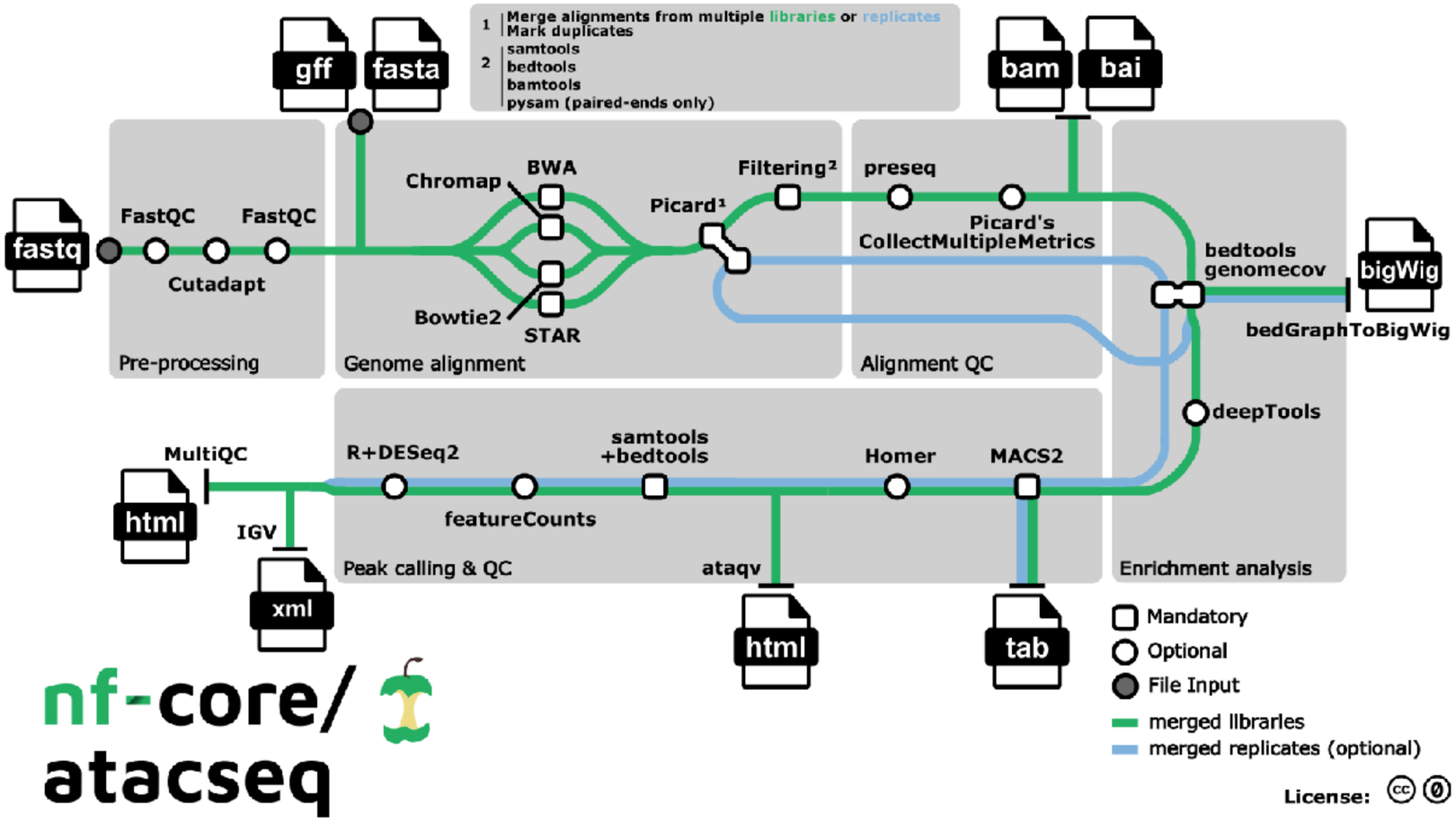


nf-core/atacseq

Chris Hakkaart - Seqera Labs

Credits

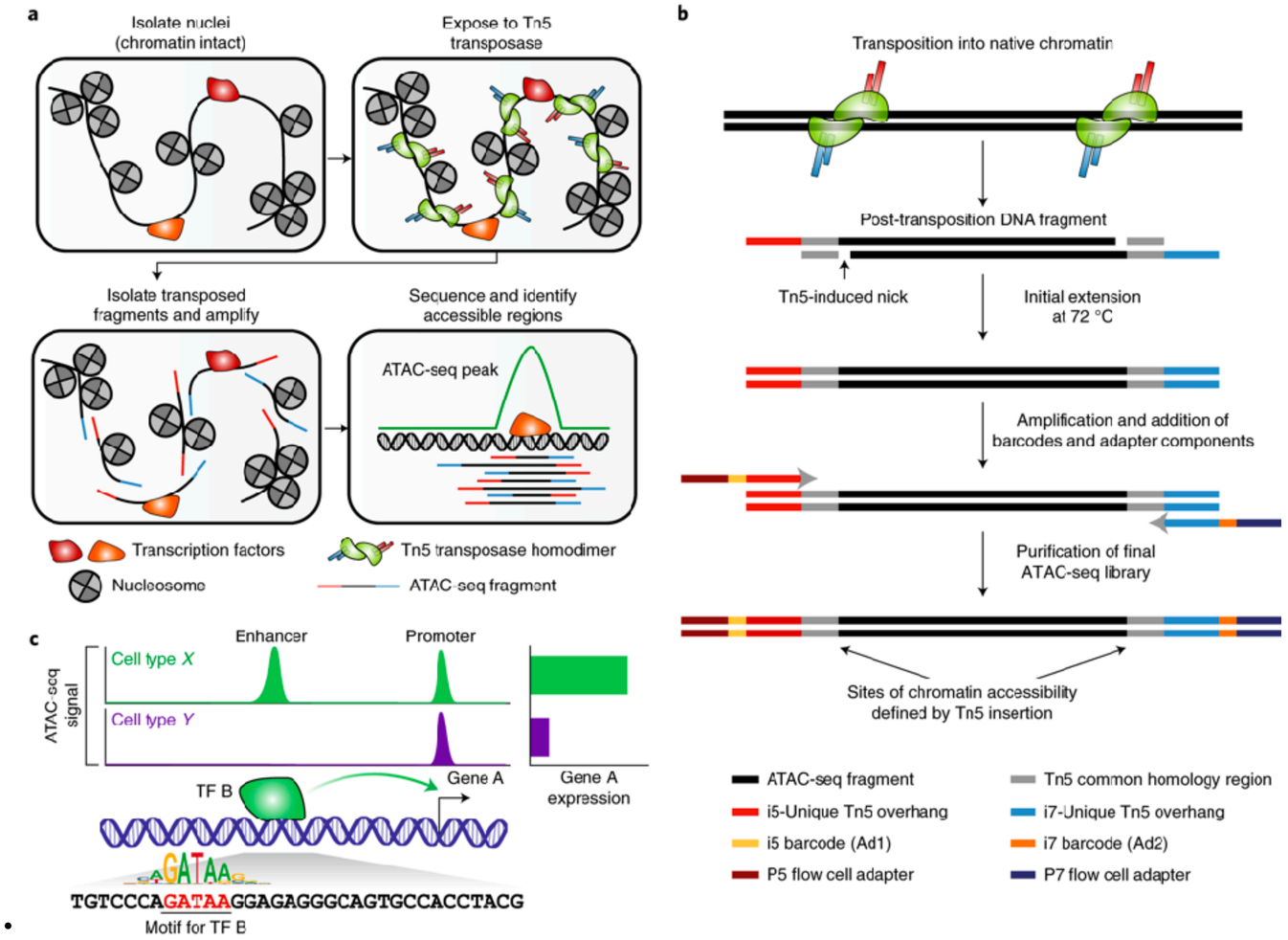
The pipeline was originally written by Harshil Patel (@drpatelh) from Seqera Labs, Spain and converted to Nextflow DSL2 by Björn Langer (@bjlang) and Jose Espinosa-Carrasco (@JoseEspinosa) from The Comparative Bioinformatics Group at The Centre for Genomic Regulation, Spain under the umbrella of the BovReg project.

Many thanks to others who have helped out and contributed along the way too, including (but not limited to): @ewels, @apeltzer, @crickbabs, drewjbeh, @houghtos, @jinmingda, @ktrns, @MaxUlysse, @mashehu, @micans, @pditommaso and @sven1103.

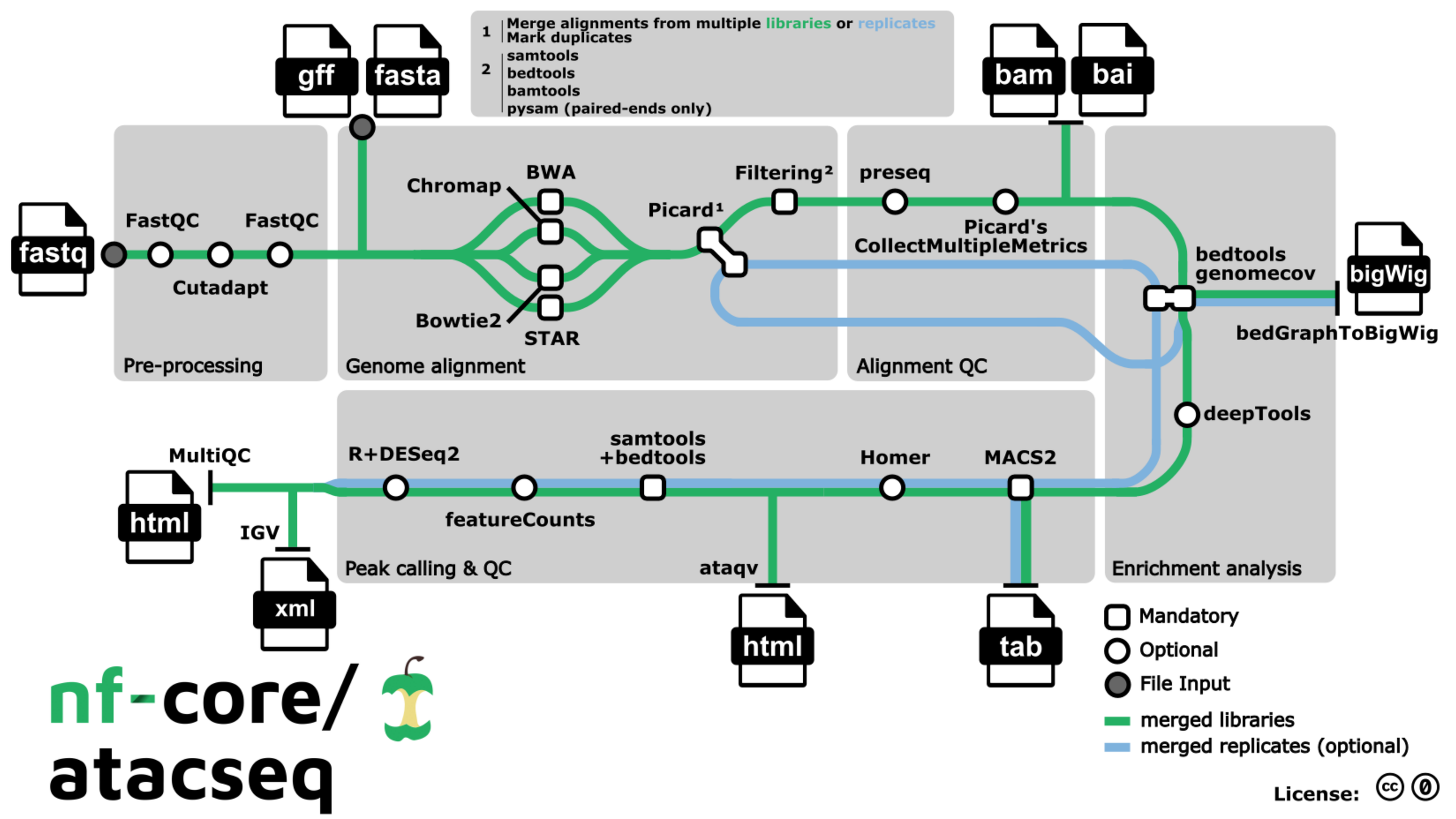
Overview

- nf-core/atacseq
  - Pipeline overview
  - The run command
  - How to build your samplesheet
  - Available parameters
  - Tips for running atacseq
- The work directory
  - A reminder about the cache and -resume functionality
  - Hidden files and how you can use them to troubleshoot

ATAC-seq



https://www.nature.com/articles/s41596-022-00692-9


1 Merge alignments from multiple libraries or replicates
Mark duplicates
2 samtools
bedtools
bamtools
pysam (paired-ends only)
gff
fasta
bam
bai
fastq
FastQC
FastQC
Cutadapt
Chromap
BWA
Bowtie2
STAR
Picard¹
Filtering²
preseq
Picard's CollectMultipleMetrics
bedtools genomecov
bigWig
bedGraphToBigWig
deepTools
Pre-processing
Genome alignment
Alignment QC
Enrichment analysis
MultiQC
html
IGV
xml
R+DESeq2
featureCounts
samtools +bedtools
Homer
MACS2
ataqv
html
tab
Peak calling & QC
Mandatory
Optional
File Input
merged libraries
merged replicates (optional)
nf-core/
atacseq
License:

Basic run command

```
nextflow run nf-core/atacseq /    <- Run directly from GitHub
    --input samplesheet.csv /     <- Local sample sheet (required)
    --outdir <OUTDIR> /           <- Output director (required)
    --genome GRCh37 /             <- Reference genome (required)
    --read_length <50> /          <- Read Length (optional)
    -profile <docker>             <- Software manager
    -r <2.1.1>                    <- Revision
```

--input '[path to samplesheet]'

Biological Replicates

```
sample,fastq_1,fastq_2,replicate
CONTROL,AEG588A1_S1_L002_R1_001.fastq.gz,AEG588A1_S1_L002_R2_001.fastq.gz,1
CONTROL,AEG588A1_S1_L003_R1_001.fastq.gz,AEG588A1_S1_L003_R2_001.fastq.gz,2
CONTROL,AEG588A1_S1_L004_R1_001.fastq.gz,AEG588A1_S1_L004_R2_001.fastq.gz,3
```

Technical Replicates

```
sample,fastq_1,fastq_2,replicate
CONTROL,AEG588A1_S1_L002_R1_001.fastq.gz,AEG588A1_S1_L002_R2_001.fastq.gz,1
CONTROL,AEG588A1_S1_L003_R1_001.fastq.gz,AEG588A1_S1_L003_R2_001.fastq.gz,1
CONTROL,AEG588A1_S1_L004_R1_001.fastq.gz,AEG588A1_S1_L004_R2_001.fastq.gz,1
```

The `*_T<TECHNICAL_REPLICATE_NUMBER>` suffix will be added to the sample name:
e.g. `CONTROL_REP1_T1`, `CONTROL_REP1_T2` and `CONTROL_REP1_T3` using the example above.

--input '[path to samplesheet]'

- If controls are to be used for peak calling use the **--with_control** parameter.
  - In this case, the samplesheet file needs the additional columns **control** and **control_replicate.**
    - Should be the sample identifier and sample replicate for the controls.

```
sample,fastq_1,fastq_2,replicate,control,control_replicate
CONTROL,AEG588A1_S1_L002_R1_001.fastq.gz,,1,,
CONTROL,AEG588A2_S2_L002_R1_001.fastq.gz,,2,,
CONTROL,AEG588A3_S3_L002_R1_001.fastq.gz,,3,,
TREATMENT,AEG588A4_S4_L003_R1_001.fastq.gz,,1,CONTROL,1
TREATMENT,AEG588A5_S5_L003_R1_001.fastq.gz,,2,CONTROL,2
TREATMENT,AEG588A6_S6_L003_R1_001.fastq.gz,,3,CONTROL,3
```

--input '[path to samplesheet]'

```
sample,fastq_1,fastq_2,replicate,control,control_replicate
CONTROL,AEG588A1_S1_L002_R1_001.fastq.gz,AEG588A1_S1_L002_R2_001.fastq.gz,1,,
CONTROL,AEG588A2_S2_L002_R1_001.fastq.gz,AEG588A2_S2_L002_R2_001.fastq.gz,2,,
CONTROL,AEG588A3_S3_L002_R1_001.fastq.gz,AEG588A3_S3_L002_R2_001.fastq.gz,3,,
TREATMENT,AEG588A4_S4_L003_R1_001.fastq.gz,,1,CONTROL,1
TREATMENT,AEG588A5_S5_L003_R1_001.fastq.gz,,2,CONTROL,2
TREATMENT,AEG588A6_S6_L003_R1_001.fastq.gz,,3,CONTROL,3
TREATMENT,AEG588A6_S6_L004_R1_001.fastq.gz,,3,CONTROL,3
```

- Will auto-detect whether a sample is single- or paired-end
- A final sample sheet file consisting of both single- and paired-end data may look something like the one above.
  - Biological triplicates for both the `CONTROL` and `TREATMENT` groups
  - Third replicate in the `TREATMENT` group is a technical replicate (was sequenced twice).

Parameters

- Input/output options
- Reference genome options
- Adapter trimming options
- Alignment options
- Peak calling options
- Differential analysis options
- Process skipping options
- Generic options

Tips for running nf-core/atacseq

- Start with the **test** profiles or other small files to check you are happy with the outputs and parameter combinations you want to use
- There are lots of files kept by default - make sure **storage** is available
- Use **--genome** (if you can) as it will remove the chance or error
- Use **nf-core launch** to help you generate a parameters file

The work directory

- Where the magic happens and your cache is pulled from
- Tasks isolated from each other
- Everything is staged in task directory (mostly symlinks)
- Nextflow dot files are hidden
- .command.sh is helpful when you want to check how a task has run