A Parallelized Snakemake Pipeline for Fungal Genome Assembly

Zachary Kratz, Alexander James Bradshaw,
Javier F. Tabima, David S. Hibbett

Clark University, Worcester MA

#2333266

Introduction

Problem

The Big Data Challenge

Modern bioinformatics generates massive datasets that are computationally intensive to analyze. Processing large amounts of data can take days or weeks.

Solution

Parallel Computing with Snakemake

Splits large problems into smaller, independent tasks that get executed simultaneously.

Why Snakemake?

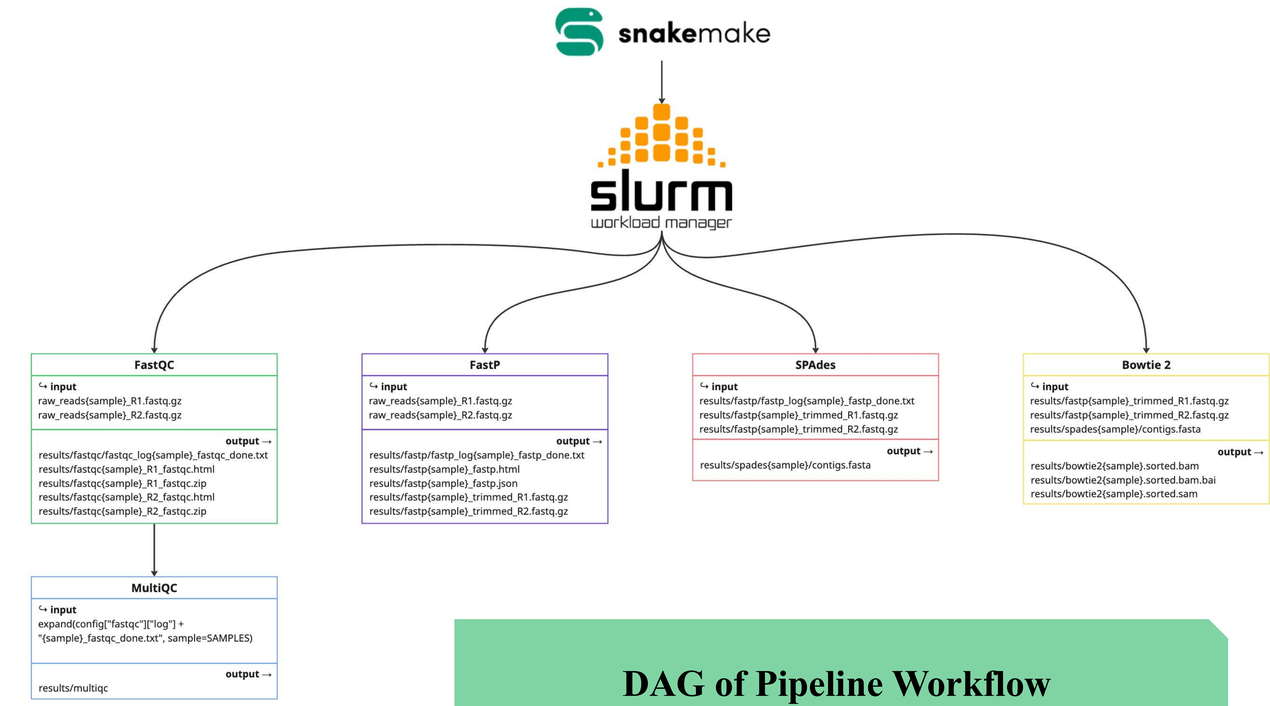
- Parallelizes per-sample and per-step maximizing efficiency
- Integration with HPC via Slurm
- Built-in support for many bioinformatic tools

Implementation

Using a Snakemake profile, we can create a template for using this pipeline on the Clark HPC

This SLURM profile acts as a bridge between Snakemake and the Clark HPC by translating workflow requirements into SLURM job submissions.

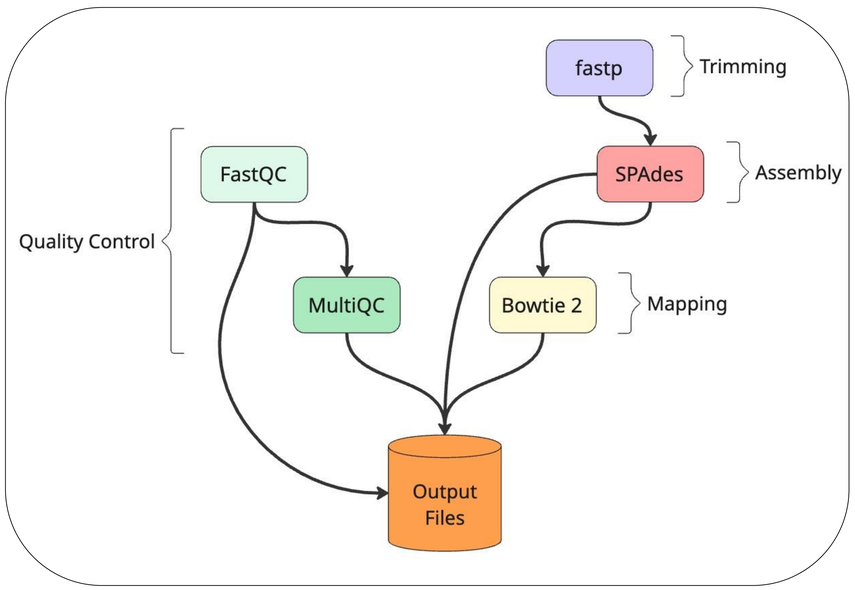
Each rule requests the CPUs, memory, and partition it needs. By keeping these settings in a profile, the workflow is portable, reproducible, and easy to adapt.



Slurm Profile

```yaml
# Basic Snakemake settings
executor: slurm
jobs: 100
latency-wait: 60
use-conda: true
rerun-incomplete: true
keep-going: true
printshellcmds: true

# Default resources for all rules
default-resources:
  slurm_partition: "short-cpu"
  slurm_time: "1:00:00"
  slurm_mem_mb: 16000
  slurm_cpus_per_task: 8
  slurm_nodes: 1
  slurm_output: "logs/slurm/%j.out"
  slurm_error: "logs/slurm/%j.err"
  slurm_job_name: "sm.{rule}"

# Rule-specific resources
set-resources:
  # FastQC - short job with moderate resource requirements
  fastqc:
    slurm_partition: "short-cpu"
    slurm_time: "1:00:00"
    slurm_mem_mb: 16000
    slurm_cpus_per_task: 8

  # MultiQC - very short job with minimal resource requirements
  multiqc:
    slurm_partition: "short-cpu"
    slurm_time: "30:00"
    slurm_mem_mb: 8000
    slurm_cpus_per_task: 4

  # FastP - short job with moderate resource requirements
  fastp:
    slurm_partition: "short-cpu"
    slurm_time: "1:00:00"
    slurm_mem_mb: 16000
    slurm_cpus_per_task: 8

  # SPAdes - long job with high resource requirements
  spades:
    slurm_partition: "day-long-cpu"
    slurm_time: "24:00:00"
    slurm_mem_mb: 64000
    slurm_cpus_per_task: 16

  # Bowtie2 mapping - medium job with moderate-high resource requirements
  bowtie2_mapping:
    slurm_partition: "day-long-cpu"
    slurm_time: "6:00:00"
    slurm_mem_mb: 32000
    slurm_cpus_per_task: 8

# Set threads for specific rules
set-threads:
  fastqc: 8
  multiqc: 4
  fastp: 8
  spades: 16
  bowtie2_mapping: 8

# Group jobs by rule for better organization
group-components:
  fastqc: 10
  fastp: 10
```

DAG of Pipeline Workflow



Acknowledgements

Clark HPC Team for time and resources

Works cited

Andrews, S. (2010). *FastQC: A quality control tool for high throughput sequence data* [Software]. Babraham Bioinformatics. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Ewels, P., Magnusson, M., Lundin, S., & Käller, M. (2016). MultiQC: Summarize analysis results for multiple tools and samples in a single report. *Bioinformatics, 32*(19), 3047–3048.

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: An ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics, 34*(17), i884–i890.

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., ... & Pevzner, P. A. (2012). SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology, 19*(5), 455–477.

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods, 9*(4), 357–359.

Yoo, A. B., Jette, M. A., & Grondona, M. (2003). SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing* (pp. 44–60). Springer.

Köster, J., & Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics, 28*(19), 2520–2522.