

Bioinformatics Containerization Tutorial

Session 1: Project Setup and Base Container

Dr. İsmail K. Sağlam

November 15, 2025

Contents

1	Overview	2
2	Project directory layout	2
3	Create the repository directory	2
4	Define the Conda environment	2
5	Write the Dockerfile	3
6	Build the container	3
7	Test the container	3
8	Run with mounted data/results	4
9	Initialize Git and push to GitHub	4
10	Using prebuilt BioContainers for QC	4
10.1	Run prebuilt <code>samtools</code>	4
10.2	<code>samtools flagstat</code> on a BAM	4
10.3	Run <code>fastqc</code> on a BAM	5
11	Summary	5

1 Overview

In this session you will:

- Create a clean project directory for the course.
- Define a Conda environment for basic population-genomics tools.
- Build a Docker image using `micromamba`.
- Test the container and mount `data/results` directories.
- Initialize a Git repository.
- Use prebuilt BioContainers (`samtools`, `fastqc`) for basic QC.

2 Project directory layout

Target layout:

```
bioinf-containers-course/  
data/                # BAMs, metadata files, lists etc.  
containers/  
  Dockerfile  
  environment.yml  
workflow/  
  compose.yaml  
  nextflow.config  
  main.nf  
scripts/            # Shell scripts, R scripts etc.  
hpc/  
  run_apptainer.sbatch  
  Apptainer.def  
results/            # All outputs from analysis will go here
```

3 Create the repository directory

From your home directory:

```
mkdir -p bioinf-containers-course/{data,results,containers,workflow,scripts,hpc}  
cd bioinf-containers-course
```

4 Define the Conda environment

Create `containers/environment.yml`:

```
name: base  
channels:  
  - bioconda  
  - conda-forge  
dependencies:  
  - python=3.11  
  - r-base=4.3  
  - samtools=1.20
```

```
- bcftools=1.20
- angsd=0.940
```

This file defines the core software stack used inside the container.

5 Write the Dockerfile

Create containers/Dockerfile:

```
# syntax=docker/dockerfile:1

FROM mambaorg/micromamba:1.5.10

COPY containers/environment.yml /tmp/environment.yml
RUN micromamba install -y -n base -f /tmp/environment.yml && \
    micromamba clean -a -y

# Ensure login shells (bash -l / bash -lc) include conda binaries
USER root
RUN printf 'export PATH=/opt/conda/bin:$PATH\n' > /etc/profile.d/00-conda-path.sh
USER mambauser

WORKDIR /workspace

ENTRYPOINT ["/usr/local/bin/_entrypoint.sh"]
CMD ["bash"]
```

6 Build the container

From the project root:

```
cd ~/bioinf-containers-course

docker build --no-cache -t isophya-course:0.1 \
    -f containers/Dockerfile .
```

7 Test the container

Interactive shell:

```
docker run --rm -it isophya-course:0.1 bash
```

Test with a login shell (important for SLURM/Apptainer later):

```
docker run --rm -it isophya-course:0.1 \
    bash -lc "echo $PATH; which angsd; which samtools; which R"
```

Expected paths (approx.):

```
/opt/conda/bin/angs
```

```
/opt/conda/bin/samtools
/opt/conda/bin/R
```

8 Run with mounted data/results

```
docker run --rm -it \
-v "$(pwd)/data:/workspace/data:ro" \
-v "$(pwd)/results:/workspace/results" \
isophya-course:0.1 \
bash -lc "samtools --version"
```

9 Initialize Git and push to GitHub

Initialize:

```
git init
git add .
git commit -m "initial project structure and container setup"
```

Add remote and push:

```
git branch -M main
git remote add origin \
https://github.com/iksaglam/bioinf-containers-course.git
git push -u origin main
```

10 Using prebuilt BioContainers for QC

Before using the custom image, it is useful to show how prebuilt BioContainers work.

10.1 Run prebuilt samtools

```
docker run --rm staphb/samtools:1.20 samtools --version
```

10.2 samtools flagstat on a BAM

Assume you have:

```
bioinf-containers-course/data/U_KAV_D04_sorted_flt.bam
```

Run:

```
docker run --rm \
-v "$(pwd)/data:/data:ro" \
staphb/samtools:1.20 \
samtools flagstat /data/U_KAV_D04_sorted_flt.bam
```

10.3 Run fastqc on a BAM

Display help:

```
docker run --rm biocontainers/fastqc:v0.11.9_cv8 fastqc --help
```

Run FastQC:

```
docker run --rm \
-v "$(pwd)/data:/data:ro" \
-v "$(pwd)/results:/results" \
biocontainers/fastqc:v0.11.9_cv8 \
fastqc /data/U_KAV_D04_sorted_flt.bam --outdir /results
```

Outputs:

- results/U_KAV_D04_sorted_flt_fastqc.html
- results/U_KAV_D04_sorted_flt_fastqc.zip

11 Summary

At the end of Session 1 you have:

- A clean project layout under `bioinf-containers-course/`.
- A reproducible Docker image `isophya-course:0.1`.
- Basic QC tools demonstrated via prebuilt containers.
- A GitHub repository tracking your setup.