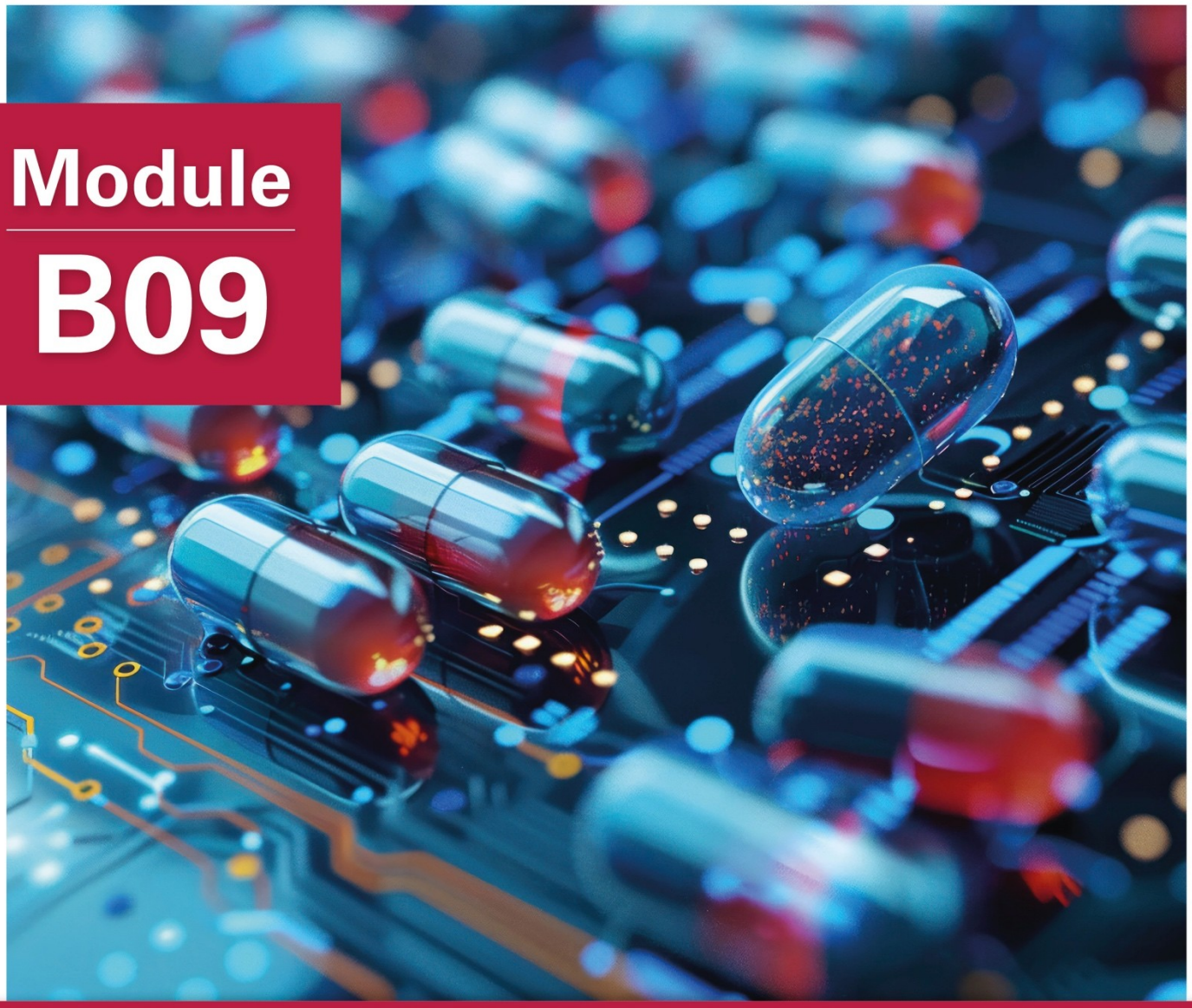
AfricaCDC
Centres for Disease Control and Prevention

Safeguarding Africa's Health



Module B09

Bioinformatics Foundational Course

Workflow Management Systems

NGS Academy for the Africa CDC

ASLM
AFRICAN SOCIETY FOR LABORATORY MEDICINE

H3ABioNet
Pan African Bioinformatics Network for H3Africa

ASLM ACADEMY

Module B09

Workflow Management Systems

back to the table of modules

Module last updated:
December 2024

Suggested or approximate number of sessions	4-5
Suggested or approximate total learning time	6-8 hours
Target audience	Bioinformaticians
Delivery format	Lectures, practicals
Level of the module	Intermediate

Contributors

Gerrit Botha, Kirsty Lee Garson, George Githinji, John Juma, Davis Kuchaka, Tony Li, Perceval Maturure, Nicola Mulder, Eric Murimi, Brian Ogoti and Evalyne Wambugu.

Suggested prerequisite module(s)

- Module B02. Introduction to Version Control
- Module B08. Virtual Environments and Containerization

Module description

Data analysis typically involves a sequence of tasks, such as gathering, cleaning, and processing data. This sequence of tasks is referred to as a workflow or a pipeline. Workflow management systems allow for the development, monitoring, and execution of pipelines. In this module, participants are also introduced to the following topics and/or concepts:

- Key features of workflow management systems include:
  - Run time management
  - Software management
  - Portability and interoperability
  - Reproducibility
  - Re-entrancy

Workflows can be developed using different workflow languages, including Nextflow, Snakemake, Workflow Description Language (WDL) etc. Trainers can decide which they want to focus on as examples. Some platforms exist that allow users to create workflows through web interfaces, such as Terra or Galaxy.
Nextflow and Snakemake are robust workflow management systems for bioinformatics; however, they employ distinct methodologies. Nextflow is suitable for pipelines that are intricate and contain numerous linked stages, as it employs a dataflow programming paradigm that involves the connection of processes through channels. For those who are accustomed to Make-like systems, Snakemake's declarative, rule-based approach, which is influenced by GNU Make, may be more intuitive. Snakemake is a workflow language created by Johannes Köster and first introduced in 2012. It structures workflow execution using a directed acyclic graph (DAG), which maps the relationships between files and tools in a data analysis pipeline. This DAG outlines the steps data must follow from start to finish and can be visualized for a clear workflow overview.
While Nextflow is developed in Groovy (JVM-based), Snakemake is based on Python. Thus, existing programming proficiency is an important factor in deciding between the two.

When learning about workflow management systems, participants are introduced to the following topics and/or concepts:

- Overview of workflow management systems
- Platforms which offer existing analysis workflows (Terra, Galaxy, etc)
- The process of developing an analysis workflow
- Comparison of Snakemake and Nextflow for different use cases - including side-by-side examples of the same workflow implemented in Snakemake and Nextflow to highlight their similarities and differences
- Introduction to Snakemake (AND/OR, depending on choice of Snakemake versus Nextflow)
- Introduction to Nextflow and resources available to Nextflow users
- The components of a Nextflow script
- Key components of a Nextflow workflow:
  - Processes are tasks to be completed, which have defined inputs and outputs.
  - Channels are asynchronous queues, used to manipulate the flow of data from one process to the next.
  - The workflow section defines the interactions between processes, and determines the flow of execution of a pipeline.
- Running Nextflow scripts
- Components of a Snakemake workflow:
  - Snakefile core script in a Snakemake workflow that defines the computational steps (rules) needed to process data. It specifies input files, output files, and commands required to transform inputs into outputs.
  - Config file (config.yaml) used to store configuration parameters, such as file paths, sample names, reference genomes, and other variables.
  - Rules define tasks to be completed, with specified inputs and outputs.
  - Wildcards serve as placeholders in rules, allowing for pattern matching in file paths.
  - The workflow is implicitly defined by the dependencies between rules, forming a directed acyclic graph (DAG).
- Running a Snakemake workflow

Module learning outcomes

On completion of this module, participants will have a basic knowledge of, or will be able to:

- Understand what a workflow management system is
- Describe the benefits of using a workflow management system
- Explain the benefits of using Nextflow or Snakemake as part of their bioinformatics workflow
- Develop a workflow
- Explain the components of a Nextflow script
- Explain the components of a snakefile
- Run a Nextflow script
- Run a Snakemake workflow

Module assessments

Module practical: Suggestion - participants should be given an assignment to develop an example workflow
Module quiz: Assessment questions available on the ASLM platform

Module resources

- PHA4GE | GitHub - Pipeline resources
- SANBI | Slides - SARS-CoV-2 sequence analysis workflows
- PHA4GE | GitHub - Bioinformatics Solutions for SARS-CoV-2 Genomic Analysis
- nf-core | GitHub - Nextflow analysis pipeline
- Theiagen | GitHub - Public health bioinformatics
- Dockstore | Webpage - An app store for bioinformatics
- WorkflowHub | Webpage - Scientific computational workflows
- Snakemake | Documentation – Official Snakemake documentation

Acknowledgements

We would like to thank the following individuals, in alphabetical order of last name, for their valuable time and effort spent in designing (i.e., drafting, reviewing, and refining) this module: **Gerrit Botha, Kirsty Lee Garson, George Githinji, John Juma, Davis Kuchaka, Yiqun Li, Perceval Maturure, Nicola Mulder, Eric Murimi, Brian Ogoti and Evalyne Wambugu.**

Furthermore, we would like to thank the following institutions, societies, journals and individuals from whom we sourced open-access resources, used in this module:
Dockstore, Nf-core, Pathogen Health Alliance for Genomic Epidemiology, South African National Bioinformatics Institute, Theiagen Genomics, WorkflowHub; nf-core GitHub contributors, PHA4GE GitHub contributors, Theiagen contributors.