Short Read Sequencing Analysis Workshop

Day 3 – Using Compute Resources at BioFrontiers

Zach Maas and Mary Allen

Original Slides from Jonathan DeMasi

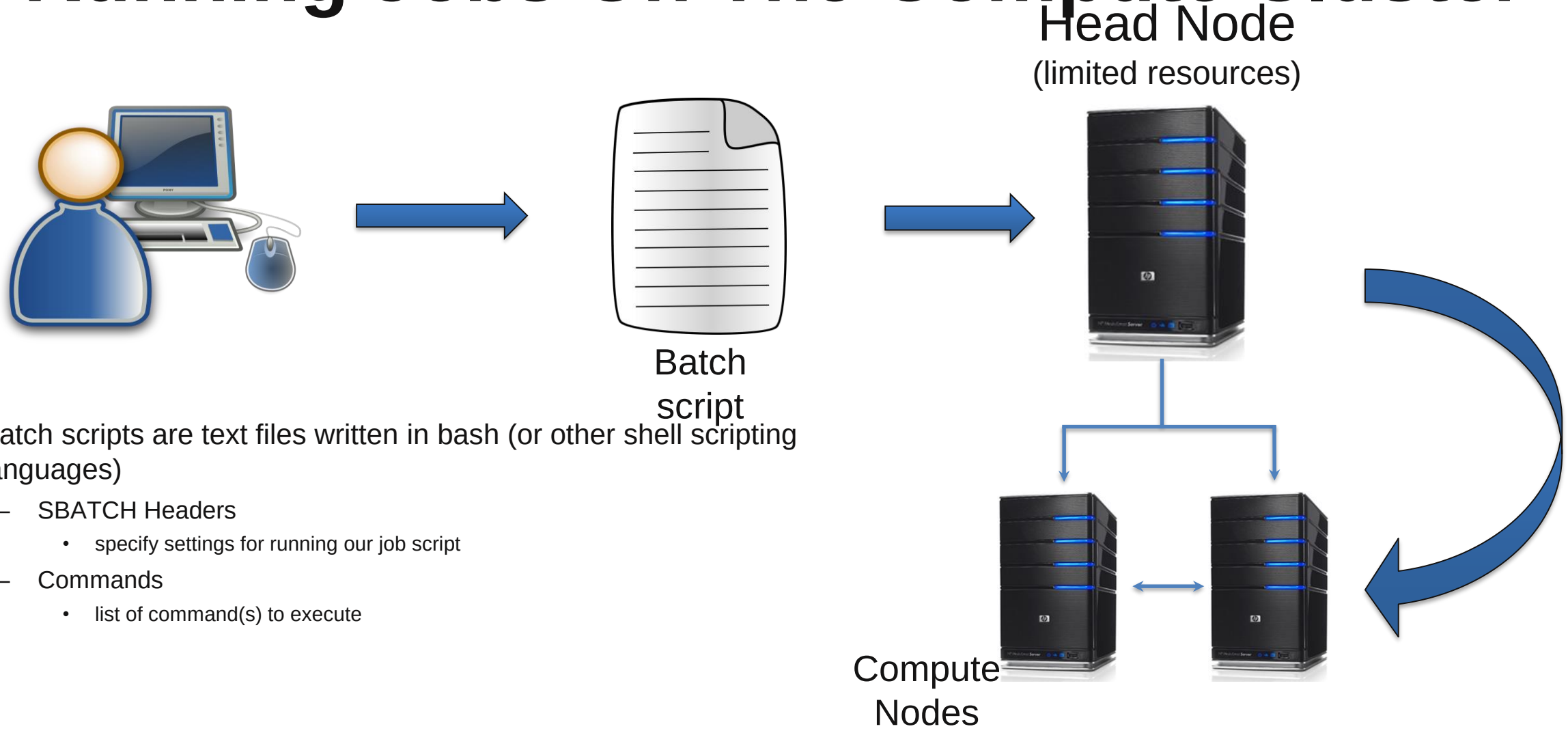
Learning Objectives

- Understand why and when we use compute clusters
- Know how to rsync data from AWS to your computer (and the other way around!)
- Download things from the Internet using wget
- Submit a job to the scheduler and interpret the output
- Diagnose job failures

Special Characters in Bash Scripts

- $ - The dollar sign character is used for representing variables
- “ “ - Double quotes will expand variables like $USER, single quotes will not
- * - The star character is a “wildcard”. It expands to all files in the directory. For example, *.txt will expand to every file ending in .txt, this is useful for pattern matching
- # - The hash sign / octothorpe starts a comment. Any text after it will not be executed
- <, >, >> - The angle bracket characters are used for redirecting command output. There are many variations using these.
- | - The pipe character takes the output from one command and feeds it in as the input into another command
- ; - The semicolon character is identical to
- & - When placed after a command, the ampersand character will send that command to the background and immediately start the next command
- \ - The backslash is an escape character. If you need to use one of these special characters, you can do so by prefixing it with a backslash. For example. \$ will be interpreted as a dollar sign character, not as the start of a variable.

Running Jobs On The Compute Cluster



- Batch scripts are text files written in bash (or other shell scripting languages)
  - SBATCH Headers
    - specify settings for running our job script
  - Commands
    - list of command(s) to execute

Understanding Modules

- Environment variables modify the way we interact with the cluster or help us to “find” things
  - Common environment variables are PATH, PYTHON_PATH, LD_LIBRARY_PATH

- Modules allow you to easily and dynamically add to and change environment variables
  - All modules are unloaded after you terminate your current session

What Causes A Job To Fail

- Numerous sources of error that can prevent successful job completion
  - Incorrect command-line setup; misuse of parameters
  - Incorrect file formats
  - Incorrect path designations
  - Issues with version compatibility
  - Incorrectly setup job script/SBATCH headers

BREAK

The way to do things

Download your data
Run programs on your data
Use any storage space you want… for instance
/Users/<username>/
Publish your data

The right way to do things

Download your data

Lock its permissions so it can never be edited!!!
Keep the raw data somewhere backed up!!!
NEVER TOUCH RAW DATA!!!

Run programs on your data

- Set up your storage system optimally
  - Make a directory on /scratch for each of your projects (/scratch/Shares/labname/ or scratch/Users/username/)
    - Make an input and output directory in that directory
      - rsync your raw data to your input directory on scratch
      - scratch is not backed up!!!!!
    - Make a scripts directory (in someplace that is backed up, for example /Users/username/)
      - Back up your scripts directory to github
      - All software you run should be in a script (not on command line!)
      - Make a README file that tells everything you would put in your lab notebook, track as you go
        - Where is the raw data
        - Which scripts did you run on it
        - What files did you make and where are they
- Keep the living room clean!
  - After you run a program, you may have an intermediate file you want to keep
    - When you get intermediate files you want to back up rsync them to somewhere backed up
    - Always check your results

      QUALITY, QUANITY (NUMBER OF READS), VISUALIZE
  - Delete stuff on /scratch frequently (Data on scratch costs more and clogs up the system)
- Upload the raw data and the final processed files to NIH GEO

Publish all your data

- All versions of all programs used must be noted in the methods section
- github can be used to share the code you used

Extensive Publicly Available Data and Tools

Publicly available datasets

- NCBI Gene Expression Omnibus (GEO)
  - NCBI FTP site and SRA database
- Genbank
- UCSC Table Browser
- Ensembl
- Model organism specific site like SGD

Programs/packages

- SourceForge
- GitHub

The End

Questions??

Don’t forget the homework.

Watch videos for Day 4

Help sessions: 1-3PM in E1B11

IT Questions? Email bit-help@colorado.edu

Acknowledgments

Workshop Coordinator: Mary Allen

Funding: BioFrontiers Institute and Colorado Office of Economic Development and International Trade

Additional Acknowledgments

Compute Resources: BioFrontiers IT Staff

Robin Dowell and Dowell Lab