Please keep the expanded **RNAseq_Pipeline** directory in your **DOWNLOADS** directory. Doing so will maintain paths included in script files that we will run in the teaching demonstration. The **RNAseq_Pipeline** directory contains the following:

- Demo_ENV.yml – installs specific compatible versions of tools used in the demo
- RawCountData.txt – contains raw gene expression count data for use in preliminary gene expression analysis
- README_SetupForDemonstration – current file, contains descriptions and setup instructions
- RNAseq_Analysis.html –description, code, and output of preliminary differential expression analysis
- RNAseq_Analysis.R – R script we will run in the demo (time permitting) that performs preliminary gene expression analysis. This script will serve as a starting point for future class periods.
- RNAseq_Pipeline.Rproj – R project object that maintains project directory structure
- SampleList.txt – text file with sample names and corresponding experiment metadata
- SampleSyllabus.pdf – example syllabus with course schedule. The current lesson materials come from day 2 of week 11
- SRR_urls.txt – text file with URL addresses for each of the subsampled data files used in the demo
- STEP1_RNAseq_config.sh – shell script preparation homework that generates a directory structure and establishes variables
- STEP2_RNAseq_Pipeline.sh – shell script that downloads sample data and the reference transcriptome, and implements RNAseq pipeline tools to estimate gene expression

To participate in the bioinformatics teaching demonstration, you will need the following programs (operating system-dependent):

Mac:

- R (v4.1.3) https://cran.r-project.org/bin/macosx/ **choose R-4.1.3.pkg, **NOT** R-4.1.3-arm64.pkg even if you have an M1 Mac (due to Bioconductor incompatibilities for packages used in demo)
- Rstudio (v2022.02.0 build 443) https://www.rstudio.com/products/rstudio/download/#download
- Open RNAseq_Analysis.R (in any R project or base R) and run lines 11-31 to install packages. Be sure to run this line by line as some packages require input during installation. Answer “yes” to all prompts to compile from source and “a” to all prompts to update packages.
- Text Editor (we will use Atom in class) https://atom.io/
- Environment Manager (Anaconda) https://docs.anaconda.com/anaconda/install/mac-os/
  - Please type each command highlighted in grey below (don’t copy-paste).
    - Test installation in Terminal by running: conda –help
    - If no errors occur, run the following in Terminal and confirm tool versions after installation (python = 3.7.12, fastp = 0.22.0, multiqc = 1.12, salmon = 1.8.0):
    - cd ~/Downloads/RNAseq_Pipeline
    - conda env create -f Demo_ENV.yml
    - conda activate Demo_ENV
    - python --version
    - fastp --version
    - multiqc --version
    - salmon --version
    - conda deactivate

Windows:

- R (v4.1.3) https://cran.r-project.org/bin/windows/base/
- Rstudio (v2022.02.0 build 443) https://www.rstudio.com/products/rstudio/download/#download
- Open RNAseq_Analysis.R (in any R project or base R) and run lines 11-31 to install packages. Be sure to run this line by line as some packages require input during installation. Answer “yes” to all prompts to compile from source and “a” to all prompts to update packages.
- Text Editor (we will use Atom in class) https://atom.io/
- Bash Shell Interpreter. If you don’t have one already installed, below are instructions for installing Git Bash.
- PLEASE NOTE: some tools used in the demonstration **aren’t available for Windows OS**. In a real class, students will run tools on a remote computer such as the University HPC, which will overcome system incompatibilities. For the demonstration, I have provided the output file (RawCountData.txt) used in the analysis using R.

1. Download the Git for Windows installer.
2. Run the installer and follow the steps below:
   1. Click on "Next" four times (two times if you've previously installed Git). You don't need to change anything in the Information, location, components, and start menu screens.

2. **From the dropdown menu, "Choosing the default editor used by Git", select "Use the Nano editor by default" (NOTE: you will need to scroll up to find it) and click on "Next".**
3. On the page that says "Adjusting the name of the initial branch in new repositories", ensure that "Let Git decide" is selected. This will ensure the highest level of compatibility.
4. Ensure that "Git from the command line and also from 3rd-party software" is selected and click on "Next". (If you don't do this Git Bash will not work properly, requiring you to remove the Git Bash installation, re-run the installer and to select the "Git from the command line and also from 3rd-party software" option.)
5. Select "Use bundled OpenSSH".
6. Ensure that "Use the native Windows Secure Channel Library" is selected and click on "Next".
7. Ensure that "Checkout Windows-style, commit Unix-style line endings" is selected and click on "Next".
8. **Ensure that "Use Windows' default console window" is selected and click on "Next".**
9. Ensure that "Default (fast-forward or merge) is selected and click "Next"
10. Ensure that "Git Credential Manager" is selected and click on "Next".
11. Ensure that "Enable file system caching" is selected and click on "Next".
12. Click on "Install".
13. Click on "Finish" or "Next".

3. If your "HOME" environment variable is not set (or you don't know what this is):
    1. Open command prompt (Open Start Menu then type `cmd` and press `Enter`)
    2. Type the following line into the command prompt window exactly as shown:

        ```
        setx HOME "%USERPROFILE%"
        ```

    3. Press `Enter`, you should see `SUCCESS: Specified value was saved.`
    4. Quit command prompt by typing `exit` then pressing `Enter`