SHORT_TITLE: Server Use END_SHORT_TITLE

USING THE CTFS ANALYTICAL SERVER

R. CONDIT

DESCRIPTION: These are instructions for using the CTFS analytical server to run R programs on CTFS plot data. The principal advantage is in running analyses that take hours to days that would otherwise be difficult on your own desktop or laptop. The processors are not especially faster, but there are 16 processors, so many programs can be running at once.

It's a Unix server, so you need to know a few Unix commands to start out, then R will be executed at the command-line, with no Gui. END_DESCRIPTION

1. Software

On Windows, you will need Putty and WinSCP to make the connection. Mac and Unix already have all you need.

2. Login

You need a user account and associated password. This can only be granted by CTFS Computer Administrators. A single account for each plot can be set up, then shared among any users working with that site.

From a Putty terminal, or a Mac or Unix terminal, enter *ssh USER@euler.arnarb.harvard.edu*, where USER is the name of your plot. You will be prompted to enter the password.

3. The Unix command prompt

3.1. A few unix commands.

- *ls* list entries and directories within your home folder
- *mkdir NEWDIR* to create new subfolder within your home folder
- *cd NEWDIR* to change directories into that new folder (or any other folder)
- *logout* to reattach one of several screens, where ##### is the number of the screen shown by *screen -ls*

3.2. Other users. THe unix command *top* shows all other users and whether there are processes running. Look in the column *%CPU* to see any job using many processes. If a one of the %CPU ~ 100%, then it is using essentially all of one of the processors. But there are 16 processors, so 16 jobs can be underway simultaneously. If you login and find close to 16 other users, you might consider waiting until some are finished. However, it is possible to run > 16 jobs at once; it just slows them all down because some must be sharing processors.

To exit *top* simply type the letter *q*.

4. Transferring files

With the Mac or Unix file browser, choose the option for connecting to a server, then connect to euler.arnarb.harvard.edu. You will need to enter a user name and password. WinSCP on Windows works the same way. Depending on which program, you may be asked to choose the folder and port, but these should work with their default values.

Once connected to the server, a file-browser link to the CTFS server will be open. If your account is *plotname*, then there should be an indication that you are in the folder */home/plotname*. You

---

Date: July 29, 2012.

can now copy files off your own computer by drag-and-drop, delete files, create and delete folders and subfolders, just as you would on your own local computer.

Copy all databases and programs across that you need for execution.

If you intend to work with CTFS R Analytical Tables that are already created by the CTFS Database, you do not need to copy them across. The tables will already be there in the folder for whatever plot you are using. If they are not, contact the CTFS Computer Administrators.

5. The CTFS R Package

You do not need to transfer the CTFS R Package into your folder. It is already installed there, and you can attach it once you run R (see instructions below).

6. The *screen* app

There is a simple Unix utility called *screen* that you need to learn in order to execute analysis 'in the background'. It's the way to start a long program, logout and shutdown the connection while the execution continues, then return later to check progress or get the results. If you simply start a program without running *screen*, the execution will be killed as soon as you close the connection.

There are three commands you need to learn for starting the program. Each of these is run from the Unix command prompt. Beyond these, check help online for some other options.

6.1. Opening and entering a screen.

- *screen* to open a new screen
- *screen -ls* to list screens that are open, showing which are attached and which are detached
- *screen -r* to (re-)attach a screen you opened earlier
- *screen -r #####* to reattach one of several screens, where ##### is the number of the screen shown by *screen -ls*

6.2. Something that can go wrong.

- *screen -ls* may show that a screen is attached, but you cannot get in it (happens sometimes when a connection is cut off while you are working inside an attached screen)
- *screen -x #####* to (re-)attach this screen
- *screen –help* for help

There are three options you need to learn for working within a screen. All options involve typing control-a then one more key. This means holding control down while typing 'a', then releasing control while typing the next key.

6.3. Working within a screen (one that is attached).

- *control-a d* to detach the screen and get back to the main unix prompt; this is what you do after you start the program you want to run
- *control-a c* to open a second screen within, a sub-screen you could call it; you can do this again to open a third screen, etc.
- *control-a space* to cycle through all sub-screens within this over-arching attached screen

6.4. **Finishing a screen (for good).** When you want to kill a screen, meaning close it for good, type *exit* at the unix command from within that screen. That is, attach the screen, then exit. It will no longer show up with *screen -ls*.

7. RUNNING R

From within any screen, use R exactly as you would at a command-line in Windows or Mac. There are no menu options, and no GUI, but commands are otherwise identical. If you plot, the graph will not appear (a pdf is created though), and you can plot to pdf's or jpg's; you cannot view the graphs though. So your strategy must be to create data files or graphs, then transfer back to your own computer to view.

To source the CTFSRPackage, type at the R command prompt *attach('/home/CTFS/CTFSRPackage.rdata')*.

Once you start execution of the program you wish to run, close the screen (control-a d), then type *logout*.

Later, log back in using Putty or a Mac/Unix terminal, enter *screen -r* to reattach the screen where program is running. When it is finished, save the result and use WinSCP, or the Mac/Linux browser, to copy to your local computer.