**ACI-ICS Lab Manual**

*Author: Daniel Elbich*

*Contact: delbich10@gmail.com*

*Getting Started*

ACI is a high powered computing system currently in use at PSU. This is a Linux based system and is best used through the command line (e.g. Terminal, bash shell scripting). There is a GUI (graphical user interface) access to the system which will take you to a Linux desktop for more point and clicker interfacing. It is critical to note that the GUI interface is where low resource tasks should be performed, such as viewing MRI data or working on scripts/data pipelines. More high resource tasks, such as preprocessing a large batch of subject data (e.g. things that will take hours, not minutes) should **NOT** be run here as they take away computing resources from all users and is against ICS protocols.

You can get by using the GUI for most things, but at some point it will become more efficient to use the command line to work here. While the system has a specific organization/setup for the university, this type of server is commercially available and employed by many universities. As such, if this manual or the PSU website (<https://ics.psu.edu/computing-services/ics-aci-user-guide/>) is missing a piece of information, it is very likely you can find it through online searches including ‘aci’ (e.g. “aci loading software”).

*Connecting to ACI*

There are three server addresses that can be used to access the ACI cluster, each with a different purpose:

* **aci-b.aci.ics.psu.edu** – This address is used to connect to the batch system through terminal. This is where you would submit jobs (see below) for high powered computing. It can only be used with the command line (e.g. no point and click). To connect, enter the following into terminal

ssh [ACCESSID@aci-b.aci.ics.psu.edu](mailto:ACCESSID@aci-b.aci.ics.psu.edu)

where ‘ACCESSID’ is your PSU access ID (e.g. dbe5007). You will be prompted to enter your password and also authenticate with Duo.

* **aci-i.aci.ics.psu.edu** – This address is used to connect to the GUI system to be used for viewing data or code testing. Computationally intensive jobs should NOT be run here, and if this is ignored you risk ICS terminating your job/session & warning you not to do this again. Use the program Exceed OnDemand to connect to the interactive desktop, available for [download here](https://ics.psu.edu/userguide/05-00-basics-aci-resources/05-04-connecting-aci/). You will be required to enter your access ID, password, and authenticate with Duo to connect.
* **datamgr.aci.ics.psu.edu** – This address is used to transfer files to and from ACI. This is the only address that can be accessed with any FTP server (e.g. Filezilla, Cyberduck, Transmit). The aci-b address can also be used to transfer data via the command line, but will not work in any program except terminal. You will be required to enter your access ID, password, and authenticate with Duo to connect.

*Basic Linux Commands*

**Important note:**

As mentioned, ACI is a Linux based system and as such all possible tasks can be run through the command line. While very useful, this can present some potential issues when beginning to learn to use this utility. The most important rule is to always be sure you are running the correct command to the best of your knowledge. Be sure you are deleting the correct file, moving the correct folder, or renaming the correct set of data. Because this is not point and click, there is no ‘undo’ command – once its done, its done. When learning this skill or testing out some new code, it is recommended to test whatever you’re writing on a separate set of data or files (e.g. **NOT** raw data) to be sure the outcomes are what you intended, so as to avoid uncorrectable errors. Data on ACI is backed up to protect from sudden losses, but this should not always be expected to happen.

Here are some basic tips for starting out using terminal:

* **Spaces are not your friend!**
  + Spaces are the natural enemy to anything done or read on the command line and should be avoided at all costs. There are some ways to work with them, but 99/100 times they will cause whatever command you’re trying to run to not work and crash. Instead, use an underscore to provide some break in the text but keeping the name contiguous.
* **Tabs**
  + Because the command line is not point and click, you (normally) need to type out all folders and files you want to access. One way to speed this up is to use the tab key. Using tab will autocomplete some or all of the text with an associated letter or prefix. For example, if you are in a directory with 1 folder beginning with ‘s’, and you begin to type the name to change to it, simply typing ‘s’ and hitting tab will complete the folder name without you having to type it all out. If there are multiple folders beginning with that character, hitting tab will list out the possibilities of what would complete the name for you to more easily type out.

Here are some basic commands for navigating through terminal (also a helpful [website here](https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners)). Many of these can provide more information by adding ‘--help’ after the command or through internet searches:

* **ls** – List all folders and files in that directory. Folders are listed first in numerical then alphabetical order, followed by files in the same order.
* **pwd** – Gives you the directory (folder) you are currently in, the present working directory.
* **cd** – Change directory. Use this followed by the folder name or path to move to that directory.

Commands for folder/file manipulation

* **mkdir** – Creates a directory of a given name (e.g. mkdir newFolder)
* **rm** – Removes/deletes a file or directory. The flag ‘-r’ must be included after this to delete a directory.
* **cp** – Copy a file/folder to a new location (e.g. cp newText.txt /new/folder/here).
* **mv** – Move a file/folder to a new location (e.g. mv newText.txt /new/folder/here)
* **touch** – Create an empty file, such as a text or zip file.

Below are some intermediate commands. They have more specialized purposes and will not be needed as much as the commands above

* **echo** – Used to print text or variable information into terminal console. Can also be used to print text into separate text file (e.g. echo text line here >> newFile.txt)
* **cat** – Print file contents into terminal window
* **nano/vi** – Text editor inside bash environment
* **zip/unzip** – Compress or uncompress file
* **chmod** – Change permissions of file or folder

Intermediate Tips

* **Syntax**
  + Because linux is also a programming language like Matlab or R, you can set items like variables to help yourself do tasks. For example, file paths can be set to variables in order to navigate faster:

Long way:

cd /path/to/long/directory/and/the/sub/folder/within/it

Variable:

folder = /path/to/long/directory/and/the/sub/folder/within/it; cd $folder