Workflow management introduction

FAS Informatics

Fall 2024

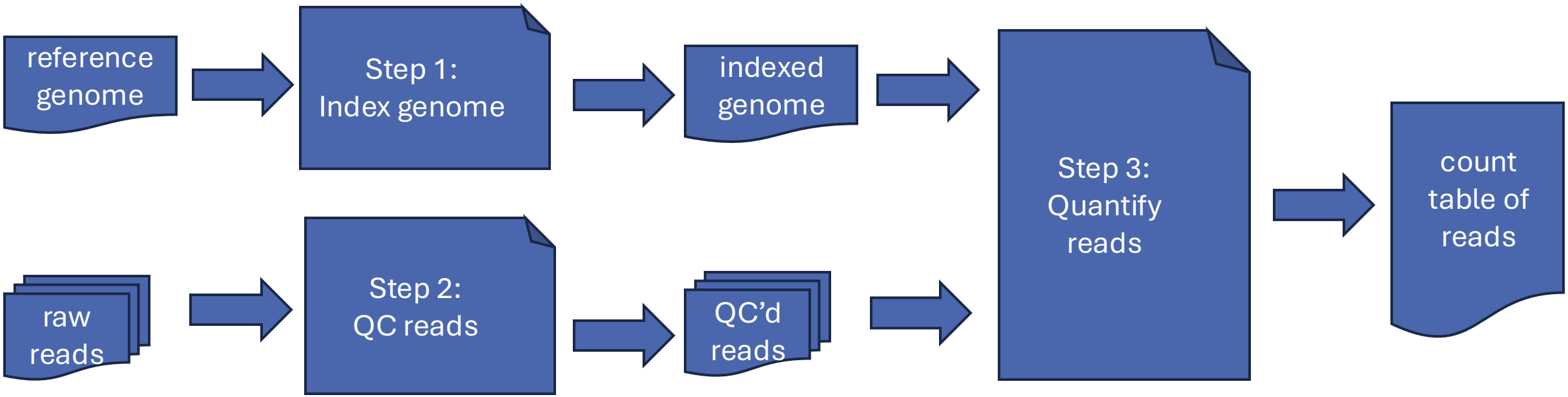
Covered in this seminar

- What and Why
- Conceptual overview of how it works
- Comparison of snakemake and nextflow
- Demonstration of nextflow on real pipeline and more detail on cool features
- Where to learn for yourself

Not covered

- Syntax on snakemake or nextflow
- How to configure workflows for SLURM or Cannon
- Details on any other workflow managers
- How to run community made workflows

What is a bioinformatics/data workflow?



- Computational actions organized into steps
- Can be in separate scripts or in a single long file
- Steps may be repeated across multiple files or variables

What if a program could do all this for you?

- Execute your workflow
  - multiple times reproducibly
  - stop/start in the middle
  - add files without re-running steps for old files
  - automatically
- Document your workflow
  - exact code run
  - software version used
  - order of operations and input/output
- Debug your workflow
  - benchmarking resource use
- Share/publish your workflow

COMMON
WORKFLOW
LANGUAGE

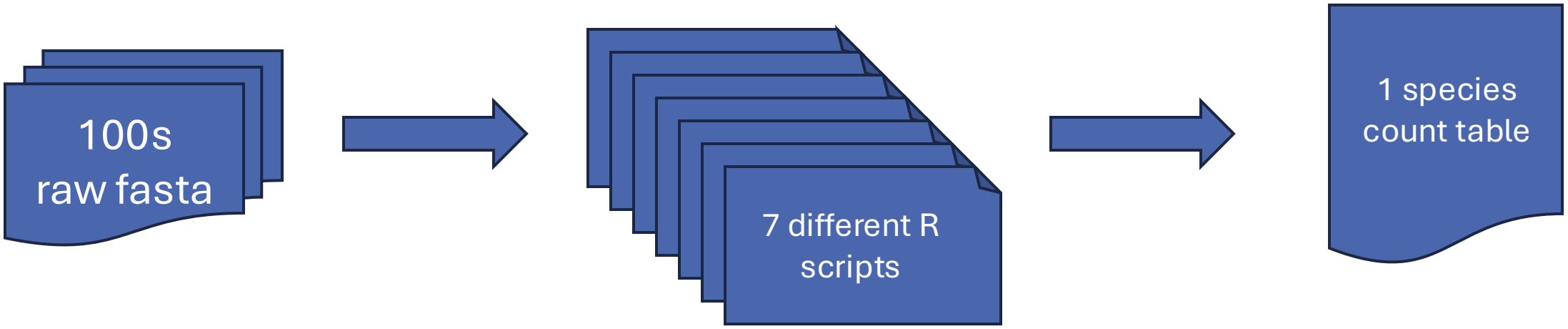
snakemake

nextflow

It’s not AI/LLMs…It’s workflow managers

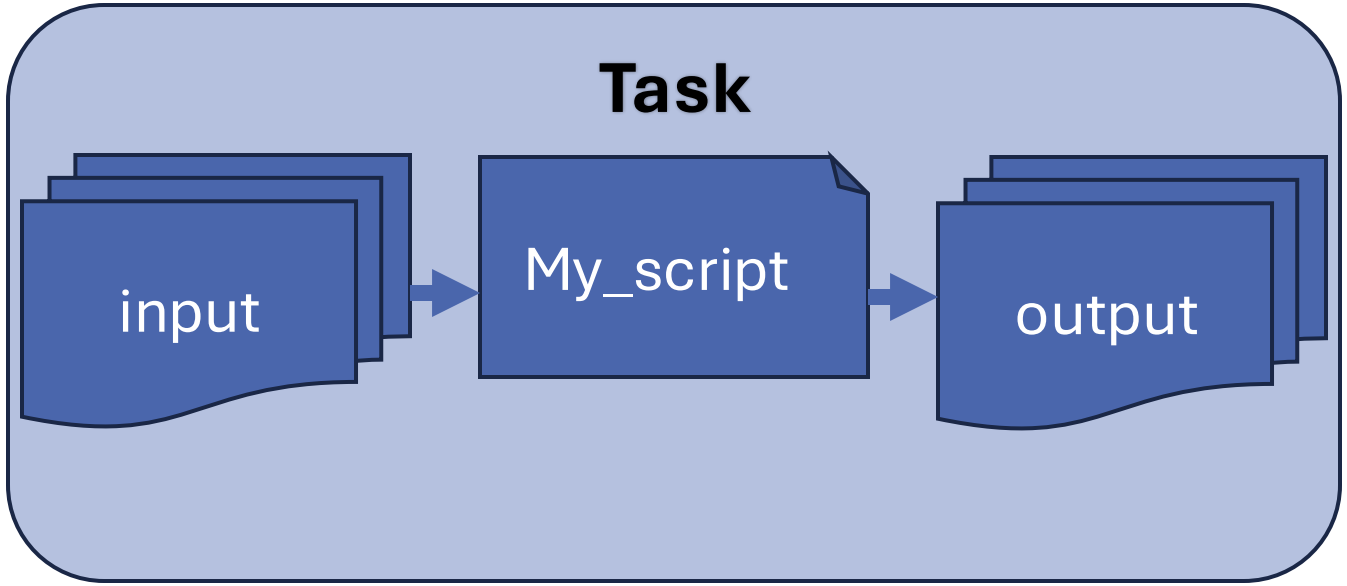
Workflow managers like snakemake and nextflow will make you love data processing and feel like a god

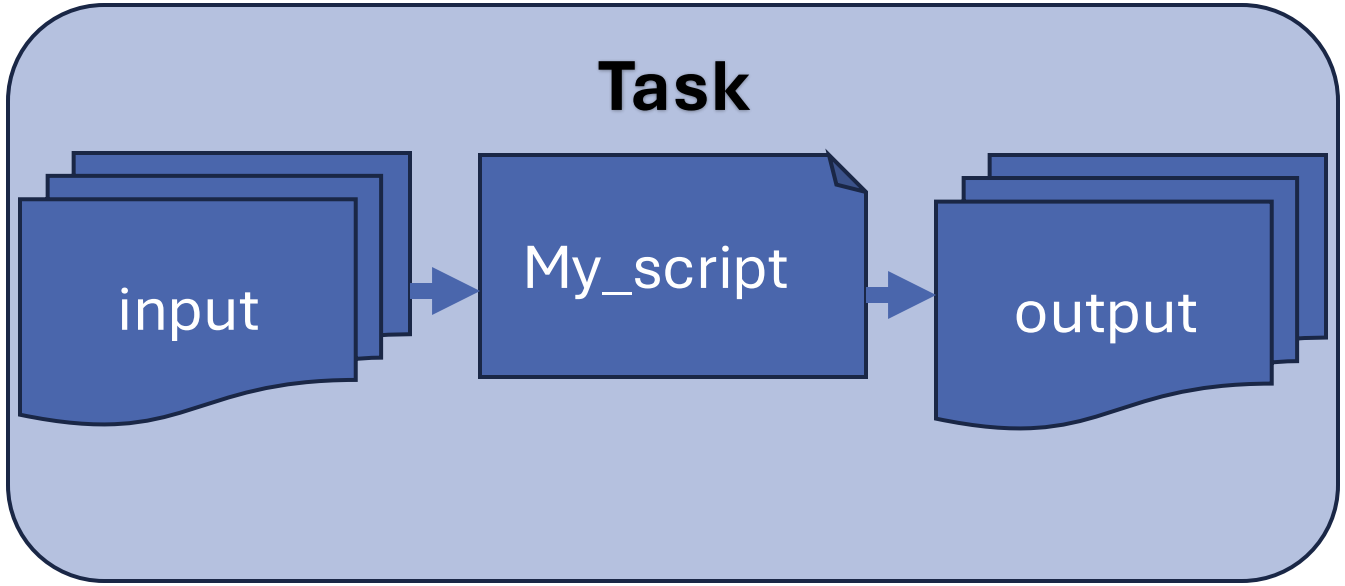
A story about my first workflow



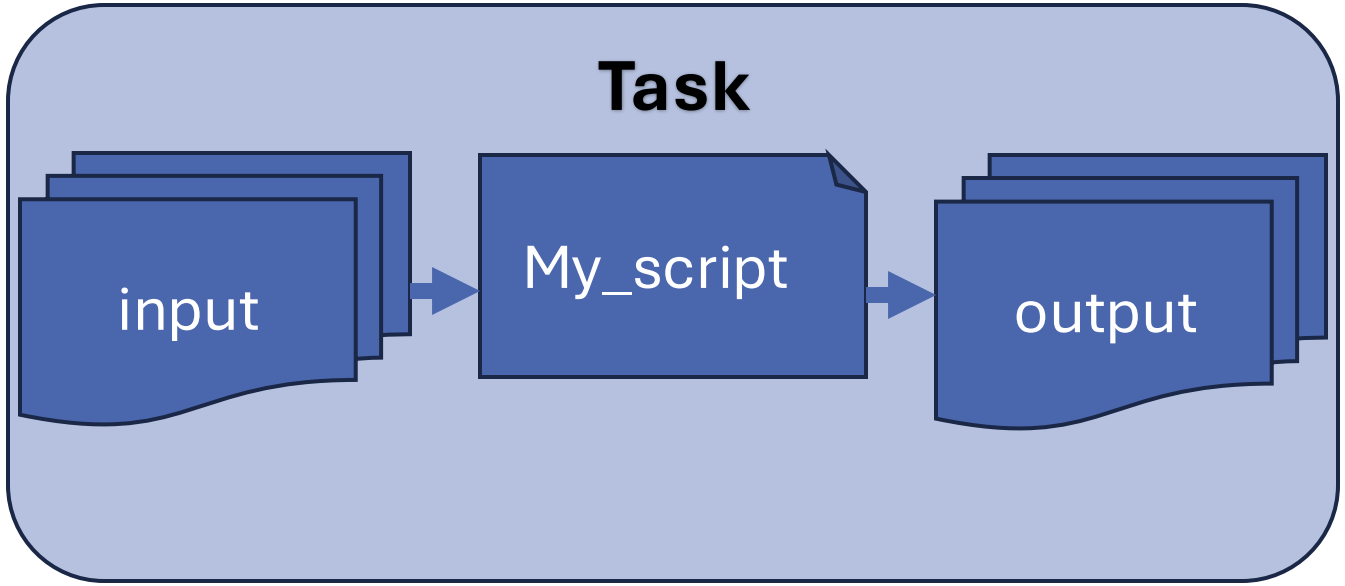

```
Lei@HPC ~/proj_dir $ snakemake
> ...
> done!
```

A workflow is made from distinct tasks






```
bwa mem data/genome.fa data/samples/file1.fastq | samtools view -Sb - > file1.bam
```




```
bwa mem data/genome.fa data/samples/file1.fastq | samtools view -Sb - > file1.bam
```

```
bwa mem {input}| samtools view -Sb - > {output}
```

A rule (aka task) in snakemake

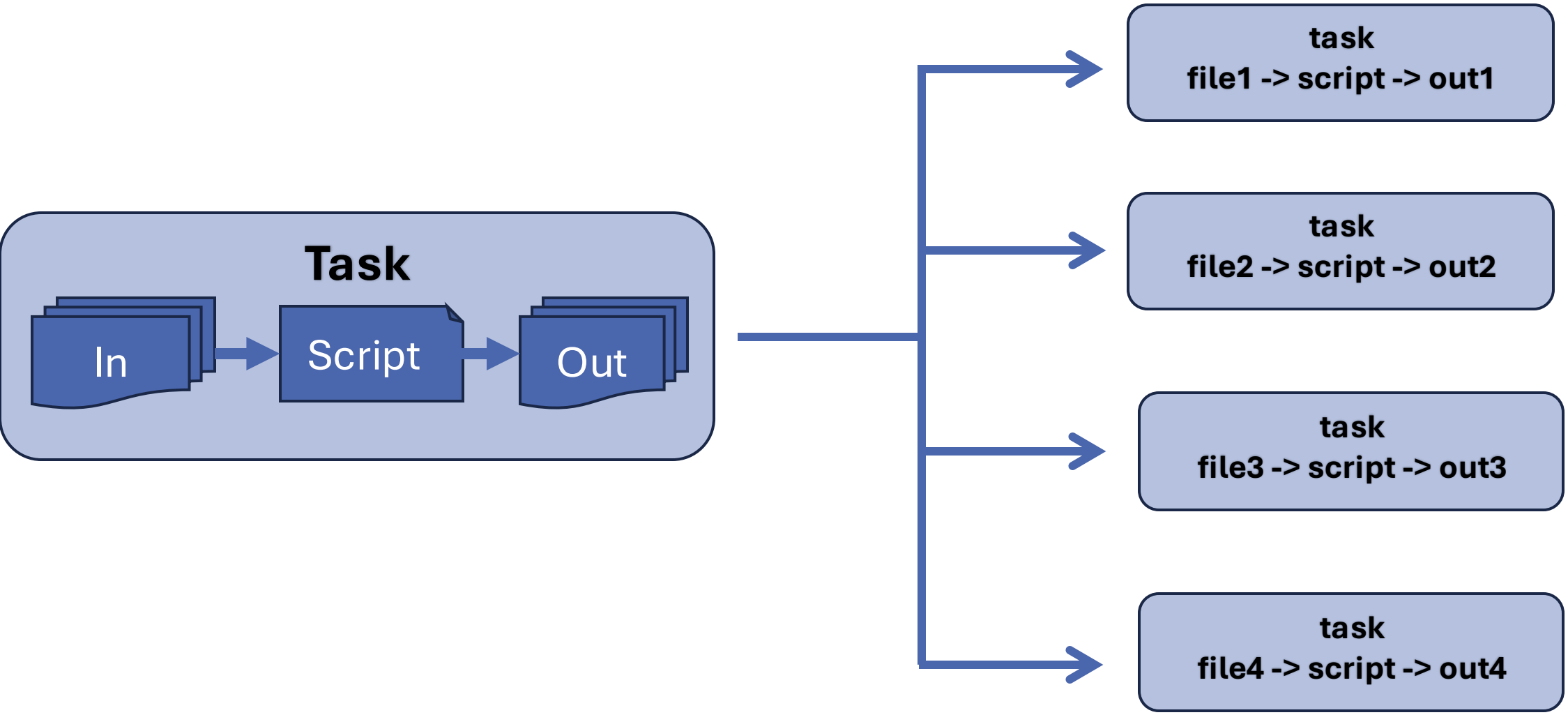
```
rule bwa_map:
    input:
        genome = "data/genome.fa",
        sample = "data/samples/{sample}.fastq"
    output:
        "mapped_reads/{sample}.bam"
    shell:
        "bwa mem {input.genome} {input.sample} |
        samtools view -Sb - > {output}"
```

A process (aka task) in Nextflow

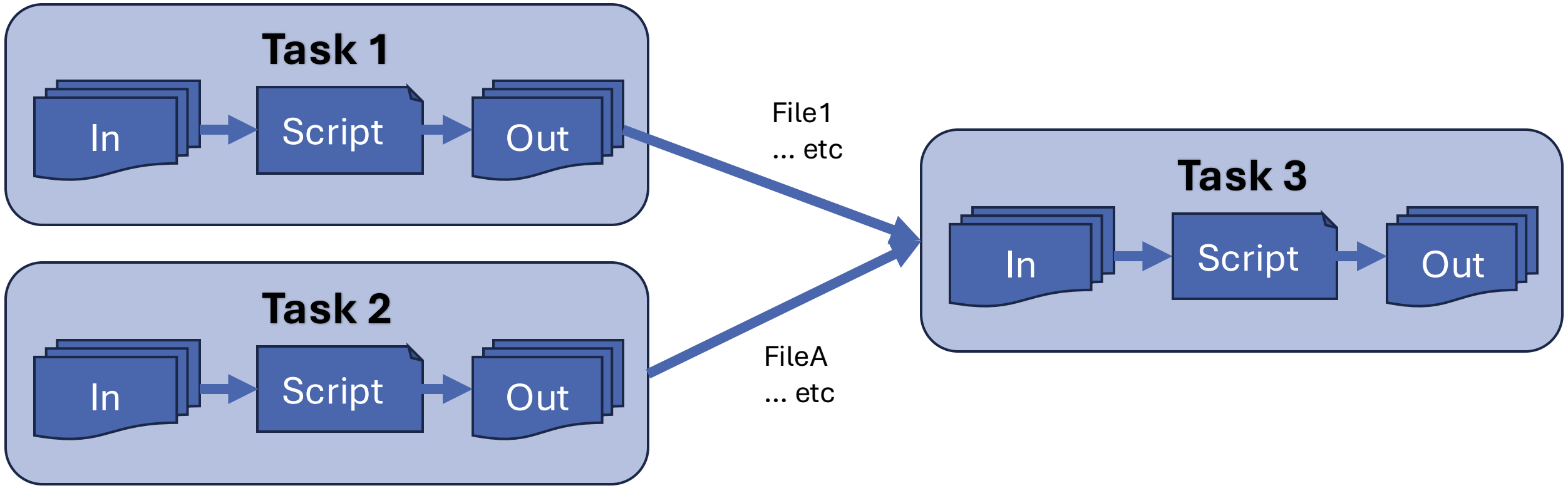
```
process bwa_map {
  input:
    path genome from 'data/genome.fa’
    path sampleFile from 'data/samples/*.fastq’
  output:
    path "mapped_reads/${sampleFile}.bam"

  script:
    """
    bwa mem $genome $sampleFile | \
    samtools view -Sb - > mapped_reads/${sampleFile}.bam
    """
}
```

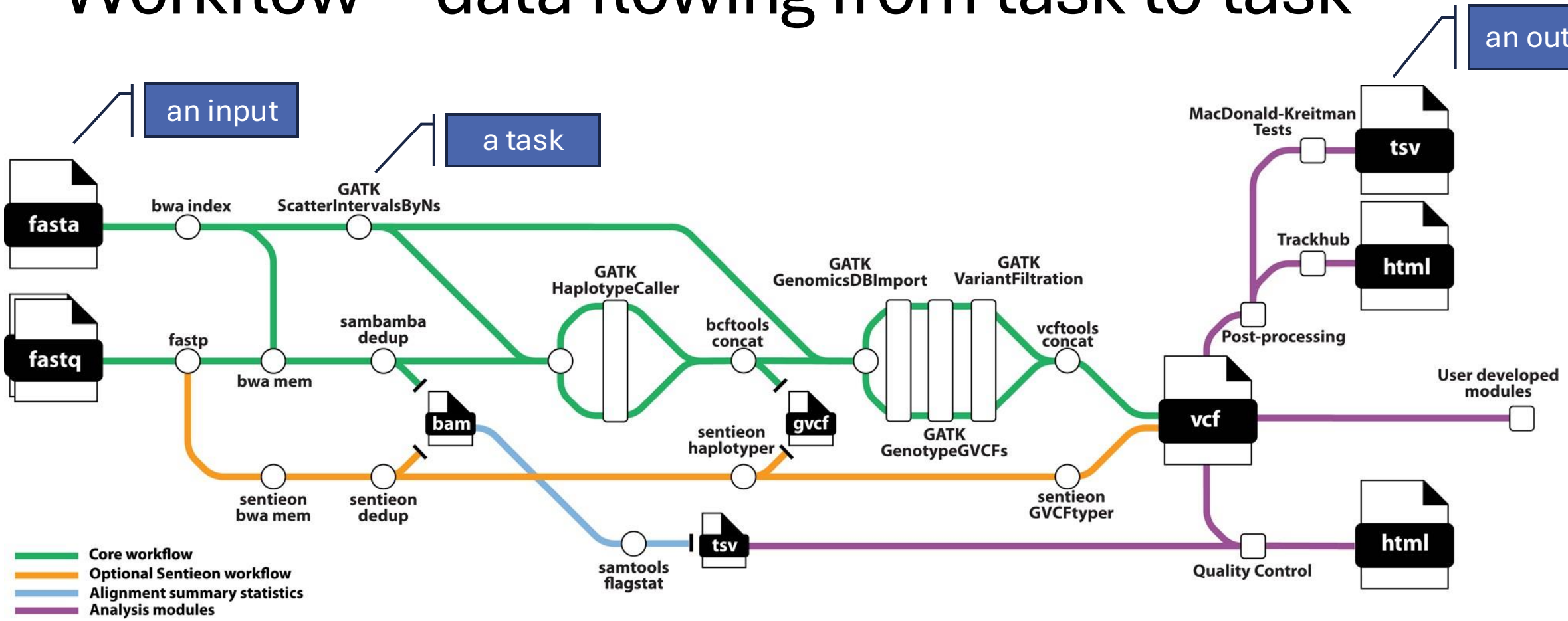
Tasks are executed independently



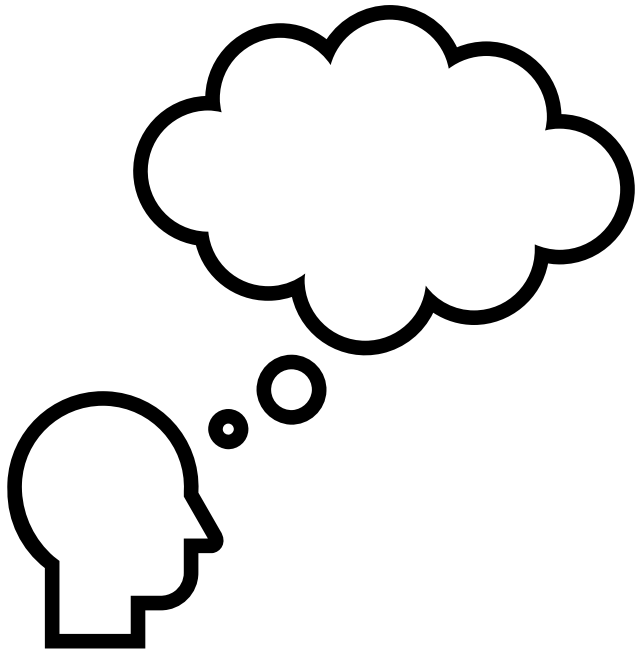
Inputs and outputs are managed dynamically



Workflow = data flowing from task to task



SNPArcher workflow from FAS Informatics



Take a moment to think about what your workflow looks like

array job vs nextflow

```
#!/bin/bash
#SBATCH -J fastqc
#SBATCH --array=0-7
#SBATCH -c 4
#SBATCH -p serial_requeue
#SBATCH -t 00:10:00
#SBATCH --mem=8000
#SBATCH -o logs/fastqc_%A_%a.out

readarray -t files < <(ls raw/*.fastq)
file=${files[${SLURM_ARRAY_TASK_ID}]}

echo "Working on file ${file}"

fastqc --outdir output --threads 4 $file
```

```
params.output_dir = 'output'
params.input_dir = 'raw/*.fastq'

process fastqc {
  executor = 'slurm'
  queue = 'shared'
  cpus = 4
  time = '10m'
  memory = '8 GB'

  input:
    path file

  output:
  path "${params.output_dir}/${file.simpleName}_fastqc.*"

  script:
  """
  fastqc --outdir ${params.output_dir} \\
    --threads ${task.cpus} $file
  """
}

workflow {
  files_ch = Channel.fromPath(params.input_dir)
  fastqc(files_ch)
}
```

The scope of a single task

- a single, well-defined task
  - e.g. fasterq-dump: downloads genomes
- a logically grouped set of closely related operations
  - e.g. bedtools sort then bedtools intersect: you want your intervals in order before you combine the files
- minimal dependencies
  - use just one software or a pair of closely related softwares
- matched resource needs
  - don’t combine a high memory task with a low memory task

Pros/cons of snakemake vs nextflow

Snakemake

- Written in python
- Syntax is simpler
- Simpler features -> easier to debug
- Used frequently in academia
- Smaller userbase/community
- Worse documentation & training
- modules

Nextflow

- Written in Groovy (Java)
- More complicated syntax
- More features -> harder to debug
- Used frequently in industry
- Large userbase/community
- Comprehensive online training available
- nf-core

Features/Design of snakemake vs nextflow

Snakemake

- “pull” philosophy: works backwards from end file
- Can do dry runs because every task has defined outputs
- Inputs are files exclusively
- Workflow executed in working directory
- Tracks file changes

Nextflow

- “push” philosophy: start with inputs and push through tasks
- Can’t do dry runs because outputs can be variable
- Inputs can be files/values/variables
- Each instance of a task has its own directory
- Tracks file, code, & other changes

Resources to learn about workflow managers

Snakemake

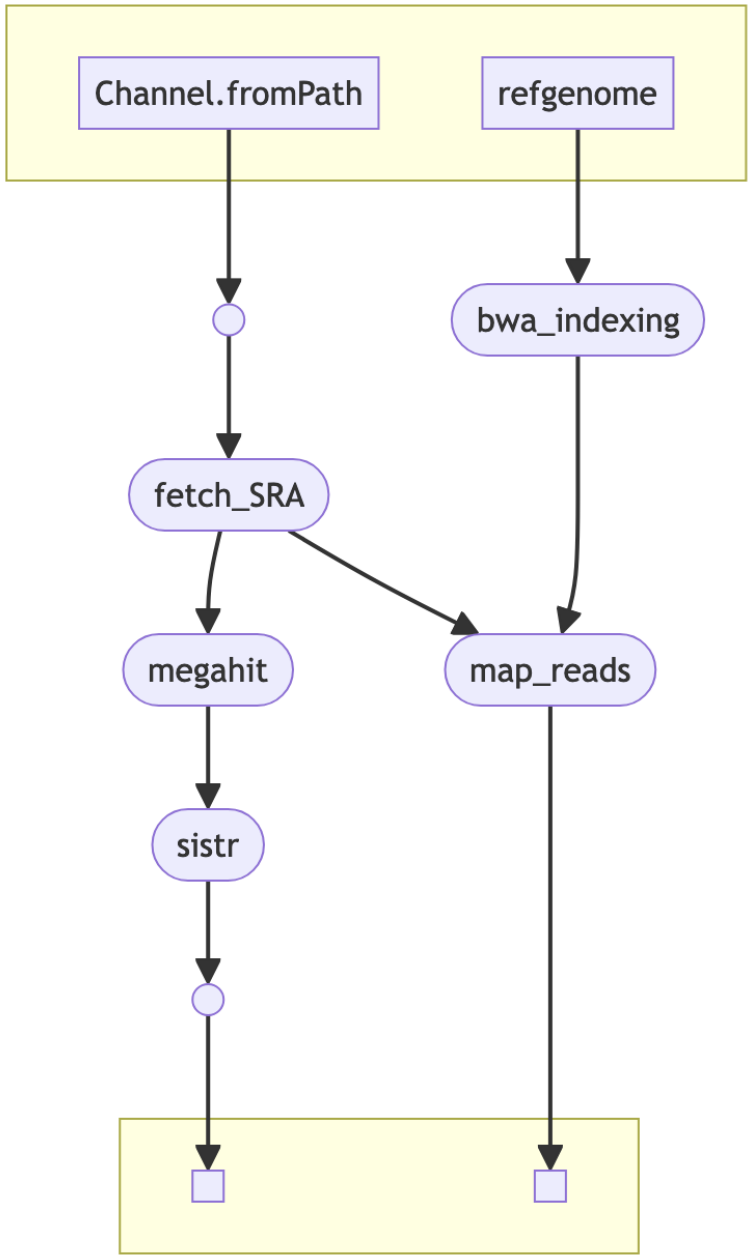
- https://snakemake.readthedocs.io/en/stable/tutorial/tutorial.html
- https://carpentries-incubator.github.io/snakemake-novice-bioinformatics/index.html
- https://uppsala.instructure.com/courses/51980/pages/snakemake-1-introduction?module_item_id=243089

Nextflow

- https://training.nextflow.io/
- https://carpentries-incubator.github.io/workflows-nextflow/index.html
- https://seqera.io/blog/learn-nextflow-in-2023/

Nextflow Demo

Diagram of workflow



How nextflow works

- Head job (nextflow runtime)
- Each sub-process has its own work folder
  - can be in netscratch or local scratch
  - input files are staged in that folder thru symlinks (default)
- Optionally ‘publish’ output files over to a separate directory
- Each task is logged and cached
  - caching is based on a hash generated from the task and file characteristics
  - allows resuming from partially complete tasks
- Software can be conda/spack/containers

Outline of demo

- Config file
- SRA input file
- Explain nf file
- Run nextflow in interactive session
- Show nextflow tower/seqera platform
- Show working directory vs publish directory
- Rerun with additional file (SRR20634159) to show caching

Vocabulary

- Workflow: a sequence of data processing steps
- Workflow management system: software that can automatically run a workflow, among other features
- Pipeline: synonym for workflow
- DAG: directed acyclic graph representation of bioinformatics pipeline