

# Bioinformatics Containerization Tutorial

## Session 5: Nextflow Mini-Pipeline

Dr. İsmail K. Sağlam

November 15, 2025

### Contents

<b>1 Overview</b>	<b>2</b>
<b>2 Design of the mini-pipeline</b>	<b>2</b>
<b>3 main.nf: minimal pipeline</b>	<b>2</b>
<b>4 nextflow.config: local Docker use</b>	<b>3</b>
<b>5 Running Nextflow locally with Docker</b>	<b>4</b>
<b>6 Running Nextflow on KUACC with Apptainer + SLURM</b>	<b>5</b>
<b>7 Summary</b>	<b>5</b>

# 1 Overview

In this session you will:

- Wrap the ANGSD and PCAngsd pipelines into a minimal Nextflow workflow.
- Use your existing Docker image locally.
- See how to lift the same pipeline to KUACC with Apptainer + SLURM.

We assume you already have:

- `isophya-course:0.1` built.
- `scripts/01_call_genotypes.sh`, `scripts/02_pcangsd_pipeline.sh`.
- A working `compose.yaml` from Session 2.

## 2 Design of the mini-pipeline

We build two processes:

- `call_genotypes`: runs `01_call_genotypes.sh`.
- `pcangsd_pipeline`: runs `02_pcangsd_pipeline.sh`.

The workflow is:

1. `call_genotypes` runs first.
2. `pcangsd_pipeline` depends on its completion.

We do not pass many files explicitly; instead we rely on the same directory layout and environment variables as in Docker/Apptainer.

## 3 `main.nf`: minimal pipeline

Create `workflow/main.nf`:

```
#!/usr/bin/env nextflow

nextflow.enable.dsl = 2

params.pop      = params.pop ?: 'isophya71'
params.threads  = params.threads ?: 8
params.workspace = params.workspace ?: '/workspace'
params.scriptdir = "${params.workspace}/scripts"

process call_genotypes {

    tag { params.pop }

    publishDir '/results', mode: 'copy', overwrite: true

    input:
    val pop from Channel.value(params.pop)

    output:
    val pop into geno_done
```

```

script:
"""
cd ${params.workdir}
export POP=${pop}
export THREADS=${params.threads}

./scripts/01_call_genotypes.sh

echo "${pop}" > /results/.geno_done_${pop}
"""
}

process pcangsd_pipeline {

    tag { params.pop }

    publishDir '/results', mode: 'copy', overwrite: true

    input:
    val pop from geno_done

    script:
"""
cd ${params.workdir}
export POP=${pop}
export THREADS=${params.threads}

./scripts/02_pcangsd_pipeline.sh

echo "${pop}" > /results/.pcangsd_done_${pop}
"""
}

workflow {

    call_genotypes()
    pcangsd_pipeline()
}

```

---

## 4 nextflow.config: local Docker use

Create workflow/nextflow.config:

```

profiles {

    docker_local {

        process.executor = 'local'
        process.container = 'isophya-course:0.1'
        docker.enabled = true

        workDir = 'work'

        process {
            withName: 'call_genotypes' {
                cpus = 8

```

```

        memory = '32 GB'
    }
    withName: 'pcangsd_pipeline' {
        cpus = 8
        memory = '32 GB'
    }
}

docker.runOptions = """
-v ${baseDir}../data:/data:ro
-v ${baseDir}../scripts:/workspace/scripts:ro
-v ${baseDir}../results:/results:rw
-w /workspace
"""
}

kuacc_apptainer {

    process.executor = 'slurm'
    workDir = '/scratch/$USER/nextflow-work'

    process.container = '/home/iksaglam/oulu/isophya-course_0.1.sif'

    singularity.enabled = true
    singularity.autoMounts = false

    process {
        withName: 'call_genotypes' {
            cpus = 8
            memory = '32 GB'
            time = '8h'
            clusterOptions = '--partition=short'
        }
        withName: 'pcangsd_pipeline' {
            cpus = 8
            memory = '32 GB'
            time = '8h'
            clusterOptions = '--partition=short'
        }
    }

    singularity.runOptions = """
--bind /home/iksaglam/oulu/data:/data:ro
--bind /userfiles/utopalan22/isophya/new_bams:/data/bams:ro
--bind /userfiles/utopalan22/isophya/references:/data/ref:ro
--bind /home/iksaglam/oulu/results:/results
--bind /home/iksaglam/oulu/scripts:/workspace/scripts:ro
--pwd /workspace
"""
    }
}

```

## 5 Running Nextflow locally with Docker

From `bioinf-containers-course/workflow`:

---

```
cd ~/bioinf-containers-course/workflow

nextflow run main.nf -profile docker_local \
--pop isophya71 \
--threads 8
```

---

Nextflow will:

- Pull `isophya-course:0.1` if needed.
- Run `call_genotypes` inside the container.
- Then run `pcangsd_pipeline`.

Check results:

---

```
ls ../results
```

---

## 6 Running Nextflow on KUACC with Apptainer + SLURM

Once KUACC's Apptainer and SLURM integration is working, you can run:

---

```
cd ~/oulu/workflow

nextflow run main.nf -profile kuacc_apptainer \
--pop isophya71 \
--threads 8
```

---

Nextflow will:

- Submit `call_genotypes` as a SLURM job using the `isophya-course_0.1.sif`.
- After completion, submit `pcangsd_pipeline`.
- Use the same bindings as in Session 4.

## 7 Summary

After Session 5 you can:

- Express your ANGSD and PCAngsd workflows as a Nextflow pipeline.
- Run the entire analysis locally with Docker from one command.
- Reuse the same container and bindings on KUACC with Apptainer + SLURM.