American Museum of Natural History

American Museum of Natural History
Richard Gilder Graduate School

RGGS Comparative Genomics 2 – Computational Methods (Session 3)

Jose Barba

Gerstner Scholar in Bioinformatics & Computational Biology

AMNH Arachnology Lab
Institute for Comparative Genomics
jbarba@amnh.org
josebarbamontoya.github.io

September 19, 2024
ICG conference room

Session 3 outline

- Questions on any topics or tutorials from session 2?
- Command line computing basics 2
- Introduction to GitHub
- Additional matters

List of basic Unix commands – updated

Archiving and Compression

tar - Archive files into a tarball (or extract them)
zip/unzip - Compress and extract files in zip format

File and Directory Operations

cat - Concatenate and display file contents
cd - Change directory
cp - Copy files or directories
head - Display the first few lines of a file
ls - List files and directories
mkdir - Make directory
mv - Move or rename files or directories
pwd - Print working directory
rm - Remove files or directories
rmdir - Remove directory
sort - Sort the lines of a file
tail - Display the last few lines of a file
touch - Create an empty file or update a file's timestamp
tr - Translate or delete characters

File and Directory Search

find - Search for files and directories within a hierarchy
grep - Search for patterns within text

File Permissions and Ownership

chmod - Change file permissions
chown - Change file owner and group

File Space and Disk Usage

df - Display disk space usage
du - Estimate file space usage

Networking and Remote Operations

curl - Download and transfer data from remote servers
exit - Close current shell, terminal prompt, interactive program
scp - Secure copy, used for transferring files between hosts
sftp - Securely transfer files between computers over a network
ssh - Secure shell, used for logging into a remote machine
wget - Download and transfer data from remote servers

Operators

***** - Mange multiple files and directories
& - Run a command in the background
> - Redirect the ouput of a command to a file
| - Connect the uputput of one command to the input of another

Operators are used to manage data and control how commands interact with files and each other in a shell environment

Process Management

kill - Terminate processes by ID
ps - Display information about running processes
top - Interactively manage processes

Terminal Utilities

clear - Clear the terminal screen
help - Show information about built-in shell commands
man - Display manual for a command

Text Processing

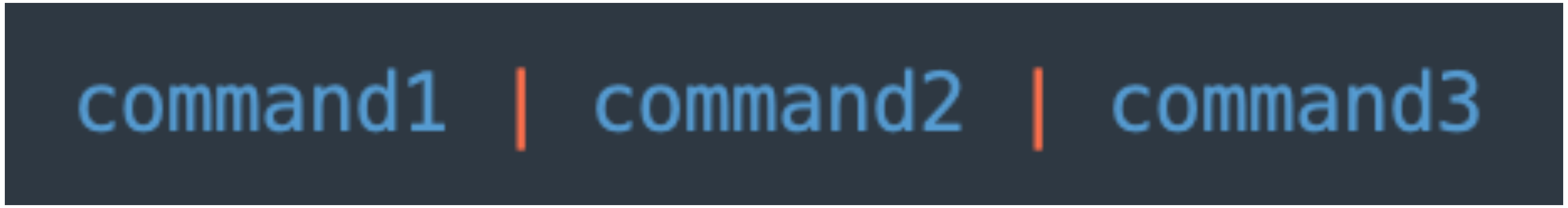
awk - Pattern scanning and processing language
cut - Extract specific sections of text
echo - Display a line of text or variables
nano - Text editor
sed - Stream editor for filtering and transforming text
vi - Text editor

Command line computing basics 2

- **A tutorial on shell pipelines and loops will give you a strong foundation for processing data, automating tasks, and manipulating text files efficiently in Unix-based systems**
- **Instructions to download the command line computing basics 2 tutorial to the home directory:**
  1. Open the terminal
  2. Type `cd ~`
  3. Enter the following command `wget https://raw.githubusercontent.com/josebarbamontoya/rggs_comparative_genomics_2/main/session_03/command_line_computing_basics_2_tutorial.sh`, if `wget` is not available, use `curl -0` instead

Command line computing basics 2

- **Shell pipelines**
  - A pipeline in a Unix shell is used to connect the output of one command to the input of another. It allows you to chain commands together, making it a powerful feature for handling complex tasks
  - Allow you to chain commands together to process outputs efficiently
  - `|` is the pipe operator, which passes the output of command1 as input to command2, and so on.
  - Basic syntax:

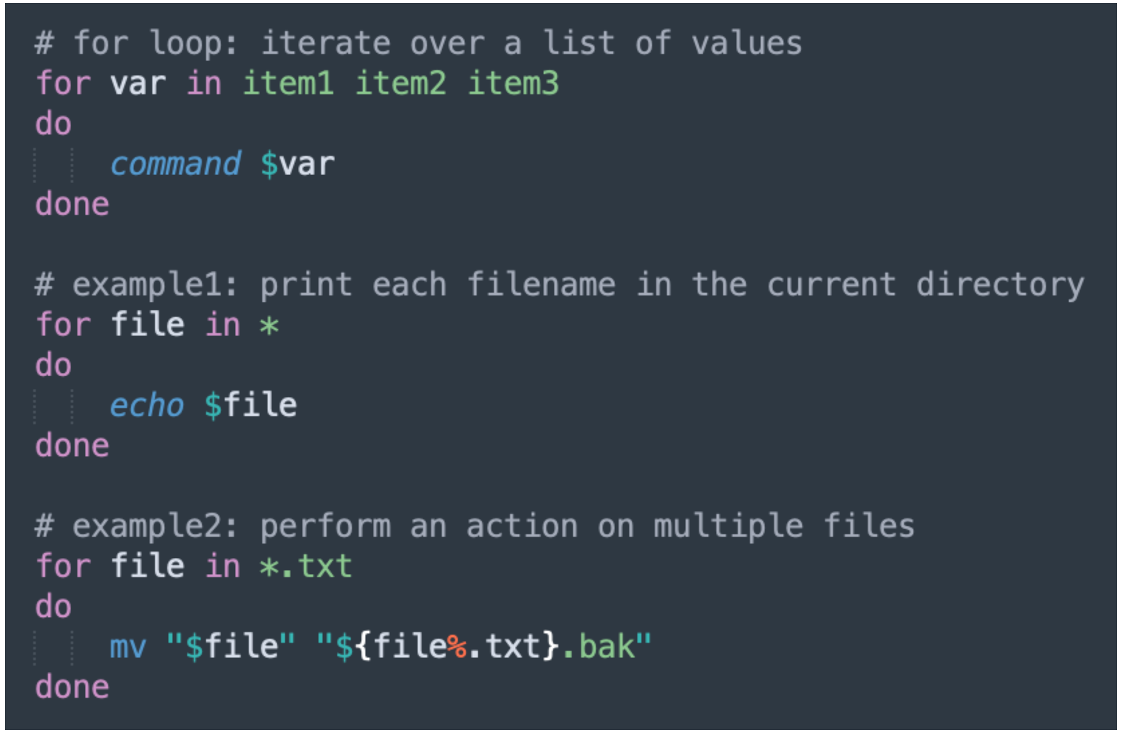


```
command1 | command2 | command3
```

Command line computing basics 2

- **Loops in shell**
  - Shell loops allow you to iterate over lists or run commands repeatedly. There are two common types of loops:

For loops

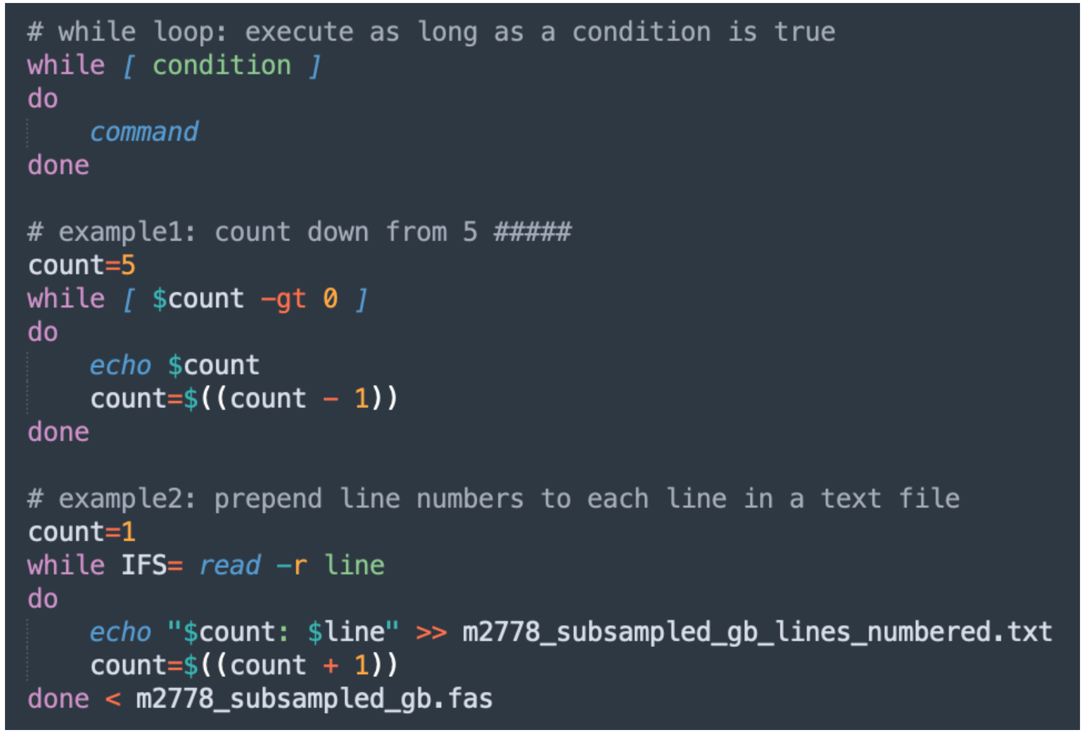


```
# for loop: iterate over a list of values
for var in item1 item2 item3
do
    command $var
done

# example1: print each filename in the current directory
for file in *
do
    echo $file
done

# example2: perform an action on multiple files
for file in *.txt
do
    mv "$file" "${file%.txt}.bak"
done
```

While loops



```
# while loop: execute as long as a condition is true
while [ condition ]
do
    command
done

# example1: count down from 5 #####
count=5
while [ $count -gt 0 ]
do
    echo $count
    count=$((count - 1))
done

# example2: prepend line numbers to each line in a text file
count=1
while IFS= read -r line
do
    echo "$count: $line" >> m2778_subsampled_gb_lines_numbered.txt
    count=$((count + 1))
done < m2778_subsampled_gb.fas
```

Introduction to GitHub

- GitHub is a platform for version control and collaboration, primarily used by developers and scientists to manage code and data
- It is built on Git, a distributed version control system that tracks file changes, enabling simultaneous collaboration without overwriting work

Introduction to GitHub

- **Key Concepts:**
  - Repository: A project space where your files and the entire project history are stored
  - Commit: A snapshot of your project at a particular time, along with a message describing the changes
  - Branch: A parallel version of your repository, used to work on new features without affecting the main project
  - Pull request: A request to merge changes from one branch into another, often reviewed before approval
  - Clone: A local copy of a repository
  - Fork: A personal copy of someone else's repository, which you can modify independently

Introduction to GitHub

- **How GitHub works:**
  - Create a repository: Start a new project or upload an existing one
  - Clone/download the repository: Work on the project locally using Git commands
  - Commit changes: After making changes, commit them to track progress
  - Push to GitHub: Send your committed changes to the GitHub repo
  - Collaborate: Use branches, pull requests, and issue tracking to collaborate with others

Introduction to GitHub

- **Basic git commands:**
  - `git init`: Initialize a Git repository
  - `git clone <repo>`: Clone a repository to your local machine
  - `git add <file>`: Stage a file for a commit
  - `git commit -m "message”`: Commit changes with a message
  - `git push`: Push local changes to the GitHub repository
  - `git pull`: Fetch and merge changes from the remote repository to your local one

Introduction to GitHub

- **Create your own GitHub account:**
  - Visit https://github.com
  - Follow instructions from https://docs.github.com/en/get-started/onboarding/getting-started-with-your-github-account#
- **Create a repository for the CG2 course:**
  - Follow instructions from https://docs.github.com/en/repositories/creating-and-managing-repositories/quickstart-for-repositories

Additional matters

- **Session 13 — November 28**
  - Reschedule the class to either November 25 or 26