Mount point:
Make Permanent

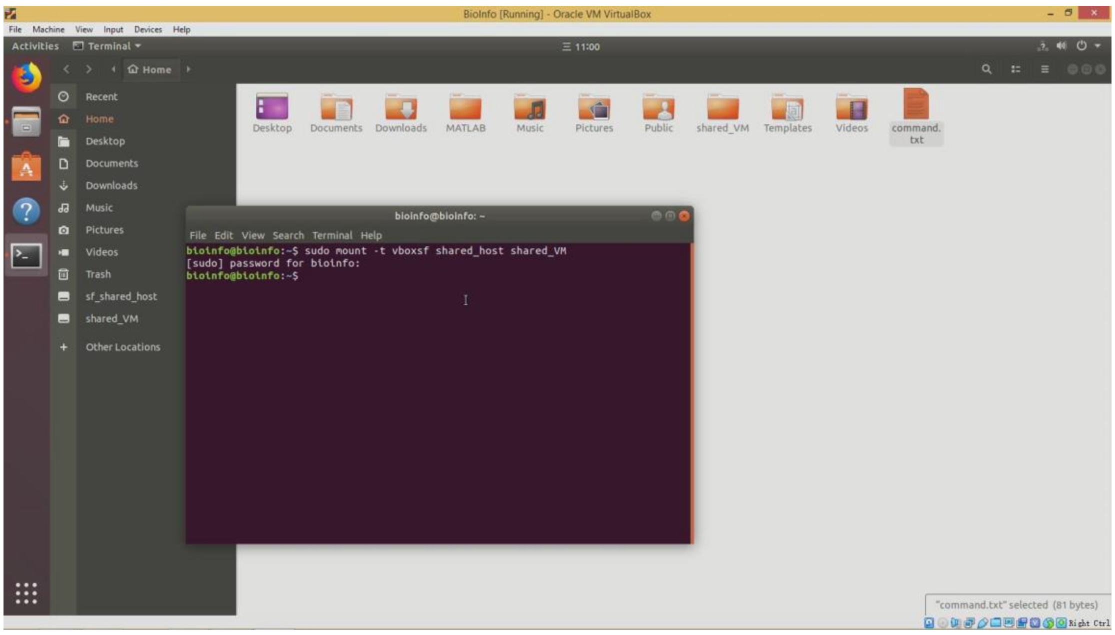
Step 4: Restart the virtual machine.



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
```

Note, you should replace "shared_host" and "shared_VM" to the shared folders' name that you specify. **The password of the virtual machine is 1.**



```
bioinfo@bioinfo:~$ sudo mount -t vboxsf shared_host shared_VM
[sudo] password for bioinfo:
bioinfo@bioinfo:~$
```

Step 6: Now, you can exchange files between virtual machine and 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.

Note:

If you want to run multiple tasks at the same time (either on physical host or virtual machine), please copy DeePhage package into different folders and run different tasks under different folders. Do not run different tasks under the same folder.