
# Install:

## Analyzing microbial communities using high-throughput 16S rRNA sequencing data.

J. Gregory Caporaso[1]*, Justin Kuczynski[2]*, Jesse Stombaugh[1]*, Kyle Bittinger[3], Frederic D. Bushman[3], Elizabeth K. Costello[1], Noah Fierer[4], Antonio Gonzalez Peña[5], Julia K. Goodrich[5], Jeff I. Gordon[6], Gavin Huttley[7], Scott T. Kelley[8], Dan Knights[5], Jeremy E. Koenig[9], Ruth E. Ley[9], Cathy A. Lozupone[1], Daniel McDonald[1], Brian D. Muegge[6], Megan Pirrung[1], Jens Reeder[1], Joel R. Sevinsky[10], Peter J. Turnbaugh[6], Will Van Treuren[1], William A. Walters[2], Jeremy Widmann[1], Tanya Yatsunenko[6], Jesse Zaneveld[2] and Rob Knight[1,11**]

1. Department of Chemistry and Biochemistry, UCB 215, University of Colorado, Boulder, CO 80309
2. Department of Molecular, Cellular and Developmental Biology, UCB 347, University of Colorado, Boulder, CO 80309
3. Department of Microbiology, Johnson Pavilion 425, University of Pennsylvania, Philadelphia, PA 19104
4. Cooperative Institute for Research in Environmental Sciences, University of Colorado, Boulder, CO 80309, USA.; Department of Ecology and Evolutionary Biology, University of Colorado, Boulder, CO 80309, USA.
5. Department of Computer Science, University of Colorado, Boulder, Colorado, USA.
6. Center for Genome Sciences, Washington University School of Medicine, St. Louis, MO 63108
7. Computational Genomics Laboratory, John Curtin School of Medical Research, The Australian National University, Canberra, Australian Capital Territory, Australia.
8. Department of Biology, San Diego State University, San Diego CA 92182
9. Department of Microbiology, Cornell University, Ithaca NY 14853
10. Luca Technologies, 500 Corporate Circle, Suite C, Golden, Colorado 80401
11. Howard Hughes Medical Institute

# 1. Installing QIIME

To use this tutorial the user should install the following programs and set the appropriate environment variables, before downloading and installing QIIME.

## 1.1 Tutorial Software Dependencies

The following programs and datasets were used to generate this tutorial and it is recommended to use the same versions as shown below.

- Python 2.6
- PyCogent 1.4.0 or later with ReportLab
- PyNAST (from svn)
- Numpy 1.3
- MatPlotLib 0.98.5.2
- jre1.6.0_16
- rdp_classifier-2.0
- greengenes core set data file (http://greengenes.lbl.gov/Download/Sequence_Data/Fasta_data_files/core_set_aligned.fasta.imputed)
- greengenes alignment lanemask file (http://greengenes.lbl.gov/Download/Sequence_Data/lanemask_in_1s_and_0s)
- blast-2.2.22 (please note that this refers to legacy BLAST from NCBI, not BLAST+)
- fasttree 2.0
- cd-hit 3.1

## 1.2 Installing QIIME

### *1.2.1 Stable Pre-Release*

Currently the most stable version of QIIME is our 0.9 pre-release, which you can download from:

https://sourceforge.net/projects/qiime/files/releases/Qiime-0.9.tar.gz/download

### *1.2.2 Latest Development Version*

To get the latest development version of QIIME, you should check it out of our Sourceforge repository. While this code is subject to minor changes in interface, it will provide access to the latest and greatest features. The official web documentation is likely to be out-of-date with respect to the development software. You should instead refer to the svn documentation in Qiime/doc. Check out the latest version of QIIME using svn with the command:

```
svn co https://qiime.svn.sourceforge.net/svnroot/qiime/trunk Qiime
```

The user can update QIIME, by using the following command:

```
svn update /path/to/QIIME/
```

## 1.3 Environment Variables

Make sure the following environment variable are set:

### *1.3.1 PATH Environment Variable*

Your $PATH environment variable should contain the following:

- /path/to/python/bin
- /path/to/cd-hit-est
- /path/to/FastTree
- /path/to/blast-2.2.21/bin/

- /path/to/PyNAST/scripts/
- /path/to/jre1.6_0_16/

For all path description throughout this tutorial, the "/path/to/" refers to the physical location of each program on your local computer. For instance, the "/path/to/python/bin/" refers to "/Library/Frameworks/Python.framework/Versions/2.5/bin" on Mac OS X version 10.5.

In the bash shell, you can use the following command (example only shows the path to python, so other softwares can be added to the PATH, using a similar approach):

- export PATH=/path/to/python/bin/:$PATH

#### *1.3.2 PYTHONPATH Environment Variable*

Your PYTHONPATH should contain the following:

- /path/to/PyCogent
- /path/to/QIIME
- /path/to/PyNAST

In the bash shell, you can use the following command (example only shows the path to QIIME, so other softwares can be added to the PYTHONPATH, using a similar approach):

- export PYTHONPATH=/path/to/QIIME/:$PYTHONPATH

#### *1.3.3 RDP_JAR_PATH Environment Variable*

The user should also define an RDP_JAR_PATH variable, since this tutorial uses the RDP Classifier:

- /path/to/rdp_classifier-2.0.jar

In the bash shell, you can use the following command:

- export RDP_JAR_PATH=/path/to/rdp_classifier-2.0.jar

### 1.4 Testing QIIME Install

Once the source code is downloaded, the user should test QIIME to be sure all essential software is properly installed and the correct environment variables are set.

In a terminal window the user, should cd to their qiime/test directory using the following command:

```
cd /path/to/QIIME/tests/
```

Then run the following test command:

```
python all_tests.py -v
```

If all tests run properly, then QIIME was properly installed. Some test scripts may fail, due to optional third party applications not being installed. Test failures which contain the text ApplicationNotFoundError can be safely ignored. The following scripts may generate errors:

1. **test_align_seqs.py** - will fail if PyNAST, BLAST, muscle or infernal are not installed. PyNAST requires BLAST, and is the default sequence aligner. If you are not using PyNAST, you can safely ignore fails regarding BLAST and/or PyNAST. If you are not using muscle or infernal, you can safely disregard those error messages.
2. **test_pick_otus.py** – will fail if cd-hit and/or blast are not installed. If you are not planning to use these methods for OTU picking, you do not need to worry about these failures.

3. **test_pyronoise.py** - Will fail in PyroNoise is not installed.