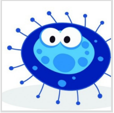


Sep 24, 2019

Use IPython to run bioconda tools in jupyter V.4

Alise J. Ponsero[1]

[1]University of Arizona

1 Works for me dx.doi.org/10.17504/protocols.io.7mjhk4n

Metagenomics class 2019

Alise J. Ponsero
University of Arizona

ABSTRACT

Learn how to run shell commands directly in your jupyter notebook using IPython

BEFORE STARTING

For this protocol, you'll need Anaconda/miniconda and jupyter installed on your machine.

Looking at bioconda

1 Meet Bioconda. Bioconda is a project aiming to integrate bioinformatic tools in the conda package management system. They have over 6000 bioinformatic tools available and a repository of recipe on how to use them.

Yes, Bioconda is your new best friend.

You can browse for your tool of interest in Bioconda through their tool repository. For this tutorial we will install and run vsearch.

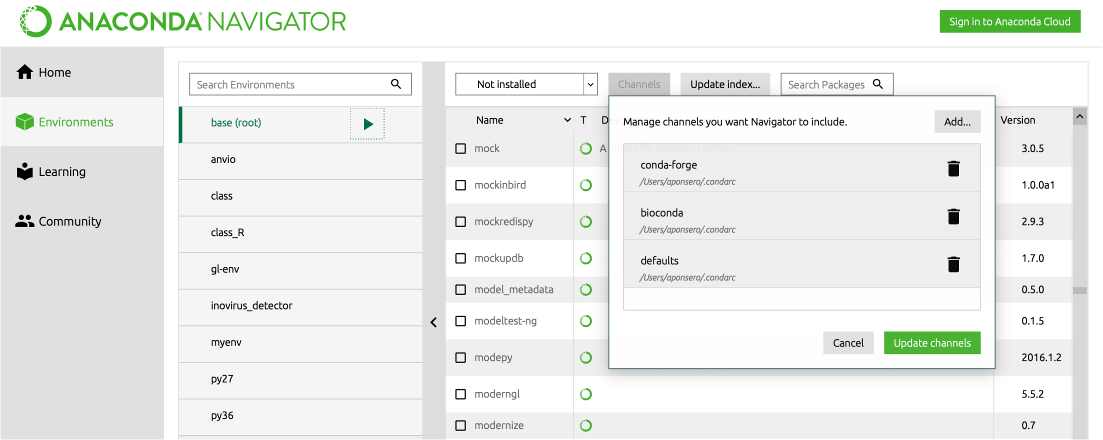
installing vsearch

2 Open Anaconda Navigator and go to your environment manager. Select the class environment and search for the vsearch package in the "not installed" packages list.

Install vsearch in the environment. If you don't remember how to do this. Do not panic and go back to the Anaconda protocol (https://www.protocols.io/view/first-steps-using-a-jupyter-notebook-6uphevn).

The installation will take some time, therefore you can read this very serious paper about Inuit poop knives.

If you don't find vsearch in the "not installed" or the "all" list of packages in the conda navigator look at your channels in the conda navigator :



Manage channels you want Navigator to include.

conda-forge
/Users/aponsero/.condarc

bioconda
/Users/aponsero/.condarc

defaults
/Users/aponsero/.condarc

Name	Version
☐ mock	3.0.5
☐ mockinbird	1.0.0a1
☐ mockredispy	2.9.3
☐ mockupdb	1.7.0
☐ model_metadata	0.5.0
☐ modeltest-ng	0.1.5
☐ modepy	2016.1.2
☐ moderngl	5.5.2
☐ modernize	0.7

If you don't see these three channels, go to step 2.2

2.1 For Miniconda users, open your terminal and type :

```
source activate class_2019
```

You should have your environment correctly loaded. Then type :

```
conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
```

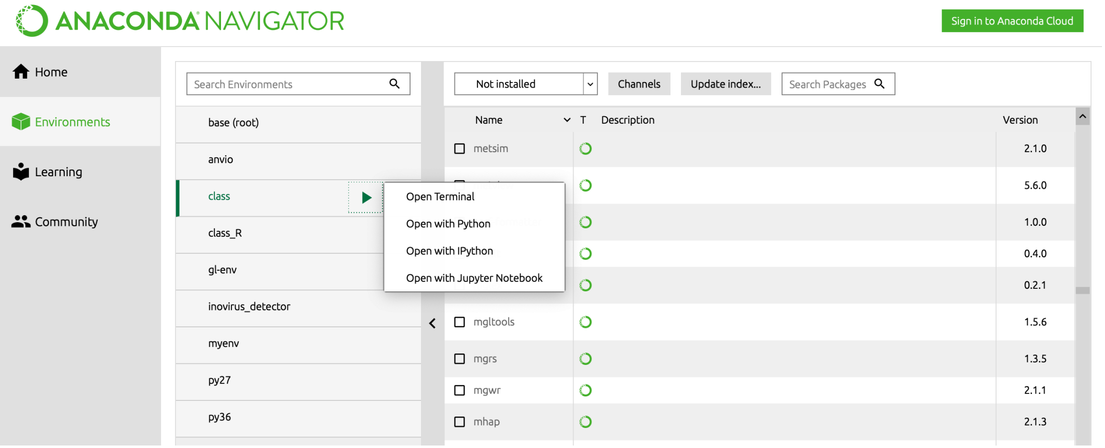
That should have set up your conda channels correctly. Now, let's install mothur.

```
conda install vsearch
```

Confirm the installation by typing "y" and wait patiently for the installation.

2.2 Let's fix your channels (these are the channels that conda look at when they look for packages to install).

In the conda navigator, load your "class_2019" environment, then click on the "triangle" logo and ask to open a terminal :



In the terminal type :

```
conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
```

then :

```
conda install vsearch
```

When the prompt asks you if you really want to install the packages, type "y". Once the install is complete, close the terminal and close the anaconda navigator. Reopen the anaconda navigator to have the changes taken correctly into account.

Using IPython

3 Is vsearch is installed in your conda environment? Great! Now open a Python jupyter notebook in your Document folder.

4 Think of IPython as a bridge between Python and shell commands. It's basically allowing you to run shell command lines directly into your Python Notebook ! Are you excited? Ok, let's try it!

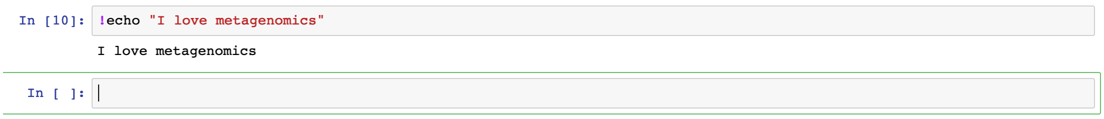
As you (should) know, in shell you can print a text using the "echo" command. Like this :

```
echo "I love metagenomics"
```

Well, let's say you REALLY want to use the "echo" command in your notebook, then IPython allows you to just type :

```
!echo "I love metagenomics"
```

Now, try it and run the cell ! You should see



```
In [10]: !echo "I love metagenomics"
         I love metagenomics

In [ ]: 
```

And the same idea applies to ls, pwd, mkdir... all your favorite commands are there! All? Well, almost all...

5 If you play with IPython long enough, you'll quickly notice that !cd doesn't exactly work as it is supposed to. If you try to run

```
!cd ..
```

You won't see an error, but you won't have changed directory either... This is because each cell from the notebook is run as a temporary sub-shell. It opened a shell, ran the cd .. and then closed it. So you're back to square one.

To run these commands, you need to use a % instead of the !

```
%cd ..
```

Now, it should work ! And even better, if you have the "automagic" IPython function installed, you don't need to use the "%". You can simply run :

```
cd ..
```

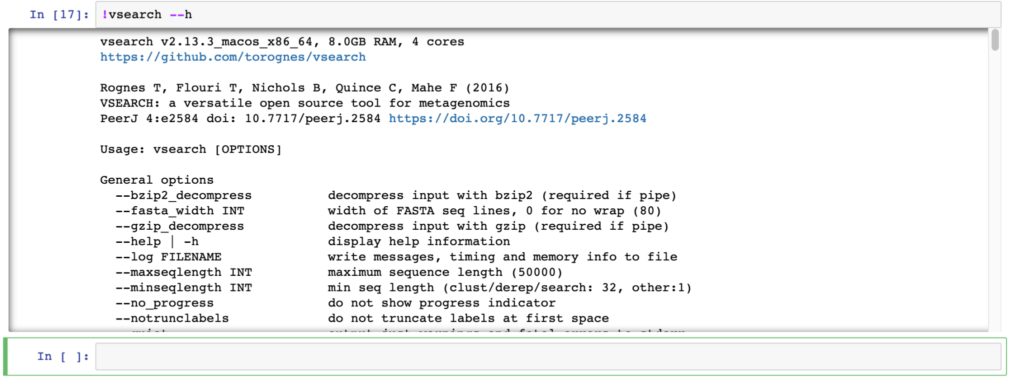
You have automagic functions for the main shell commands (ls, cd, cat, man, mkdir, mv, pwd, rm and rmdir).

Running vsearch

6 So let's use IPython to run vsearch.

```
!vsearch --h
```

You should see the following output:



```
In [17]: !vsearch --h

         vsearch v2.13.3_macos_x86_64, 8.0GB RAM, 4 cores
         https://github.com/torognes/vsearch

         Rognes T, Flouri T, Nichols B, Quince C, Mahe F (2016)
         VSEARCH: a versatile open source tool for metagenomics
         PeerJ 4:e2584 doi: 10.7717/peerj.2584 https://doi.org/10.7717/peerj.2584

         Usage: vsearch [OPTIONS]

         General options
           --bzip2_decompress          decompress input with bzip2 (required if pipe)
           --fasta_width INT           width of FASTA seq lines, 0 for no wrap (80)
           --gzip_decompress           decompress input with gzip (required if pipe)
           --help | -h                 display help information
           --log FILENAME              write messages, timing and memory info to file
           --maxseqlength INT          maximum sequence length (50000)
           --minseqlength INT          min seq length (clust/derep/search: 32, other:1)
           --no_progress               do not show progress indicator
           --notrunclabels             do not truncate labels at first space

In [ ]:
```

If the command didn't run and gave back an error saying that vsearch isn't found, go to step 7.

7 If the previous step command didn't work, you need to find where conda installs your packages. To do so, mac users can right-click on the Anaconda icon in the Application folder and open the "get info" panel. Under "original" you see the path to the anaconda3 installation folder. For windows users, right-click on the Anaconda navigator shortcut and "Open file location". It should open a new windows to the folder where Anaconda3 is installed.

In my case the path to this folder looked like this : '/Users/[USERNAME]/anaconda3/envs/class_2019/bin'

Once you know where is you bin folder, you can run vsearch using the command (here showing the output of the help command) :

```
!/Users/[USERNAME]/anaconda3/envs/class_2019/bin/vsearch --h
```

The above command can become very long and painful to write. It would be awesome to be able to save "/Users/[USERNAME]/anaconda3/envs/class_2019/bin" in a variable called "bin_dir", right? Well, let's do this !

First, let's create the variable:

```
bin_dir="/Users/[USERNAME]/anaconda3/envs/class_2019/bin"
```

Now, we can to pass this variable in our IPython command using the {variable} syntax:

```
!{bin_dir}/vsearch --h
```