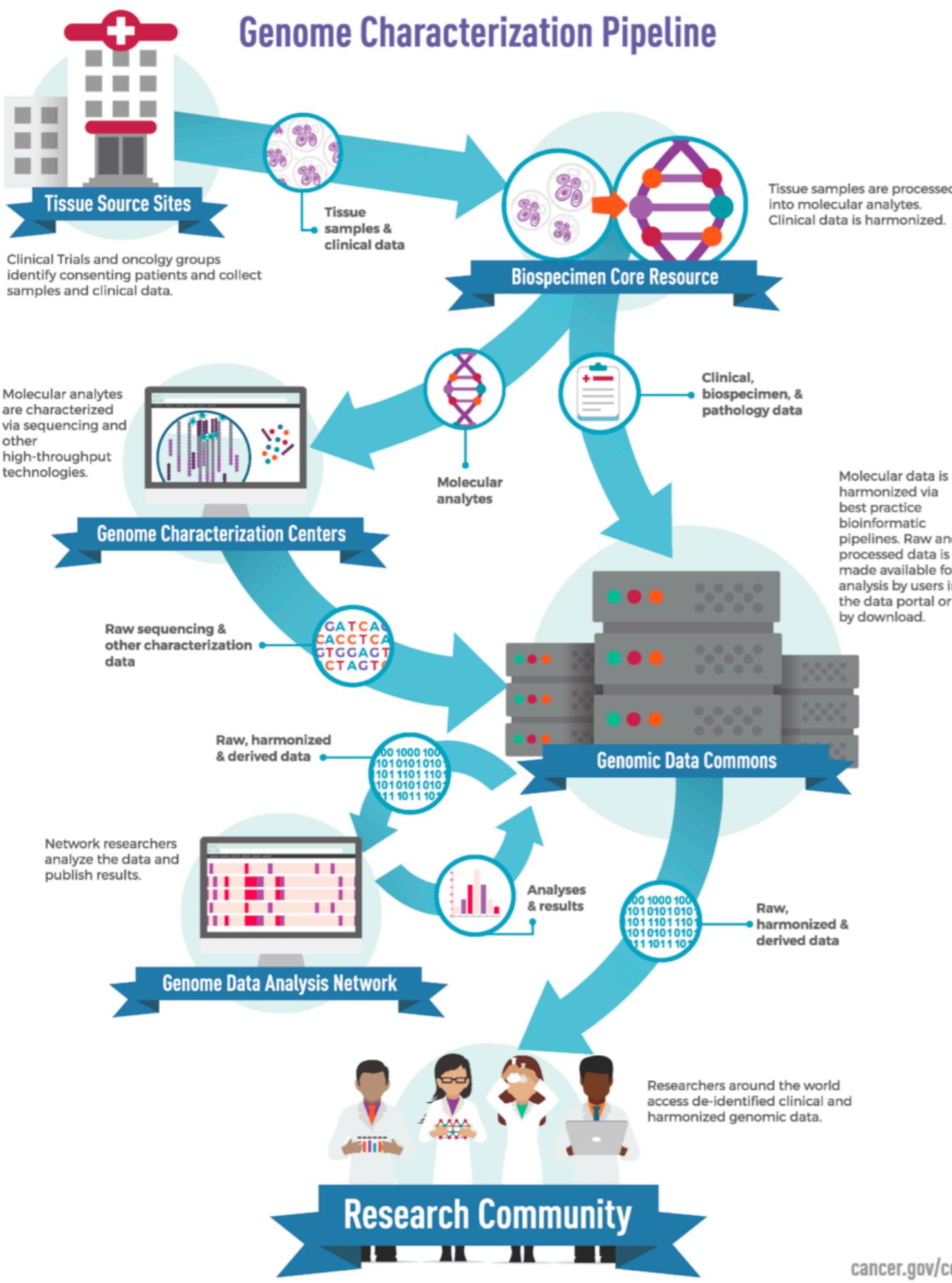


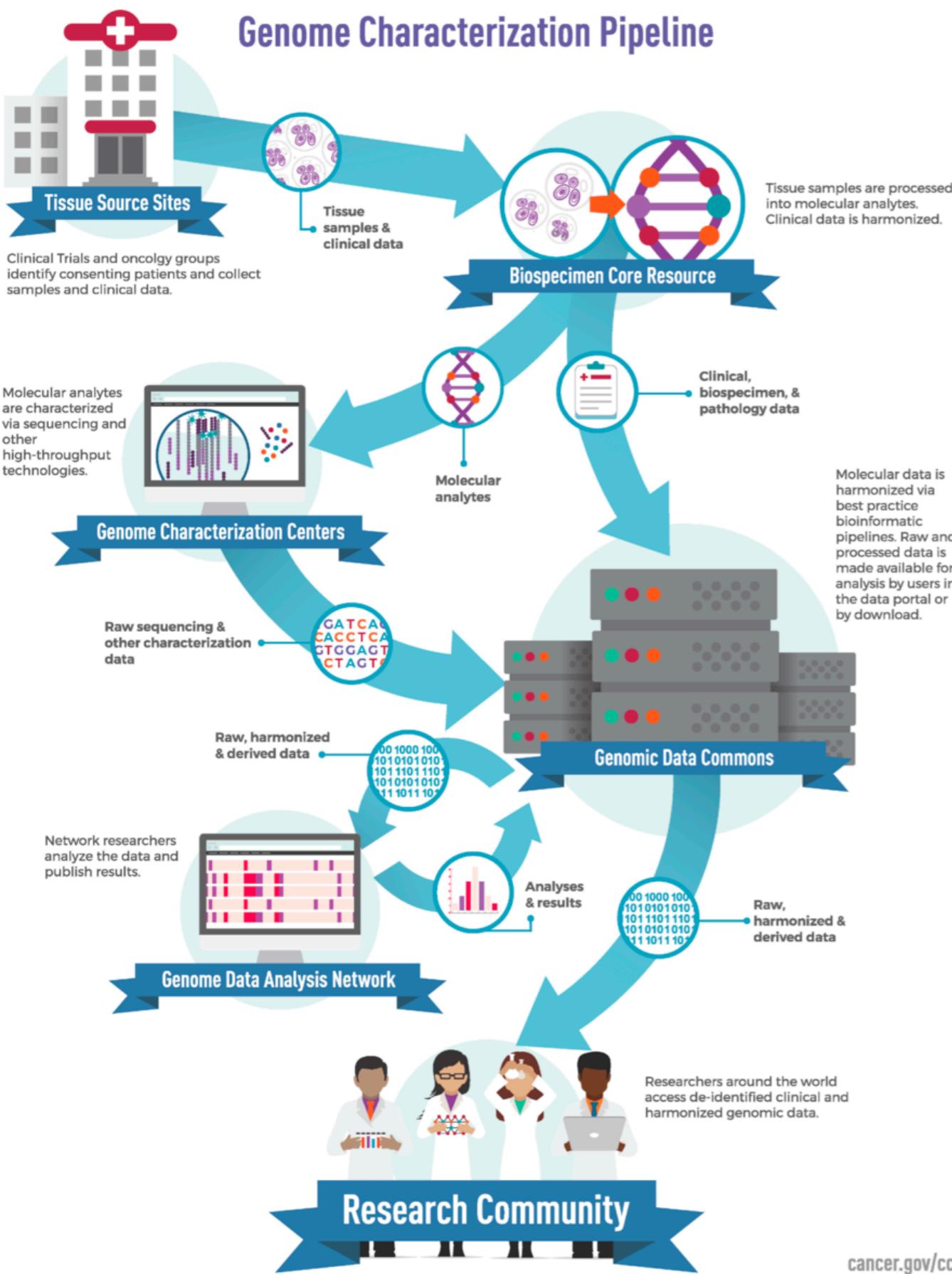
Nextflow: Distribuible escalable, y reproducible computación científica.

Alex Di Genova



Nextflow

Genómica



Nextflow

Genómica

Requisitos de pipelines para proyectos genómicos a gran escala

- **Reproducible:** los resultados genómicos deben ser totalmente reproducibles
- **Escalable:** Facil ejecucion en cluster HPC o cloud.
- **Portátil:** puede ejecutarse en varias infraestructuras (diferente sistema operativo, cloud)
- **Manejar la heterogeneidad:** funciona con dependencias de software en conflicto y varios requisitos de recursos.

Pipelines para datos genómicos

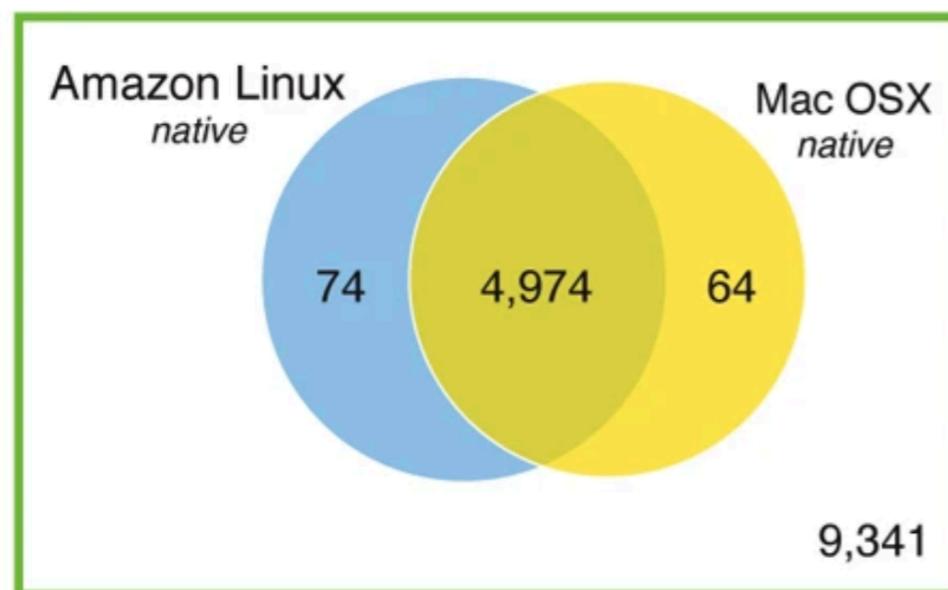
Solución utilizar un lenguaje de dominio específico (DSL)

Un lenguaje específico de dominio (DSL) es un lenguaje de programación especializado en un dominio de aplicación particular (pipelines).

- **Nextflow:** Basado en Groovy (java); Desarrollado en el Centro de regulacion del genoma (Barcelona, España).

C

Transcript quantification and differential expression with Kallisto and Sleuth



nature biotechnology

Explore content ▾ About the journal ▾ Publish with us ▾

[nature](#) > [nature biotechnology](#) > [correspondence](#) > [article](#)

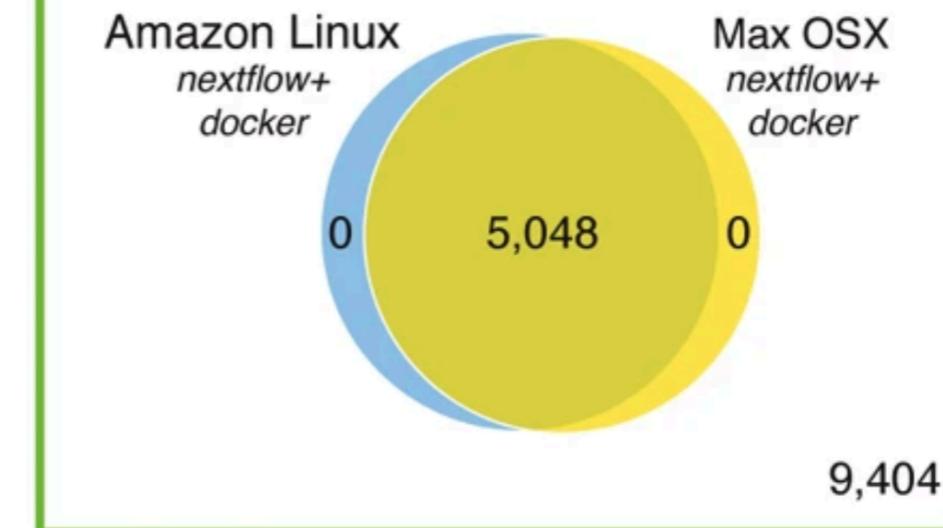
Published: 11 April 2017

Nextflow enables reproducible computational workflows

[Paolo Di Tommaso](#), [Maria Chatzou](#), [Evan W Floden](#), [Pablo Prieto Barja](#), [Emilio Palumbo](#) & [Cedric Notredame](#)✉

Nature Biotechnology 35, 316–319 (2017) | [Cite this article](#)

14k Accesses | 590 Citations | 122 Altmetric | [Metrics](#)



Nextflow

Diseño y características

1. Write code in any language

```
trim_galore --paired --fastqc --gzip \  
--basename sample1 -j 2 pair_1.fq pair_2.fq
```

2. Define succession of processes using dataflow programming

Trimming | Mapping | Quantification

3. Define software dependencies

Conda, Docker, Singularity

3. Define configuration

Executor (local, HPC scheduler),
error management, etc

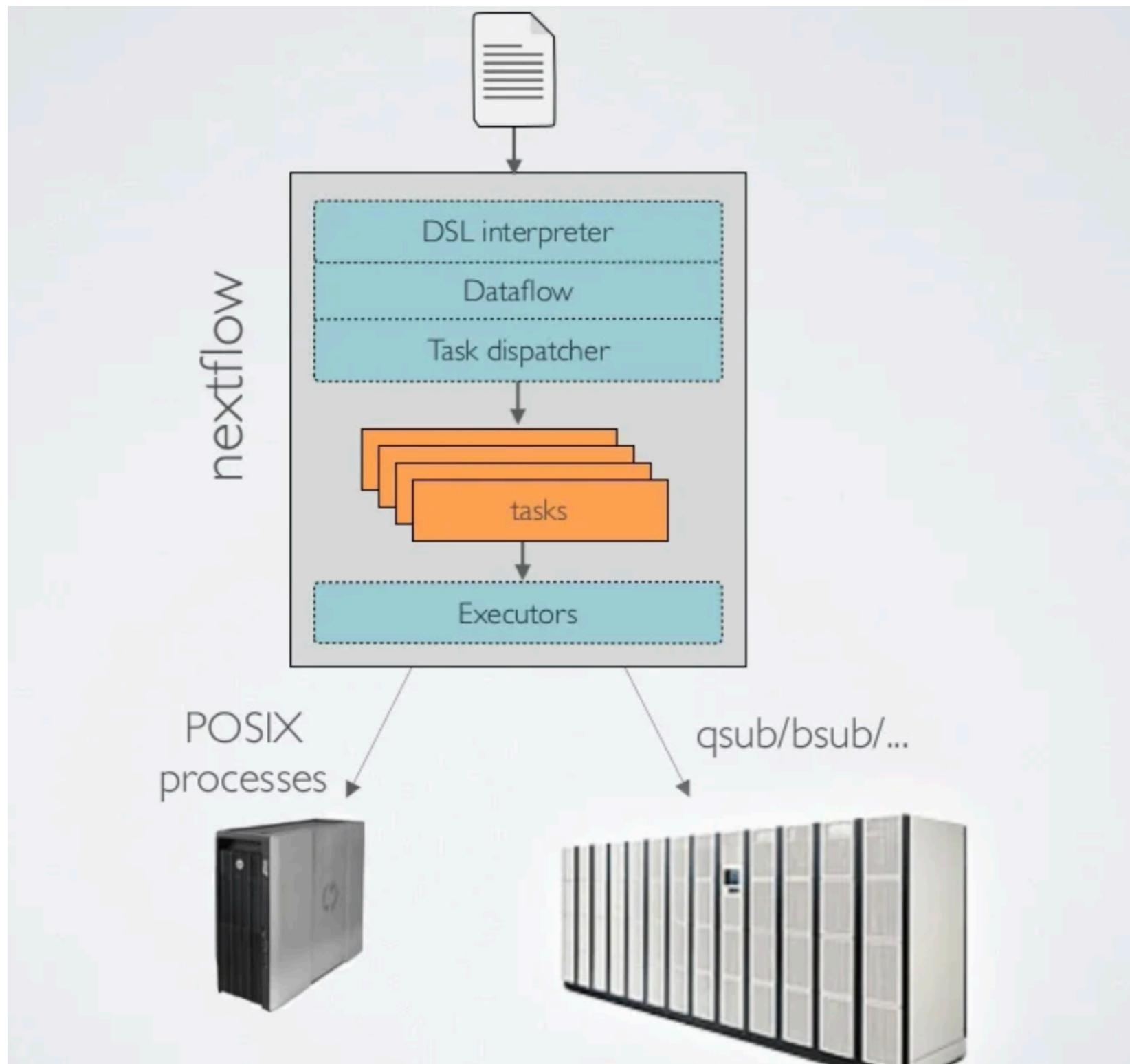
4. Run

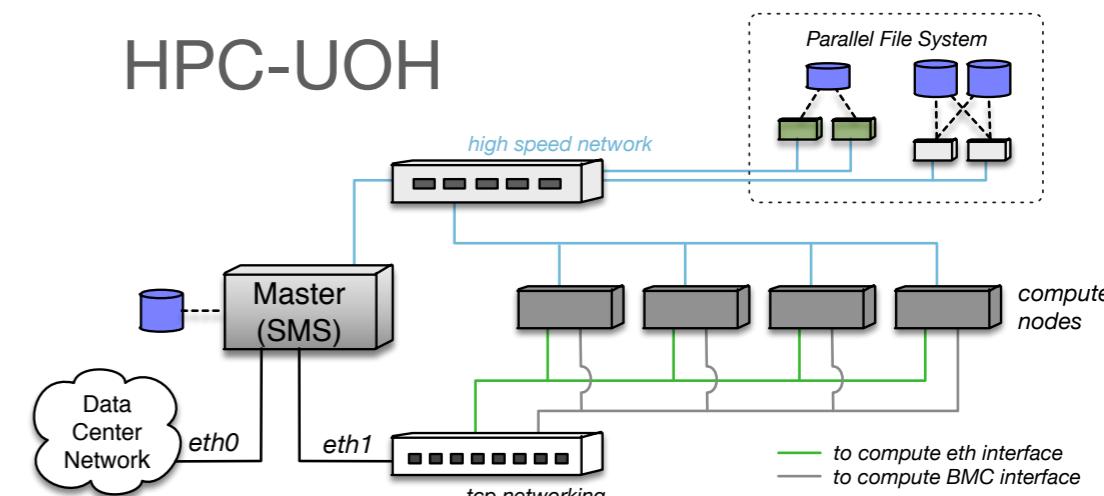
Lenguaje de pipeline con paralelización implícita y programación de tareas:

- El mismo conjunto de tareas se aplica a todos los archivos de entrada
- Las tareas se ejecutan automáticamente en el orden correcto dadas sus entradas y salidas
- Maneja automáticamente el envío de tareas a distintos sistemas administradores de recursos.

Nextflow

Independiente de la plataforma



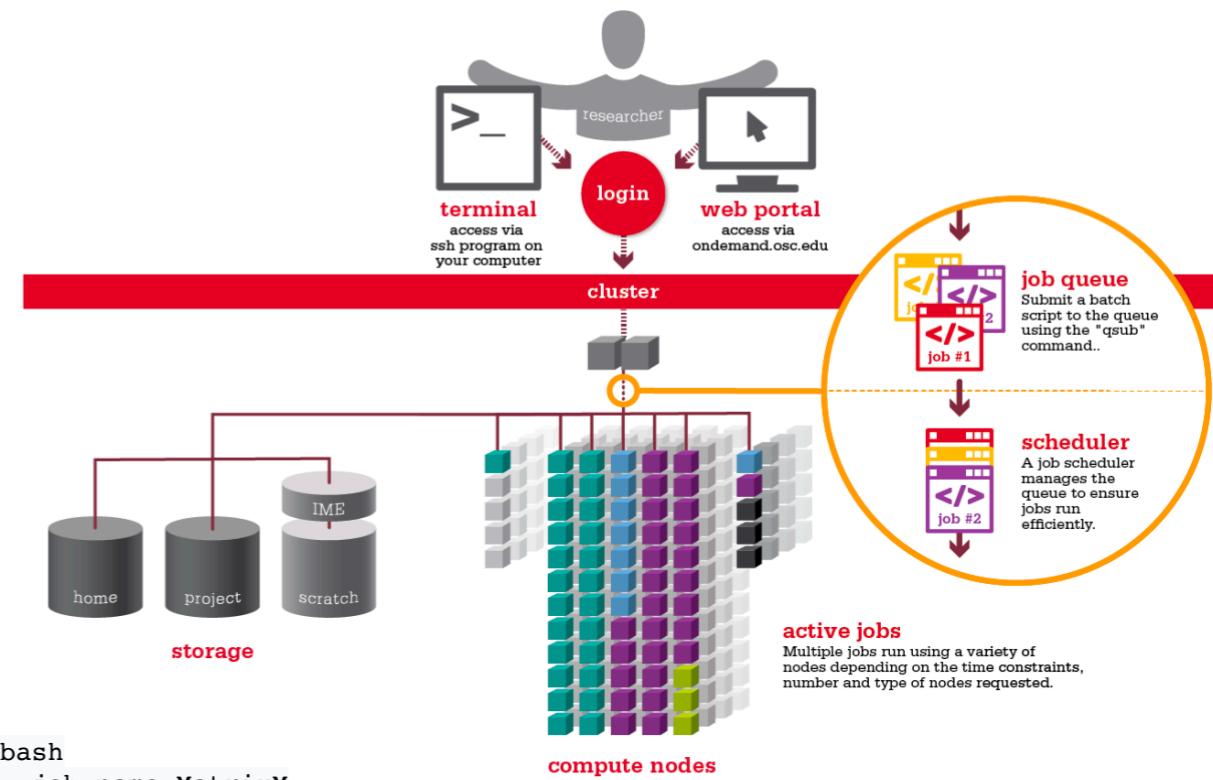
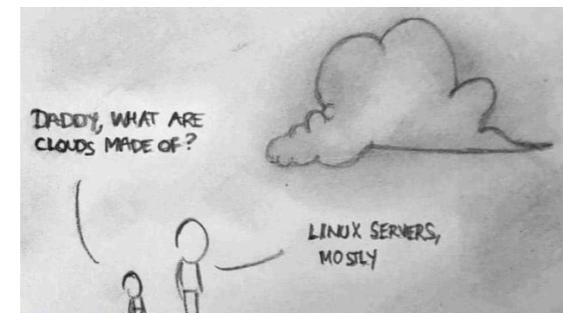


Nextflow

Local, cluster, cloud

- Manejadores de recursos:

- **Local** (Ejecuta los procesos en la computadora local)
- **SLURM** (<https://slurm.schedmd.com/documentation.html>)
- **LSF**(https://en.wikipedia.org/wiki/IBM_Spectrum_LSF)
- **Moab** (https://en.wikipedia.org/wiki/Moab_Cluster_Suite)
- **OAR** (<https://oar.imag.fr/>)
- **PBS/Torque** (http://en.wikipedia.org/wiki/Portable_Batch_System)
- **SGE** (<http://gridscheduler.sourceforge.net/>)
- **Clouds:**
 - **AWS Batch** (<https://aws.amazon.com/batch/>)
 - **Azure Batch** (<https://azure.microsoft.com/en-us/services/batch/>)
 - **Google Cloud Batch** (<https://cloud.google.com/batch>)
 - **Google Life Sciences** (<https://cloud.google.com/life-sciences>)



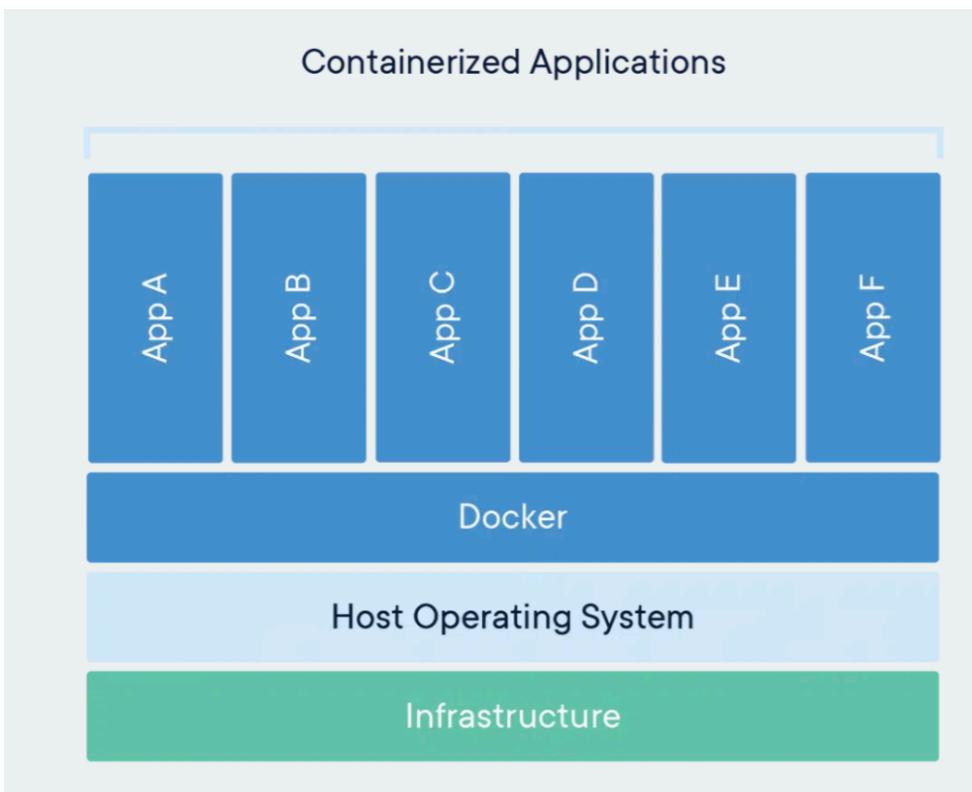
```
#!/bin/bash
#SBATCH --job-name=MatrixM
#SBATCH --output=LAMMPS_%j.out
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=alex.digenova@uoh.cl
#SBATCH --nodes=4           # Number of nodes
#SBATCH --ntasks=8          # Number of MPI ranks
#SBATCH --ntasks-per-node=2 # Number of MPI ranks per node
#SBATCH --ntasks-per-socket=1# Number of tasks per processor socket on the node
#SBATCH --cpus-per-task=8   # Number of OpenMP threads for each MPI process/rank
#SBATCH --mem-per-cpu=2000mb # Per processor memory request
#SBATCH --time=4-00:00:00    # Walltime in hh:mm:ss or d-hh:mm:ss
```

```
## bash code
date;hostname;pwd
module load intel/2018 openmpi/3.1.0
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
srun --mpi=pmi_v1 /path/to/app/matrix_multiplication
date
```

Nexflow

Contenedores (Docker/singularity)

- Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias, por lo que la aplicación se ejecuta de forma rápida y fiable de un entorno informático a otro.
- Una imagen de contenedor de Docker es un paquete de software ligero, independiente y ejecutable que incluye todo lo necesario para ejecutar una aplicación: código, binarios, herramientas del sistema, bibliotecas del sistema y configuración.



```
#####
# BASE IMAGE #####
FROM mambaorg/micromamba:0.15.3
##site to test docker configuration files
#####
# METADATA #####
LABEL base_image="mambaorg/micromamba"
LABEL version="0.15.3"
LABEL software="k-count-nf"
LABEL software.version="1.1"
LABEL about.summary="Container image containing all requirements for k-count-nf"
LABEL about.home="https://github.com/digenoma-lab/k-count-nf"
LABEL about.documentation="https://github.com/digenoma-lab/k-count-nf/README.md"
LABEL about.license_file="https://github.com/digenoma-lab/k-count-nf/LICENSE.txt"
LABEL about.license="MIT"

#####
# MAINTAINER #####
MAINTAINER Alex Di Genova <digenova@gmail.com>
#####
# INSTALLATION #####
USER root
#the next command install the ps command needed by nexflow to collect run metrics
RUN apt-get update && apt-get install -y procps
USER micromamba
COPY --chown=micromamba:micromamba environment.yml /tmp/environment.yml
RUN micromamba create -y -n k-count -f /tmp/environment.yml && \
    micromamba clean --all --yes
ENV PATH /opt/conda/envs/k-count/bin:$PATH
```

name: k-count
channels:
- conda-forge
- bioconda
- defaults
dependencies:
- kmc=3.1.1rc1
- genomescope2=2.0

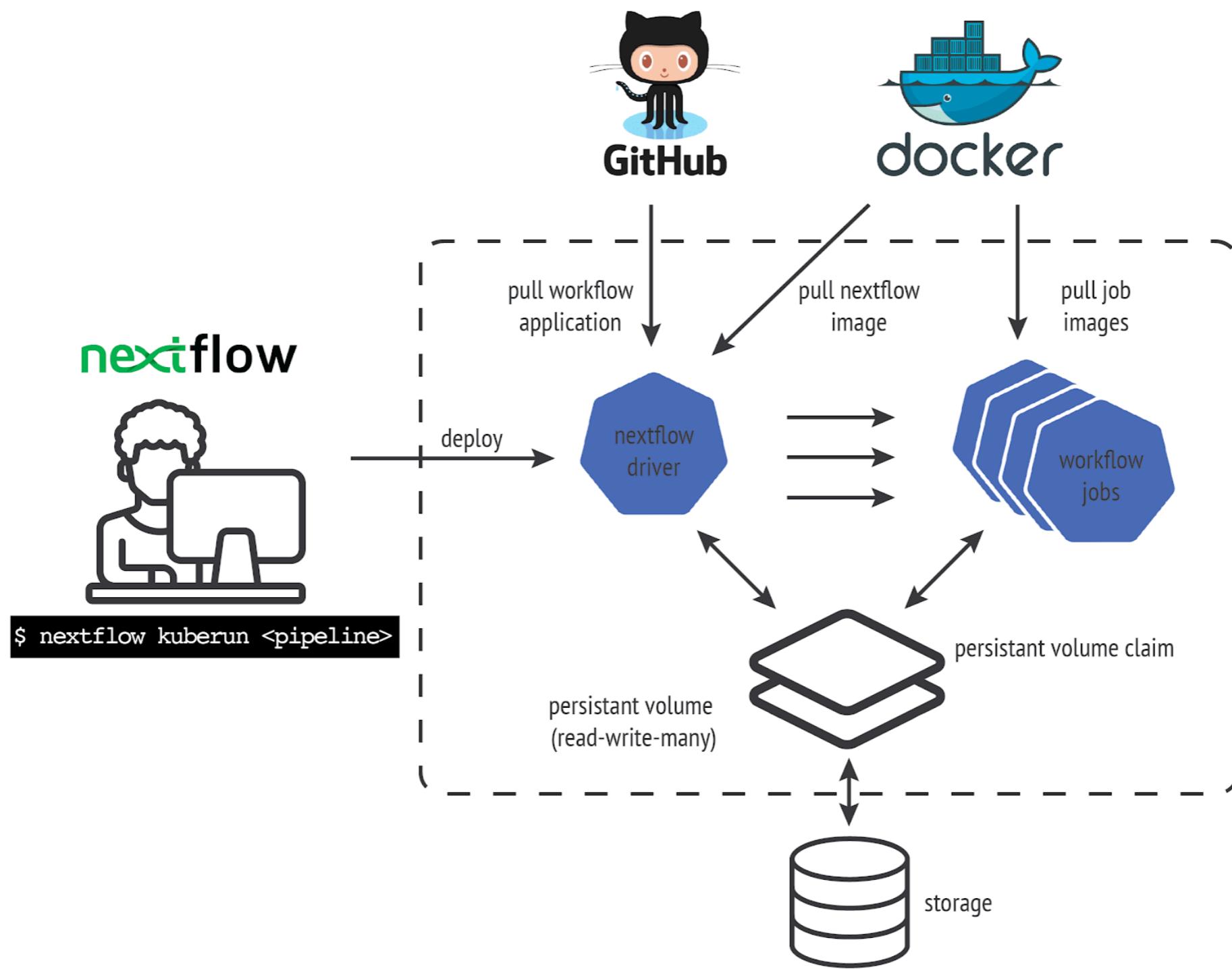
<https://hub.docker.com/>
<https://biocontainers.pro/>



```
singularity pull biocontainers/fastqc:v0.11.9_cv8
singularity shell fastqc_v0.11.9_cv8.sif
```

Nextflow

Tecnología



GitHub

Nextflow pipeline

The screenshot shows a GitHub repository page for 'digenoma-lab/longreadstats'. The repository is public and contains 1 branch and 1 tag. The main commit is by 'c-valenzuelac' titled 'Update base.config' with a timestamp of '1b9077b 15 days ago' and 8 commits. The repository has no description, website, or topics provided. It has 0 stars, 1 watching, and 0 forks. There is one release, 'longreadstats v1.0dev', which is the latest version and was published 15 days ago. No packages have been published.

digenoma-lab / longreadstats Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main · 1 branch · 1 tag

Go to file Add file Code

About

No description, website, or topics provided.

Readme MIT license Cite this repository

0 stars 1 watching 0 forks

Releases 1

longreadstats v1.0dev Latest
15 days ago

Packages

No packages published Publish your first package

File	Commit Message	Time Ago
base.config	c-valenzuelac Update base.config	15 days ago
assets	adding nanoplot pipeline	17 days ago
bin	adding nanoplot pipeline	17 days ago
conf	Update base.config	15 days ago
docs	adding nanoplot pipeline	17 days ago
lib	adding nanoplot pipeline	17 days ago
modules	adding nanoplot pipeline	17 days ago
reads	adding nanoplot pipeline	17 days ago
subworkflows/local	adding nanoplot pipeline	17 days ago
workflows	adding nanoplot pipeline	17 days ago
CHANGELOG.md	adding nanoplot pipeline	17 days ago
CITATION.cff	adding nanoplot pipeline	17 days ago
CITATIONS.md	adding nanoplot pipeline	17 days ago
LICENSE	adding nanoplot pipeline	17 days ago

<https://github.com/digenoma-lab/longreadstats>

Nextflow

Corriendo un pipeline

```
nextflow run digenoma-lab/longreadstats -profile <singularity> -input samplesheet.csv
```

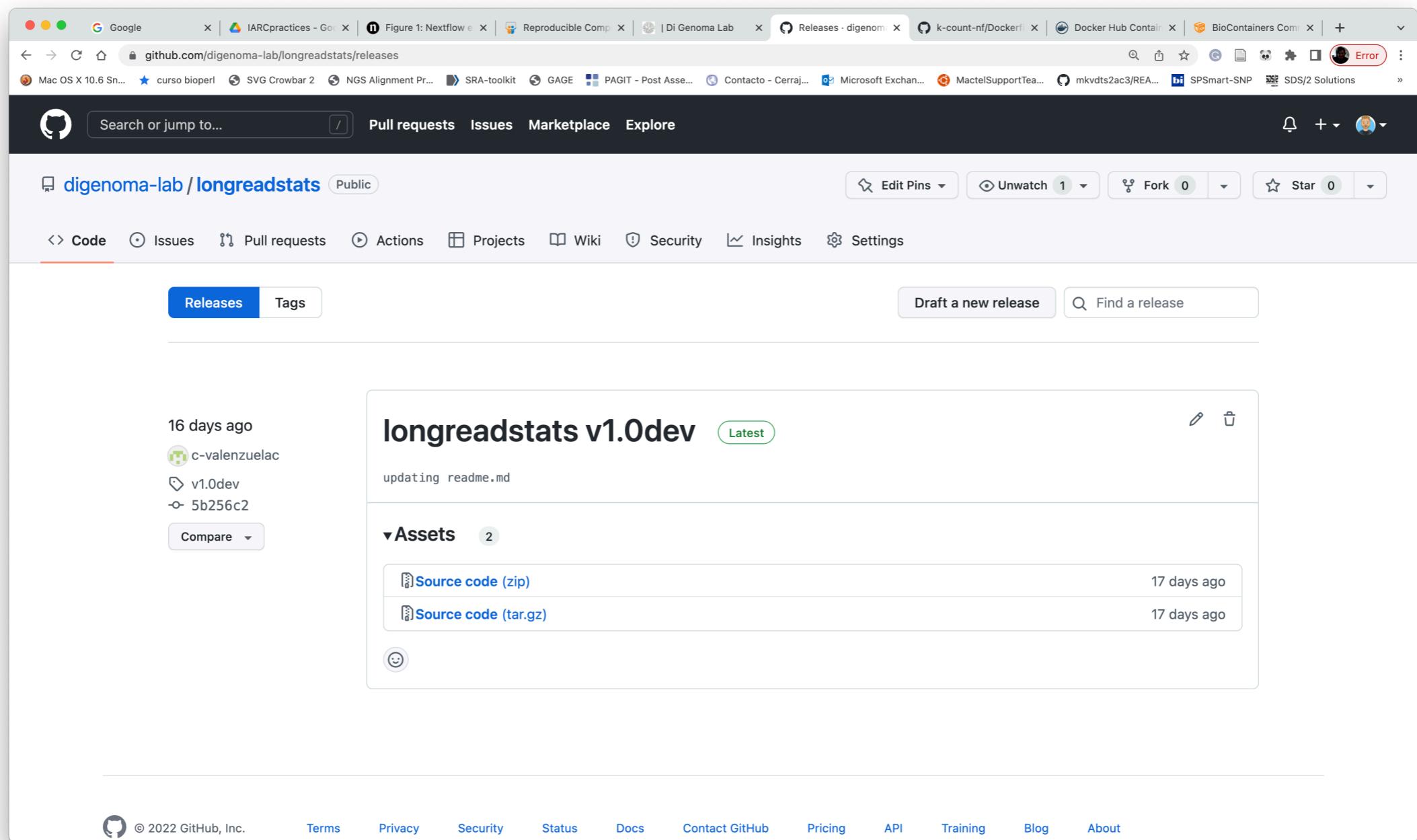
Nombre del pipeline: ruta a un script de *nextflow script*, una ruta a un directorio con un archivo de configuración de *nextflow* que indica un script *nextflow*, o un repositorio en *github* (por ejemplo, <https://github.com/digenoma-lab/longreadstats>)

Opciones de nextflow (“-”): por ejemplo, perfiles definiendo el uso de contenedores o el software local.

sample,fastq_1,fastq_2
A01,/home/adigenova/digenoma-lab-longreadstats/reads/tmp1.fastq.gz,
A02,/home/adigenova/digenoma-lab-longreadstats/reads/tmp2.fastq.gz,
A03,/home/adigenova/digenoma-lab-longreadstats/reads/tmp3.fastq.gz,

Github releases

```
nextflow run digenoma-lab/longreadstats -r v1.0dev --profile <singularity> --input samplesheet.csv
```



Nextflow

Convirtiendo la pesadilla de las dependencias en...

Para correr un pipeline usualmente debemos instalar muchos programas:

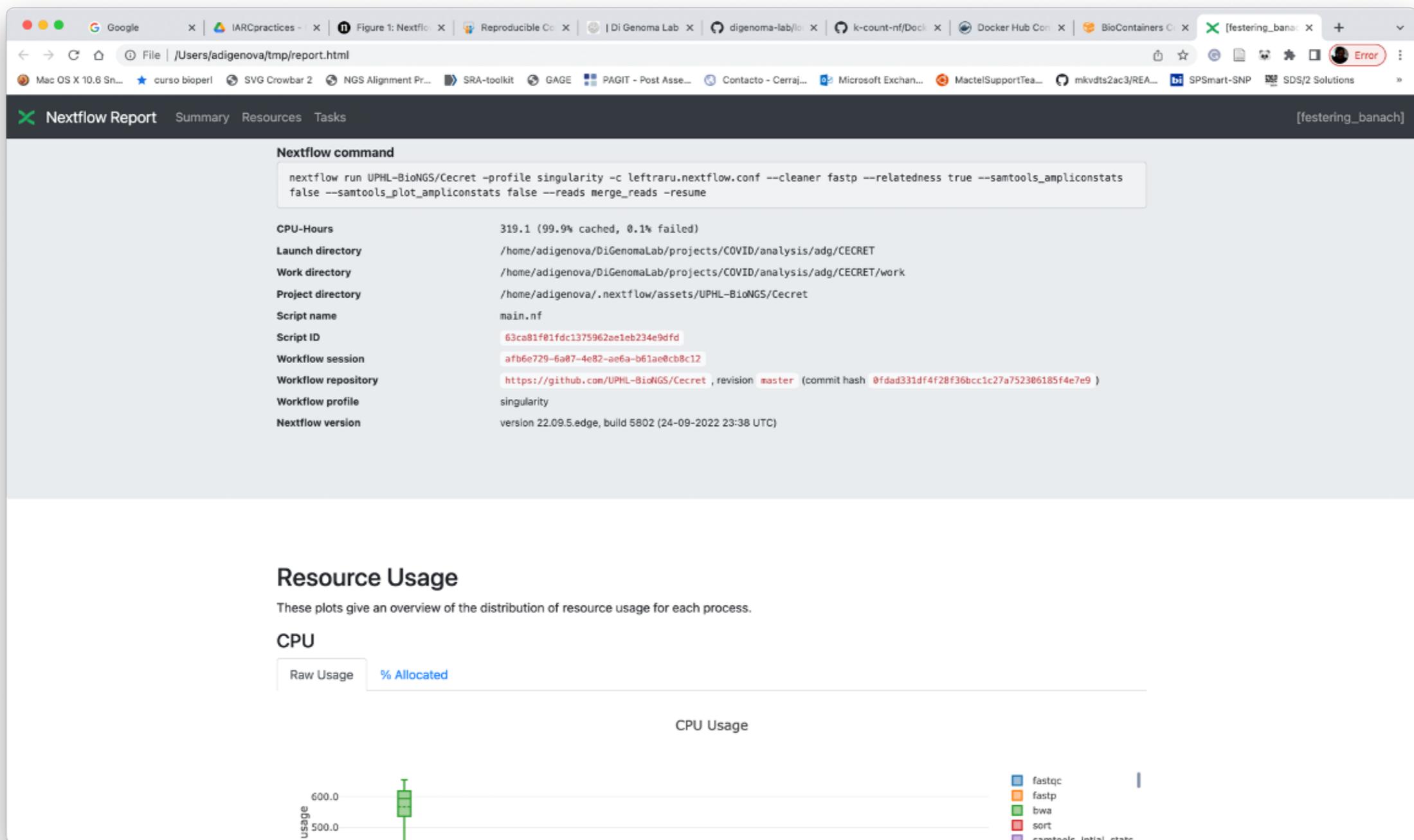
gatk4, fastqc, trim-galore, samtools, star, rseqc, ldc, sambamba, samblaster, multiqc, htseq, R with ggplot2, gsalib, reshape package....

..... Un gran sueño.

```
nextflow run digenoma-lab/longreadstats -r v1.0dev --profile <singularity> --input samplesheet.csv
```

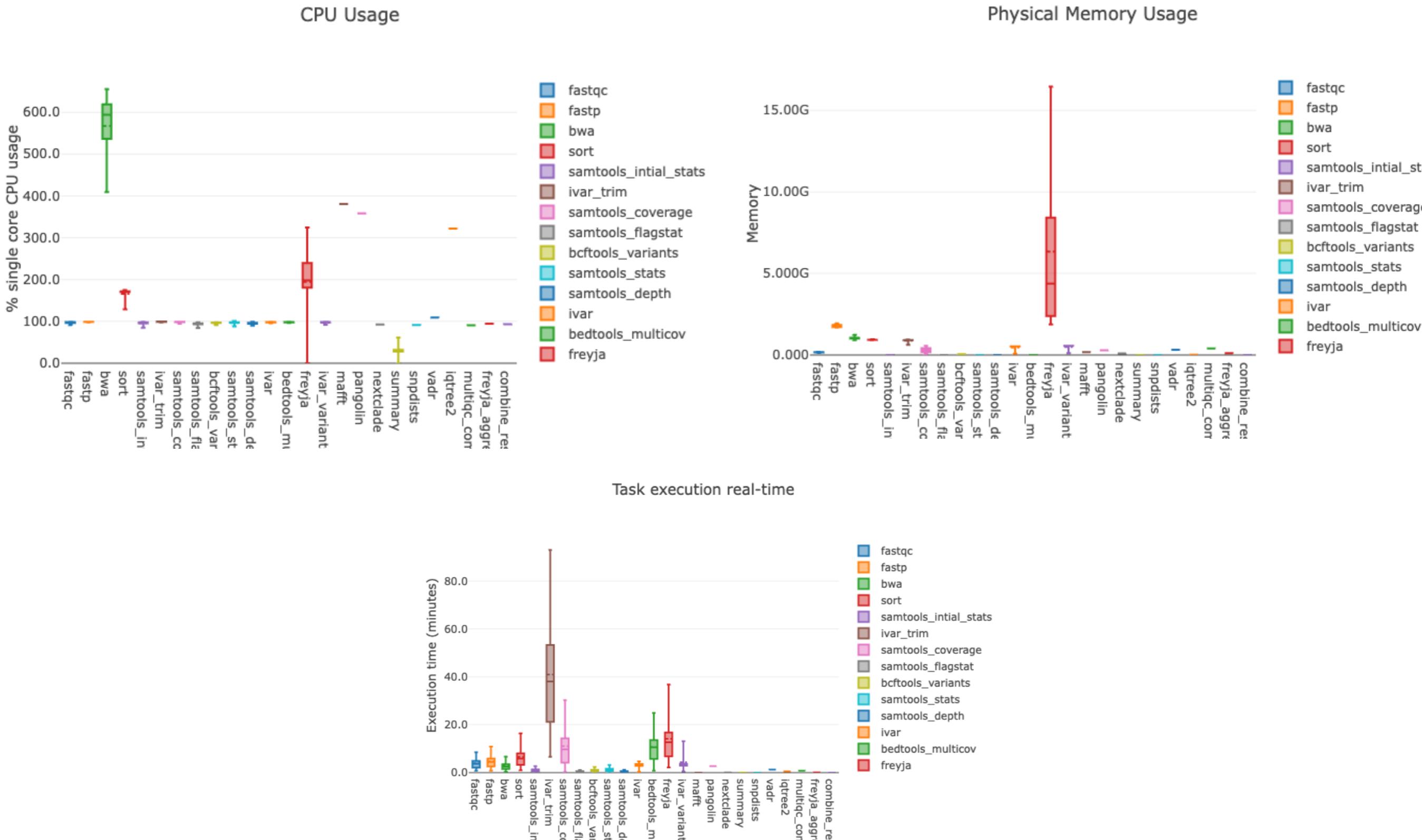
Nextflow

Reportes



Nextflow

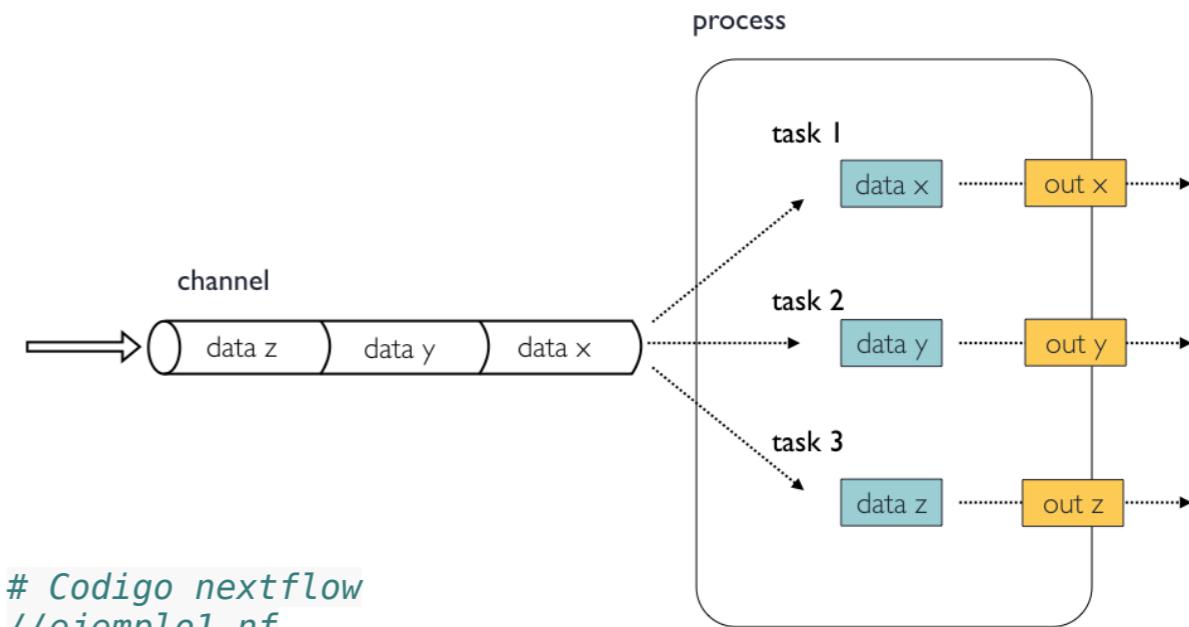
Reportes



Nextflow

Bases de programación

- Un script de Nextflow se crea uniendo diferentes **procesos**. Cada proceso se puede escribir en cualquier lenguaje de comandos que pueda ejecutar la plataforma Linux (Bash, Perl, Ruby, Python, etc.).
- Los **procesos** se ejecutan de forma independiente y están aislados entre sí, es decir, no comparten un estado común (de escritura). La única forma de comunicación es a través de colas FIFO asíncronas, llamadas **canales** en Nextflow.
- Cualquier proceso puede definir uno o más **canales** como entrada y salida. La interacción entre estos procesos y, en última instancia, el propio flujo de ejecución del pipeline, se define implícitamente mediante estas declaraciones de entrada y salida.



```
# Código nextflow
//ejemplo1.nf
nextflow.enable.dsl=2

process NUM_LINEAS {
    input:
        path read

    script:
        """
        printf '${read}'
        gunzip -c ${read} | wc -l
        """
}

reads_ch = Channel.fromPath( 'reads/ref*.fq.gz' )

workflow {
    NUMLINES(reads_ch)
}

# Bash
nextflow run ejemplo1.nf -process.echo

# output
[cd/77af6d] process > NUMLINES (1) [100%] 6 of 6 ✓
ref1_1.fq.gz 58708
ref3_2.fq.gz 52592
ref2_2.fq.gz 81720
ref2_1.fq.gz 81720
ref3_1.fq.gz 52592
ref1_2.fq.gz 58708
```

Nextflow

Primer pipeline

Procesos

```
//pipeline_01.nf
nextflow.enable.dsl=2

process INDEX {
    input:
        path transcriptome
    output:
        path 'index'
    script:
        """
            salmon index -t $transcriptome -i index
        """
}

process QUANT {
    input:
        each path(index)
        tuple(val(pair_id), path(reads))
    output:
        path pair_id
    script:
        """
            salmon quant --threads $task.cpus --libType=U \
                -I $index \
                -1 ${reads[0]} -2 ${reads[1]} -o $pair_id
        """
}

//Definición del pipeline
workflow {
    //canales de entrada
    transcriptome_ch =
        channel.fromPath('transcriptome/*.fa.gz',checkIfExists: true)
    read_pairs_ch =
        channel.fromFilePairs('reads/*_{1,2}.fq.gz',checkIfExists: true)
    //el proceso INDEX toma el canal transcriptome_ch
    index_ch = INDEX(transcriptome_ch)
    //el proceso QUANT toma dos canales como argumentos: Indice y las lecturas
    QUANT( index_ch, read_pairs_ch ).view()
}
```

Nextflow

Groovy

- Apache Groovy es un lenguaje de programación orientado a objetos compatible con la sintaxis de Java para la plataforma Java.

The screenshot shows a web browser displaying the Apache Groovy Syntax documentation at groovy-lang.org/syntax.html. The page has a dark header with the Apache Groovy logo and navigation links for Learn, Documentation, Download, Support, Contribute, Ecosystem, Socialize, and a search icon. A red banner in the top right corner says "Fork me on GitHub". The main content area is titled "Syntax" and contains a "Table of contents" sidebar on the left with sections like Comments, Keywords, Identifiers, Strings, and Interpolation. The main content area starts with a section on "Comments", specifically "Single-line comment", showing code examples like `// a standalone single line comment` and `println "hello" // a comment till the end of the line`. It then moves to "Multiline comment", showing code like `/* a standalone multiline comment spanning two lines */` and `println "hello" /* a multiline comment starting at the end of a statement */`. Finally, it covers "Groovydoc comment", explaining how they are multiline comments starting with `/**` and ending with `*/`, and listing associated types like classes, interfaces, enums, and annotations.

- <http://groovy-lang.org/documentation.html>

- <https://www.nextflow.io/docs/latest/script.html>

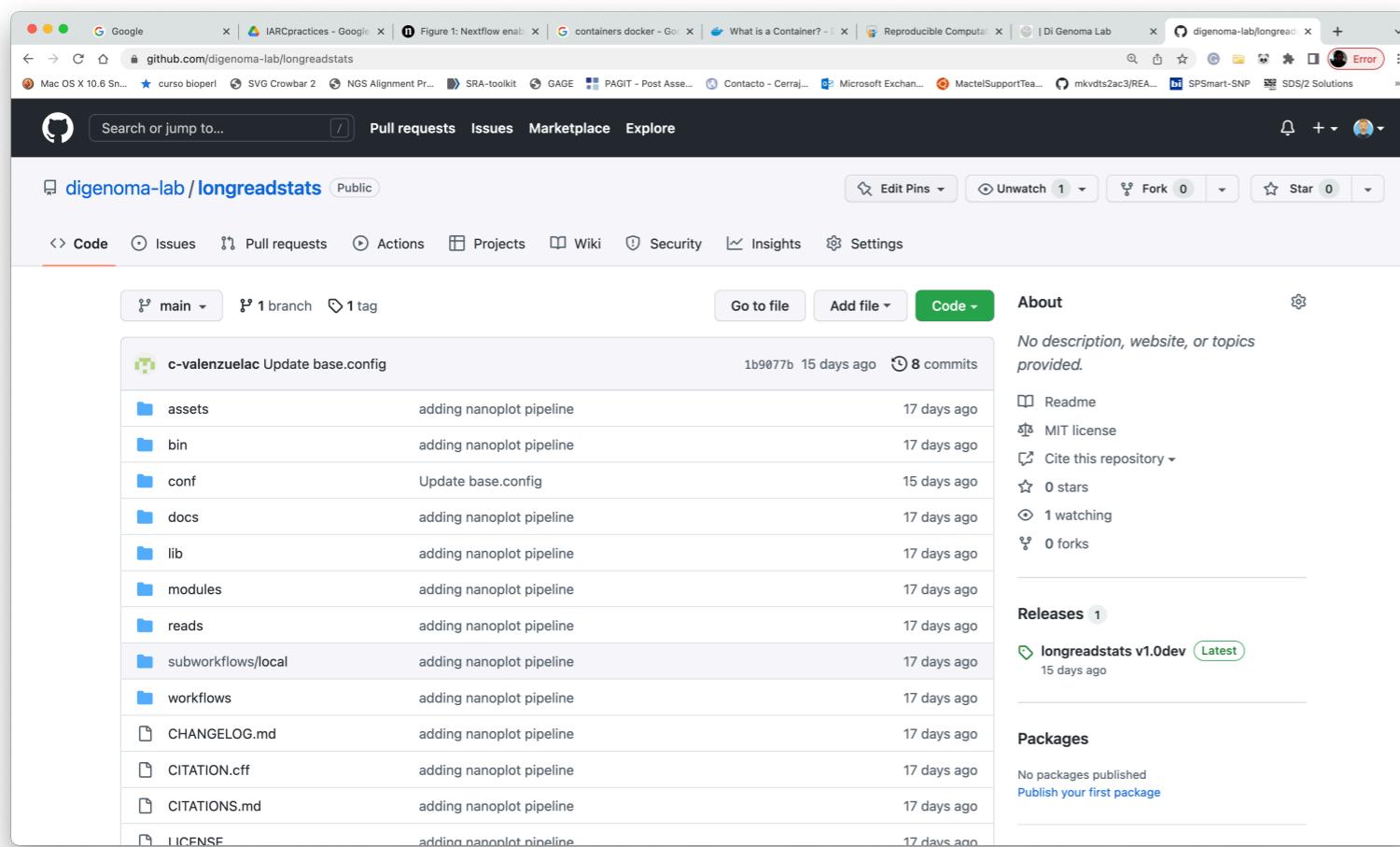
Nextflow

Creando pipelines con el estandar nf-core

```
nf-core create -n long-read-stats -d "pipeline for computing long-read statistics" -a "Alex Di Genova"
```

```
nf-core modules list remote
```

```
nf-core modules install nanoplot
```



```
[adigenova@leftraru1 ~]$ nf-core modules list remote
```



```
nf-core/tools version 2.5.1 – https://nf-co.re  
There is a new version of nf-core/tools available! (2.6)
```

INFO Modules available from nf-core/modules (master):

Module Name
nf-core/abacas
nf-core/abricate/run
nf-core/abricate/summary
nf-core/adapterremoval
nf-core/adapterremovalfixprefix
nf-core/agrvate
nf-core/allelecounter
nf-core/ampir
nf-core/amplify/predict
nf-core/amps
nf-core/amrfinderplus/run
nf-core/amrfinderplus/update
nf-core/angsd/docounts
nf-core/antismash/antismashlite
nf-core/antismash/antismashliteownloadadbases
nf-core/aria2
nf-core/ariba/getref
nf-core/ariba/run

Preguntas?