

Incorporating Snakemake into High Performance Computing Workflows

Texas A&M HPRC

Wes Brashear

28 April 2026



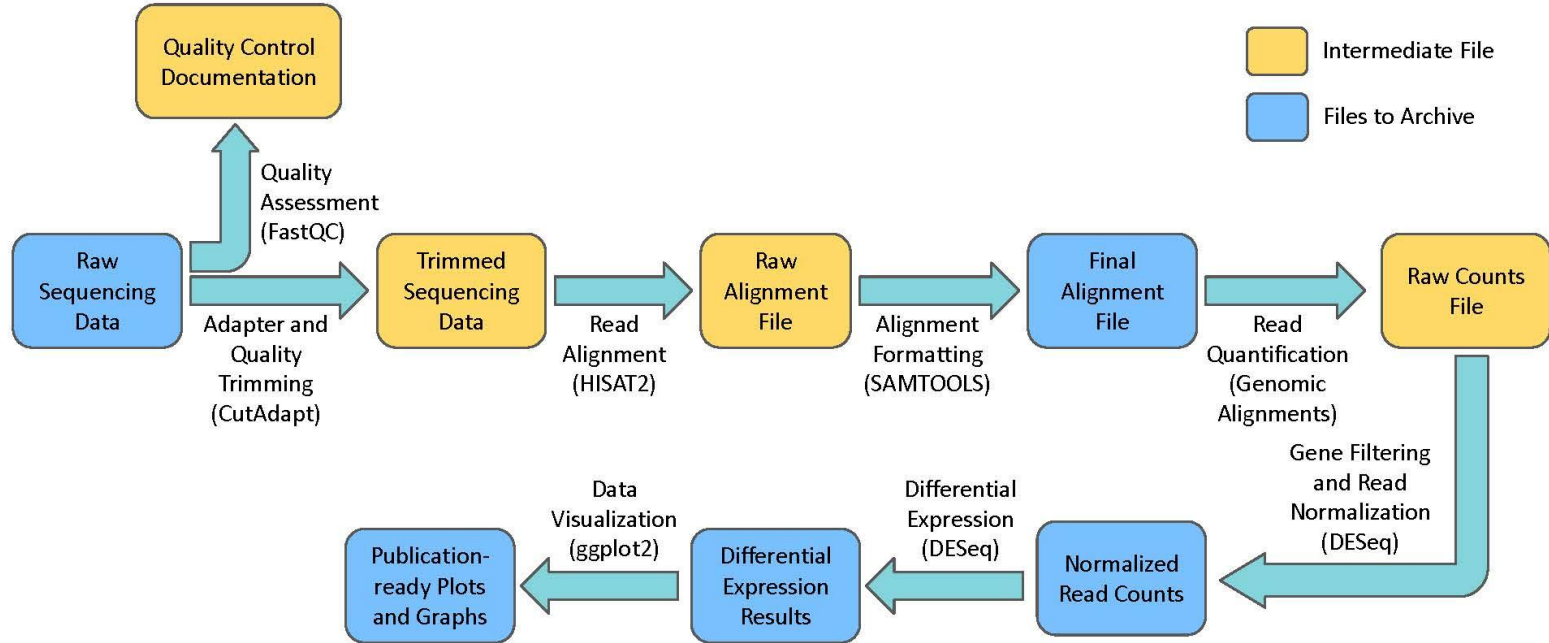
High Performance
Research Computing

DIVISION OF RESEARCH

Course Outline

- Brief Introduction
- Getting on ACES
- Our first Snakemake job
- Understanding the order of rule execution
- A Complete Bioinformatics Workflow
- Submitting Snakemake jobs through Slurm
- Generating reports
- Snakemake parallelization

The Need for Workflow Management



Snakemake

- Python-based workflow management system that allows users to easily create and share workflows
- Broad user community that is rapidly growing
- Provide access to curated and citable workflows
- Integrates with Jupyter (data visualization, report generation)
- Integrates well with batch scheduling systems



Accessing the HPRC ACES Portal

The screenshot displays the HPRC website interface. At the top, the Texas A&M logo and the text "TEXAS A&M HIGH PERFORMANCE RESEARCH COMPUTING" are visible. A navigation bar includes links for Home, User Services, Resources, Research, Policies, Events, Training, About, and Portal. The "Portal" link is highlighted with a red box. A dropdown menu from the Portal link shows options for Grace Portal, FASTER Portal, ACES Portal (ACCESS) (highlighted with a red box), and Launch Portal (ACCESS). On the left, a "Quick Links" section lists various user resources, and a "User Guides" section lists guides for Launch, ACES, FASTER, Grace, Portal, and Galaxy. The main content area features a diagram illustrating the process of studying amyloid-forming peptides. The diagram shows a 3D model of an "Amyloid-forming CPP" (colored purple, blue, and orange) interacting with a "Plasmid DNA" (green circle). This forms a "pDNA/peptide complex" (shown as a grey textured mass). The complex then interacts with a "Cell", specifically with the "Nucleus" and "membrane", leading to "internalization". The diagram is labeled "Interaction with the negatively charged cell membrane and internalization". Below the diagram, the text "peptide fibrils" and "Using computational methods to study, engineer and design" are visible.

Quick Links

- New User Information
- Accounts
 - Apply for Accounts
 - Manage Accounts
- User Consulting
- Training
- Knowledge Base
- Software
- FAQ

User Guides

- Launch
- ACES
- FASTER
- Grace
- Portal
- Galaxy

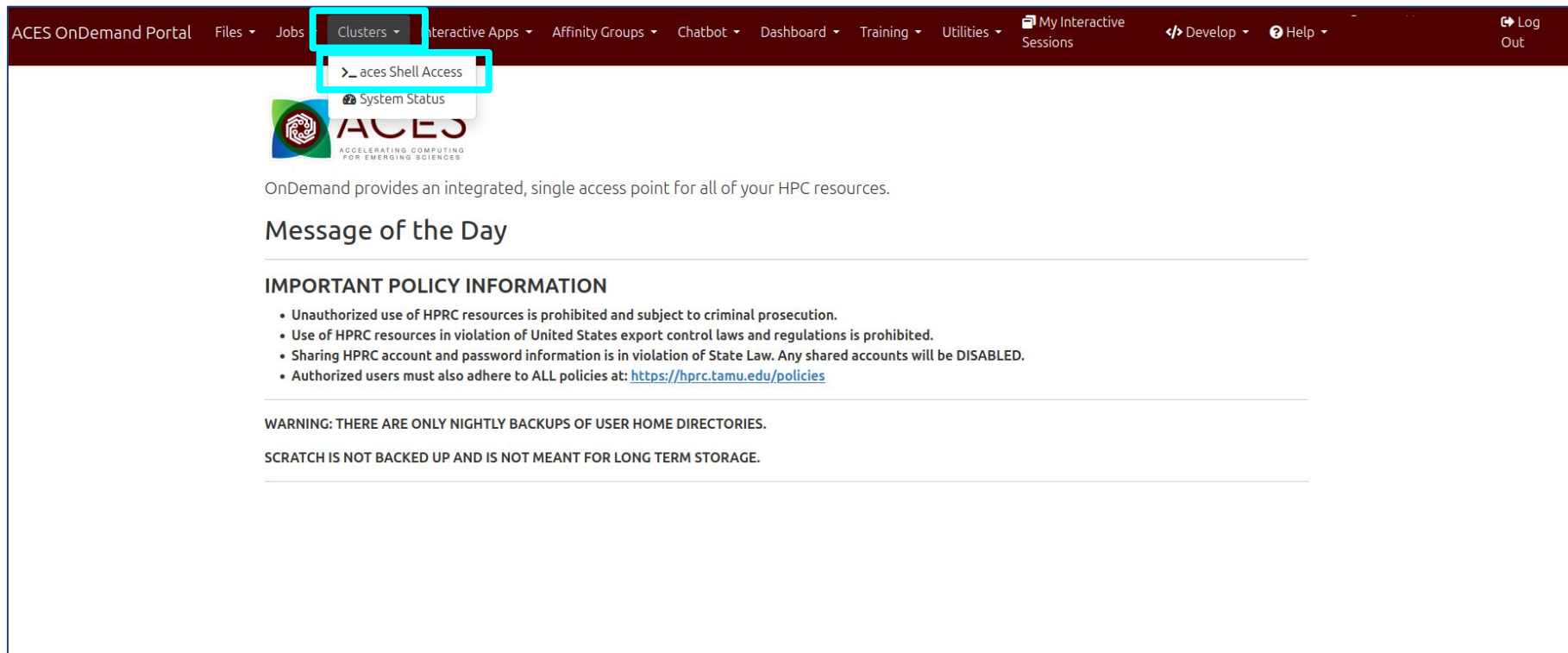
Cluster Status

[Launch](#)

Diagram Labels:

- Amyloid-forming CPP
- Plasmid DNA
- pDNA/peptide complex
- peptide fibrils
- Using computational methods to study, engineer and design
- Interaction with the negatively charged cell membrane and internalization
- Cell
- Nucleus

Accessing ACES Shell in OOD Portal



ACES OnDemand Portal Files Jobs Clusters Interactive Apps Affinity Groups Chatbot Dashboard Training Utilities My Interactive Sessions Develop Help Log Out

aces Shell Access

System Status

 **ACES**
ACCELERATING COMPUTING
FOR EMERGING SCIENCES

OnDemand provides an integrated, single access point for all of your HPC resources.

Message of the Day

IMPORTANT POLICY INFORMATION

- Unauthorized use of HPRC resources is prohibited and subject to criminal prosecution.
- Use of HPRC resources in violation of United States export control laws and regulations is prohibited.
- Sharing HPRC account and password information is in violation of State Law. Any shared accounts will be DISABLED.
- Authorized users must also adhere to ALL policies at: <https://hprc.tamu.edu/policies>

WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES.

SCRATCH IS NOT BACKED UP AND IS NOT MEANT FOR LONG TERM STORAGE.

Accessing ACES Shell in OOD Portal

```
portal-aces.hprc.tamu.edu/pun/sys/shell/ssh/login.aces
Host: login.aces

hosts.
*****
This computer system and the data herein are available only for authorized
purposes by authorized users. Use for any other purpose is prohibited and may
result in disciplinary actions or criminal prosecution against the user. Usage
may be subject to security testing and monitoring. There is no expectation of
privacy on this system except as otherwise provided by applicable privacy laws.
Refer to University SAP 29.01.03.M0.02 Acceptable Use for more information.
*****

Last login: Thu Apr 23 15:14:31 2026 from 10.71.1.6

=====
Texas A&M University High Performance Research Computing
=====
Consulting:      help@hprc.tamu.edu (preferred) or (979) 845-0219
Website:         https://hprc.tamu.edu
Knowledgebase:   https://hprc.tamu.edu/kb/
YouTube Channel: https://www.youtube.com/texasamhprc
=====

*****
=== IMPORTANT POLICY INFORMATION ===
* - Unauthorized use of HPRC resources is prohibited and subject to
*   criminal prosecution.
* - Use of HPRC resources in violation of United States export control
*   laws and regulations is prohibited.
* - Sharing HPRC account and password information is in violation of
*   Texas State Law. Any shared accounts will be DISABLED.
* - Authorized users must also adhere to ALL policies at:
*   https://hprc.tamu.edu/policies/
*****

!! WARNING: THERE ARE ONLY NIGHTLY BACKUPS OF USER HOME DIRECTORIES. !!
!! SCRATCH IS NOT BACKED UP AND IS NOT MEANT FOR LONG TERM STORAGE. !!

Please restrict usage to 8_CORES across ALL login nodes.
Users found in violation of this policy will be SUSPENDED.

To see these messages again, run the motd command.

Your current disk quotas are:
Disk      Disk Usage  Limit  File Usage  Limit
/home/u.wb109972      3.4G    10.0G    4033    10000
/scratch/user/u.wb109972  2.3T    5.0T   175205   250000
/scratch/group/p.tra230003.000    4K    1.0T      1   500000
/scratch/group/p.tra220029.000   12.5G   1.0T   10287   500000
/scratch/group/p.sta220004.000    3.7G   1.0T   21746   500000
/scratch/group/p.bio240022.000    3.1G   1.0T     695   500000
Type 'showquota' to view these quotas again.
[u.wb109972@aces-login3 ~]$
```

Get course data and set up the environment

- Run the following commands to change your working directory and copy the material for today's course:

```
cd $SCRATCH
```

```
cp -r /scratch/training/snakemake_course .
```

- Run the following command to load the required modules

```
module load GCC/13.2.0 OpenMPI/4.1.6 R_tamu/4.4.1 snakemake/8.28.0
```


Exercise 1 - Our first snakemake job

- Navigate to the ex01 directory:

```
cd snakemake_tutorial/ex01-03
```

- List the contents of the directory

```
ls -lth
```

Output:

```
total 28K
-rw-r--r-- 1 u.wb109972 u.wb109972 20K Apr 27 14:54 penguin.csv
-rw-r--r-- 1 u.wb109972 u.wb109972 781 Apr 27 14:54 plot_penguins.R
-rw-r--r-- 1 u.wb109972 u.wb109972 264 Apr 27 14:54 Snakefile
```

Exercise 1 - Our first snakemake job

- Look at the penguin.csv and plot_penguins.R files in the Launch file browser
- Run the following command to execute Snakemake:

```
snakemake --cores 1 penguin_plot.pdf
```

Exercise 1

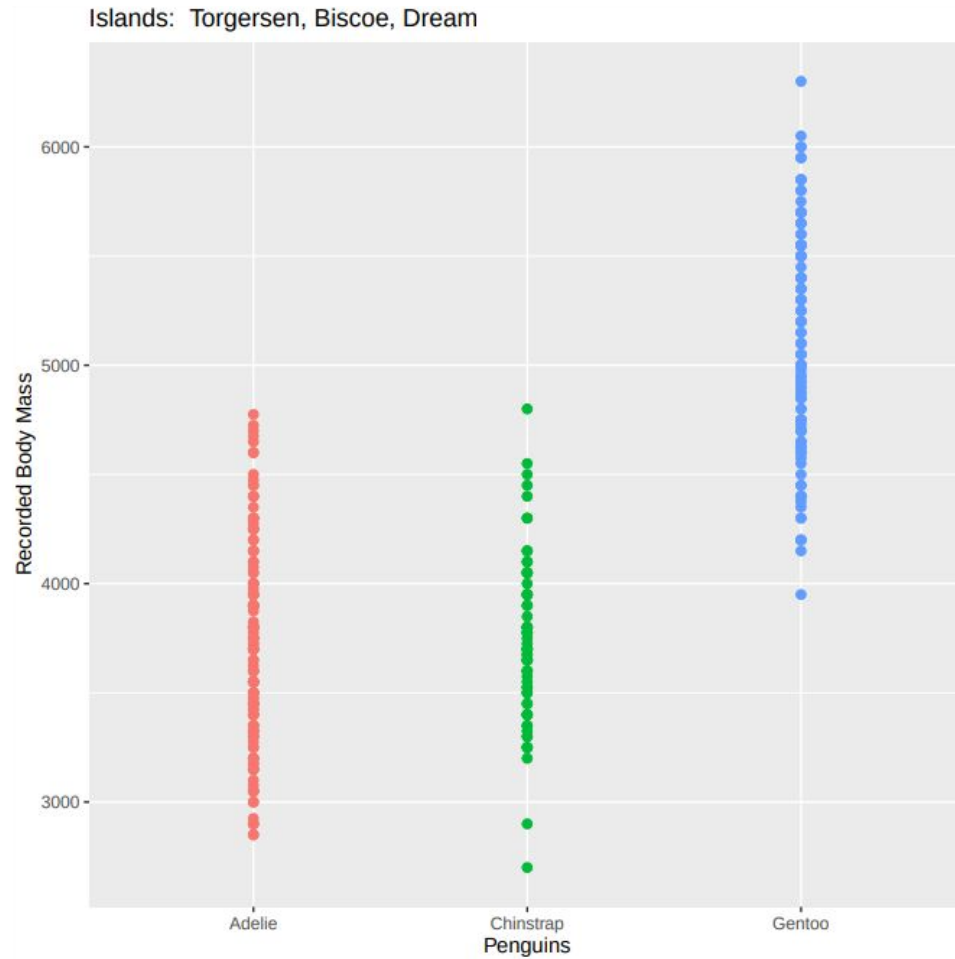
```
Assuming unrestricted shared filesystem usage.
host: alogin2.cluster
Building DAG of jobs...
Using shell: /usr/bin/bash
Provided cores: 1 (use --cores to define parallelism)
Rules claiming more threads will be scaled down.
Job stats:
job      count
-----
plot      1
total     1

Select jobs to execute...
Execute 1 jobs...

[Mon Apr 27 16:35:58 2026]
localrule plot:
  input: penguin.csv
  output: penguin_plot.pdf
  jobid: 0
  reason: Missing output files: penguin_plot.pdf
  wildcards: data=penguin
  resources: tmpdir=/tmp

Saving 7 x 7 in image
[Mon Apr 27 16:36:03 2026]
Finished job 0.
1 of 1 steps (100%) done
Complete log: .snakemake/log/2026-04-27T163557.943573.snakemake.log
```

Exercise 1



Exercise 1

Snakefile:

```
rule plot:
    output: "{data}_plot.pdf"
    input:  "{data}.csv"
    shell:  "Rscript plot_penguins.R -i {input} -o {output}"

rule biscoe_island:
    output: "{data}_biscoe.csv"
    input:  "{data}.csv"
    shell:  "grep -e species -e Biscoe {input} > {output}"
```

Command:

```
snakemake --cores 1 penguin_plot.pdf
```

Exercise 1

Snakefile: **Snakemake recognizes the {data} == penguin from our command**

```
rule plot:
    output: "{data}_plot.pdf" → penguin_plot.pdf
    input:  "{data}.csv" → penguin.csv
    shell:  "Rscript plot_penguins.R -i {input} -o {output}"

rule biscoe_island:
    output: "{data}_biscoe.csv"
    input:  "{data}.csv"
    shell:  "grep -e species -e Biscoe {input} > {output}"
```

Command:

```
snakemake --cores 1 penguin_plot.pdf
```

Exercise 1 - Why isn't the second rule executed?

Snakefile:

```
rule plot:
    output: "{data}_plot.pdf"
    input:  "{data}.csv"
    shell:  "Rscript plot_penguins.R -i {input} -o {output}"

rule biscoe_island:
    output: "{data}_biscoe.csv"
    input:  "{data}.csv"
    shell:  "grep -e species -e Biscoe {input} > {output}"
```

Snakemake builds the simplest DAG (directed acyclic graph) to produce the results:

{data}.csv $\Longrightarrow$ **Rscript** $\Longrightarrow$ **{data}_plot.pdf**

Exercise 2 - Changing requested output to run both rules

- Run the following command to execute Snakemake:

```
snakemake --cores 1 penguin_biscoe_plot.pdf
```


Exercise 2

```
Assuming unrestricted shared filesystem usage.  
host: alogin1.cluster  
Building DAG of jobs...  
Using shell: /usr/bin/bash  
Provided cores: 1 (use --cores to define parallelism)  
Rules claiming more threads will be scaled down.  
Job stats:
```

job	count
-----	-----
biscoe_island	1
plot	1
total	2

```
Select jobs to execute...  
Execute 1 jobs...
```

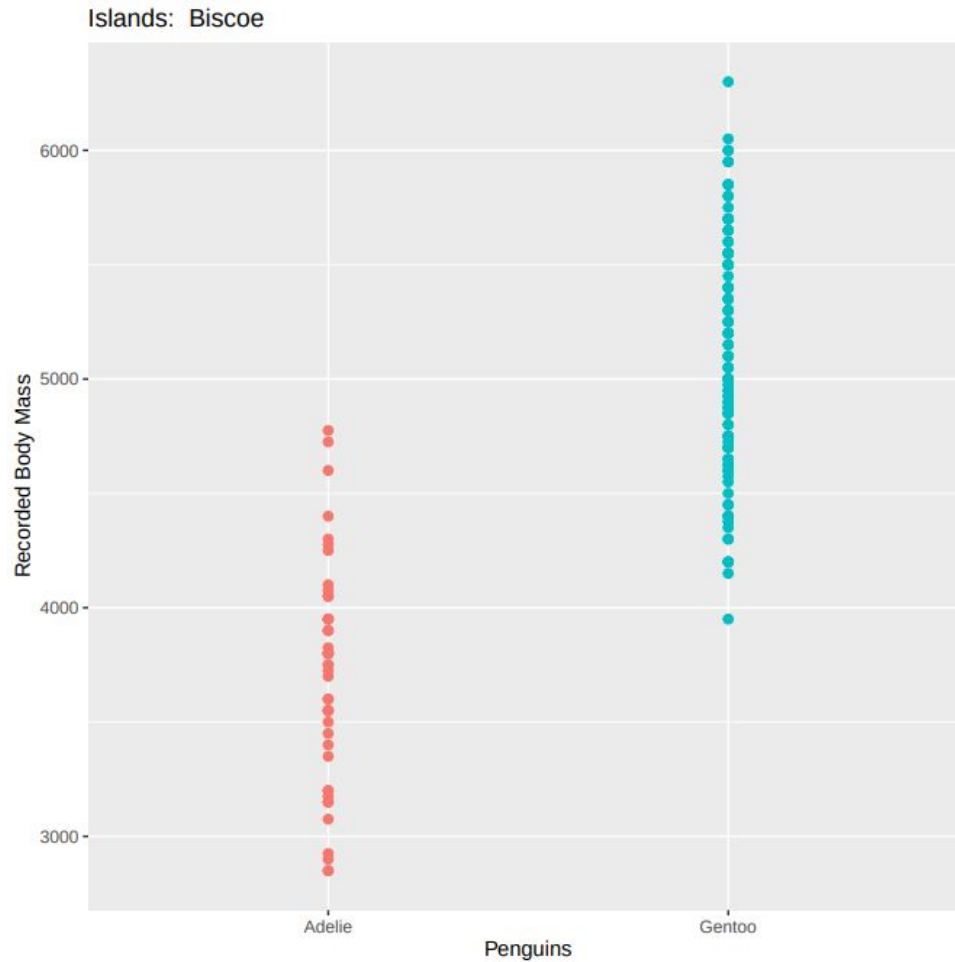
```
[Mon Apr 27 16:56:54 2026]  
localrule biscoe_island:  
    input: penguin.csv  
    output: penguin_biscoe.csv  
    jobid: 1  
    reason: Missing output files: penguin_biscoe.csv  
    wildcards: data=penguin  
    resources: tmpdir=/tmp
```

```
[Mon Apr 27 16:56:55 2026]  
Finished job 1.  
1 of 2 steps (50%) done  
Select jobs to execute...  
Execute 1 jobs...
```

```
[Mon Apr 27 16:56:55 2026]  
localrule plot:  
    input: penguin_biscoe.csv  
    output: penguin_biscoe_plot.pdf  
    jobid: 0  
    reason: Missing output files: penguin_biscoe_plot.pdf;  
Input files updated by another job: penguin_biscoe.csv  
    wildcards: data=penguin_biscoe  
    resources: tmpdir=/tmp
```

```
Saving 7 x 7 in image  
[Mon Apr 27 16:56:58 2026]  
Finished job 0.  
2 of 2 steps (100%) done  
Complete log:  
.snakemake/log/2026-04-27T165654.923742.snakemake.log
```

Exercise 2



Exercise 2 - Why were both rules executed?

Snakefile:

```
rule plot:
    output: "{data}_plot.pdf"
    input:  "{data}.csv"
    shell:  "Rscript plot_penguins.R -i {input} -o {output}"

rule biscoe_island:
    output: "{data}_biscoe.csv"
    input:  "{data}.csv"
    shell:  "grep -e species -e Biscoe {input} > {output}"
```

DAG:

{data}.csv $\Rightarrow$ rule biscoe_island $\Rightarrow$ {data}_biscoe.csv $\Rightarrow$ rule plot $\Rightarrow$ {data}_plot.pdf

Exercise 3

- Run the following command again:

```
snakemake --cores 1 penguin_plot.pdf
```

Exercise 3

- Run the following command again:

```
snakemake --cores 1 penguin_plot.pdf
```

- Output:

```
Building DAG of jobs...  
Nothing to be done (all requested files are present and up to date).
```

Exercise 3 (continued)

- Delete the last line of penguin.csv and run the following command again:

```
snakemake --cores 1 penguin_plot.pdf
```

- Output:

```
host: alogin1.cluster
Building DAG of jobs...
Using shell: /usr/bin/bash
Provided cores: 1 (use --cores to define parallelism)
Rules claiming more threads will be scaled down.
Job stats:
job      count
-----  -
plot      1
total    1

Select jobs to execute...
Execute 1 jobs...
```

```
[Mon Apr 27 17:14:02 2026]
localrule plot:
  input: penguin.csv
  output: penguin_plot.pdf
  jobid: 0
  reason: Updated input files: penguin.csv
  wildcards: data=penguin
  resources: tmpdir=/tmp

Saving 7 x 7 in image
[Mon Apr 27 17:14:06 2026]
Finished job 0.
1 of 1 steps (100%) done
Complete log: .snakemake/log/2026-04-27T171402.906747.snakemake.log
```

Exercise 4 - A Complete Bioinformatics Workflow

1. Change directories to ../ex04

```
cd ../ex04
```

2. Purge the modules that are currently loaded and load the modules necessary to run Snakemake, FastQC, Trim_Galore!, HISAT2, and SAMtools (Hint: you will need to ensure that all of the loaded modules have compatible dependencies).

```
# Use module avail <pkg> to find compatible versions
module load GCC/13.3.0 OpenMPI/5.0.3 snakemake/8.27.0 \
    FastQC/0.11.9-Java-11 Trim_Galore/0.6.11 \
    HISAT2/2.2.1 SAMtools/1.22
```

Exercise 4 - Snakefile

```
# Threads per step
FASTQC_THREADS = 2
TRIM_THREADS   = 8
ALIGN_THREADS  = 8
SORT_THREADS   = 8

# Samples
SAMPLES = ["Control1"]

# Ensure full pipeline is run
rule all:
    input:
        expand("FastQC/{sample}_R1_fastqc.html", sample = SAMPLES),
        expand("trimming/{sample}_R1_val_1.fq.gz", sample = SAMPLES),
        expand("align/{sample}.sam", sample = SAMPLES)
```


Exercise 4 - Snakefile (continued)

```
# Run FastQC on raw illumina data
rule fastqc:
    input:
        r1 = "raw_fastqs/{sample}_R1.fastq.gz",
        r2 = "raw_fastqs/{sample}_R2.fastq.gz"
    output:
        "FastQC/{sample}_R1_fastqc.html",
        "FastQC/{sample}_R1_fastqc.zip",
        "FastQC/{sample}_R2_fastqc.html",
        "FastQC/{sample}_R2_fastqc.zip"
    threads: FASTQC_THREADS
    shell:
        """
        mkdir -p FastQC
        fastqc -t {threads} -o FastQC {input.r1} {input.r2}
        """
```

Exercise 4 - Snakefile (continued)

```
# Trim libraries with Trim_Galore!
rule trim_galore:
    input:
        r1 = "raw_fastqs/{sample}_R1.fastq.gz",
        r2 = "raw_fastqs/{sample}_R2.fastq.gz"
    output:
        val1 = "trimming/{sample}_R1_val_1.fq.gz",
        val2 = "trimming/{sample}_R2_val_2.fq.gz",
        rep1 = "trimming/{sample}_R1.fastq.gz_trimming_report.txt",
        rep2 = "trimming/{sample}_R2.fastq.gz_trimming_report.txt"
    threads: TRIM_THREADS
    shell:
        """
        mkdir -p trimming
        trim_galore --paired --fastqc --output_dir trimming --cores {threads} {input.r1} {input.r2}
        """
```

Exercise 4 - Snakefile (continued)

```
# Map trimmed reads to genome with HISAT2
rule hisat:
    input:
        r1 = "trimming/{sample}_R1_val_1.fq.gz",
        r2 = "trimming/{sample}_R2_val_2.fq.gz"
    output: "align/{sample}.sam"
    threads: ALIGN_THREADS
    params:
        index = "../mm39/GCF_000001635.27_GRCm39_genomic"
    shell:
        """
        mkdir -p align
        hisat2 -x {params.index} -1 {input.r1} -2 {input.r2} -p {threads} \
            --no-mixed --no-discordant -S {output}
        """

# Sort sam file
rule samtools:
    input: "align/{sample}.sam"
    output: "align/{sample}_sorted.bam"
    threads: SORT_THREADS
    shell: "samtools sort -@ {threads} -O bam -o {output} {input}"
```

Exercise 4

Run snakemake:

```
snakemake --cores 8
```

Exercise 5 - Submitting snakemake jobs to Slurm

Navigate to the ex05 directory. Create a batch script to run the snakemake job from exercise 2. Submit the job to Slurm.



Exercise 5 - Submitting snakemake jobs to Slurm

Solution:

```
#!/bin/bash
#SBATCH --job-name=plotting_penguins
#SBATCH --time=00:02:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=1G
#SBATCH --output=%x.%j.stdout
#SBATCH --error=%x.%j.stderr

# Set up the environment
module purge
module load GCC/13.2.0 OpenMPI/4.1.6 R_tamu/4.4.1 snakemake/8.28.0

snakemake --cores 1 penguin_biscoe_plot.pdf
```

Exercise 6 - Submitting snakemake jobs to Slurm (multiple cores)

Create a batch script to run our RNA workflow snakemake job. Submit the job to Slurm.



Exercise 6 - Submitting snakemake jobs to Slurm (multiple cores)

Solution:

```
#!/bin/bash
#SBATCH --job-name=rna_workflow
#SBATCH --time=00:05:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=8
#SBATCH --mem=10G
#SBATCH --output=%x.%j.stdout
#SBATCH --error=%x.%j.stderr

# Set up the environment
module purge
module load GCC/12.3.0 OpenMPI/4.1.5 snakemake/8.4.2 FastQC/0.12.1-Java-11
module load Trim_Galore/0.6.10 HISAT2/2.2.1 SAMtools/1.18

snakemake --cores 8
```


Exercise 7 - Generating a simple report

1. Change your working directory to ex07
2. Run snakemake:

```
snakemake --cores 8
```

3. Generate the report:

```
snakemake --report pipeline_report.html
```

Exercise 7 - Generating a simple report

Snakefile:

```
# Threads per step
FASTQC_THREADS = 2
TRIM_THREADS    = 8
ALIGN_THREADS   = 8
SORT_THREADS    = 8

# Samples
SAMPLES = ["Control1", "Control2"]

# Ensure full pipeline is run
rule all:
    input:
        expand("FastQC/{sample}_R1_fastqc.html", sample=SAMPLES),
        expand("FastQC/{sample}_R2_fastqc.html", sample=SAMPLES),
        expand("trimming/{sample}_R1.fastq.gz_trimming_report.txt", sample=SAMPLES),
        expand("trimming/{sample}_R2.fastq.gz_trimming_report.txt", sample=SAMPLES),
```

```

# Run FastQC on raw illumina data
rule fastqc:
    input:
        r1 = "../fastqs/{sample}_R1.fastq.gz",
        r2 = "../fastqs/{sample}_R2.fastq.gz"
    output:
        report("FastQC/{sample}_R1_fastqc.html"),
        "FastQC/{sample}_R1_fastqc.zip",
        report("FastQC/{sample}_R2_fastqc.html"),
        "FastQC/{sample}_R2_fastqc.zip"
    threads: FASTQC_THREADS
    shell:
        """
        mkdir -p FastQC
        fastqc -t {threads} -o FastQC {input.r1} {input.r2}
        """

# Trim libraries with Trim_Galore!
rule trim_galore:
    input:
        r1 = "../fastqs/{sample}_R1.fastq.gz",
        r2 = "../fastqs/{sample}_R2.fastq.gz"
    output:
        "trimming/{sample}_R1_val_1.fq.gz",
        "trimming/{sample}_R2_val_2.fq.gz",
        report("trimming/{sample}_R1.fastq.gz_trimming_report.txt"),
        report("trimming/{sample}_R2.fastq.gz_trimming_report.txt")
    threads: TRIM_THREADS
    shell:
        """
        mkdir -p trimming
        trim_galore --paired --fastqc --output_dir trimming --cores {threads} {input.r1} {input.r2}
        """

```

Exercise 8 - Examining Runtimes

1. Copy the Snakefile from Exercise 7
2. Modify the input to include the following samples from the “fastqs” folder in the parent directory:
 - Control 1
 - Control 2
 - Control 3
3. Add mapping with HISAT2 as an additional rule (make sure to include the results in the report)
4. Create a batch script to submit the job to slurm using 48 cores and 120Gb of memory
5. You will need to load the module WebProxy for `snakemake --report` to work

Exercise 8 - Snakefile modification

```
# Map trimmed reads to genome with HISAT2
rule hisat:
    input:
        r1 = "trimming/{sample}_R1_val_1.fq.gz",
        r2 = "trimming/{sample}_R2_val_2.fq.gz"
    output:
        sam = "align/{sample}.sam",
        summary = report("align/{sample}_summary.txt")
    threads: ALIGN_THREADS
    params:
        index = "../mm39/GCF_000001635.27_GRCm39_genomic"
    shell:
        """
        mkdir -p align
        hisat2 -x {params.index} -1 {input.r1} -2 {input.r2} -p {threads} --no-mixed \
            --no-discordant -S {output.sam} --summary-file {output.summary}
        """
```

Exercise 8 - Slurm script

```
#!/bin/bash
#SBATCH --job-name=rna_workflow
#SBATCH --time=00:30:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=48
#SBATCH --mem=120G
#SBATCH --output=%x.%j.stdout
#SBATCH --error=%x.%j.stderr

# Set up the environment
module purge
module load GCC/13.3.0 OpenMPI/5.0.3 snakemake/8.27.0 FastQC/0.11.9-Java-11
module load Trim_Galore/0.6.11 HISAT2/2.2.1 SAMtools/1.22

snakemake --cores 48
snakemake --report
```

Exercise 9 - Examining Runtimes

1. Copy the Snakefile and batch script from Exercise 8
2. Add the following samples to the Snakefile:
 - Treatment 1
 - Treatment 2
 - Treatment 3
3. Compare runtimes and resource utilization between the runs for Exercise 8 and 9

Exercise 9 - Examining Runtimes

Three Samples:

```
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 48
CPU Utilized: 05:30:27
CPU Efficiency: 39.12% of 14:04:48
core-walltime
Job Wall-clock time: 00:17:36
Memory Utilized: 12.01 GB
Memory Efficiency: 3.24% of 371.00 GB
```

Six Samples:

```
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 48
CPU Utilized: 12:06:51
CPU Efficiency: 60.49% of 20:01:36
core-walltime
Job Wall-clock time: 00:25:02
Memory Utilized: 20.83 GB
Memory Efficiency: 5.61% of 371.00 GB
```




**HIGH PERFORMANCE
RESEARCH COMPUTING**
TEXAS A&M UNIVERSITY

<https://hprc.tamu.edu>

HPRC Helpdesk:

help@hprc.tamu.edu

Phone: 979-845-0219

Take our short course survey!



HPRC Survey

https://u.tamu.edu/hprc_shortcourse_survey

Help us help you. Please include details in your request for support, such as, Cluster (FASTER, Grace, ACES, Launch), NetID (UserID), Job information (Job id(s), Location of your jobfile, input/output files, Application, Module(s) loaded, Error messages, etc), and Steps you have taken, so we can reproduce the problem.

