Building Basic Workflows

Concept slides

Who am I?

Who are you?

Building basic workflow

Systemisation of bioinformatic commands, to enable swift execution.

Advantages

- Quick usage
- Reproducibility

Disadvantages

- Mostly specialists who can develop and maintain
- No-one documents your workflow for you

How comfortable are you with *bioinformatics*?

A bit about this Course

- Split into several parts
- Short thematic presentations
- Exercises -> Learning by ~~DOING?~~ ~~failing~~ trying!

Expectations

What will be your take away?

- A very basic modular workflow
- Ability to create modules
- Ability to implement modules into your workflow

What you WON'T learn

- Which bioinformatic tools does what
- How to utilize pipeline tools
- How to swim

Agenda – Day 1

Hour	Topics
09:00-09:35	Introductions
09:35-10:20	Virtual environments
Time for a	Break
10:35-11:35	Scripts
Time for a	Break
11:45-12:20	Making a module

Have you installed the *software requirements* and watched the *recommended online courses*?

01 What are Virtual environments?

Virtual Environments are an isolated subsystem where:

- Software doesn't interact with rest of the system
- Settings and configurations are separate from the rest of the system

01 Why Virtual Environments?

- Ensure compatibility with version-specific bioinformatics tools
- Prevent software conflicts, corruption, and errors from deprecated packages
- Allow multiple software versions to coexist seamlessly
- Simplify dependency management for a stable workflow

01 What is Anaconda

Anaconda is an online platform which hosts several software

- Founded for bringing Python programming into business data analytics

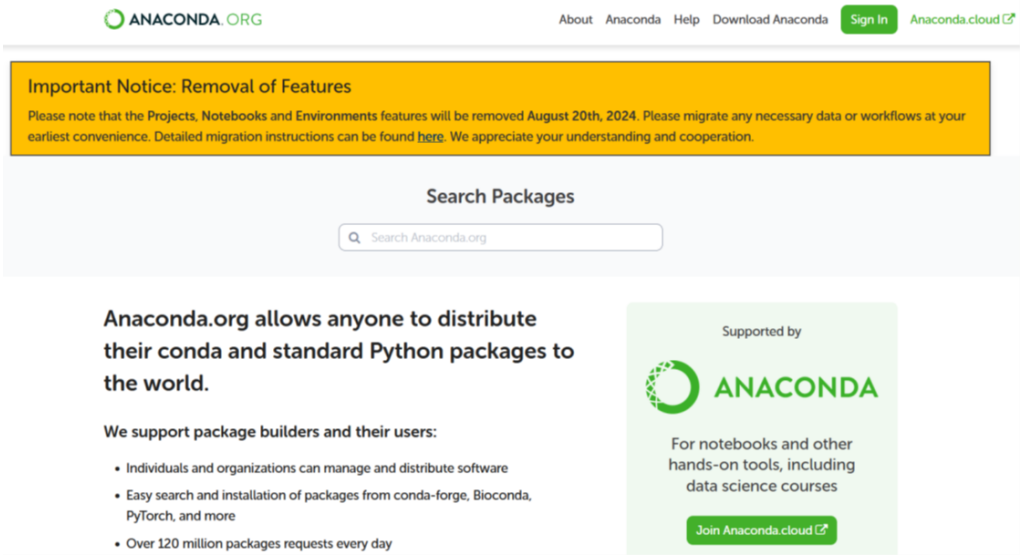
Rapidly expanded, hosting thousands of software packages

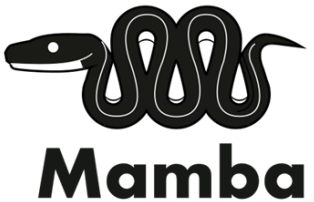
Several package managers:

- Anaconda (300+ preinstalled packages)
- Conda (160+ preinstalled packages)
- Miniconda (Few preinstalled)
- Mamba & Micromamba (Optimized versions of Conda and Miniconda)

01 Anaconda Platform

- Anaconda Hosting Platform
- Consist of several channel





01 What is a channel

- A subsection of the Anaconda platform, containing packages
- Channels are mostly community driven

- Common packages (n = 26000+)

- Bioinformatic centered packages

01 What is a package

A package holds a specific software and:

- Installation instructions (for the computer)

- All required software (called Dependencies)

Installing a package leads to installation of all relevant dependencies as well

01 Environments in Micromamba

command [subcommand] [-options] [arguments]

1) Create an empty virtual environment called Bobby

micromamba create -n Bobby

2) Install the package Joe from the awesome channel in the Bobby environment

micromamba install -n Bobby –c awesome Joe

01 Environments in Micromamba using YAML

Create the virtual environment Bobby using YAML file

micromamba create -f Bobby.yaml

Bobby.yaml

```
name: Bobby
channels:
 - awesome
dependencies:
- Joe
```

01 Environments in Micromamba

Run from Joe tool the help option in the Bobby environment

micromamba run -n Bobby Joe –help

Why are virtual environments important in bioinformatics?

03 Introduction on BASH scripting

- A series of commands contained into a text file is defined as **script**
- The shell executes the file's commands as if they were entered directly in the command line.
- Anything that can be run normally on the terminal can be put into a script and will do the same task
- By convention the bash script files possess the «**.sh**» extension

03 Example

run_amrfinder.bash

```
# Setup variables
assembly="E_coli.fasta"
output="output_ecoli.tsv"
identity=0.9
coverage=0.5

#Download database
amrfinder –-update

#Run AMRFinderPlus on assemblies
amrfinder --ident_min $identity \
        --coverage_min $coverage \
        --organism Escherichia \
        --nucleotide $assembly \


        --output $output
```

```
> bash run_amrfinder.sh
```

Time to make a Script

03 Positional arguments

1. **Essential for scripting tasks** – Arguments make scripts flexible and dynamic instead of static and hard-coded.

2. **Enable user input at runtime** – Passed when executing the script, allowing customization.

3. **Essential for automation** – Enable efficient and scalable scripting for repetitive tasks.

03 Example

run_amrfinder.sh

```
# Setup variables
assembly="E_coli.fasta"
output="output_ecoli.tsv"
identity=0.9
coverage=0.5

#Download database
amrfinder –-update

#Run AMRFinderPlus on assemblies
amrfinder --ident_min $identity \
        --coverage_min $coverage \
        --organism Escherichia \
        --nucleotide $assembly \


        --output $output
```

> **bash run_amrfinder.sh**

03 Example

run_amrfinder.sh

```
# Setup variables
assembly=$1
output=$2
identity=$3
coverage=$4

#Download database
amrfinder –-update

#Run AMRFinderPlus on assemblies
amrfinder --ident_min $identity \
        --coverage_min $coverage \
        --organism Escherichia \
        --nucleotide $assembly \


        --output $output
```

> **bash run_amrfinder.sh** E_coli.fasta output_ecoli.tsv 0.9 0.5

Today's achievement

Learned how to create and install virutal environment for specific tasks
Executing bioinformatic commands from environments
Converting bioinformatic commands into scripts
Converting Hard-coded values to positional arguments
Generating a script using above approach

What is the least clear right now?

Don't forget to feedback us!