

Making reproducible workflows with

nextflow

Nextflow features

- Based on **Groovy** and Java
- **Portable**
- **Scalable**
- **Platform-agnostic**
- Large, active **community**

Differences between Snakemake and Nextflow

	Snakemake	Nextflow
Language	Python	Groovy
Data	Everything is a file	Can use both files and values
Execution	Working directory	Each job in its own directory
Philosophy	"Pull"	"Push"
Dry-runs	Yes	No
Track code changes	No	Yes

- **Question:** But, which one is the best?
- **Answer:** Both - it's mostly up to personal preference!

Anatomy of a Nextflow workflow

```
workflow {  
  // Define SRA input data channel  
  Channel  
    .fromList ( ["SRR935090", "SRR935091", "SRR935092"] )  
    .set      { ch_sra_ids }  
  
  // Define the workflow  
  GET_SRA_BY_ACCESSION (   
    ch_sra_ids  
  )  
  RUN_FASTQC (   
    GET_SRA_BY_ACCESSION.out  
  )  
  RUN_MULTIQC (   
    RUN_FASTQC.out.collect()  
  )  
}
```

```
process GET_SRA_BY_ACCESSION {  
  input:  
  val(sra_id)  
  
  output:  
  tuple val(sra_id), path("${sra_id}.fastq.gz")  
  
  script:  
  """  
  fastq-dump ${sra_id} > ${sra_id}.fastq.gz  
  """  
}
```

```
$ nextflow run main.nf
```

The aims of this tutorial

- Give a brief **overview** of how Nextflow works through a recreation of the MRSA Snakemake workflow
- Show **similarities** and **differences** between Snakemake and Nextflow
- Give you an opportunity to **try out both** Snakemake and Nextflow
- The Nextflow tutorial is shorter and less in-depth compared to the Snakemake tutorial, given these aims

Questions?