

Aug 10, 2018

Ubuntu on Windows for computational biology

DOI

<https://dx.doi.org/10.17504/protocols.io.sfuebnw>



ubuntu[®]

James Lloyd¹

¹University of Western Australia



James P B Lloyd

University of Western Australia

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.sfuebnw>

External link: <https://www.badgrammargoodsyntax.com/compbio/2017/10/28/compbio-009-ubuntu-on-windows>

Protocol Citation: James Lloyd 2018. Ubuntu on Windows for computational biology. protocols.io

<https://dx.doi.org/10.17504/protocols.io.sfuebnw>

License: This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited



Protocol status: Working

We use this protocol and it's working

Created: August 07, 2018

Last Modified: August 10, 2018

Protocol Integer ID: 14548

Keywords: bioinformatics, biology, bash, compbio, Python, RNA, bioinformatics software, using cygwin, ubuntu on window, windows 10 machine, computational biology with the fall creator, linux, ubuntu, cygwin, computational biology, window, binaries of software, installing binary, software, windows app store, python

Abstract

With the Fall Creators update to **Windows 10** comes the ability for all to install **Ubuntu** on your Windows 10 machine by simply downloading the app from the official **Windows App Store**. Rather than using a tool like **Cygwin**, which replicates a **UNIX(-like)** experience with some of the same command line tools, **Ubuntu on Windows** is the full **Linux** experience. Unlike using Cygwin, you can install bioinformatics software like **Salmon** or **Cufflinks** with ease. This protocol from **Bad Grammar, Good Syntax** provides a detailed method of how to install Ubuntu, setting up Python, and installing binaries of software for compbio.

Guidelines

In the past, you could try to dual boot your system so that it has two operating systems or run Ubuntu in a virtual machine within Windows, but I think this method is nice and clean. Others will disagree.

What you won't get is the Ubuntu graphical user interface. You are still stuck with running Windows 10. If that is a problem for you, then dual booting or a virtual machine is probably your best option. Instead, what Ubuntu on Windows offers is a simple way to get access to the command line from Ubuntu. Not only does that mean that you can run software designed and compiled to run on Linux, but you can also use all the cool tools of Ubuntu like apt-get to seamlessly install programs, keep them up-to-date, and fetch dependences.

An alternative guide to Linux on Windows using Conda:

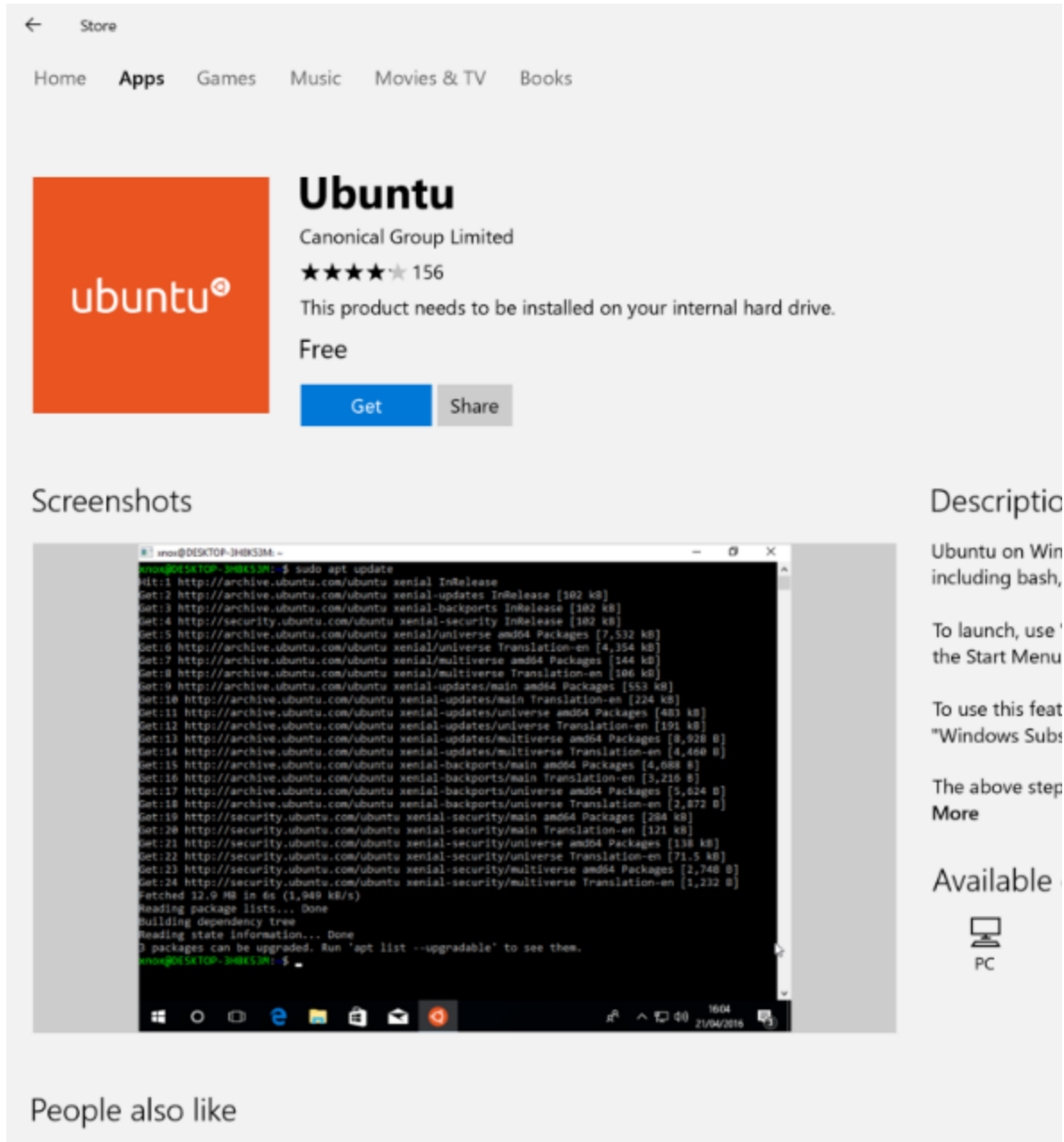
<https://github.com/kapsakcj/win10-linux-conda-how-to/blob/master/README.md>

Before start

You need a computer with Windows 10 installed and kept up to date.

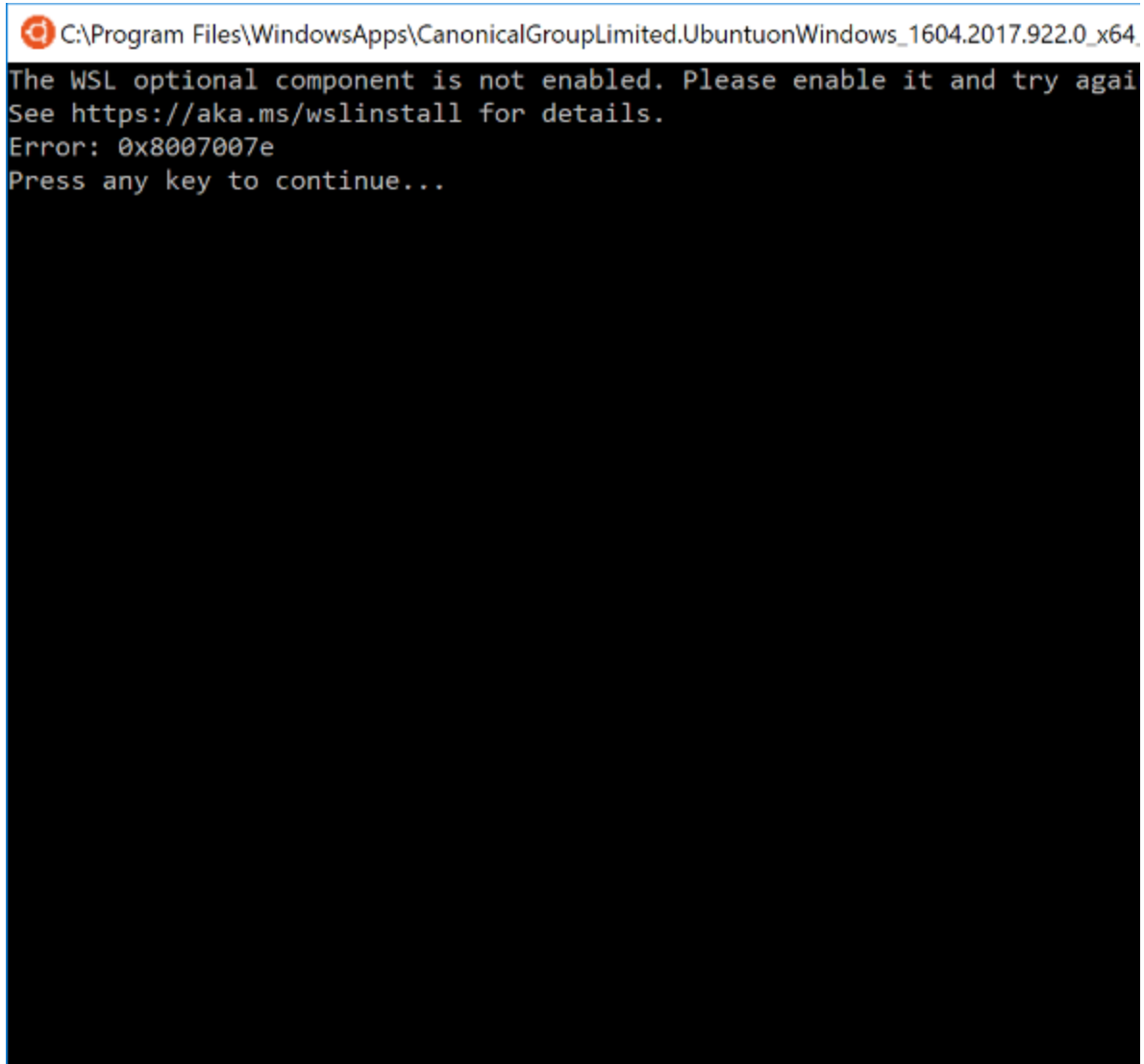
Installing Ubuntu

- 1 Ensure your Windows10 has the 2018 Fall Creators update (version 1709 or greater).
- 2 Go to the Windows App store and download Ubuntu app:



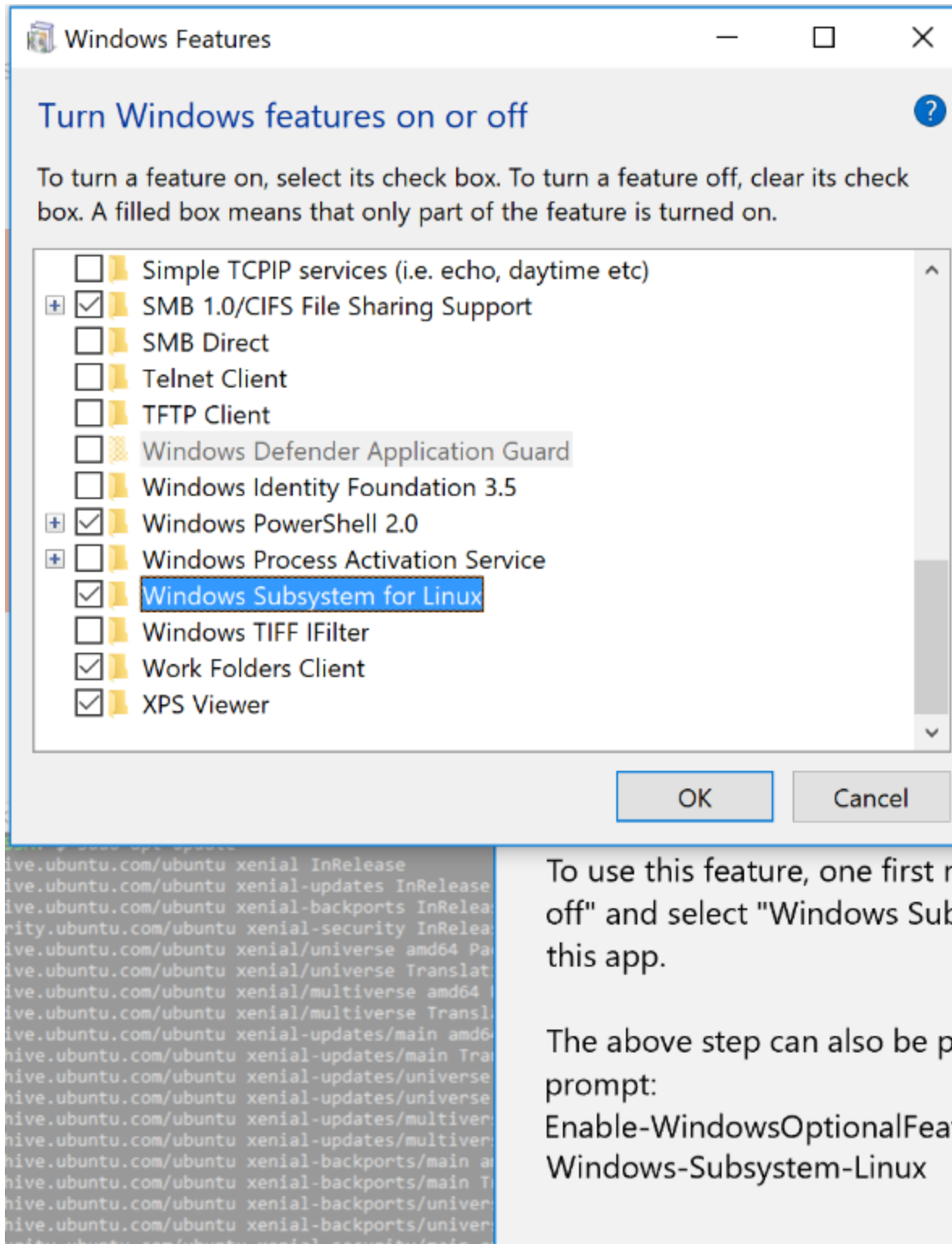
The screenshot shows the Windows App Store interface. At the top, there's a navigation bar with 'Store', 'Home', 'Apps', 'Games', 'Music', 'Movies & TV', and 'Books'. The main content area features the Ubuntu app card. The card has an orange square icon with the 'ubuntu' logo. To the right of the icon, the text reads 'Ubuntu', 'Canonical Group Limited', a 4.5-star rating with 156 reviews, and a note: 'This product needs to be installed on your internal hard drive.' Below this, it says 'Free' and has 'Get' and 'Share' buttons. Under the app card, there's a 'Screenshots' section showing a terminal window with the command 'sudo apt update' and its output. To the right of the screenshots is a 'Description' section with text about installing Ubuntu on Windows, including instructions to use the Start Menu and Windows Subsystem for Linux. Below the description is an 'Available' section with a PC icon.

- 3 Once the app is installed, open it up. You will see this terminal appear:



C:\Program Files\WindowsApps\CanonicalGroupLimited.UbuntuonWindows_1604.2017.922.0_x64,
The WSL optional component is not enabled. Please enable it and try again.
See <https://aka.ms/wslinstall> for details.
Error: 0x8007007e
Press any key to continue...

- 4 You need to enable an option in Windows first. Open the Start menu on Windows and start typing 'Windows Features' until a program called **Turn Windows Features on or off** appears. Click on it and find 'Windows Subsystems for Linux', then enable it (tick the box):



5 Restart your PC:

←  Windows Features

Windows completed the requested changes.

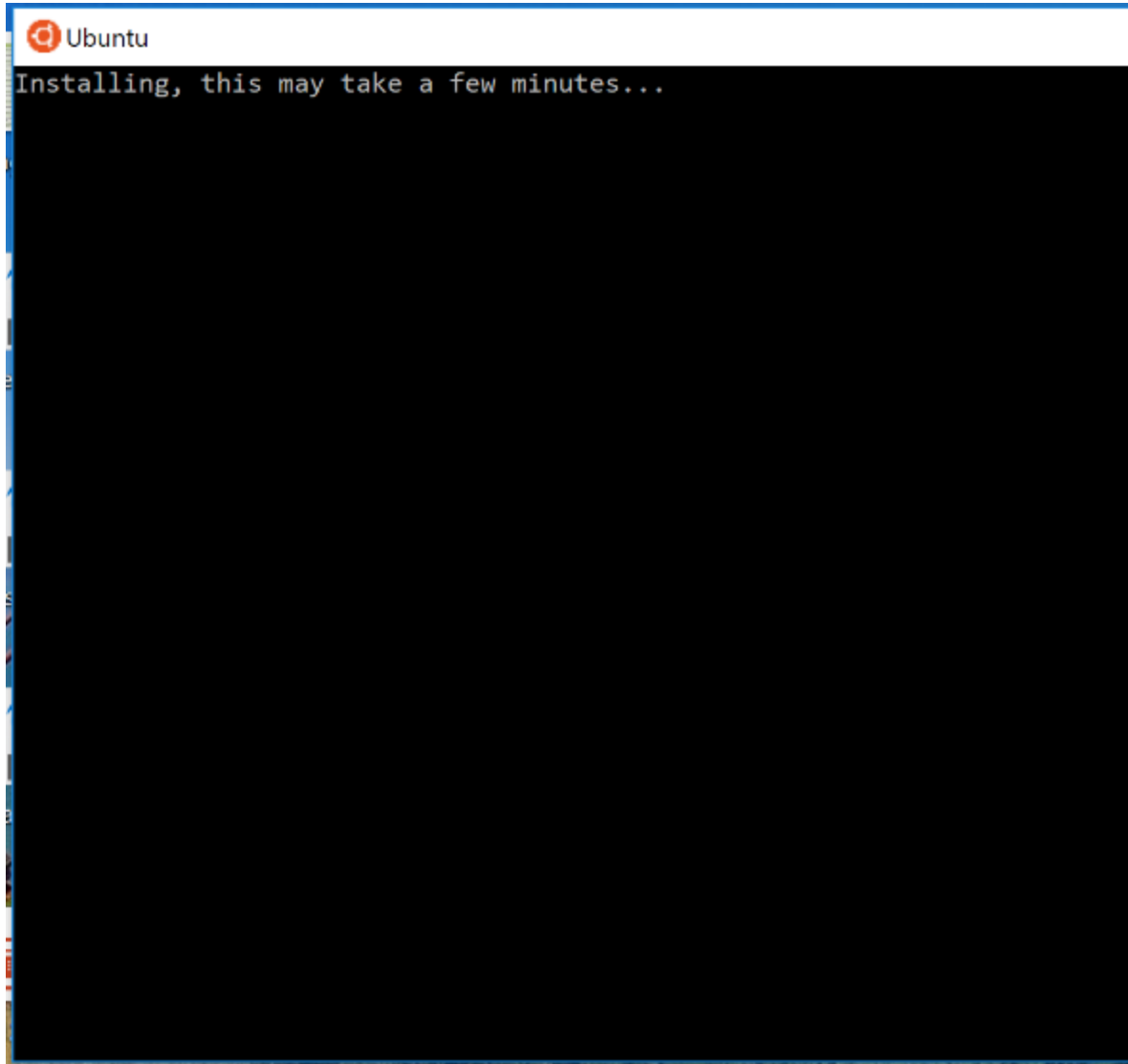
Windows needs to reboot your PC to finish installing the requested changes.

Restart now

```
et:9 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64-
et:10 http://archive.ubuntu.com/ubuntu xenial-updates/main Tra
et:11 http://archive.ubuntu.com/ubuntu xenial-updates/universe
et:12 http://archive.ubuntu.com/ubuntu xenial-updates/universe
et:13 http://archive.ubuntu.com/ubuntu xenial-updates/multiver
et:14 http://archive.ubuntu.com/ubuntu xenial-updates/multiver
et:15 http://archive.ubuntu.com/ubuntu xenial-backports/main a
et:16 http://archive.ubuntu.com/ubuntu xenial-backports/main T
et:17 http://archive.ubuntu.com/ubuntu xenial-backports/univer
et:18 http://archive.ubuntu.com/ubuntu xenial-backports/univer
et:19 http://security.ubuntu.com/ubuntu xenial-security/main a
```

The above step can also
prompt:
Enable-WindowsOptiona
Windows-Subsystem-Lin

- 6 Now try opening the Ubuntu app again. This time it should take a few moments to install itself:



- 7 Once it has finished installing, it will ask for your to select a username and password. These are independent from your Windows username and password. Select a username and passphrase that you will remember (or store in a password manager) and then enter them. Then you can check the exact version you are using with this command:

Command

```
$ lsb_release -a
```



Note



jamesl@DESKTOP-K2H5OME: ~

```
Installing, this may take a few minutes  
Installation successful!
```

```
Please create a default UNIX user account  
For more information visit: https://ubuntu.com/server/docs/installing-ubuntu-server
```

```
Enter new UNIX username: jamesl
```

```
Enter new UNIX password:
```

```
Retype new UNIX password:
```

```
passwd: password updated successfully
```

```
Default UNIX user set to: jamesl
```

```
To run a command as administrator (use sudo), see the man page for sudo.
```

```
See "man sudo_root" for details.
```

```
jamesl@DESKTOP-K2H5OME:~$ lsb_release
```

```
No LSB modules are available.
```

```
Distributor ID: Ubuntu
```

```
Description:    Ubuntu 16.04.3 LTS
```

```
Release:        16.04
```

```
Codename:       xenial
```

```
jamesl@DESKTOP-K2H5OME:~$
```



- 8 Now you are ready to start using your Ubuntu on Windows.

Note

One thing that you might notice is that you cannot do Ctrl+C or Ctrl+P for copy/paste. This is a pain but to copy text that you have highlighted in the Ubuntu app, you can right-click with your mouse. When you have no text highlighted, right-click will then paste what you have copied.

- 9 Right now, you are in a special drive for your Ubuntu files that are separate from your normal storage drive. To access your normal files (what is normally the C drive), go to where they are mounted at:

Command

```
$ /mnt/c
```

Note

This is where you wish you had never used spaces in file/folder names, as bash does not handle spaces in filenames well. If this is a big issue for you, I would suggest going back and renaming files/folders you plan to access from the command line.

- 10 Now update the Ubuntu OS that is running.

Command

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Note

While my version number did not increase, doing this cleared up an issue I was having with some other software I was trying to install - so I advise that you do update it now with the two commands above.

If you are new to using a UNIX(-like) system, sudo makes you a superuser: you are able to change important things in your system you cannot do normally. It will ask for your password you set Ubuntu up with.

Setting up Python

- 11 If you use a lot of **Python** in your work, it is rather simple to get Python running on Ubuntu. There are many ways to install Python. One way would be to use Ubuntu's apt-get. Another way is to use Conda. Each of these methods installs Python in a different environment, so just be aware. Ubuntu comes with an install of Python 3 out of the box

Command

```
$ which python3
/usr/bin/python3
$ python3 --version
Python 3.5.2
```

- 12 Given that a lot of the software and code used is from Python version 2.7, it is worth having a legacy version around. To install Python 2.7, use apt-get like so:

**Command**

```
$ sudo apt install python
```

Note

This will ask for the password that you set Ubuntu up with.

- 13 Now you have Python 2.7 (which you can call with python or python2 as opposed to python3):

Command

```
$ which python
/usr/bin/python
$ python --version
Python 2.7.12
```

- 14 With advice from [here](#), install these very useful tools for Python on Ubuntu:

Command

```
$ sudo apt-get install build-essential
$ sudo apt-get install python-dev
$ sudo apt-get install python3-dev
```

- 15 Now for something really important, to install **pip**. pip is the official package manager for Python modules.

Command

```
$ sudo apt-get install python-pip  
$ sudo apt-get install python3-pip
```

Note

It makes installing new packages for Python SOOOO much easier in the long run.

The issue with apt-get is that while it deals with all of the dependency issues (which is really important), it usually installs a really out of date version of the python package. In constrast, pip doesn't deal with the dependencies but it does install the latest version of the package. So I suggest initally installing with apt-get and then updating with pip (see below).

- 16 Now to install pandas using apt-get:

Command

```
$ sudo apt-get install python-pandas  
$ sudo apt-get install python3-pandas
```

Note

By installing pandas with apt-get, we also install many other important Python packages like numpy and matplotlib as dependances, which is a massive timesaver, compared to going after each on individually.

- 17 The latest scipy version is 1.0.0. If you look at the version of scipy that has been installed, you can see that we are well out of date:

Command

```
$ pip show scipy
Name: scipy
Version: 0.17.0
Summary: SciPy: Scientific Library for Python
Home-page: http://www.scipy.org
Author: SciPy Developers
Author-email: scipy-dev@scipy.org
License: BSD
Location: /usr/lib/python2.7/dist-packages
Requires:
```

- 18 You can update scipy from version 0.17.0 to 1.0.0 with pip:

Command

```
$ sudo pip install scipy --upgrade
```

- 19 Now you have the latest version of scipy:

Command

```
$ pip show scipy
Name: scipy
Version: 1.0.0
Summary: SciPy: Scientific Library for Python
Home-page: https://www.scipy.org
Author: SciPy Developers
Author-email: scipy-dev@python.org
License: BSD
Location: /usr/local/lib/python2.7/dist-packages
Requires: numpy
```

- 20 This is great, but beware that the scipy for Python3 has NOT been updated! So do not forget to do that one too!

Command

```
$ sudo pip3 install scipy --upgrade
```

Installing binaries of software for compbio

- 21 There are a number of different programs out there for computational biologists that are not going to be downloadable with apt-get. With these, you might need to download the source code and compile the software yourself. Or you can install them with Conda/bioconda (highly recommended when possible) or you can download a pre-compiled binary. You can download a binary and unzip it like so:

**Command**

```
$ cd /usr/local/bin
$ sudo wget https://github.com/COMBINE-
lab/salmon/releases/download/v0.8.2/Salmon-0.8.2_linux_x86_64.tar.gz
$ sudo tar -xvzf Salmon-0.8.2_linux_x86_64.tar.gz
```

Note

Luckily, many developers of these tools put out binaries for systems (Mac, Linux, and sometimes even Windows). A binary is a pre-compiled file that you can call to run the program. So here I will go through the simple task of downloading the latest binary for the RNA-seq quantification software **Salmon**. But a similar approach should work for other programs available as binaries.

- 22 To run Salmon, simply enter the path to the binary in the terminal.

Command

```
/usr/local/bin/Salmon-0.8.2_linux_x86_64/bin/salmon
Salmon v0.8.2
Usage: salmon -h|--help or salmon -v|--version or salmon -c|--cite or
salmon [--no-version-check] <COMMAND> [-h | options]
Commands: cite Show salmon citation information index Create a salmon
index quant Quantify a sample swim Perform super-secret operation
```

- 23 Now you can run Salmon on your Windows PC (through Ubuntu)!



Note

The issue here is having to remember this long path every time you wanted to run Salmon. You could modify your `bashrc` to add Salmon to your path. But one advantage of entering the path every time, is that you know exactly which version of Salmon (or any software) you are calling every time you use it.

Note

Hopefully now you can start to perform some of your work on your local (Windows) machine but with the comfort of having all of the tools for Linux at your fingertips.