Training materials hackathon

Containers & Workflow Pipelines

This is a draft document for assembling materials, information, etc. in function of the training materials hackathon for containers & workflow pipelines.

Meeting platform & details

Day 1: **10 February 2021, online**
Meeting platform link: https://zoom.us/j/96985097738?pwd=SHc4VGJZc3h4WlV6ZTZTdWx0WkNQZz09
Password: 441148

Content of this document

Situation

Please refer to our GitHub repository for the broader scope. Ultimately, we would like to assemble the materials here in an open and version controlled manner.

ELIXIR Feedback survey

ELIXIR Feedback survey for the hackathon is now open. Here is a link to a short registration form that you need to fill in to get access to the survey. A link to the survey is in the registration form.
https://elixir.mf.uni-lj.si/course/view.php?id=63

Participants

Please adjust your name and affiliation if our guess is incorrect!

*In remarks: Feel free to write something you would like to share with the group e.g. what you preferably want to achieve with the hackathon, or what you're currently doing yourself on these topics.

Name	Affiliation	Remarks*
Alexandre Francisco	Instituto Superior Técnico Lisboa, PT gh:	working with ELIXIR PT on infrastructure, deployment for tech infrastructure to life sciences
Alexandros Kanterakis	Institute of Computer Science, FORTH, GR gh: kantale	bioinformatician, experience reproducible pipelines, have a lot of use cases (text mining regarding COVID), snakemake, nextflow, CWL No training material yet, but have internal pipelines in: (1) snakemake pipelines for biomedical text parsing in BioC, (2) snakemake pipeline for variant annotation with VEP + Annovar.
Alexander Botzki	VIB Bioinformatics Core, ELIXIR BE, BE gh: abotzki	bioinformatician, teaches docker, keen interest in reproducibility
Ana Portugal Melo	Instituto Superior Técnico Lisboa, PT gh:	
Brane Leskosek	ULjubljana Medicinska Fakulteta, ELIXIR SI, SI	apologies for the first day
Geert van Geest	SIB, CH gh: GeertvanGeest	joined SIB Switzerland in July 2020, UBern, snakemake, nextflow, teach NGS courses, containers, improve materials

		We're currently developing an online course on containers (Docker and Singularity): https://github.com/sib-swiss/containers-introduction-training and is hosted at https://sib-swiss.github.io/containers-introduction-training/ We also have gained experience in deploying AWS cloud services for teaching/training in an environment mimicking a shared computer environment (e.g. a HPC).
John Sundh	SciLifeLab, SE gh: johnne	at NBIS, metagenomics expertise, snakemake workflows, containers (to some degree), NBIS course tools for reproducibility, 3 years (twice per year) We've been running a course on Tools for reproducible research for close to 3 years now. It includes lectures and tutorials on git, conda, snakemake, jupyter, rmarkdown, docker and singularity. Last time, in November 2020, we ran it online.
Jose Espinosa-Carrasco	Centre for Genomic Regulation, Barcelona, ES gh:	CRG, with C. Notredame, participating in nextflow courses, working together with CRG Bioinfo Core. Developer of Nextflow pipelines. Last time we run the course we used these materials https://bovreg.github.io/nf-workshop20/
Julia Ponomorenko	Centre for Genomic Regulation, Barcelona, ES gh:	
Luca Cozzuto	Centre for Genomic Regulation, Barcelona, ES gh: lucacozzuto	Bioinformatician at bioinformatics core, developer of NF pipelines. We have organized several courses about linux containers and nextflow. https://biocorecrg.github.io/
Marko Vidak	ULjubljana Medicinska Fakulteta, ELIXIR SI, SI gh: marko_whatever	Tech Coordinator for ELIXIR SI We have organized several ELIXIR training courses (e.g genome assembly and annotation) that used

		containerized tools for hands-on exercises.
Mateusz Kuzak	eScience Center, NL gh: mkuzak	plan to teach https://carpentries-incubator.github.io/docker-introduction/ soon Python - one episode about snakemake is available
Maxime Garcia	nf-core, SciLifeLab, Karolinska Institutet, SE gh: MaxUlysse	Bioinfo, developing pipelines in Nextflow. Part of the nf-core core group that manages the community. We held hackathons (about using nf-core pipelines, with tutorials on how to use Nextflow...), and the next one is in March (22-24). We work closely with the Nextflow devs. Experienced in Nextflow, AWS.
Michael R. Crusoe	CWL, ELIXIR-NL gh:	
Pedro Fernandes	Instituto Gulbenkian de Ciência, PT gh:	The Gulbenkian Training Programme in Bioinformatics, materials development, training course design https://github.com/GTPB https://github.com/GTPB/ARANGS16
Renuka Kudva	nf-core, SciLifeLab, SE gh: renbot-bio	I co-ordinate some of the outreach activities of nf-core where I work with Maxime Garcia and others in the nf-core team. We have our next hackathon March 22-24 and are looking to develop course materials. Biochemist and Molecular Biologist by training - not a bioinformatician. Looking to learn by doing.
Sarah Bonnin	Centre for Genomic Regulation, Barcelona, ES gh: sarahbonnin	We (bioinformatics unit at CRG) are developing pipelines in Nextflow that use Docker/Singularity containers. We have been teaching those topics several times. I am quite a beginner in those topics myself, but I am quite involved in training in general so hopefully I can contribute in the “getting started” aspect of the training material.
Toni Hermoso Pulido	Centre for Genomic Regulation, Barcelona, ES gh:	

Tuur Muyldermans,	VIB Bioinformatics Core, ELIXIR BE, BE gh: tmuylder (vibbits)	We're looking forward to teaching these topics ourselves, as there is a high interest. We could benefit from discussing technical setups (make sure that all students have the same environment/OS), as well as more example exercises, eventually use-cases.

Preliminary program

Here is a suggestion of how the first day will look like. This program serves merely as a guideline for getting ourselves organised. We might have to dividdevelop Docker ime in different groups depending on interests and expertises (e.g. Containers vs Workflows).

Time	Program	Detail
9:30 - 10:00	**Welcome & Tour de table**	Who are you, why are you here, what is your expertise, what would you like to achieve…
10:00 - 11:00	**Discuss materials that exist**	- Discuss list with existing materials - Identify current gaps & opportunities e.g. technical set-up, training materials, hands-on exercises / use-cases.
11:00 - 11:15	Break	
11:15 - 11:45	**Define learners' profiles (target audience)**	- How should we extend on current materials depends on who will follow the workshops e.g. people that want to use it, or people that want to develop their own containers/pipelines.
11:45 - 12:30	**Define learning objectives and outcomes**	With the goal of knowing how and what we will create ourselves.
12:30 - 1:30	Lunch	
1:30 - …	Continue the above if unfinished.	
1:30 - 4:30	**Translate into actions**	In groups probably (depending on interests, expertise, etc.) with regular check-ins: translate the learning objectives & outcomes into action points so we can focus on developing materials on the second day.

		Ultimately we can already start developing during the afternoon.
4:30 - 5:00	**Wrap-up**	Wrap-up the day, define tasks for next hackathon day & decide on the next date

As a reference: repository for the training materials hackathon on Machine Learning (here)

Existing material

@all: feel free to add related content you are already aware of here:

Containers

1. https://biocorecrg.github.io/ELIXIR_containers_nextflow/
   Note: ELIXIR course "Containers and Workflow Pipelines for reproducible and automated data analysis" (26-27 October, 2020): Two full day hands-on course (run online). The course has been organized and supported by the VIB Bioinformatics core.
   25 people attended the course. The course materials are available on GitHub.

2. Workshop e-learning materials & exercises (Docker and Singularity)

   Notes: theory materials for Docker & Singularity in 'e-learning' format.
   introductory material with exercises linked to RNAseq
   Accompanied with exercises that focus on using containers rather than how to build them. These materials should hence familiarize students on how to use these technologies.
   exercises -

3. https://sib-swiss.github.io/containers-introduction-training/ , which is inspired by https://carpentries-incubator.github.io/docker-introduction/, and added Singularity material cherry picked since combination of docker and singularity

   tech setup: NGS courses set up with Amazon cloud server, with login ssh (bash code for setup here: https://github.com/geertvangeest/AWS), all tools provided with conda. Conda conda.yml provided to participants. Example for providing conda env, docker images to do a course: https://sib-swiss.github.io/NGS-variants-training/day1/server_login/

4. https://ome.github.io/training-docker/
5. https://carpentries-incubator.github.io/docker-introduction/
   Mateusz does not contribute to the lesson development

6. https://carpentries-incubator.github.io/singularity-introduction

7. https://nbis-reproducible-research.readthedocs.io/en/latest/
   course covers conda, snakemake, git, Jupyter, R markdown, singularity
8. https://github.com/orchid00/The_Carpentries_info/blob/master/carpentries_style_shared_lessons.md#docker--singularity---containers

9. GTPB
   professional team at the U Lisboa provides infrastructure
   Openstack, ansible, terraform / volatile environment

   issue with participants can work further on materials
   'portable VM'

Workflow pipelines

1. **https://biocorecrg.github.io/ELIXIR_containers_nextflow/**

   Note: ELIXIR course "Containers and Workflow Pipelines for reproducible and automated data analysis" (26-27 October, 2020): Two full day hands-on course (run online). The course has been organized and supported by the VIB Bioinformatics core. 25 people attended the course. The rse materials are available on GitHub.
   web-based course with
   introduction to containers - why docker, why singularity in HPC, theory and demos
   introduction to nextflow - intro, playing with exercises,
   material in DSL2 (workflows, modules)
   examples with RNAseq sequence, mapping, QC

   technical setup: Ubuntu on GCE, everyone had access to the same machine, VNC

   Jose:
   alternative: Amazon , browser, terminal also in the browser
   https://aws.amazon.com/cloud9/

2. VIB BITS Nextflow workshop

   Notes: workshop materials (mainly) in DSL2 aiming to get familiar with the Nextflow syntax by explaining basic concepts and building a simple RNAseq pipeline. Highlights also reproducibility aspects with adding containers (docker & singularity). Slides available here.
   also starting with DSL1 and continue with DSL2

3. https://carpentries-incubator.github.io/workflows-snakemake

4. https://snakemake.readthedocs.io/en/stable/tutorial/tutorial.html

5. https://edwards.sdsu.edu/research/snakemake-tutorial/

6. https://ulhpc-tutorials.readthedocs.io/en/latest/bio/snakemake/

7. https://nbis-reproducible-research.readthedocs.io/en/latest/
   course covers conda, snakemake, git, Jupyter, R markdown, singularity

8. https://seqera.io/training/

   FYI https://www.nextflow.io/blog/2020/learning-nextflow-in-2020.html

9. https://nf-co.re/usage/nf_core_tutorial
   community is the driver of the pipelines
   nf-core intro -
   how to run pipelines
   tiny on intro for git (CI with github actions)
   create nextflow profile
   create a nf core pipeline
   touch on conda environment
   with DSL2 - more atomize approach with containers (-> biocontainers), eventually not up to date with all respects to DSL2 (as we're updating our pipelines)

   participants level: advanced users (for developing pipelines), but beginners also for users who want to run pipelines
   slack 1000 participants, 500 contributors on github, video introductions planed
   mentorship system will be introduced

10. Analysis pipelines with Python (hpc-carpentry.github.io)
    snakemake is also in the course as an example

11. Parallel Programming in Python (escience-academy.github.io)
    contains a section about snakemake as a use case for parallisations

12. Linux containers and Nextflow (biocorecrg.github.io/CRG_Containers_Nextflow/)

    Two days course on docker and singularity + nextflow DLS2. From scratch to real use using simple examples that become more and more complex adding features that are explained during the course. We start directly with DLS2 (workflows and modules) and explain the integration with BioContainers.

13. https://bovreg.github.io/nf-workshop20/
    Nextflow training including a containers section

based on sequera course, 4 days, online, section for Groovy
a lot of persons were beginners, seems to be difficult sometimes

14. Recording computational steps - Reproducible research (coderefinery.github.io)

comment from Jose:
workflow manager - which are the differences between the two
snakemake - nextflow

(Short) discussion

Regarding what could/should be valuable to work on.

- Self standing course on the topics (beginners, advanced or specific)
- Embedding into other courses

presentation on top of each exercises / with group discussion after the exercises

library of exercices:

skeleton: building blocks with exercises - all together they fulfill the overall objectives

design requirement:

- short 'learning' objects / under 10 min
- video give guidelines how to find the material
  please read this material on github
- Freestanding, short duration videos & text which people can use.

exercises as learning outcome
progressive and good exercises

learning outcomes for basic concepts / advanced concepts

focus on which technologies: docker, singularity, nextflow, snakemake

starting with the exercises / compiling video material in a later stage

basic concepts:
- why containers are needed?
- automate tasks -> scalability

target audiences:

workflows tools
- distinction between 'end user' only people who are interested in developing pipeline

make containers - use containers (more tightly connected to each other)
- no clear distinctions between users/developers, security issues

Learner's profiles

In break-out rooms

1. Who are they?
2. What problem are they having?
3. How will the tutorial/workshop help them?

breakout room 1:

Containers (users and creating containers - more tightly connected)
Alexander, Marko, Mateusz, Geert, Pedro

Persona 1

who are they?

Researcher / RSE who wants to reuse others analysis / pipeline / software which has been containerised.

What challenges are they facing?

They don't know how to use / run containers, how to troubleshoot issues with running them.

The incomplete documentation of containers they would like to use.

How will the lesson/workshop help them?

Being able to run a basic container from a registry

Assessment

Persona 2

Who are they?

Researcher / RSE who wants to make their analysis / pipeline / software more reusable and reproducible.

What challenges are they facing?

They have challenges to understand containers conceptually

Current documentation from Docker/Singularity is not for beginners

How will the lesson/workshop help them?

Make sure their documentation is clear and complete.

Give a template on which they can build further

Assessment

1. Who are they?
   - people have heard of it and what to use it because they want to use pipeline

   - people who want to reuse others' software which have been containerized

   - people that have recently discovered that they have to work more reproducibly (aiming at automation, mass processing)

   - people realise that others cannot reproduce their analysis (installation, dependencies, testing, example)
   mybinder / Jupyter / docker file

   - go from images to dockers / tweak existing container recipes

2. What problem are they having?

   persona 1:

   develop Docker images locally
   singularity for deployment - HPC - command line / production pipeline
   aiming at automation for mass processing
   docker / conda - hard to understand the concept of docker vs conda environment

   persona 2:

   persona 3:

3. How will the tutorial/workshop help them?

   containers do not solve the problem of correctness of analyses
   isolated environment / share image after tinkering a live images
   documentation of docker recipes

prerequisites: basics, 'system admin, installation', depending on the background

breakout room 2: Nextflow users (more introductory - concepts why - exercises explaining fundamentals)
Sarah, Toni, Tuur, Renuka

- Who are they?
  - Already have some background with bioinformatics data analysis ⟶ "Applied" bioinformaticians?
  - Linux users ⟶ they know how to use CLI
- What problem are they having?
  - Big data, or very similar data and do same process over and over again. They have some (basic) experience in CLI/Linux, so they are able to run a pipeline and understand what's happening or tune the parameters etc.
- How will the tutorial/workshop help them?
  - Understand conceptually, Understand how to run, debugging - know what which
  - errors might mean what, resuming/retry, workdir, if process fails ⟶ where to find fails (.command.*), .nextflow.pid
    - Reproduce failing process from workdir (maybe coupled to run from associated container)

prerequisite is linux users, Dataflow, so conceptually, git / GitHub

breakout room 3: Nextflow creating pipelines / developers
Luca, Jose, Maxime

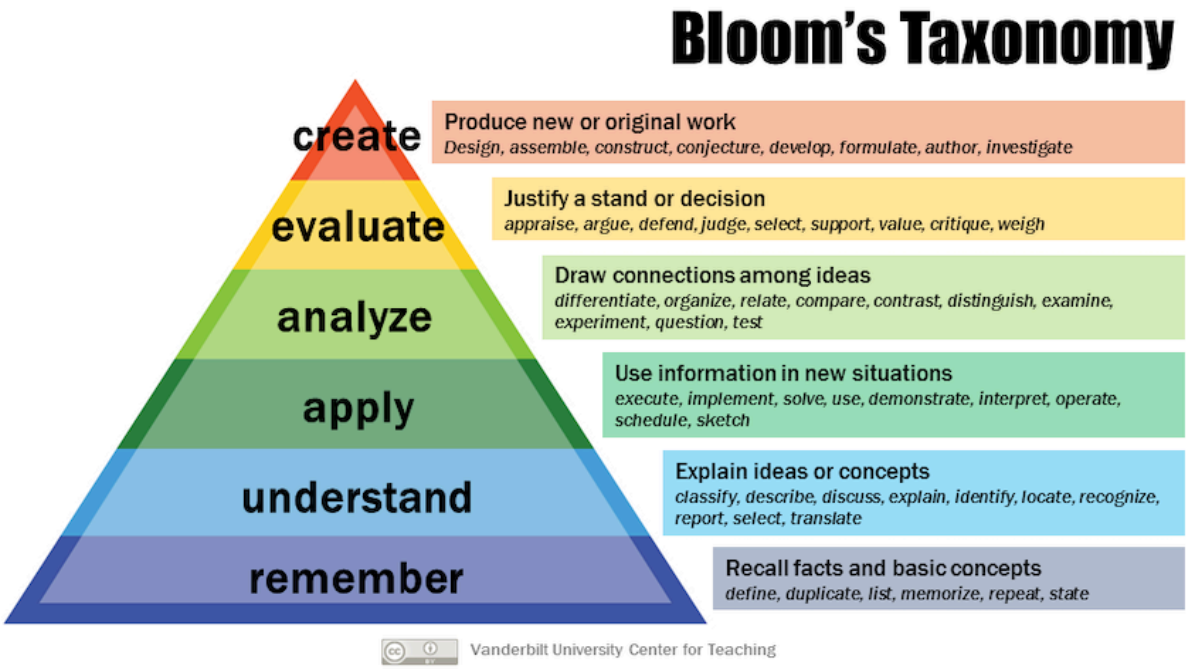
- Who are they?

    - bioinformaticians that want to develop using nextflow (both beginners and more advanced)
- What problem are they having?
    - parallelizing, reproducibility, portability etc.
    - porting pipelines to NF
    - porting DSL1 to DSL2
- How will the tutorial/workshop help them?
    - learning basics and pointers with more resources / community etc
    - they will know a list of useful tools / resources for programming / editing
    - debugging

Learning objectives and outcomes

In break-out rooms(?)

- Prerequisites for the workshop
- Define exercises with learning outcomes
- Goals: “After following one of these tutorials, learners will be able to …” - Blooms taxonomy:



Furthermore: We should be able to think of how we will address the Learning Outcomes. If there is no such way, we should re-assess the Learning Outcomes. Learning should take place in whichever set of levels without skipping intermediate levels. Likewise, assessment should not be run above the level of learning provision, e.g. if you teach at the analise level (and below) you cannot aim at finding good learning at the create and evaluate levels.

Breakout room 1: containers

- on a general note: put timing on exercises

- Prerequisites for the course/tutorial

  Persona 1. basic command line

  Persona2.

  - knowledge on command line in UNIX
  - know how to change permission/ working directory
  - package managers
  - use git
  - account on docker hub
  - understanding a Python script

- Define exercises with learning outcomes (measurable)
  - check whether docker is properly setup (docker run hello-world)
    - Participants are able to check whether their environment is running
  - running a container with docker run, outcomes:
    - Participant can run a container from dockerhub
  - tagging
  - run a command
  - process listing
  - pruning/cleaning
  - attached + interactive mode (-it)
  - detached mode
  - running web services, mapping ports
  - volumes, bind-mounts
  - parameters (working dir, user)
  - dockerfiles

- handling permissions

- Goals: “After following one of these tutorials, learners will be able to …” - Blooms taxonomy:

Persona 1.

- Find a image from docker hub containing bwa
  (ex: https://hub.docker.com/r/biocontainers/bwa)

  Exercise:

- Align reads in file x.fastq.gz to reference genome y.fa using this container with the latest BWA version from dockerhub, write the alignments to aln-pe.sam file.

  bwa mem ref.fa read1.fq read2.fq > aln-pe.sam

  in analogy :
  docker run --rm --name fastqc_albot -u="$(id -u):$(id -g)" -w="/data/"

  -v ~/workshop-janssen/data/:/data
  quay.io/biocontainers/fastqc:0.11.9--0
  /bin/bash -c "fastqc WT*.fq.gz"

  >> bash downl-data.sh or chmod u+x downl-data.sh && ./downl-data.sh && rm -f *.fa

Persona 2.

containerize something container Dockerfile

exercise

- fetch script from git repo

  - write a Dockerfile which will create an image with a specific version of [tool] build from source with resolved dependencies, other can run this as command line tool with the parameters provided on CLI
  - put it on Docker Hub

  - requires specific python dependencies

  - default command and entrypoint

Dockerfile: alpine, bwa, resolve dependencies, ...

measurable outcome:

two person exercise

- image on docker hub
- produce a visualisation which is exported to png

breakout room 2: Nextflow users

Nextflow users (more introductory - concepts why - exercises explaining fundamentals)
Sarah, Toni, Tuur, Renuka, Julia

- Prerequisites for the workshop
  - Basic knowledge of Linux/command-line. Familiar enough to navigate through the folders, inspect and change files (with nano or any other text-editor of preference), more than just running the literal commands given in the 'course'.
  - On a technical side.
- Define exercises with learning outcomes
- Goals: "After following one of these tutorials, learners will be able to …"
  - Introduction nextflow and language
    - Describe nextflow conceptually and the language.
    - Describe the concepts of channels, operators, processes, modules
  - Knowing where to find a pipeline (github, nf-core, ...) and which one to use.
  - Import a pipeline (download/clone locally or from github - but also possible to pull (nextflow pull)) https://github.com/nextflow-io/awesome-nextflow
    - Exercise nextflow-pull a specific pipeline (they need to find themselves) Extra: define release (or tag/release/commit)
    - Exercise git clone so they have a pipeline where they can work on / alternatively: create config file that will overwrite the config file.
  - Explain config files and being able to do minor modifications (eg. change version of docker container, parameters like computation power)
    - Config files: there might be parameters related to running things in the cloud/HPC which you are not allowed to change, but there might be also one related to the pipeline (so more regarding the bioinformatics tools)
    - Exercise inspect the config file and change a parameter (an easy one) and it should be clear that when you have changed it it gives a different output
    - understand how databases are used within pipeline
  - Execute a pipeline

        - Exercise nextflow run <pipeline> (toy pipelines = tiny pipelines) + fast, easy, simple. When more familiar → then go to a nf-core pipeline.
    - Locate and describe the output after running a pipeline → workdir with different files (.command.*).
        - Exercise when running the pipeline it created an output. Find where the output was created.
        - Exercise: find for that process which command was given to the computer (.command.sh)
    - Identify the different parameters : nextflow-specific versus pipeline-specific (- vs --)
        - Running in the background
        - Resume pipeline
            - Exercise delete a workdirectory and see that the pipeline resumes from somewhere else (in the beginning) people should be aware of the fact that it becomes very big. Some pipelines allow you to cp or mv the results inside the directory that you want (there is a parameter/flag for it so not necessary to change it inside the pipeline script).
            - Showcase the caching - run with a different parameter (file or the file has moved). When the file has moved → different timestamp → different cache.
        - ouputdir
            - see above
        - profile
            - Exercise with different profiles (if possible)
        - Create a report (-with-dag)
            - Exercise where they create a visual report
        - containers/singularity?
        - https://www.nextflow.io/blog/2020/cli-docs-release.html
        -
    - Analyze the error/failure of a pipeline - correct it appropriately (if technical is not relevant)
        - lack of memory and resources, file not found, docker not provided, error related to the tool (find it in the error message of the pipeline or in the workdir : .command.out/err)
            - One hyphen or two hyphen, wrong dash, file missing, missing parameter.
        - File issue on GitHub (if relevant) or contact community
            - Needs no further explanation.

Objective: Describe terminology/different concepts so you can explain to a developer what might need to be changed. To be able to find and run different existing pipelines. Do small trouble-shootings yourself (unrelated to the programming of the pipeline - at the level of config file). Navigate the workdir and find specific output(files). raise issues/contact community

breakout room 3: Nextflow pipeline developers

Nextflow creating pipelines / developers
Luca, Jose, Maxime

- prerequisite:
  - linux users, bash, CLI, know a programming language, git / GitHub
- strongly recommended
  - containers
- Define exercises with learning outcomes
  - Channel, Process and Workflow (To learn Nextflow DSL2)
  - Switching to DSL2 (For users already knowing Nextflow)
  - Assembling (sub)workflows
  - Configuration (To help with controlling the resource, parameters)
    - Change docker/conda/singularity - know what they can change
  - Debug (With some examples. To help users understanding the logic and the mindset)
- Goals: “After following one of these tutorials, learners will be able to …”
  - Recognize that reproducibility is an advantage
    - (-> using a workflow manager makes a pipeline reproducible and save time)
    - (-> using containers)
  - Remember Reproducibility
    - (Why we use workflow managers, NO MORE BASH SCRIPTS)
  - Understand dataflow programming model
  - Use configuration files (or profile)
  - Analyze what is causing an issue/bug, and where it’s coming from
  - Design their own pipeline

Warm-up event: https://nf-co.re/events/2021/hackathon-march-2021