Setting Up Your Workspace Environment

March 10, 2011

This document will help you to install and configure the Rosetta executables, database, scripts and tutorials necessary for this workshop. In this document, instructions are written in this font, and commands that should be entered into the terminal are written in `this font.`

Note for Windows Users

At this time, Rosetta 3.2 does not work properly on Windows systems. If you are using Windows at this workshop, We will provide you with the login for a linux based workstation. This workstation will already have the environment properly configured.

1 Copying the Thumbdrive

Create a new directory on your system called `workshop`, and copy the entire contents of the thumbdrive to that directory. Open up a new terminal, and cd into the directory you just created.

2 Extracting Rosetta Binaries and Database

If you already have a working copy of rosetta installed on your system, skip this section.

2.1 MacOS

Extract the rosetta executables and database by running the following two commands:

```
tar -xvzf rosetta_database.tar.gz
tar -xvzf macos.tar.gz
```

You will now have a directory called `macos` containing compiled Rosetta executables and a directory called `rosetta_database` containing the database

2.2 Linux

Extract the rosetta database by running the command:

```
tar -xvzf rosetta_database.tar.gz
```

If you are running 32 bit linux, extract the executables:

```
tar -xvzf linux_32.tar.gz
```

If you are running 64 bit linux, extract the executables:

```
tar -xvzf linux_64.tar.gz
```

You will now have a directory called `linux_64` or `linux_32` containing compiled Rosetta executables and a directory called `rosetta_database` containing the database

3 Configuring and testing the environment

3.1 Running the Setup Script

There are several python modules that need to be installed in order for our analysis scripts to work. Depending on your system, you may or may not have these installed. To determine the status of your installation, run the following command from within the `workshop` directory and pay close attention to the output of the script. This script will tell you which sections of this tutorial you need to follow to set up your environment:

```
python tutorials/setup/setup_rosetta.py --bin=linux_32\
--database=rosetta_database --scripts=scripts --thumbdrive=../workshop
```

Replacing `linux_32, rosetta_database`, `scripts` and `../workshop` with the paths to your binaries, database, scripts and workshop directories.

If you are missing any modules, refer to the section of this document listed in the script error message. After you have installed the missing module(s), rerun the script. Repeat until the script reports that it has successfully configured your environment variables.

If everything is properly configured, the setup script will output the commands necessary for setting the environment variables `ROSETTA_BIN`, `ROSETTA_DATABASE`, `ROSETTA_SCRIPTS`, `WORKHOP_ROOT`, and `ROSETTA_SUFFIX.`

The configuration script will produce a shell script with the commands for setting the environment variables. To run this script and set the environment variables, run the command:

```
source environment.sh
```

These environment variables will only be set in the terminal in which you run the above command, and will only persist until you close the terminal. To permanently set your environment variables, follow the instructions output by the setup_rosetta.py script.

3.2 Running the test script

The test script determines whether you have properly set up your environment. It checks that the environment variables have been correctly configured and attempts to run a score a protein using Rosetta to ensure that it runs correctly. To test your installation, run the following command from the `workshop` directory:

```
python tutorials/setup/test_rosetta.py
```

4 Installing python modules

4.1 Installing numpy

4.1.1 On Ubuntu linux (with root access)

```
sudo apt-get install python-numpy
```

4.1.2 On Fedora linux (with root access)

```
yum install numpy
```

4.1.3 On Gentoo linux (with root access)

```
emerge -va numpy
```

4.1.4 On a Mac, or a linux system without root access

From the `workshop` directory:

```
tar -xvzf numpy.tar.gz
cd numpy-1.5.1
```

If you have root access:

```
sudo python setup.py install
```

If you do not have root access:

```
python setup.py install --home=~
```

4.2 Installing biopython

4.2.1 On Ubuntu linux (with root access)

```
sudo apt-get install python-biopython
```

4.2.2 On Fedora linux (with root access)

```
yum install python-biopython
```

4.2.3 On Gentoo linux (with root access)

```
emerge -va biopython
```

4.2.4 On a Mac, or any other system

From the `workshop` directory:

```
unzip biopython.zip
cd biopython
```

If you have root access:

```
sudo python setup.py install
```

If you do not have root access:

```
python setup.py install --home=~
```

4.3 Installing py_rosetta_util and scripts

If you have root access, from the `workshop` directory:

```
tar -xvzf py_protein_utils.tar.gz
cd py_protein_utils

sudo python setup.py install --install-scripts=../scripts/
```

if you do not have root access, from the `workshop` directory:

```
python setup.py install --home=~ --install-scripts=../scripts/
```