Sandro Andreotti

May 02, 2025

Applied Sequence Analysis

Summer term 2025

Assignment 2
Due 2025-05-09 12:00 PM

1 Extend your workflow from last week

Extend your workflow on both ends to:

- read paired-end fastq files (again, the user should just set the folder containing the fastq files at the beginning of the Snakefile).
- perform a (paired-end) read mapping with *bowtie2* against the provided reference and proceed with the generated sam files as in last week's workflow.
- aggregate all idxstats reports: Create a single `.tsv` file summarizing the data from all samples (Hint: create a *Python* rule and use pandas):
  - **row ids:** reference sequence names
  - **columns:** reference length and one column for each sample (named by the sample) with the associated mapped read count.

2 Restructure your workflow *[1]

Restructure your extended workflow in the following ways:

- create a project structure according to the guidelines.
- split your rules into two files `rules/bowtie.smk` and `rules/samtools.smk` and import them into your `Snakefile`.
- create a *Conda* environment descripition *yaml* with the required dependencies (samtools, bowtie2) and use it in your workflow.
- *bowtie2* and *samtools* should be able to use up to 4 threads wherever possible.

[1] Submission: Submit your project (Snakefile, *Conda* environments, config, sample sheet - according to guideline structure) via KVV. The workflow must be executable on a vanilla machine (having only Snakemake and Conda installed) once samples.tsv is set by the user!

- in file `config/config.yaml` define:
  - `ref` (path to fasta file with reference sequence for *bowtie2* mapping)
  - `samples` (path to sample tsv file - see below)
  - at least two *bowtie2* parameters of your choice that will be used in the mapping rule
- for the definition of your input data define a file `samples.tsv` like:

sample	fq1	fq2
ERRx	<path to first fastq file>	<path to second fastq file>
ERRy	<path to first fastq file>	<path to second fastq file>

- parse the sample file in your Snakefile to obtain sample names and associated read files
- The sam, bam and stats files should be all named by the sample, e.g.:
  `ERRx.sam, ERRx.bam, ERRx_sorted.bam, ERRx.stats, ERRx.stats_aug`

Hints: for parsing and using the sample file:

```
#load samples into table
import pandas as pd
configfile: "config.yaml"
samples = pd.read_csv(config["samples"], index_col="sample", sep='\t')
#...
#...
#list all samples
expand("stats/{sample}.stats", sample=list(samples.index))
#...
#...
#access files for samples
r1 = lambda wildcards: samples.at[wildcards.sample,'fq1']
```