**Installing Software Using Python and R**

## Python

Python will install software packages into your /.local/bin/ directory and keep the software libraries in /.local/lib/. To call packages, add your /.local/bin/ to your .bashrc file under your environmental variables and refresh the terminal; alternatively, set this command before you start using your python packages.

export PATH=/home/cmstaley/username/.local/bin/.:$PATH

### PyPI

*PyPI* is the python package index – a repository where software packages for python are stored. PyPI packages can be installed using pip (make sure python is loaded):

module load python3[[1]](#footnote-1)

pip install *package* –-user

The –user flag directs the installation to the /.local/ directory as described above. Software dependencies[[2]](#footnote-2) will usually be installed with the package that needs them, but these installs may fail due to discrepancies in versions or the need for second-tier dependencies (*i.e.*,the dependency has a dependency). It may be necessary to manually install some dependencies.

### Conda

Conda is a more intuitive way to navigate python installs. You can also create separate environments so that python packages don’t conflict with other packages or your general computational environment. To use it similarly to the PyPI install,

conda install *package* -c *biorepository*

Conda has multiple biorepositories it can use to try to find all of the dependencies that correspond to your install. You can add these to the conda environment using the command below; this only needs to be done once. Software will still be installed in your /.local/ directory.

conda config --add channels

Available biorepositories include *biobakery, bioconda,* and *conda-forge*.

To create a conda environment, use the command below. You can also install packages and specify versions of software you want the environment to have. Packages can also be installed after the environment is created by activating the environment and installing new software into the environment.

conda create -n *env* python=3.9[[3]](#footnote-3) numpy pandas

source activate *env*

conda install *package[[4]](#footnote-4)*

|  |  |
| --- | --- |
| conda info --envs | List conda environments |
| conda update –-all -n *env* | Update all packages in an environment |
| conda update *package=ver* | Update a package (versions can be specified directly or via logic arguments *e.g.,* >2.5) |
| conda remove -n *env* --all | Remove an environment and all packages |
| source deactivate | Get out of your current conda environment |

See more conda commands in conda-cheatsheet.pdf in the Sequencing Files > Instructions folder on Box.

### Tips

* The safest thing to do would be to create separate conda environments for difficult or “major” software packages (*e.g.,* MetaPhlAn4.1 and HUMAnN3.9) and update those environments when needed.
* When trying to install software in a conda environment, watch the outputs that report the success or failure of the install and *check the filepaths*. Make sure the software is installing in the /.conda/envs/ filepath. If not, it’s probably putting it in your /.local/bin/ and your conda *environment* is a literal joke. This is a notorious issue with the MetaPhlAn4.1/HUMAnN3.9 install.

# R

R is it’s own whole other thing from python. The two do not get along well. You can’t write to the normal R module folder and will need to create a local directory to put your packages in - /R/. Installing packages across different versions of R is a challenge because things have moved quickly over the last decade. The “latest” version of R appears pretty stable at 4.3. It’s not part of MSI’s mainstream R suite, though, so the module load looks a little different.

module load R/4.3.0-openblas[[5]](#footnote-5)

R

If you want to be precious about it:

module load rstudio

rstudio[[6]](#footnote-6)

Use Bioconductor to manage your R packages.

if (!require("BiocManager", quietly = TRUE))

 install.packages("BiocManager")

BiocManager::install(version = "3.18")

Install and manage your packages; this command will also update all packages. Use the checkBuilt flag to make sure your R library maintains compatibility with itself. Versions can also be specified similar to python.

BiocManager::install(*package*, checkBuilt=TRUE)

BiocManager::install(checkBuilt=TRUE)

If you want to try recompiling your installed packages because you need to do a drastic update:

if (!require("BiocManager", quietly = TRUE))

 install.packages("BiocManager")

pkgs <- rownames(installed.packages())

BiocManager::install(pkgs, type = "source", checkBuilt = TRUE)

Some packages are only available on github and don’t have CRAN (R’s package repository) or Bioconductor packages. For that, we have devtools.

install.packages(“devtools”)[[7]](#footnote-7)

library(devtools)

devtools::install\_github(‘*gitacct/pkg*’)

### Useful R Packages

|  |  |  |
| --- | --- | --- |
| Package | Install | Description |
| [SplinectomeR](https://github.com/RRShieldsCutler/splinectomeR) | devtools::install\_github ('RRShieldsCutler/splinectomeR') | Longitudinal analysis of microbiome data |
| [Phyloseq](https://joey711.github.io/phyloseq/) | BiocManager::install(phylsoeq, checkBuilt=TRUE) | Used to manipulate data from mothur to transfer to other packages |
| [MetaPhlAn4.1](https://github.com/biobakery/MetaPhlAn/tree/master) Diversity Calc | Rscript calculate\_diversity.R(from the MetaPhlAn4.1 python install) | Installs all of the dependencies so you can use this script. You must be in the directory with the script |
| [MMUPHin](https://bioconductor.org/packages/release/bioc/vignettes/MMUPHin/inst/doc/MMUPHin.html) | BiocManager::install(MMUPHin, checkBuilt=TRUE) | Renormalizes OTU/ASV counts to account for batch effects |
| [MaAsLin 2.0](https://huttenhower.sph.harvard.edu/maaslin) | BiocManager::install(Maaslin2, checkBuilt=TRUE) | Multivariate statistical analysis of metagenomic data |
| [NetCoMi](https://github.com/stefpeschel/NetCoMi) | devtools::install\_github("stefpeschel/NetCoMi", dependencies = c("Depends","Imports","LinkingTo"),repos = c("https://cloud.r-project.org/",BiocManager::repositories())) | Co-occurrence network construction, comparison, and statistical analysis |
| [Covariance Matrix modeling](https://github.com/ajmolstad/SpPDCC) | devtools::install\_github (‘ajmolstad/SpPDCC’) | Uses a covariance matrix approach to create simple visual networks |

Find additional R and python packages for data analysis at the [bioBakery](https://github.com/biobakery/biobakery/wiki/) (or in the literature).

1. Python greatly interferes with the current version of R. It is Chris’s current recommendation to not have this set to load automatically in your .bashrc. [↑](#footnote-ref-1)
2. All of these software packages are built like houses of cards. A newer package may call on an older package to carry out a function it can accomplish instead recoding the function from the older package. [↑](#footnote-ref-2)
3. This is in no way an endorsement of python version 3.9. It’s an example. Check the documentation of your software to see if this is specified. [↑](#footnote-ref-3)
4. By now, you really should have set up your channels, but you could also set one here. [↑](#footnote-ref-4)
5. Chris has this load automatically from his .bashrc. At the present time, this has not presented a problem. [↑](#footnote-ref-5)
6. Requires an [interactive session](https://ondemand.msi.umn.edu/) with X11 forwarding for graphics. Log in using ssh -Y *server*. Chris has not verified that this works for R version 4.3. [↑](#footnote-ref-6)
7. This is how you install packages outside of Bioconductor. [↑](#footnote-ref-7)