

Accessing modules across editors

Programming for Biologists

February 17, 2015

The Problem

The purpose of installing Anaconda for this class was to simplify the installation of modules for Python. The package installation process now requires only a single line of code in the console to immediately grant you access to a new set of classes and functions! However, these modules are only accessible through Anaconda's default editor Spyder. Anyone wishing to switch back to IDLE for some programming got an unpleasant surprise when they were unable to import the module they had just been using successfully on Spyder.

```
>>> import Bio
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    import Bio
ImportError: No module named Bio
>>>
```

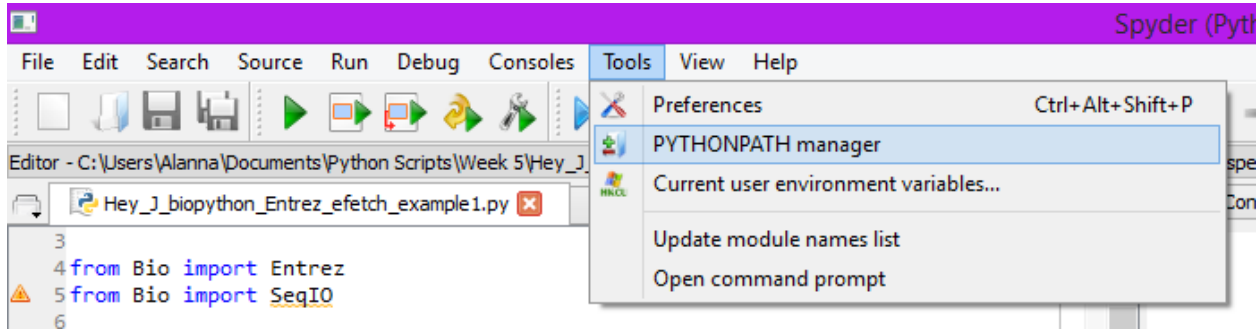
What causes this error when you know very well that the module BioPython is installed on this very machine? The problem is that IDLE doesn't know where to look for it. When Anaconda installs modules, it installs them in the Anaconda directory. You can see from the `sys.path` command that IDLE only knows to look in the standard Python directories.

```
>>>
>>> import sys
>>> sys.path
['', 'C:\\Python27\\Lib\\idlelib', 'C:\\Python27\\lib\\site-packages\\setuptools-11.3.1-py2.7.egg', 'C:\\windows\\SYSTEM32\\python27.zip', 'C:\\Python27\\DLLs', 'C:\\Python27\\lib', 'C:\\Python27\\lib\\plat-win', 'C:\\Python27\\lib\\lib-tk', 'C:\\Python27', 'C:\\Python27\\lib\\site-packages']
>>>
```

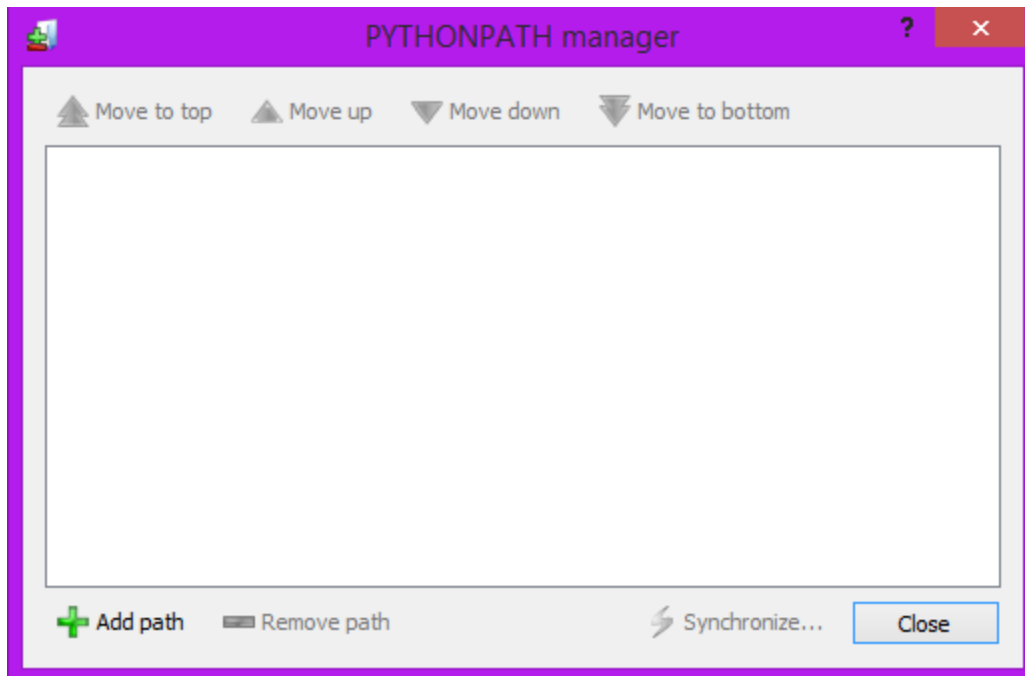
When IDLE is looking for the package you're trying to `import`, it only knows to look in these directories. Since the object `sys.path` returns is a list, you can add items to the list with `.append()` as usual and tell IDLE to check another location for the module. However, this is only a temporary fix that would disappear as soon as you restart the shell. It's probably not worth it to type it every time you open IDLE.

Windows Solution

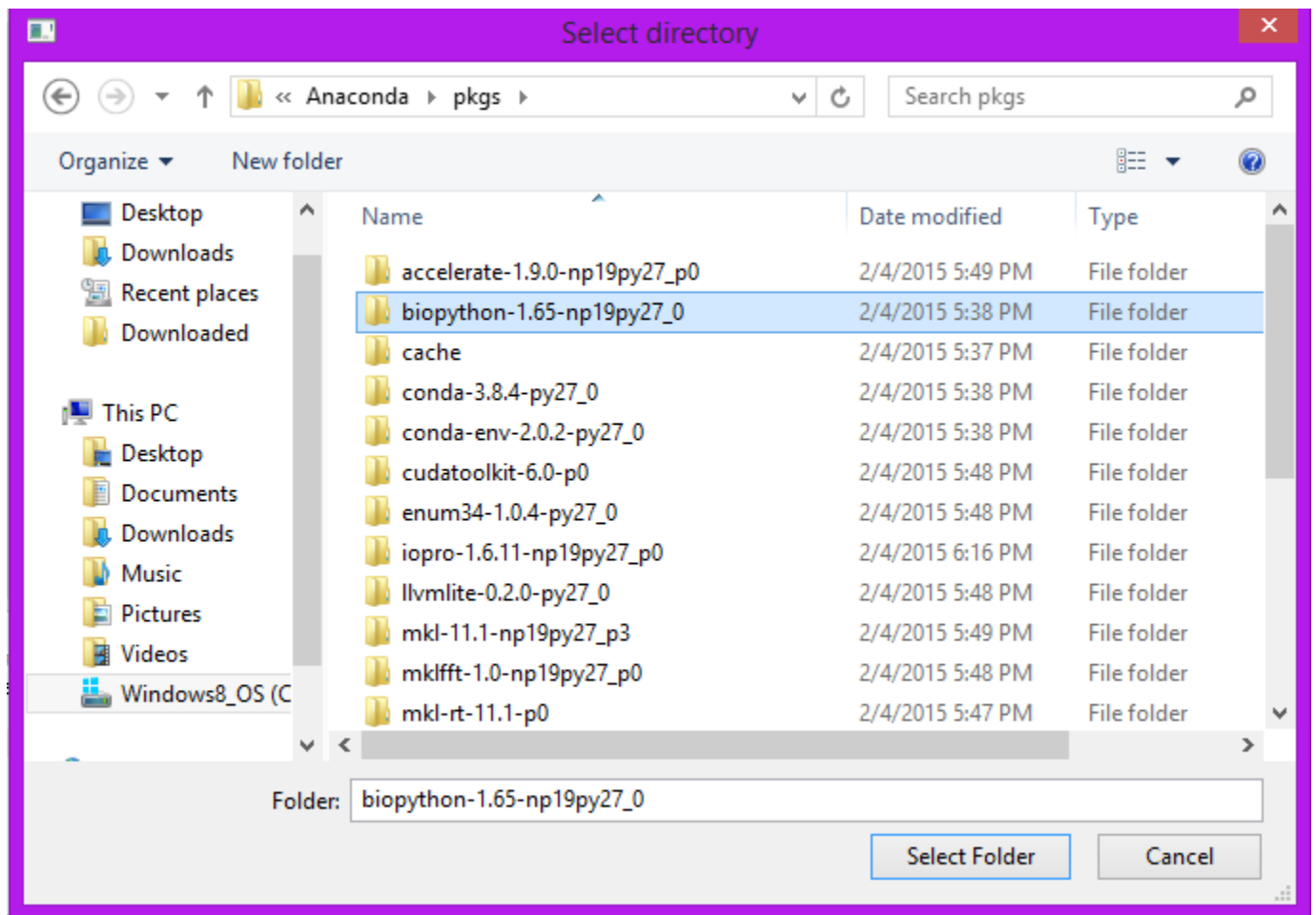
If you are using Anaconda on **Windows**, there is a feature included that allows you to fix this problem permanently. First go to the Tools menu in Spyder and click PYTHONPATH manager.



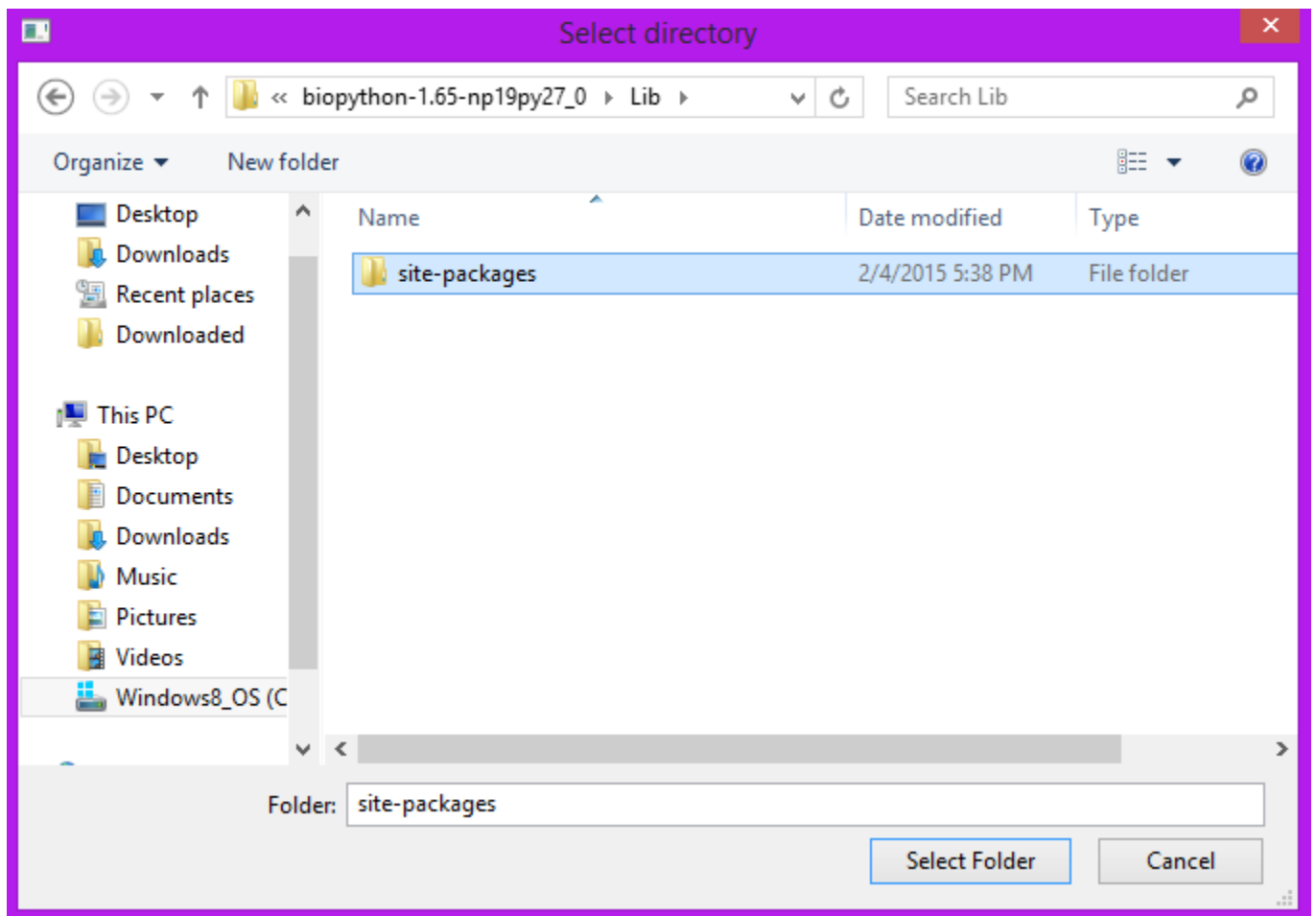
Once the PYTHONPATH manager dialog box appears, click the green plus sign to add a path.



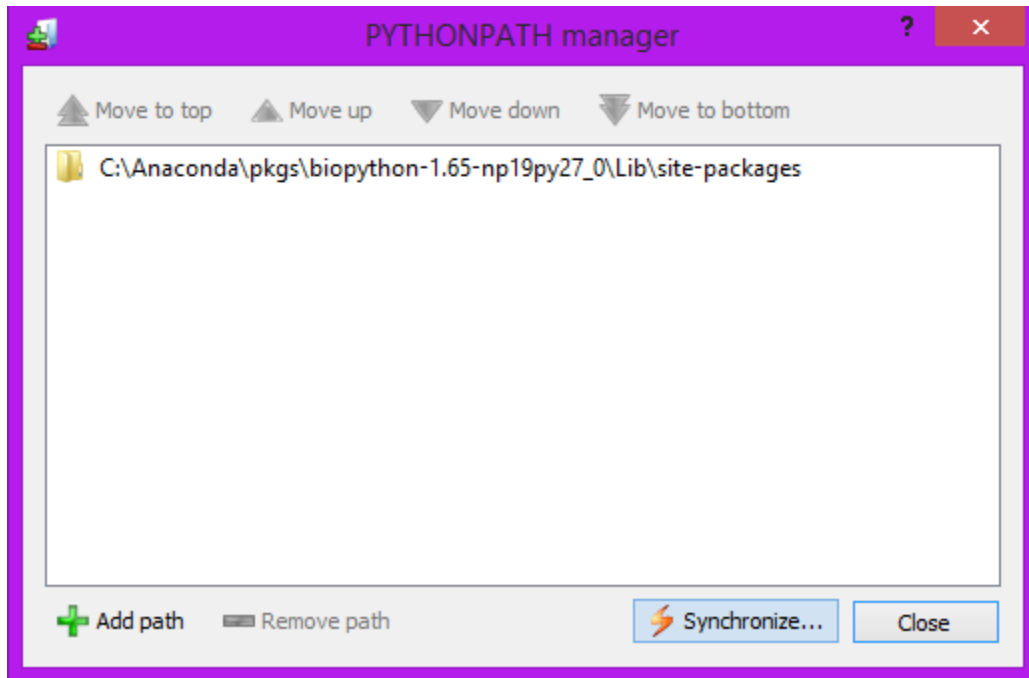
Now the PYTHONPATH manager will open a file explorer window that allows you to browse to the folder you want to add as a PYTHONPATH. In this case, we need to add the folder where Anaconda installed BioPython. If you used the default installation settings on Windows, all of the modules should be installed in the folder `C:/Anaconda/pkg/`.



However, this path isn't specific enough to help IDLE find BioPython. You need to go deeper and select the `site-packages` folder that is several levels within the main BioPython folder.



After selecting the `site-packages` folder and clicking `Select Folder`, the file explorer box will close and your new path will appear in the PYTHONPATH manager. The last step you need to take is to hit `Synchronize` on the dialog box to share this path with all other Python editors installed on your computer. It will ask you if you're sure, so respond "Yes" if you trust me.



Now when you open up IDLE again, the `sys.path` will be altered to reflect this change!

```
>>> import sys
>>> sys.path
['', 'C:\\Python27\\Lib\\idlelib', 'C:\\Python27\\lib\\site-packages\\setuptools-11.3.1-py2.7.egg', 'C:\\Anaconda\\pkgs\\biopython-1.65-np19py27_0\\Lib\\site-packages', 'C:\\windows\\SYSTEM32\\python27.zip', 'C:\\Python27\\DLLs', 'C:\\Python27\\lib', 'C:\\Python27\\lib\\plat-win', 'C:\\Python27\\lib\\lib-tk', 'C:\\Python27', 'C:\\Python27\\lib\\site-packages']
>>>
```

And since IDLE now has a new path to check for modules, it will be able to find BioPython if you've installed it through Anaconda.

```
>>>
>>> import Bio
>>>
```

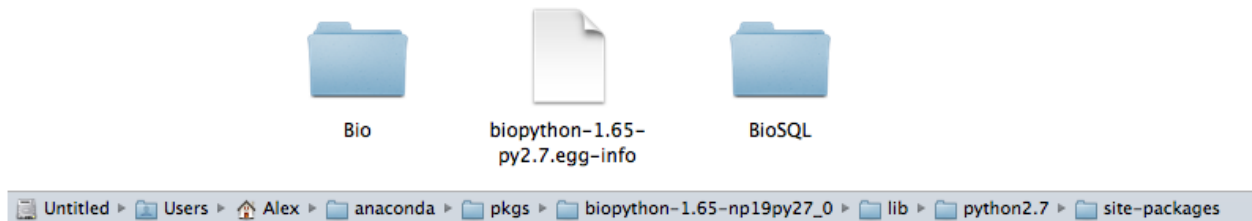
No red text = success!

Now, if you are a Mac user, you could follow along with all of these steps in the PYTHONPATH manager (found under the "python" menu in the Mac version of Spyder), but the Synchronize button is lacking on Macs. So this allows you to change where Spyder looks for modules without being able to change IDLE's `sys.path`.

Mac Workarounds

If you can't change where IDLE is looking for the modules, you can change the location of the modules instead! In a primitive workaround to this problem, one solution is to simply copy the `site-packages` folder over to a location in your `sys.path`. (This solution also works for Windows!)

The standard Mac installation of Anaconda is located in your home directory on the computer (the folder in the Users directory that has the name of your Mac). The path within Anaconda is similar to that of the Windows installation. Navigate to `Users/YourName/anaconda/pkg/biopython-1.65-np19py27_0/lib/python2.7/site-packages`. Once in the `site-packages` folder, copy all files and folders you find (example below):



All this material in the `site-packages` folder is essentially what makes up the BioPython module. To make sure IDLE knows where to find it, we'll be copying it into another location named `site-packages`. Finding IDLE's module directory is even easier than finding Anaconda's, because it's in that list `sys.path` we looked at before!

```
>>> import sys
>>> sys.path
['', '/Users/Alex/Documents', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7.zip', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-darwin', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-mac', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/plat-mac/lib-scriptpackages', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-tk', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-old', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-dynload', '/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-packages', '/Library/Python/2.7/site-packages']
>>> |
```

Navigate to this Python `site-packages` folder now and paste the BioPython module data you copied from Anaconda. See the finished product below:



README



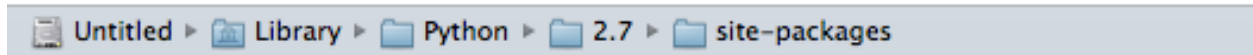
BioSQL



Bio



biopython-1.65-
py2.7.egg-info



Warning: Please remember to leave a copy of these files in the Anaconda directory as well. It may seem inefficient to duplicate the files, but I can't guarantee your computer's safety if you cut these BioPython files out of Anaconda.

Coming soon...

The drawback of these approaches is that you have to perform these actions for every module you want to use across editors. If you want to test a script that uses Scipy or Numpy in IDLE, you'll have to correct each module individually. So in the future there may be an edition of this guide that also includes how to get this done more quickly in the command line. More importantly, there will be a guide coming soon of how to install modules directly in the command line (without Anaconda) that avoids this entire problem!