Making a workflow

All files are available on my github:

https://github.com/LauritsSkov/gwf_example

Install

Install

```
conda config --add channels gwforg
conda create -n gwf_tutorial gwf
```

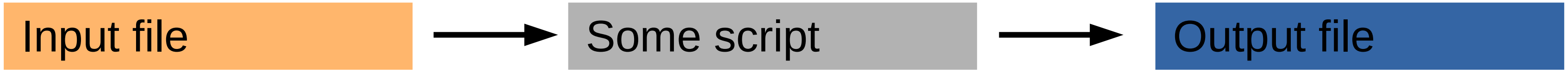
Setup slurm backend

```
gwf config set backend slurm
```

Tutorial

https://gwf.app/guide/tutorial/

Making a workflow

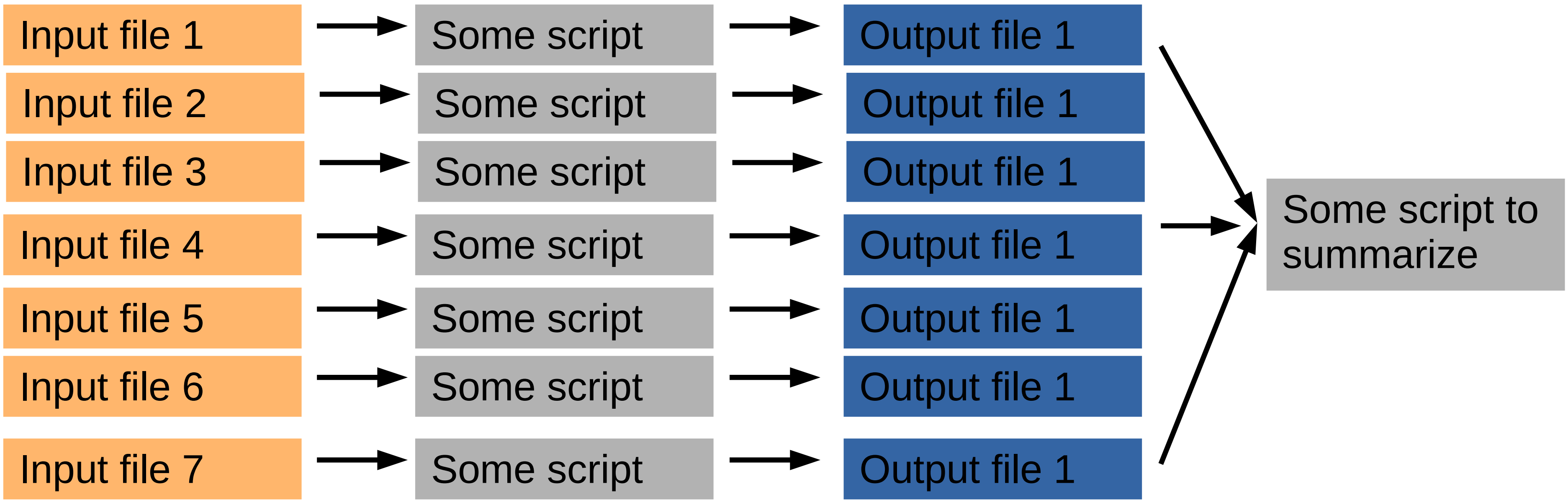


Input file

Some script

Output file

Making a workflow



Input file 1 → Some script → Output file 1

Input file 2 → Some script → Output file 1

Input file 3 → Some script → Output file 1

Input file 4 → Some script → Output file 1

Input file 5 → Some script → Output file 1

Input file 6 → Some script → Output file 1

Input file 7 → Some script → Output file 1

Some script to summarize

```
#!/bin/sh
#SBATCH --job-name=jobname
#SBATCH --account=co_moorjani
#SBATCH --partition=savio3_htc
#SBATCH --ntasks-per-node=1
#SBATCH --cores=1
#SBATCH --time=12:00:00
#SBATCH --mem 4gb

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

```
> sbatch -o log ./post.sh
```

Why not use snake make?

Snakemake

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

Why not use snake make?

Snakemake

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

gwf

```
def bwa_map(ref_genome, fastq_file, output):
    inputs = [ref_genome, fastq_file]
    outputs = [output]
    options = {
        'cores': 1,
        'memory': '1g',
    }

    spec = f"""
    bwa mem {ref_genome} {fastq_file} | samtools view -Sb - > {output}
    """

    return AnonymousTarget(inputs=inputs, outputs=outputs, options=options, spec=spec)

# call it like this
gwf.target_from_template(“jobname”, bwa_map("data/genome.fa", "data/samples/A.fastq",
"mapped_reads/A.bam"))
```

```
def bwa_map():

    inputs = [reads.fastaq.gz, Refgenome.fa]
    outputs = [out.bam]
    options = {
        'cores': 1,
        'memory': '4g',
        'walltime': '12:00:00',
        'queue': 'savio3_htc',
        'account': 'co_moorjani',
    }
    spec = '''
    bwa mem Refgenome.fa reads.fastaq.gz | samtools view -Sb - >
out.bam
    '''
    return AnonymousTarget(inputs=inputs, outputs=outputs,
options=options, spec=spec)

gwf.target_from_template('jobname', bwa_map())
```

```
#!/bin/sh
#SBATCH --job-name=jobname
#SBATCH --account=co_moorjani
#SBATCH --partition=savio3_htc
#SBATCH --ntasks-per-node=1
#SBATCH --cores=1
#SBATCH --time=12:00:00
#SBATCH --mem 4gb

bwa mem Refgenome.fa reads.fastaq.gz
| samtools view -Sb - > out.bam
```

```
> gwf run
```