

A tidy bioinformatics environment to the rescue!

A little treat to your fellows and future self

Introduction

How to draw an owl

1.



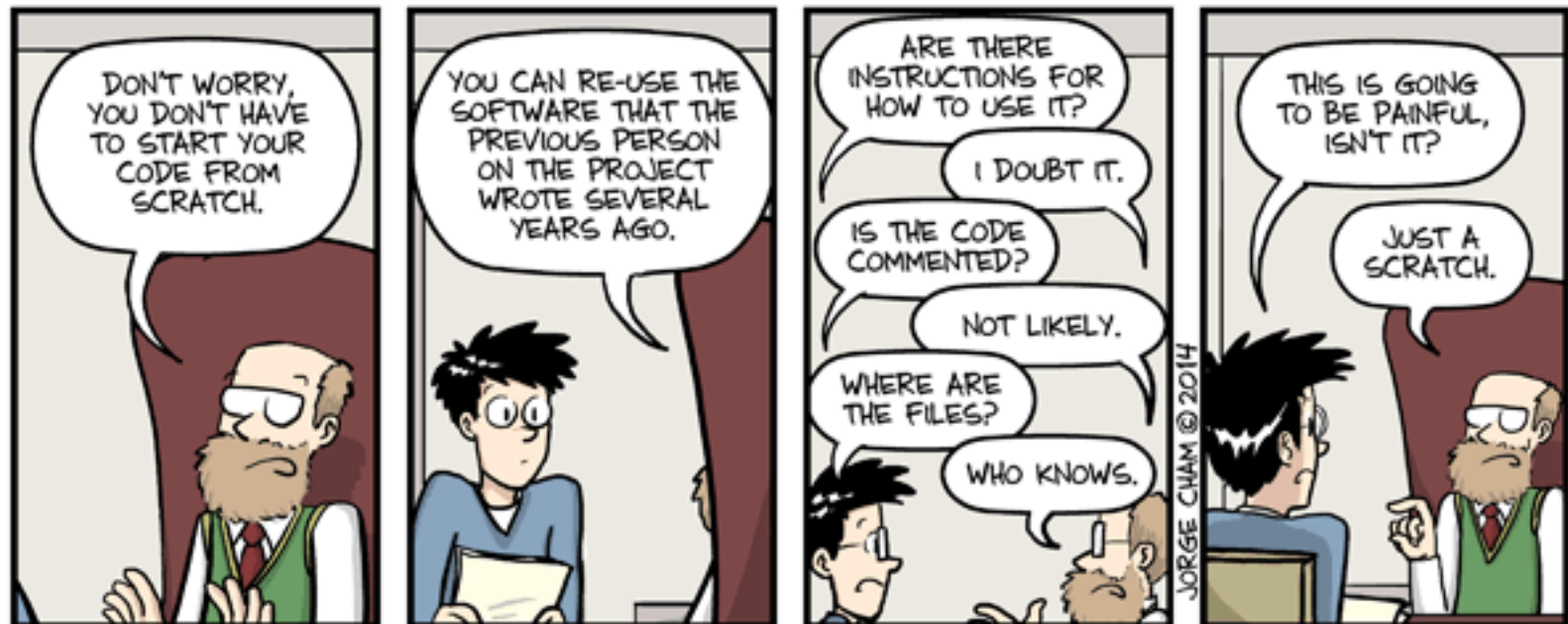
1. Draw some circles

2.



2. Draw the rest of the fucking owl

Piled Higher and Deeper *by Jorge Cham*



title: "Scratch" - originally published 3/12/2014 WWW.PHDCOMICS.COM

Environment management
Set up and manage the project environment

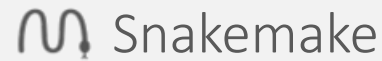
CONDA

Start here!

Version control
Track and backup your project history



Workflow management
Move from separate scripts to a connected analysis



Reports
Connect code, output and text in fancy reports



Notebooks
Document your exploratory analysis



Containerization
Make your project self-contained and distributable

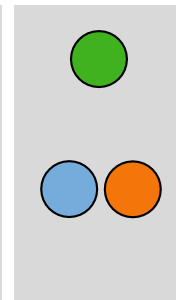
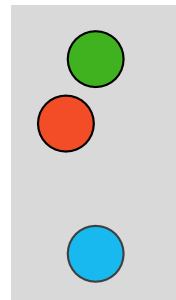
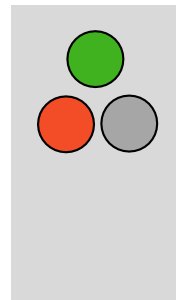
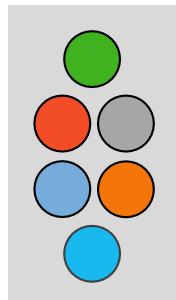


Do it all!

Workflow

Reproducible
environment

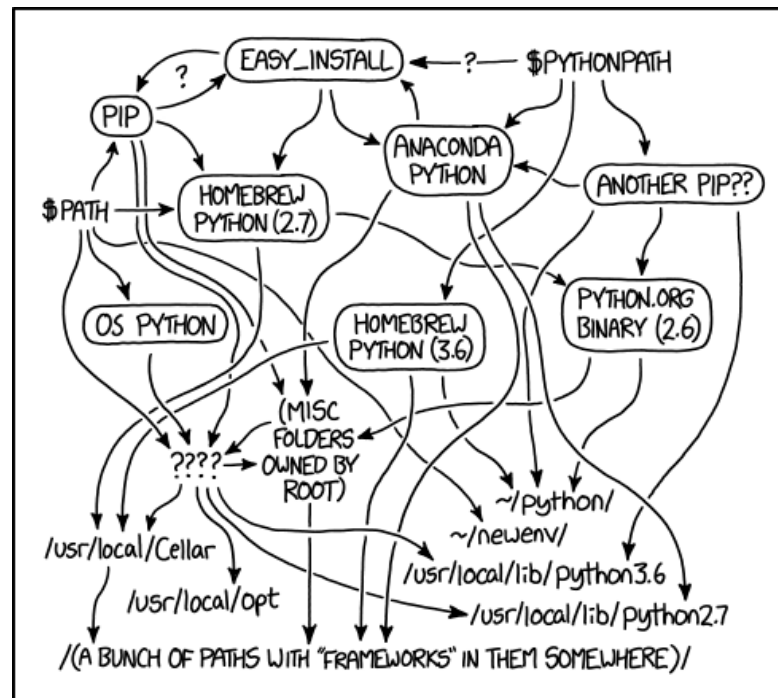
Interactive
notebooks



Package and environment managers

Pip & Conda

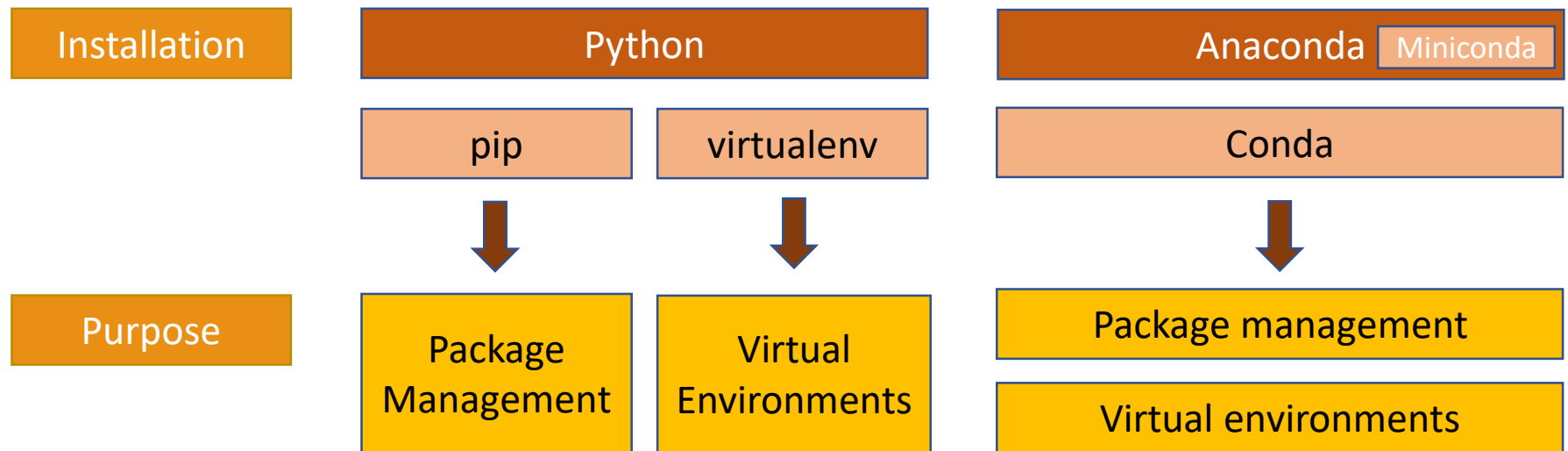
Package dependency problems



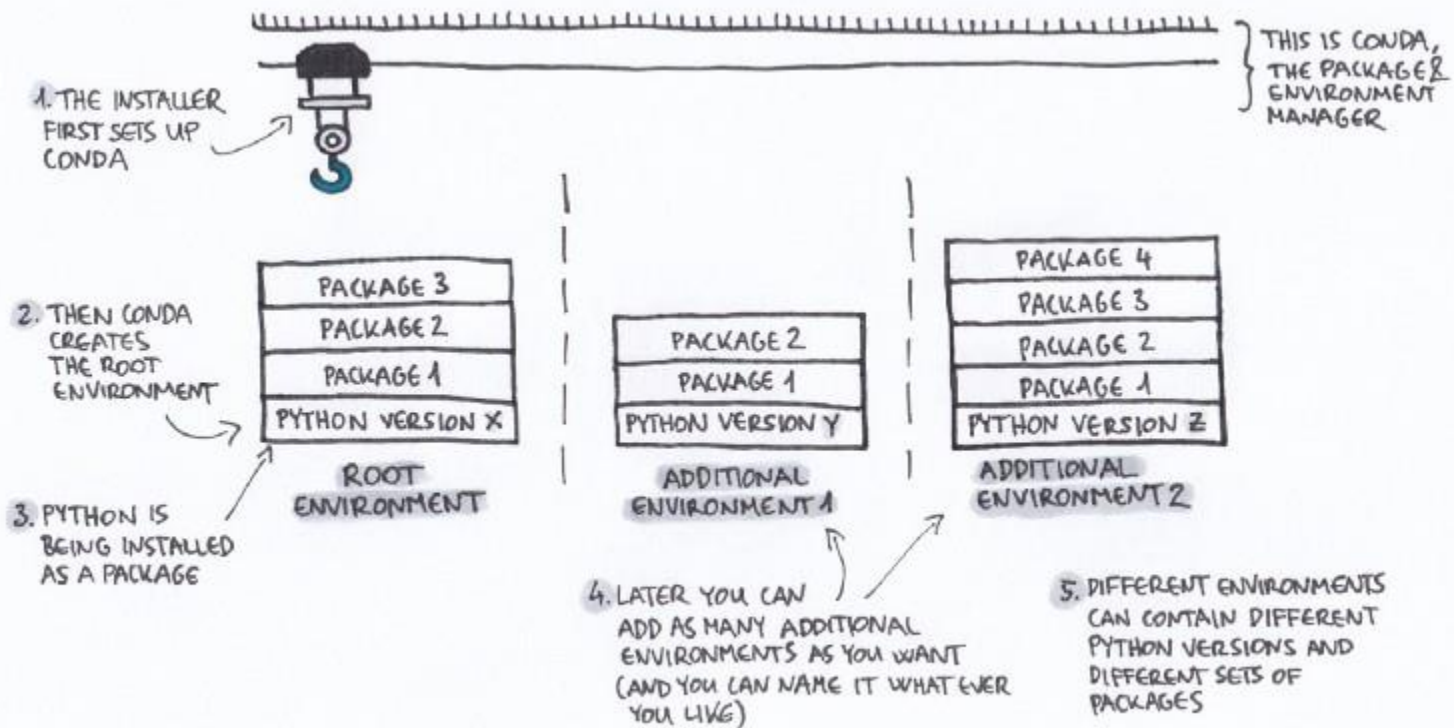
MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.



Package and environment managers



academind.com



freecodecamp.org



*Package, dependency and environment management for any language
Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN, and more.*

What is Conda?

- Finds, installs and updates packages
- Switch between environments for different versions
- Few commands make a totally separate environment with different versions of packages
- Combined with CI systems to provide frequent and automated testing of code

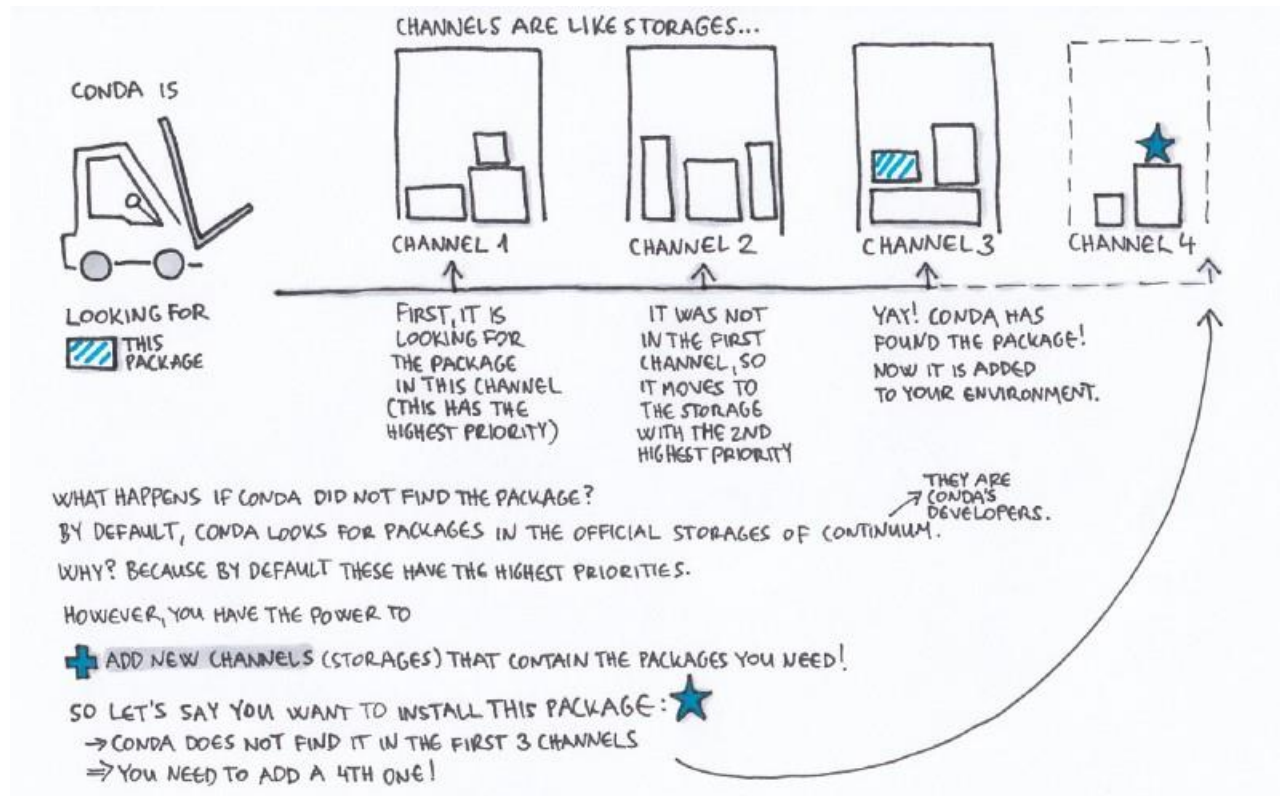
Packages

- Compressed tarball file (.tar.bz2) or .conda file
 - system-level libraries,
 - Python or other modules,
 - Executable programs, or other components
 - Metadata
 - Installation files

```
drwxr-xr-x  4 root root      4096 Jan 17 14:06 curl-7.67.0-hbc83047_0/  
-rw-r--r--  1 root root    136810 Jan 17 14:06 curl-7.67.0-hbc83047_0.conda  
drwxr-xr-x  5 root root      4096 Jan 17 14:07 fastqc-0.11.8-2/  
-rw-r--r--  1 root root   10021668 Jan 17 14:07 fastqc-0.11.8-2.tar.bz2
```

- Format is identical across platforms and operating systems

Channels



Channels



- >7000 packages
- A community-led collection of recipes, build infrastructure and distributions for the conda package manager
- e.g.: numpy, Scipy, CRAN packages, etc.



- >6000 packages
- Specializing in bioinformatics software
- e.g. samtools, fastqc, salmon, cutadapt, etc.

- Add your own channel

bioconda / packages / bwa 0.7.17



2

The BWA read mapper.

Conda

Files

Labels

Badges



License: GPL3



Home: <https://github.com/lh3/bwa>



244899 total downloads



Last upload: 1 month and 25 days ago

Installers

Info: This package contains files in non-standard labels.

conda install ?



linux-64

v0.7.17



osx-64

v0.7.17

To install this package with conda run one of the following:

```
conda install -c bioconda bwa
```

```
conda install -c bioconda/label/cf201901 bwa
```

Description

Environments

- Directory with specific collection of packages
- Switch between environments with *activate* and *deactivate*
- Directory structure
 - ROOT_DIR : where Ana/Miniconda was installed
 - /pkgs : decompressed packages
 - /envs : system location for additional conda environments

Pinning

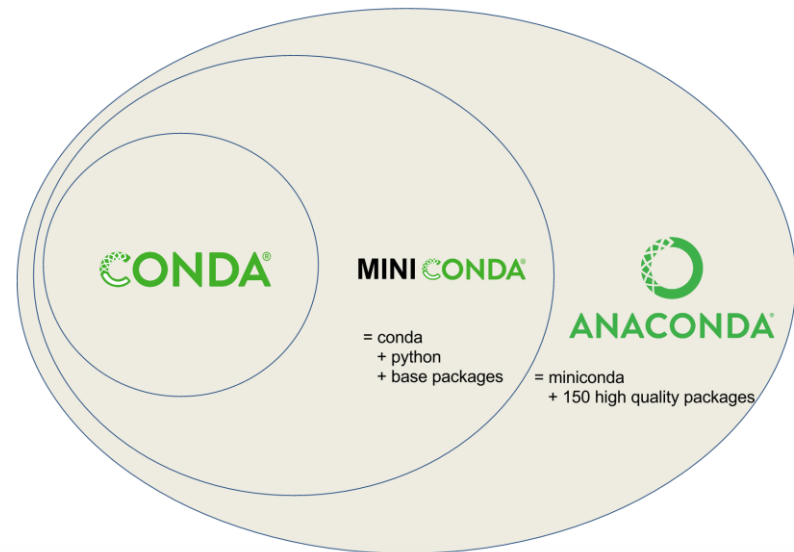
- Preventing packages from updating
- In the environment's conda-meta directory
 - File named *pinned* that includes a list of the packages that you do not want updated.

How it works

1. Installation & config
2. Channels
3. Environment
4. Packages
5. Reference & further reading

Installation

- Three different installers:
 - Miniconda
 - Anaconda
 - Anaconda Enterprise platform



- Miniconda
 - Conda (package & environment management system)
 - “root environment” with certain version of Python and few basic packages
- Anaconda
 - All of the above, and
 - 150+ packages
 - Navigator (GUI)

```
wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh && bash Miniconda2-latest-Linux-x86_64.sh
```

.condarc

- Conda configuration file

- `conda config --show`

- Add channels

- `conda config --add channels conda-forge`
 - `conda config --add channels bioconda`

Channels

- List channels
 - `conda config --get channels`
- Add a channel with lowest priority
 - `conda config --append channels newchannel`
- Add a channel with highest priority
 - `conda config --prepend channels newchannel`
- In order to install a package from a channel:
 - `conda install -c <channel> <package>`
- In order to automatically select channels you need to change your `.condarc`:
 - `conda config --add channels <my_channel>`

Environments

- Create a new environment and install a package in it
 - `conda create -n capita-selecta python=3.8 biopython`

```
## Package Plan ##

environment location: /home/tuur/.conda/envs/capita-selecta

added / updated specs:
- biopython
- python=3.8
```

Environments

- Create a new environment and install a package in it
 - `conda create -n capita-selecta python=3.8 biopython`

The following packages will be downloaded:

package	build		
biopython-1.76	py38h516909a_0	2.6 MB	conda-forge
ld_impl_linux-64-2.33.1	h53a641e_8	589 KB	conda-forge
libgfortran-ng-7.3.0	hdf63c60_5	1.7 MB	conda-forge
numpy-1.18.1	py38h95a1406_0	5.3 MB	conda-forge
pip-20.0.2	py_2	1.0 MB	conda-forge
python-3.8.1	h357f687_2	58.2 MB	conda-forge
setuptools-45.2.0	py38_0	655 KB	conda-forge
wheel-0.34.2	py_1	24 KB	conda-forge
Total:		69.9 MB	

Environments

- Create a new environment and install a package in it
 - `conda create -n capita-selecta python=3.8 biopython`

The following NEW packages will be INSTALLED:

<code>_libgcc_mutex</code>	<code>conda-forge/linux-64::_libgcc_mutex-0.1-conda_forge</code>
<code>_openmp_mutex</code>	<code>conda-forge/linux-64::_openmp_mutex-4.5-0_gnu</code>
<code>biopython</code>	<code>conda-forge/linux-64::biopython-1.76-py38h516909a_0</code>
<code>ca-certificates</code>	<code>conda-forge/linux-64::ca-certificates-2019.11.28-hecc5488_0</code>
<code>certifi</code>	<code>conda-forge/linux-64::certifi-2019.11.28-py38_0</code>
<code>ld_impl_linux-64</code>	<code>conda-forge/linux-64::ld_impl_linux-64-2.33.1-h53a641e_8</code>
<code>libblas</code>	<code>conda-forge/linux-64::libblas-3.8.0-14_openblas</code>
<code>libcbblas</code>	<code>conda-forge/linux-64::libcbblas-3.8.0-14_openblas</code>
<code>libffi</code>	<code>conda-forge/linux-64::libffi-3.2.1-he1b5a44_1006</code>
<code>libgcc-ng</code>	<code>conda-forge/linux-64::libgcc-ng-9.2.0-h24d8f2e_2</code>
<code>libgfortran-ng</code>	<code>conda-forge/linux-64::libgfortran-ng-7.3.0-hdf63c60_5</code>
<code>libgomp</code>	<code>conda-forge/linux-64::libgomp-9.2.0-h24d8f2e_2</code>
<code>liblapack</code>	<code>conda-forge/linux-64::liblapack-3.8.0-14_openblas</code>
<code>libopenblas</code>	<code>conda-forge/linux-64::libopenblas-0.3.7-h5ec1e0e_6</code>
<code>libstdcxx-ng</code>	<code>conda-forge/linux-64::libstdcxx-ng-9.2.0-hdf63c60_2</code>
<code>ncurses</code>	<code>conda-forge/linux-64::ncurses-6.1-hf484d3e_1002</code>
<code>numpy</code>	<code>conda-forge/linux-64::numpy-1.18.1-py38h95a1406_0</code>
<code>openssl</code>	<code>conda-forge/linux-64::openssl-1.1.1d-h516909a_0</code>
<code>pip</code>	<code>conda-forge/noarch::pip-20.0.2-py_2</code>
<code>python</code>	<code>conda-forge/linux-64::python-3.8.1-h357f687_2</code>

Environments

- Create a new environment and install a package in it
 - `conda create -n capita-selecta python=3.8 biopython`
- To use or “activate” the new environment
 - `conda activate capita-selecta`

Environments

- To see a list of all your environments

- `conda info --envs`

```
# conda environments:
```

```
#
```

```
capita-selecta /home/tuur/.conda/envs/capita-selecta
```

```
base * /usr/local/Miniconda3-4.7.12.1-Linux-x86_64
```

Environments

- Export

- In order to export current environment:

- `conda env export > capita-selecta.yml`

- Or, to export any other environment:

- `conda env export -n capita-selecta > capita-selecta.yml`

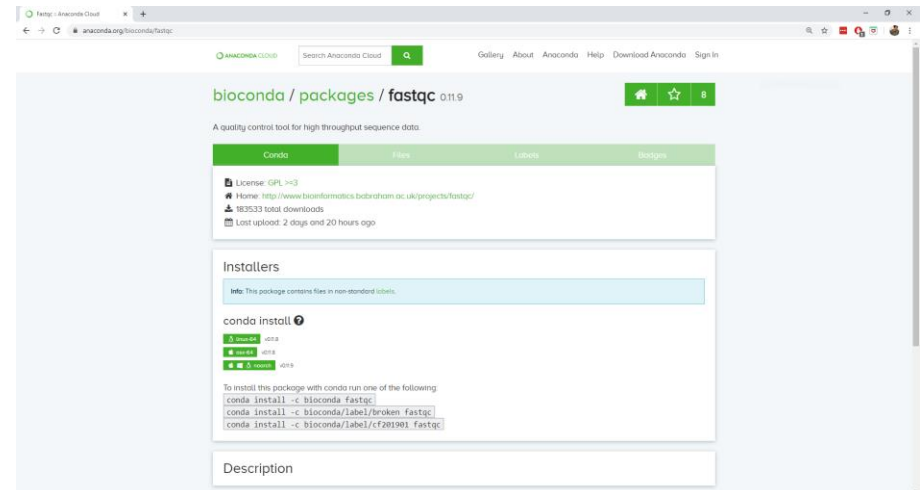
- Import

- New environment from an environment definition:

- `conda env create -n capita-selecta-from-file -f capita-selecta.yml`

Packages

- List all installed packages
 - `conda list`
- Search for a package:
 - If you're not sure if your package is available from conda, just google it!
 - `conda search fastqc`



Packages

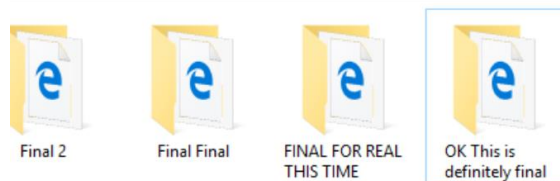
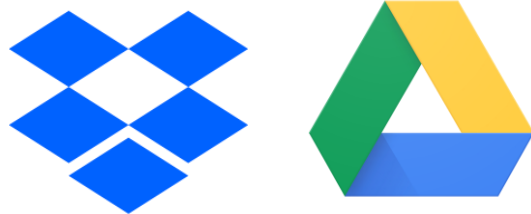
- List all installed packages
 - `conda list`
- Search for a package:
 - If you're not sure if your package is available from conda, just google it!
 - `conda search fastqc`
- Installing
 - If any other package is required, can be installed using conda:
 - `conda install seaborn`
 - `conda install fastqc=x.y.z`
 - Not in channel list:
 - `conda install -c conda-forge fastqc`

References and reading:

- <https://docs.conda.io/en/latest/>
- <https://github.com/ifosch/conda-intro>
- <https://www.freecodecamp.org/news/why-you-need-python-environments-and-how-to-manage-them-with-conda-85f155f4353c/>

Version controlling

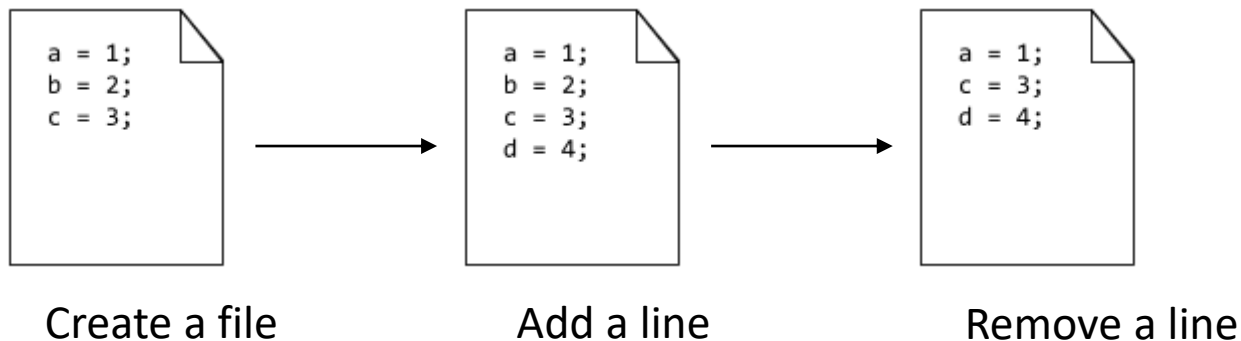
Git & GitHub





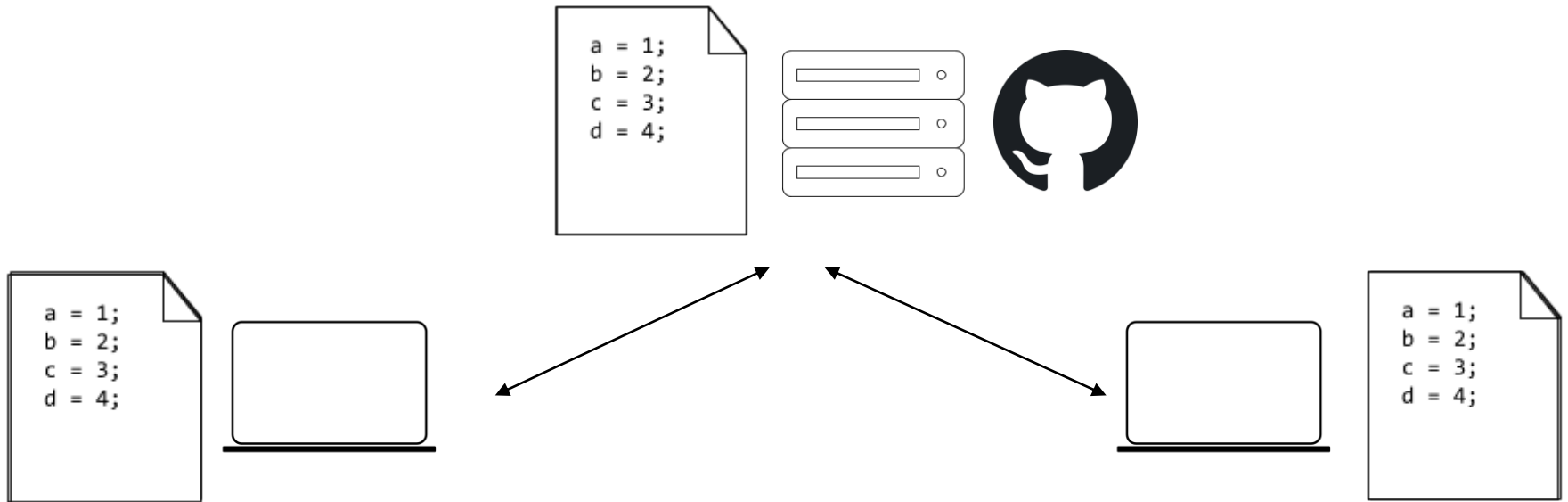
Introduction

- What is Git used for?
 - ▶ Keep track of changes to your code



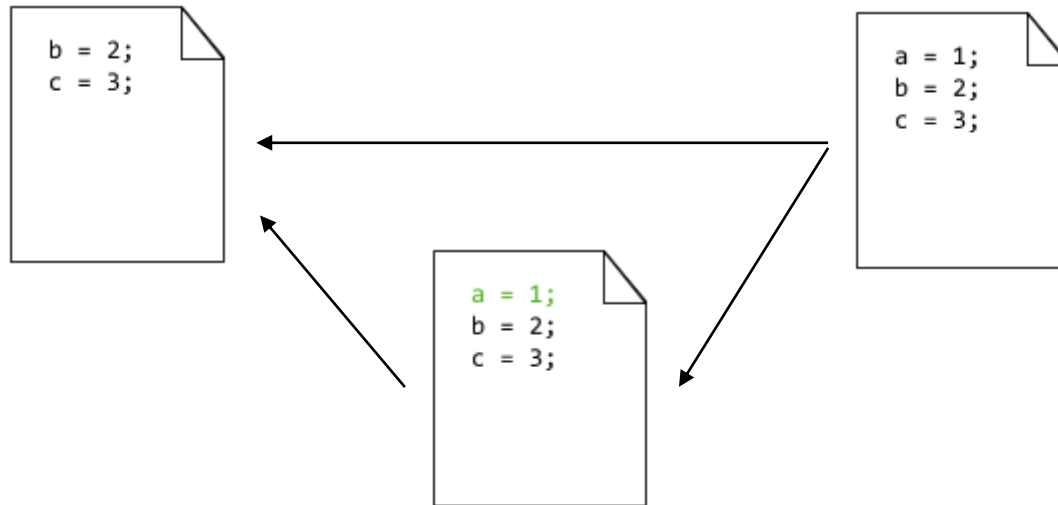
Introduction

- What is Git used for?
 - ▶ Keep track of changes to your code
 - ▶ Synchronize code between different people



Introduction

- What is Git used for?
 - ▶ Keep track of changes to your code
 - ▶ Synchronize code between different people
 - ▶ Testing new code

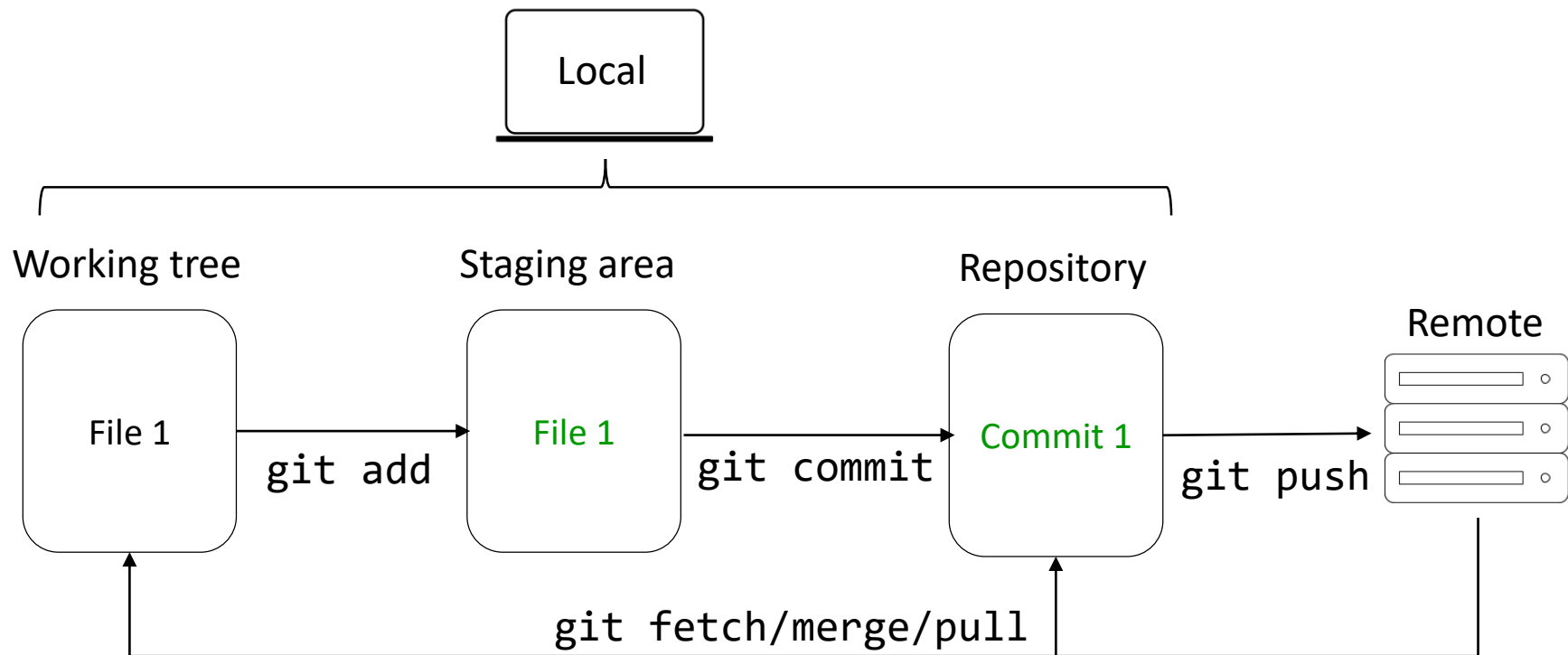


Introduction

- What is Git used for?
 - ▶ Keep track of changes to your code
 - ▶ Synchronize code between different people
 - ▶ Testing new code
 - ▶ Reverting back changes

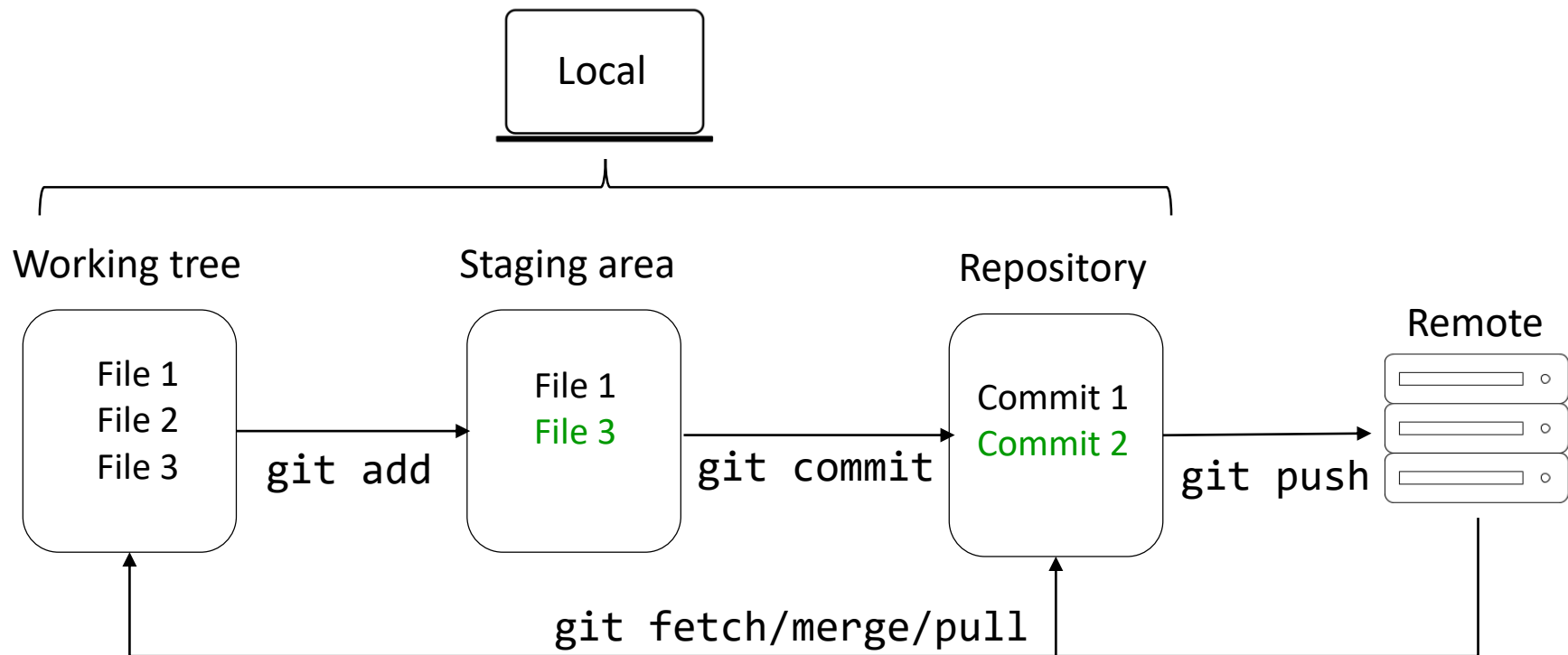
Save changes

- Three conceptual areas of a repository



Save changes

- Three conceptual areas of a repository



.gitignore

- Ignore certain files or directories in repository
- E.g.: data files, results files, temporary files
- * wildcards

```
# Ignore R project information:
.Rproj.user
.Rhistory
.RData
.Ruserdata

# Ignore directories that contain data:
results/
data/

# Ignore temporary files:
*.tmp
```

Workflow pipelines

Nextflow

Bash pipeline

```
#!/bin/bash
blastp -query sample.fasta -outfmt 6 \
    | head -n 10 \
    | cut -f 2 \
    | blastdbcmd -entry - > sequences.txt
```

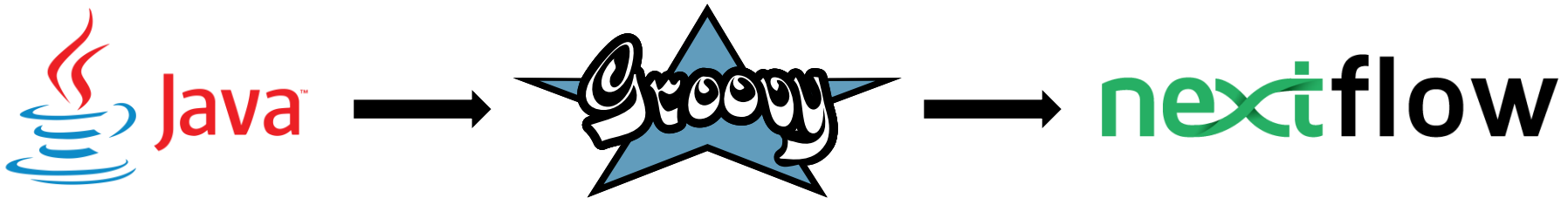
nextflow



Snakemake

Nextflow

- Reactive workflow framework and a programming DSL that eases the writing of data-intensive computational pipelines.
- Scripting language:



Why (not)?

nextflow

- + Parallelization
- + Highly scalable and portable
- + Reproducible (native support of containers)
- + Continuous checkpoints for resuming / expanding pipelines
- Groovy
- Not made for simple pipelines



A community effort to collect a curated set of analysis pipelines built using Nextflow.

[VIEW
PIPELINES](#)

For facilities

Highly optimised pipelines with excellent reporting. Validated releases ensure reproducibility.

For users

Portable, documented and easy to use workflows. Pipelines that you can trust.

For developers

Companion templates and tools help to validate your code and simplify common tasks.

nf-core is now published in [Nature Biotechnology!](#) [Read the full text here.](#)

Processes

- Five definition blocks

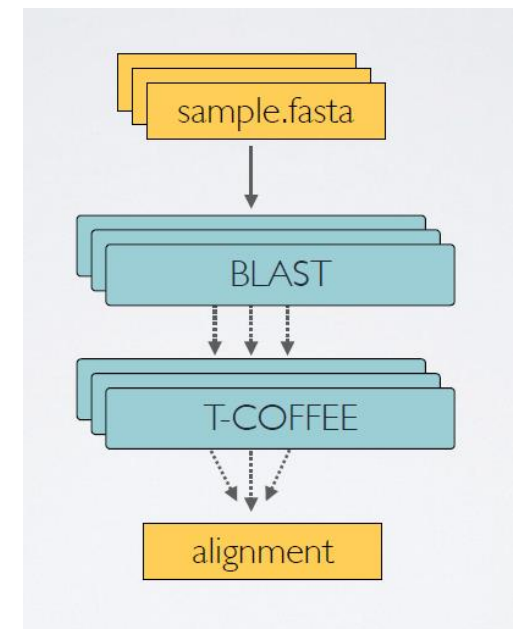
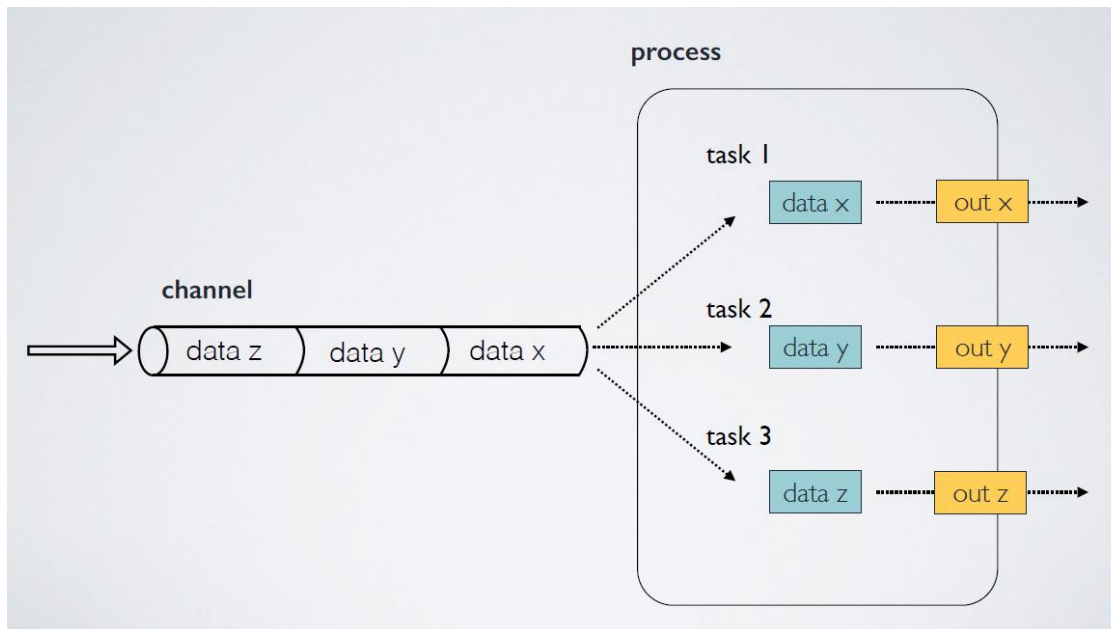
- ▶ Directives
- ▶ Inputs
- ▶ Outputs
- ▶ When clause
- ▶ Process script

```
process < name > {  
  
    [ directives ]  
  
    input:  
        < process inputs >  
  
    output:  
        < process outputs >  
  
    when:  
        < condition >  
  
    [script|shell|exec]:  
        < user script to be executed >  
  
}
```

- In any language (Bash, Python, Perl, Ruby, etc.)
- Executed independently & isolated
- Processes communicate via asynchronous FIFO queues = *channels*

Processes & channels

- Processes are linked via channels: one process will wait for the output of another and then runs reactively when the channel has contents



```
// Script parameters
params.query = "/some/data/sample.fa"
params.db = "/some/path/pdb"

db = file(params.db)
query_ch = Channel.fromPath(params.query)

process blastSearch {
    input:
        file query from query_ch

    output:
        file "top_hits.txt" into top_hits_ch

    """
    blastp -db $db -query $query -outfmt 6 > blast_result
    cat blast_result | head -n 10 | cut -f 2 > top_hits.txt
    """
}

process extractTopHits {
    input:
        file top_hits from top_hits_ch

    output:
        file "sequences.txt" into sequences_ch

    """
    blastdbcmd -db $db -entry_batch $top_hits > sequences.txt
    """
}
```

```
num = Channel.from( 1, 2, 3 )
```

```
process basicExample {
```

```
  input:
```

```
  val x from num
```

```
  "echo process job $x"
```

```
}
```

```
process job 3
```

```
process job 1
```

```
process job 2
```

Execution abstraction

- *Executor* determines *how* the script is run on the target system
- By default: locally
- Alternatively: HPC or cloud platforms



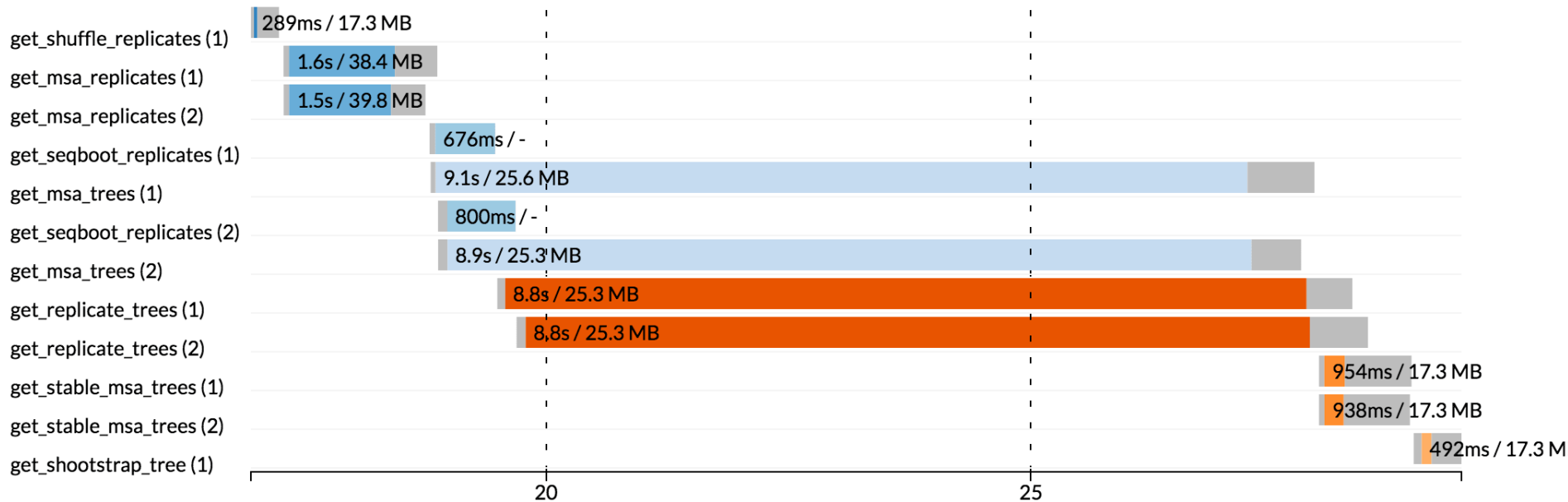
.config

- Local or cluster usage

```
executor {  
    cpus = 4  
}
```

```
process {  
    executor = 'sge'  
    penv = 'smp'  
    clusterOptions = { "-V -S /bin/bash " }  
}
```

Output & report



Output & report

Nextflow Report Summary Resources Tasks

[angry_babbage]

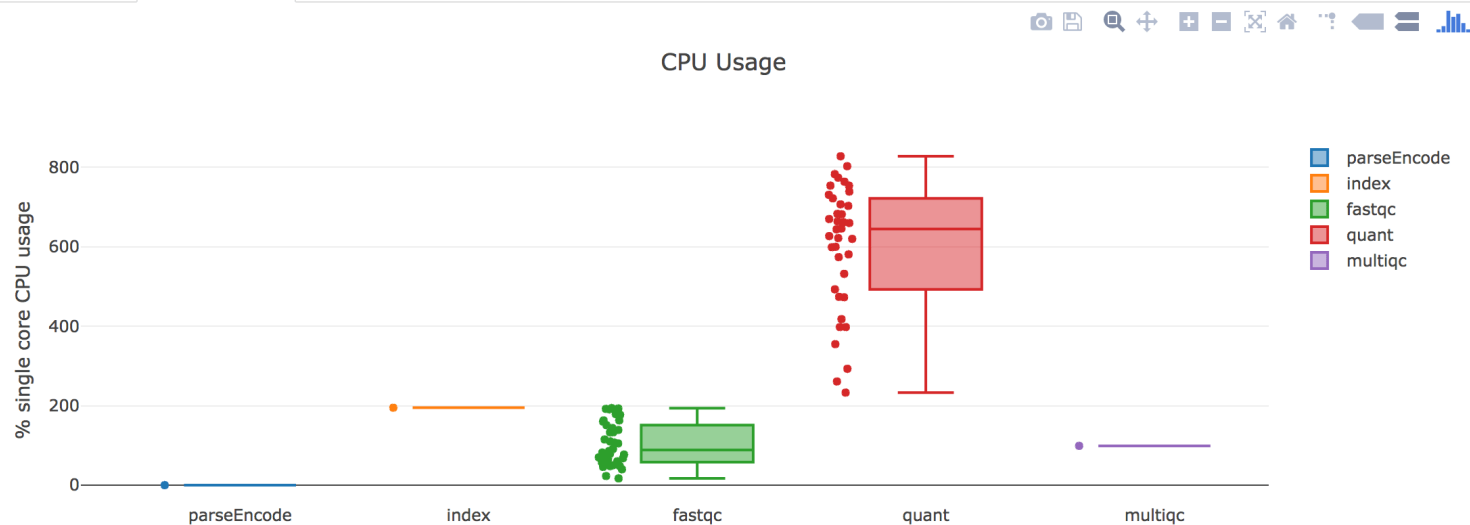
Resource Usage

These plots give an overview of the distribution of resource usage for each process.

CPU Usage

% Allocated

Raw Usage





- Web-based platform
- Built-in integration with many tools and datasets
- Little control over tasks parallelization
- Easy-to-use
- Suited for training/learning and non-experienced users



- Command-line oriented tool
- Can incorporate any tool
- Fine control over parallelization and parameters
- Learning curve
- Suited for production workloads & experienced bioinformaticians

Snakemake

- Command-line oriented tool
- Pull model
- Python based
- Compute DAG ahead
- Support for sub-workflows
- ...

nextflow

- Command-line oriented tool
- Push model
- Java/Groovy based
- Compute DAG at runtime
- Working on sub-workflows
- ...

References & further reading

- <https://www.nextflow.io/docs/latest/getstarted.html>
- <https://github.com/nextflow-io>

training.vib.be

Chan Zuckerberg Initiative

- [Pip](#)
- [Bioconda](#)
- [Docker and R](#)
- [JupyterHub](#)

Containerization

Docker & Singularity

Slides:

- <https://material.bits.vib.be/courses/?https://raw.githubusercontent.com/vibbits/material-lascrip/master/slides-docker-introduction.md#12>