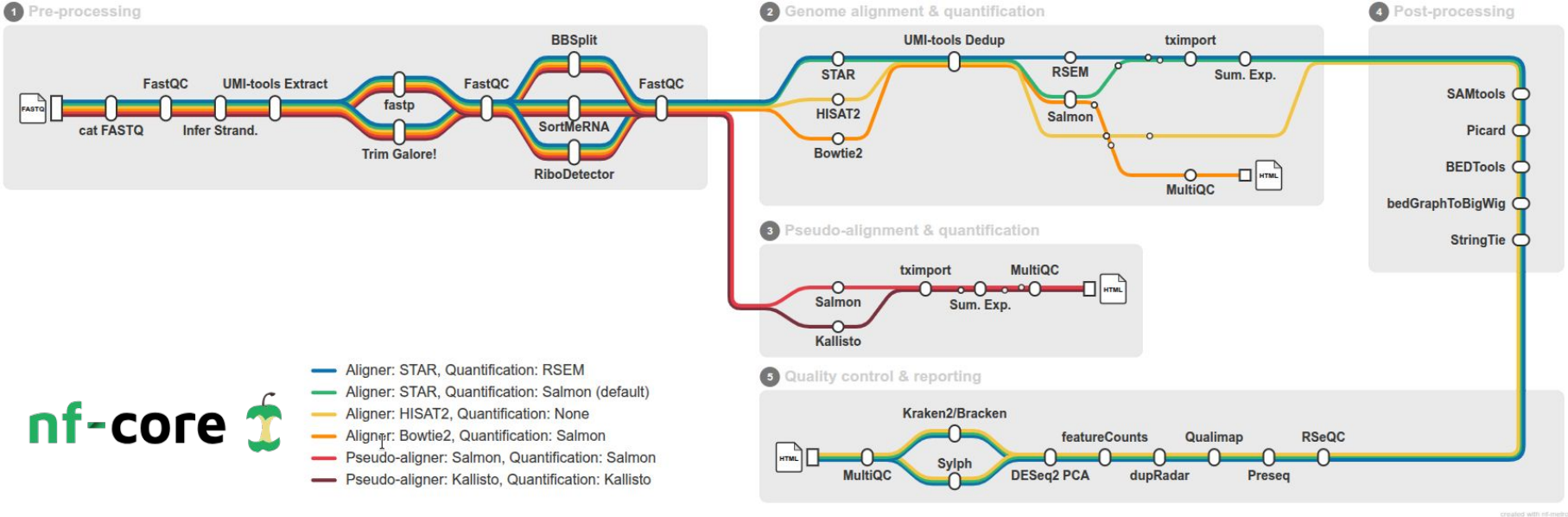
WM7

Computational and Systems Biology

RNA sequencing data analysis - practical part

Dietmar Rieder

RNA-seq analysis pipeline: nf-core/rnaseq



Nextflow - Features

Nextflow is built around the idea that Linux is the *lingua franca* of data science.

- **Fast prototyping**
  - write a computational pipeline by making it simpler to put together many different tasks
  - reuse your existing scripts and tools
  - extend with a powerful DSL
  - “no” need to learn a new language or API

- **Reproducibility and isolation of dependencies**
  - supports Docker and Singularity containers technology.
  - supports Conda envs: easy and automatic software installation
  - integration of the GitHub code sharing platform
  - abstraction layer between pipeline's logic and the execution layer
    - can be executed on multiple platforms (SGE, Amazon AWS, ...) without it changing

Nextflow - Basic pipeline

```
1  #!/usr/bin/env nextflow
2
3  params.in = "$baseDir/data/sample.fa"
4
5  /*
6   * Split a fasta file into multiple files
7   */
8  process splitSequences {
9
10     input:
11     path 'input.fa'
12
13     output:
14     path 'seq_*'
15
16     """
17     awk '/^>/{f="seq_"++d} {print > f}' < input.fa
18     """
19 }
20
21 /*
22  * Reverse the sequences
23  */
24 process reverse {
25
26     input:
27     path x
28
29     output:
30     stdout
31
32     """
33     cat $x | rev
34     """
35 }
36
37 /*
38  * Define the workflow
39  */
40 workflow {
41     splitSequences(params.in) \
42       | reverse \
43       | view
44 }
```

Synopsis

- **Line 1** The script starts with a shebang declaration. This allows you to launch your pipeline just like any other Bash script.
- **Line 3**: Declares a pipeline parameter named `params.in` that is initialized with the value `$HOME/sample.fa`. This value can be overridden when launching the pipeline, by simply adding the option `--in <value>` to the script command line.
- **Lines 8-19**: The process that splits the provided file.
- **Line 10**: Opens the input declaration block. The lines following this clause are interpreted as input definitions.
- **Line 11**: Declares the process input file, which will be named `input.fa` in the process script.
- **Line 13**: Opens the output declaration block. The lines following this clause are interpreted as output declarations.
- **Line 14**: Files whose names match the pattern `seq_*` are declared as the output of this process.
- **Lines 16-18**: The actual script executed by the process to split the input file.
- **Lines 24-35**: The second process, which receives the splits produced by the previous process and reverses their content.
- **Line 26**: Opens the input declaration block. Lines following this clause are interpreted as input declarations.
- **Line 27**: Defines the process input file.
- **Line 29**: Opens the output declaration block. Lines following this clause are interpreted as output declarations.
- **Line 30**: The standard output of the executed script is declared as the process output.
- **Lines 32-34**: The actual script executed by the process to reverse the content of the input files.
- **Lines 40-44**: The workflow that connects everything together!
- **Line 41**: First, the input file specified by `params.in` is passed to the `splitSequences` process.
- **Line 42**: The outputs of `splitSequences` are passed as inputs to the `reverse` process, which processes each split file in parallel.
- **Line 43**: Finally, each output emitted by `reverse` is printed.

nf-core/rnaseq

Basic run command:

```
nextflow run nf-core/rnaseq -r "3.26.0" \
    --input samplesheet.csv \
    --outdir results \
    --aligner star_salmon \
    [...] \
    --gencode \
    --gtf <gtf file>\
    --fasta <genome fasta file> \
    --star_index <path to star index> \
    --salmon_index <path to salmon index> \
    -profile singularity
```

prepare

complete options for QuantSeq 3'

https://nf-co.re/rnaseq/3.26.0/usage

nf-core/rnaseq

Raw data for sample sheet:

Data dir:

`/data/wm7/raw`

Raw fastq files:

untreated: `CaCo2-[1..3]_R1_001.fastq.gz`

treated: `CaCo2-[4..5]_R1_001.fastq.gz`

samplesheet.csv:

```
sample,fastq_1,fastq_2,strandedness,condition
CaCo2_1,/data/….fastq.gz,,forward,control
[...]
CaCo2_4,/data/….fastq.gz,,forward,MEKi
[...]
```

nf-core/rnaseq

Genome indexes + annotation files:

Data dir:

`/data/wm7/genome`

indexes:

- STAR: `index/star`
- salmon: `index/salmon`
- rsem: `rsem`

GTF:

`gencode.v43.primary_assembly.annotation.gtf`

FASTA:

`gencode.v43.primary_assembly.genome.fasta`

nf-core/rnaseq

Check QC and results:

- results/multiqc/star_salmon
  - # mapped reads
  - PCA
  - Biotype Counts
  - QualiMap
  - RSeQC
  - FASTQC raw/trimmed
    - Coverage Profile
    - Per Base Sequence Content
    - Adapter Content
- results/star_salmon
  - salmon.merged.gene_counts.tsv
  - salmon.merged.gene_tpm.tsv

https://nf-co.re/rnaseq/3.26.0/output

DESeq2 - Rstudio

Tasks:

- Run DESeq2 analysis
  - treated vs untreated
- Generate PCA plot of all samples
  - perform variance stabilized transformation (VST)
  - plotPCA
- Generate Volcano plot
- Generate table with all differentially expressed genes
  - fold change > 2 or < 0.5 ~ $|\log_2(\text{fold change})| > 1$
  - p.adjusted < 0.1
- Generate a heatmap of the top 30 up/down regulated genes

https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html