

# Class 3

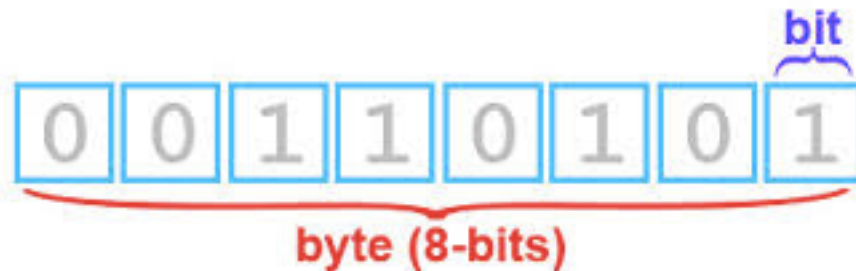
All about Computers and  
High Performance Computing

# Goals

- Understand high performance computational resources
- Newton – the UTK cluster
- Set up and manage a project directory

# Start at the beginning

- The basics of a personal computer
- Computers use a language of 0s and 1s
- Bit = a single stored unit that is either 0 or 1
- Byte = a set of 8 bits



# CPU and Processor

## CPU

- The “brain”
- Executes instructions with electronic circuits
- “Clock Speed” is measured in gigahertz (GHz)
- Billions of operations per second (sort of)
- Manufacturers include Intel and AMD

CPUs are usually 64-bit now (used to be 32-bit, this refers to the amount of data that can be operated on at one time)



# CPU and Processor

## Processor

- A processor used to contain only one CPU so the terms were essentially synonymous.
- Now processors can contain many CPUs. These are called cores. (multiple CPUs = cores)
- Dual core/ Quad core



There are 4 cores in  
this tiny processor!

3 billion instructions  
per second

# Random Access Memory (RAM)

- This is short term memory
- Fast memory
- CPUs can only load data from RAM
- Lots of RAM lets your computer do more things at once or access more data at once
- Stored information is lost if power is lost



8Gb of RAM is currently common for personal computers; 16Gb is also available

# Hard Drive

internal



external



- Long term storage
- Slow memory
- Data is not lost if power is lost
- When you “open” software or a file, the information is loaded from the hard drive to RAM
- The processor can access the information from RAM much faster
- Saving the file writes the new information back to the hard drive

# Graphical Processing Unit (GPU)

- Video card
- Specialized CPUs for image processing and computer graphics (monitor, phone screen)
- Highly parallel
- Most processing involves matrix and coordinate systems – this leads to...
- GPUs being co-opted for use in bioinformatics algorithms

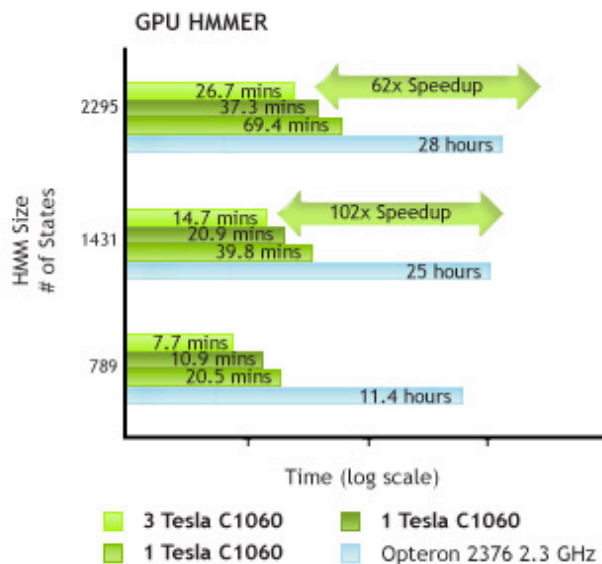




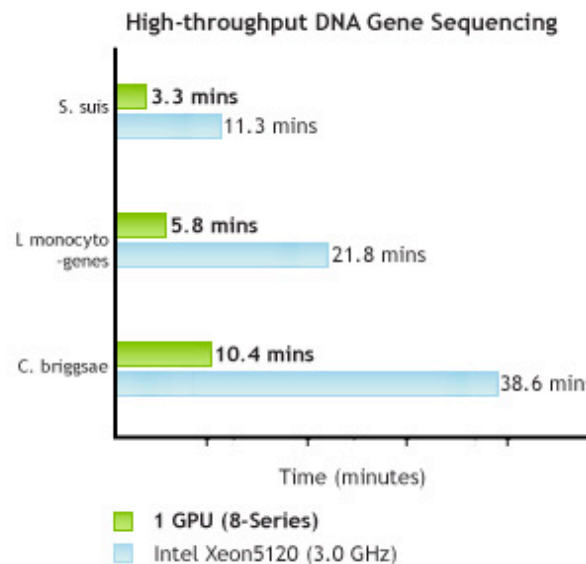
# Graphical Processing Unit (GPU)

## BIOINFORMATICS AND LIFE SCIENCES

Sequencing and protein docking are very compute-intensive tasks that see a large performance benefit by using a CUDA-enabled GPU. There is quite a bit of ongoing work on using GPUs for a range of Bioinformatics and life sciences codes.



Accelerating HMMER using GPUs  
Scalable Informatics



MUMmerGPU: High-through DNA sequence alignment using GPUs  
Schatz, et al

# Memory Review

## RAM

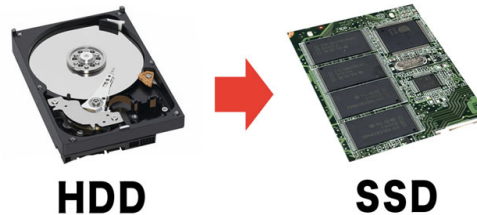
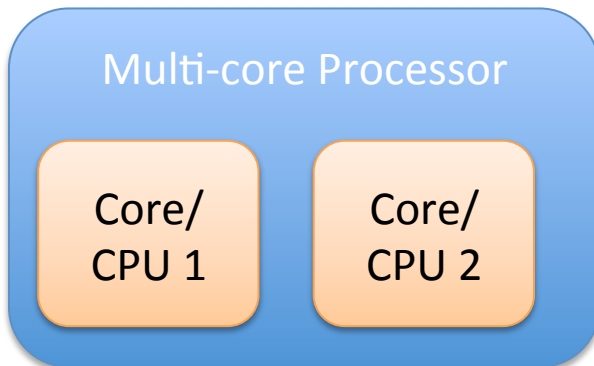
“Memory”

- Fast
- Expensive
- Volatile

## Hard drive

“Filesystem”

- Slow
- Cheap
- Not Volatile



Solid state drives are 5-10X faster than hard drives, but more expensive.

# Remote computing

- Large data is often too much for your small laptop or desktop to handle
- Move to large computers or clusters of computers – ie “remote” computers
- Vocabulary
  - “High performance computing” (HPC)
  - Supercomputer
  - Grid



Type the commands



Actual work being done



# Scaling up inside one computer

Our “workhorse” computer

- 4 processors
- 10 cores per processor
- = 40 CPUs
- 256 Gb of RAM
- ~ \$8k
- Lives in the UTK server room with other computers



This is by no means a particularly large supercomputer. I have heard of computers with 2Tb of RAM. (1Tb = 1000Gb)

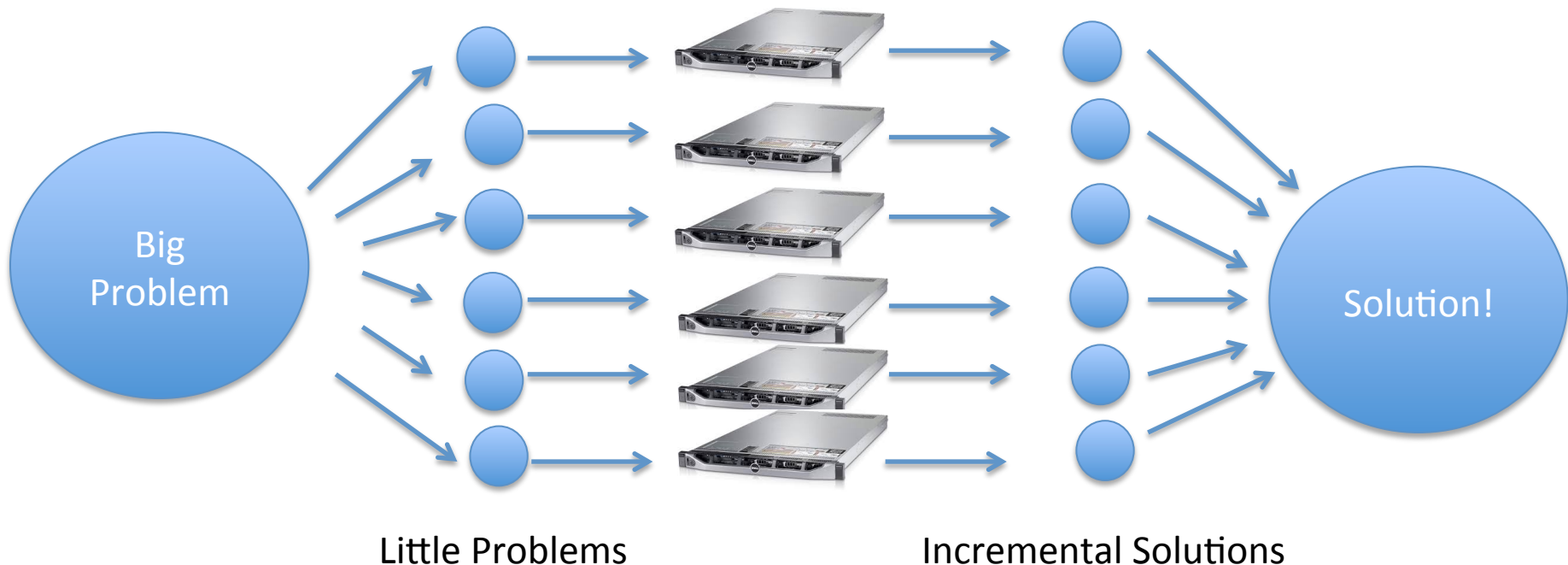
# Scaling up to many computers

- Supercomputers, grid computing
- Massively parallel
- Tens of thousands of processors across thousands of machines
- Need controller software to coordinate activities



# Parallelism

- dividing a single job among multiple processors
- Objective is to run a program in less time
- Some activities are more easy to divide than others
- There are different ways to have parallel processing



# Multi-threaded

- Many CPUs in a single computer
- All can access the same RAM
- Many people are all in the same room
- Puzzle pieces are all on a table in the middle



- People = CPUs
- Puzzle pieces = Data
- Table = RAM



# Multi-computer, or “distributed memory”

- Split the job across different computers
- RAM is not shared
- People are in different rooms
- Puzzle pieces must be divided up between each person



- People = CPUs
- Puzzle pieces = Data
- Table = RAM



# Parallel processing

- Parallel across computers is much more difficult to implement for complex problems
  - Assembly
- Fortunately we have many problems in bioinformatics that are modular and scale gracefully to many computers (“embarrassingly parallel”) – little to know work is involved in splitting the problem into small pieces
  - BLAST
  - Sequencing alignments
- If software will run across many computers, documentation will usually mention MPI (message passing interface)

# Titan

- The second largest supercomputer in the world is at ORNL (used to be first)
- 18,688 processors, each with 16 CPUs
- 18,688 GPUs
- \$97 million



# Newton

- over 6000 CPU cores
- Scientific Linux
- Any UTK affiliated researcher (student, faculty or staff) can get access
- Higher priority access for those who buy in
- New in 2016 –  
Monster Node! 1Tb of RAM shared across 48 cores



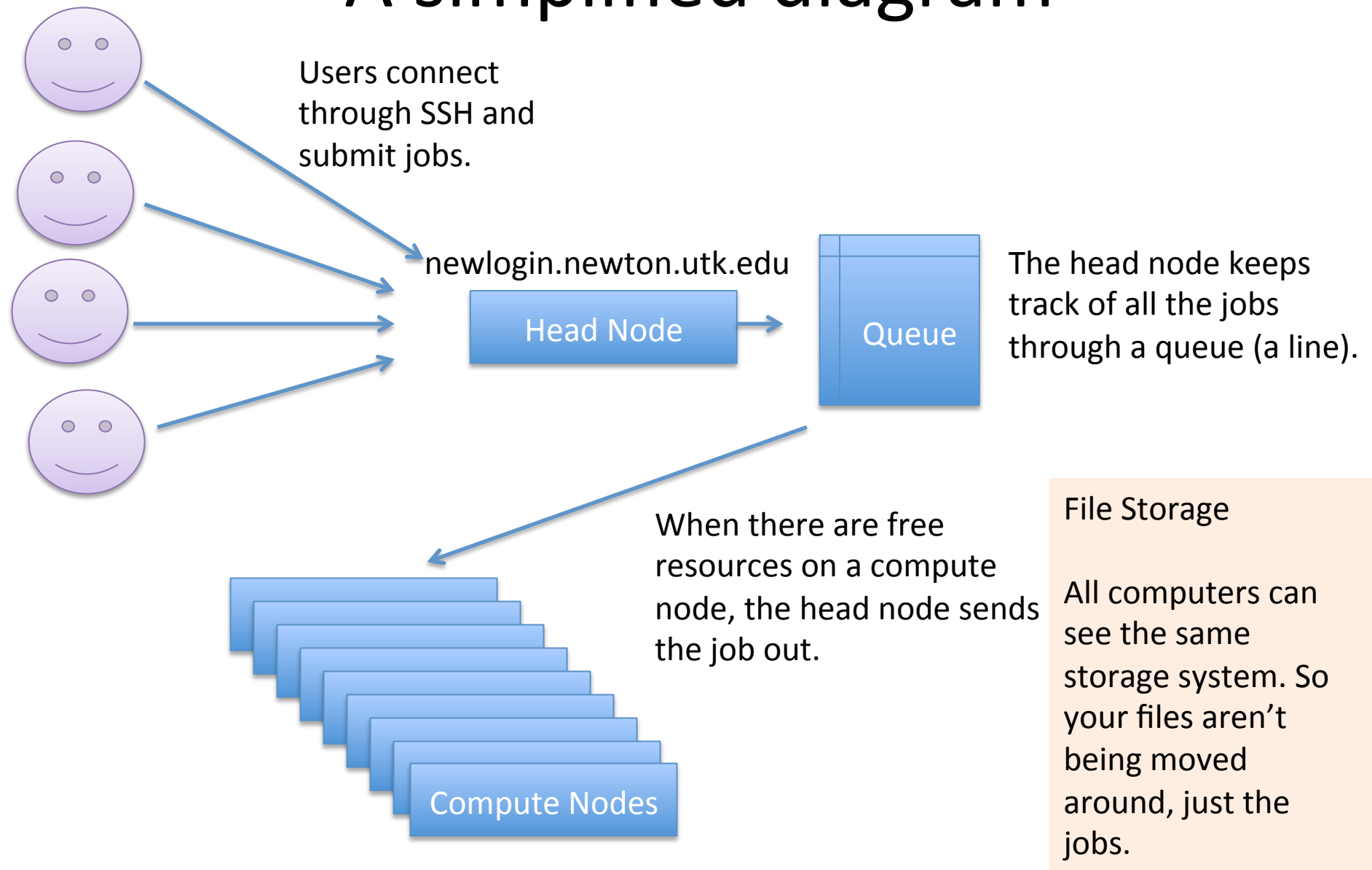
# Service features

- Technical
  - HPC cluster
  - High-performance storage
    - 150Tb filesystem
  - Remote data backup
  - Computing environment management
  - Resource management (Grid Engine)
- Professional
  - Computing support, including software installs
  - Technical advising
  - Consulting (proposal development)

Newton Program website:

<http://newton.utk.edu/>

# A simplified diagram



# SSH

- Secure shell
- Network protocol – secure channel of communication between a client and a server
- This is the most common way to connect to a remote computer
- Its encrypted and its available on all UNIX/Linux systems
- To log into newton, you should ssh into the head node:

[login.newton.utk.edu](https://login.newton.utk.edu)

# Grid Engine

- The software for distributed resource management (ie computational cluster)
- Determines when and where to execute jobs
- Manages resources (keeps jobs from trying to use the same resources)
- Determines user priority and job priority
- Optimizes – like a puzzle, tries to figure out how to get the most jobs executed in the fastest amount of time

No computation on the head node.

Anything that will run for longer than a minute or take more than 1Gb of RAM should be submitted to the scheduler

# Newton

- Sub clusters
- You can specify what you need to run your job
- <https://newton.utk.edu/doc/Documentation/Systems/>

Cluster Name	CPU Model	Nodes	Cores /Node	RAM /Node
Sigma	Intel Xeon E5-2680v3	108	24	128 GB
Rho	Intel Xeon E5-2670	48	16	32 GB
Chi	AMD Opteron 6180 SE	36	48	96 GB
Phi	Intel Xeon X5660	72	12	24 GB
<u>Monster</u>	Intel Xeon E5-2687W v4 @ 3.00GHz	1	48	1 TB



# Creating a job

1. You create a job file that describes all the resources that your calculation needs and details on how to execute your programs (a shell script). This file (resource request plus shell script) is called a "job file".
2. You submit the job file to the Grid Engine.
3. The Grid engine will find the requested resources on the compute cluster and will reserve them for you job. If enough resources are not currently free, the Grid Engine will hold your job and periodically check if sufficient resources have become available.

# Creating a job

4. Once the required resources have been reserved, the Grid Engine will automatically execute your calculation for you using the shell script that you provided as part of your job file.
5. As the calculation runs, the Grid Engine will save log files for your job.
6. When the calculation is finished, the job will be deleted from the Grid Engine and the resources that it was using will be freed. You can now look at the calculation output for your result.

# Limits

- Opportunistic users – (that's us!) – we have lower limits than priority users. You can become a priority user by buying resources.
- Time
  - Jobs can go into a medium queue or a short queue
  - Medium queue will run a job for up to 24 hours
  - Short queue will run a job for up to 2 hours
- Storage
  - Your home directory is limited to 20Gb
  - Can be increased by request
  - You'll want to keep an eye on this with the ``du -sk`` command

# Project Directories

# Organization of files

- In the course of a project, you will have
  - Data files
  - Processed data files
  - Intermediate analysis files
  - Final output files
  - Images
  - Scripts
  - Notes
  - Commands



No!

# Organization of files

- Essential for:
  - Reproducing or rerunning results
  - Sharing with other people
  - Automating tasks
  - Managing a large project with many files
  - Your sanity

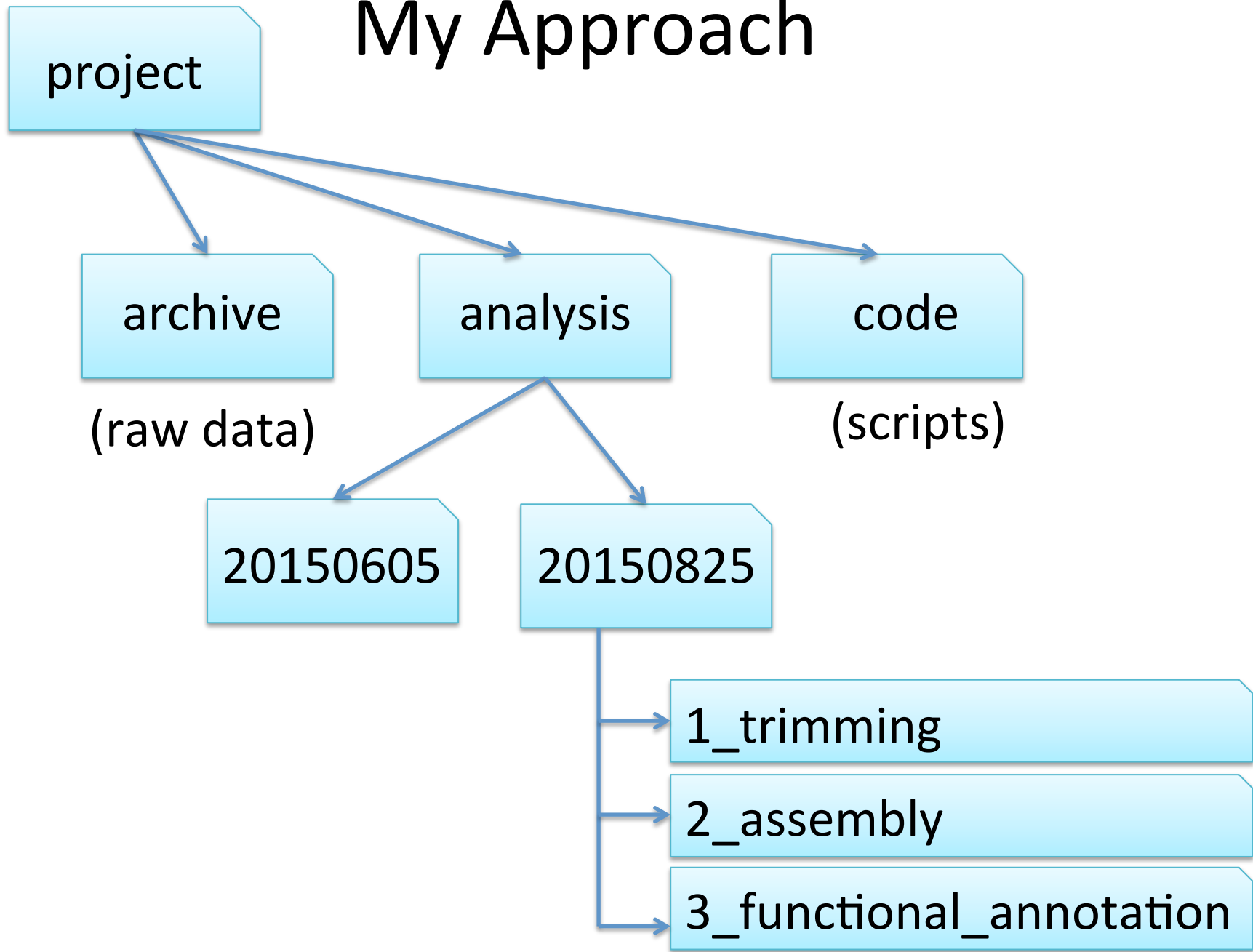
# To start

- Everything for one project will go in one master folder
- Give this folder a sensible, descriptive name

The organization inside this folder varies from lab to lab. One strategy is outlined in Noble (2009) “A Quick Guide to Organizing Computational Biology Projects”

Just be consistent.

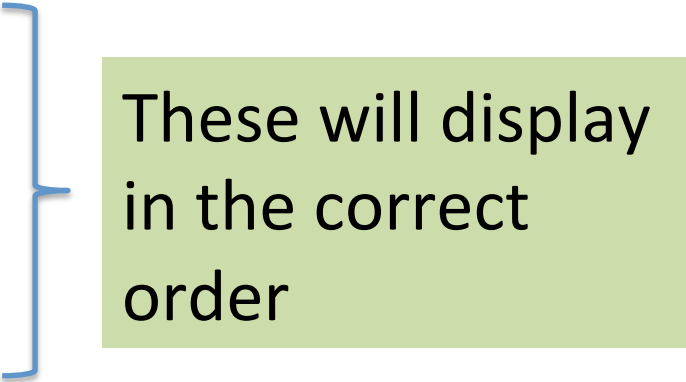
# My Approach





# More Best Practices

- Use sensible, descriptive names for all folders and files
- No spaces
- Use file extensions (.txt, .fasta, .sam)
- Use leading zeros
  - File001.txt
  - File002.txt
  - ...
  - File192.txt



These will display  
in the correct  
order

# Documentation

- Methods and workflows
  - Write down your full command lines
  - Preferably in a script
  - Software versions
- Origin of data
  - Where did it come from? Who gave it to you?
  - This includes not just your experimental data, but any data you use – a reference genome, a set of genes, a database
  - Date of download
  - Version

Simplest  
method:  
**README**  
file



# Data Integrity

- We have already seen that you can download data from the internet via wget (or curl)
- Other ways to move data
  - scp
  - ftp

# HTTP vs FTP

- HTTP = Hypertext Transfer Protocol
- FTP = File Transfer Protocol
- Both are ways to transfer files across a network (thus building the internet)
- HTTP is used for webpages, HTTPS for secure web pages (all communication encrypted)
- Both can do uploads and downloads
- FTP can be faster and more efficient for large files - sometimes



Panic for OS X



Cyberduck  
For OS X



You can use FTP to transfer files to/from your computer to newton.

# SCP

- Secure copy
- Command line interface based on SSH

From a local directory to newton:

```
scp file.txt mstaton1@login.newton.utk.edu:~/lesson2
```

Local file

Username  
on remote

Remote  
Machine

Remote  
location to  
place file

From newton to your local directory:

```
Scp mstaton1@login.newton.utk.edu:~/lesson2/file.txt .
```

Username  
on remote

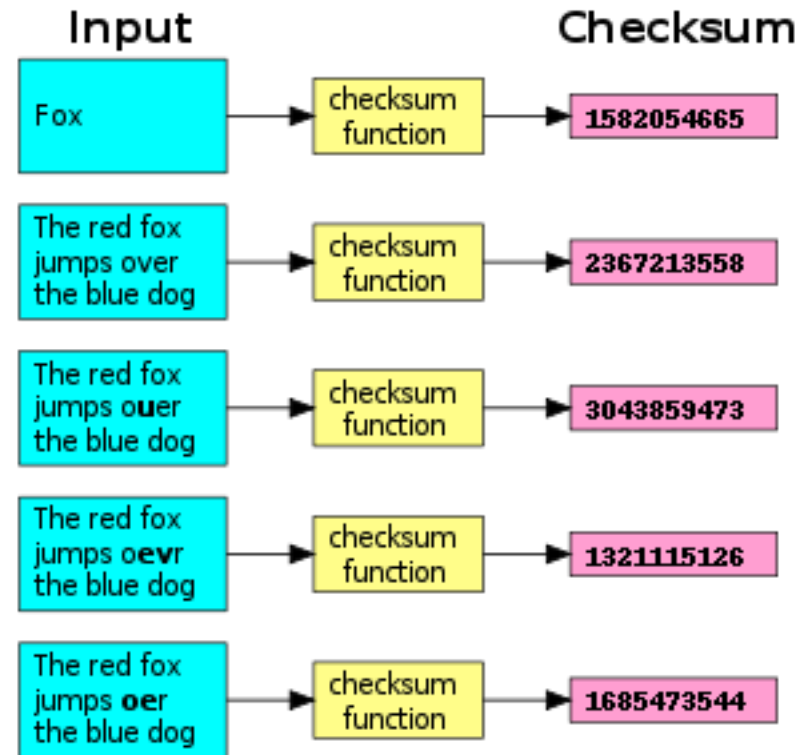
Remote  
Machine

Remote  
Location

Local  
Location

# MD5 and SHA-1 checksums

- A short(ish) set of characters generated from a file using a standard mathematical algorithm
- Used to detect errors that may have been introduced by transfer or storage
- MD5 is more common, we'll use that
- SHA-1 is newer but less common



If the server comes up with an md5 checksum and the recipient comes up with the same md5 checksum, the entire file was (most likely) transferred intact

# Checksums

- Often used by sequencing centers
- Sequencing center provides:
  - Sequence file
  - Checksum
- You download the file
  - Calculate a checksum on your version
  - Check if they match
  - If they do, your file is whole and identical to theirs

# Lesson

- Finding things in files
- Writing shell scripts