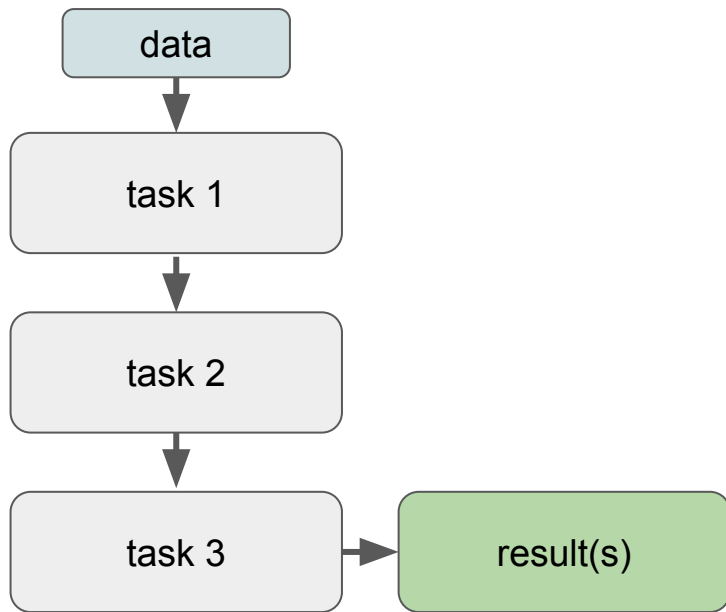


Pipelines with **nextflow**

Maxime Borry, Quentin Letourneur

Why do you need a pipeline language ?

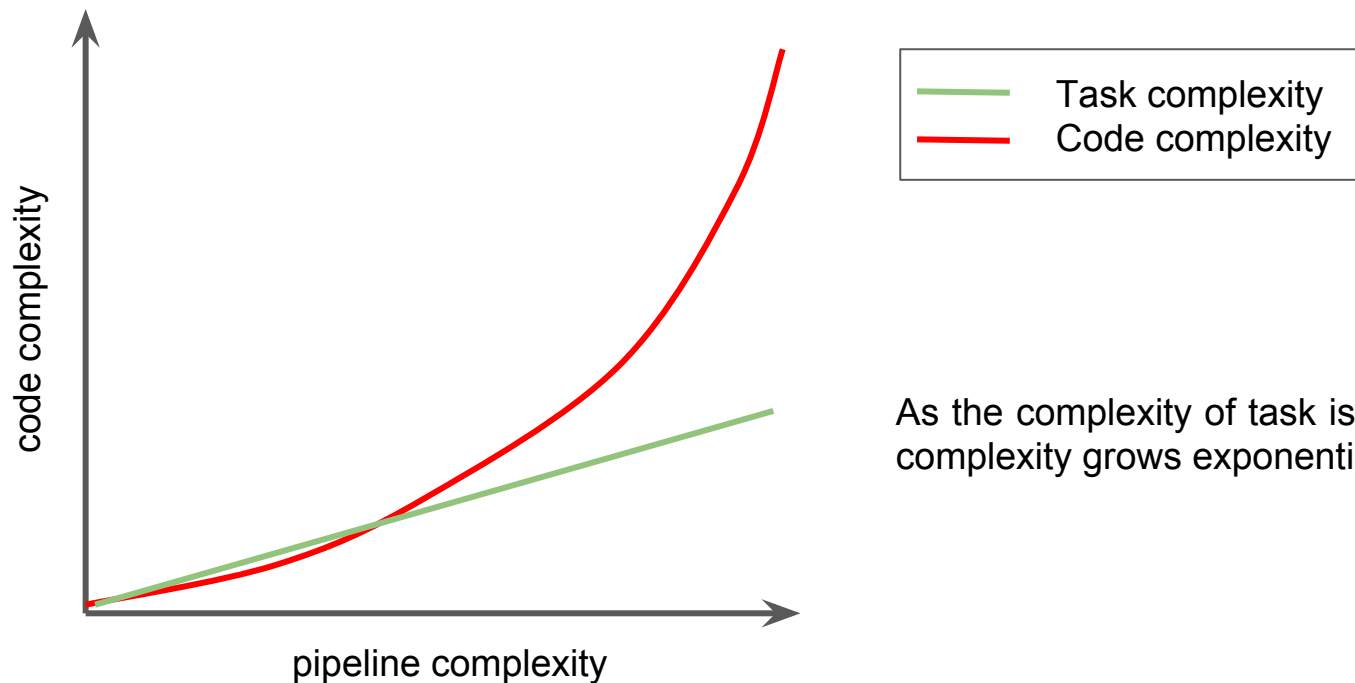


*“Easy, I can write a script in ** to do that”*

***insert programming language here*

- Runtime optimisation
- Reporting
- Code Readability
- Share-ability
- Versioning
- Dependencies
- Scalability

Pipeline vs code complexity



As the complexity of task is linear, code complexity grows exponentially !

A solution: workflow managers



nextflow



nextflow.io

Di Tommaso, Paolo, et al. "Nextflow enables reproducible computational workflows." *Nature biotechnology* 35.4 (2017): 316.

Snakemake vs Nextflow

	Nextflow	Snakemake
Language	Groovy / JVM	Python
Tasks connection	Channels	File name
Asynchronous job gestion	+++	++
Scheduler flexibility	+++	+
HTML reports (results)	N	Y



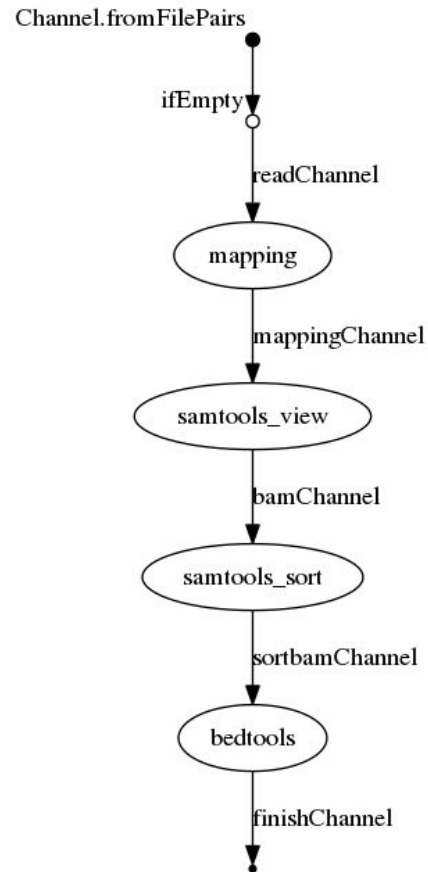
Anatomy of Nextflow pipeline

What is a channel ?

How to give inputs ?

How to define a process ?

How to define arguments/parameters ?



Channels

A First in First out (FiFo) pipe that allows processes to communicate (pass files) between themselves

Channel factory

```
myChannel=Channel.fromPath("/home/data")  
    [/my/data/SRR493366_1.fastq]
```

```
myChannel=Channel.fromFilePairs("/path/to/paired_{1,2}_data")  
    [SRR493366, [/my/data/SRR493366_1.fastq, /my/data/SRR493366_2.fastq]]
```

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

Processes

```
process toto {  
  input:  
    file(myFile) from myInputChannel  
  
  output:  
    file("myOutput.txt") into myOutputChannel  
  
  script:  
    ""  
    myscript $myFile  
    ""  
}
```

input:

<input qualifier> <input name> [from <source channel>] [attributes]

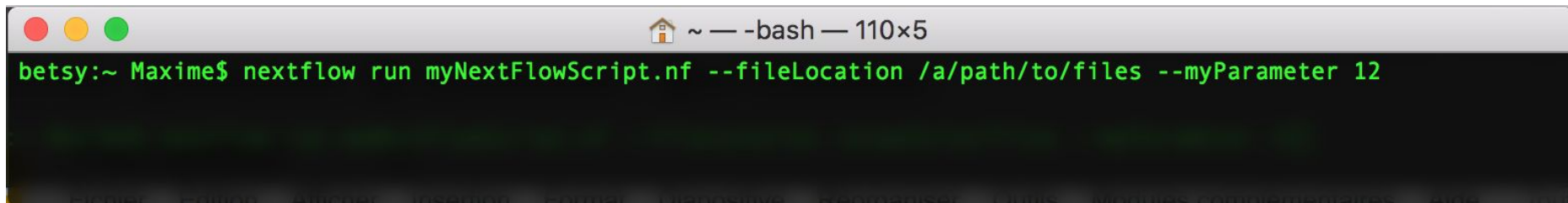
output:

<output qualifier> <output name> [into <target channel>[,channel,...]] [attribute
[,...]]

Parameters

Avoid hardcoded path and parameters !

```
params.fileLocation = "/default/path"  
params.myParameter = 8
```

A screenshot of a terminal window with a light gray title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left, a small house icon representing the home directory, and the text '~ — -bash — 110x5'. The terminal area has a black background with green text. The prompt 'betsy:~ Maxime\$' is followed by the command 'nextflow run myNextFlowScript.nf --fileLocation /a/path/to/files --myParameter 12'.

```
betsy:~ Maxime$ nextflow run myNextFlowScript.nf --fileLocation /a/path/to/files --myParameter 12
```

A simple Nextflow pipeline example

```
#!/usr/bin/env nextflow

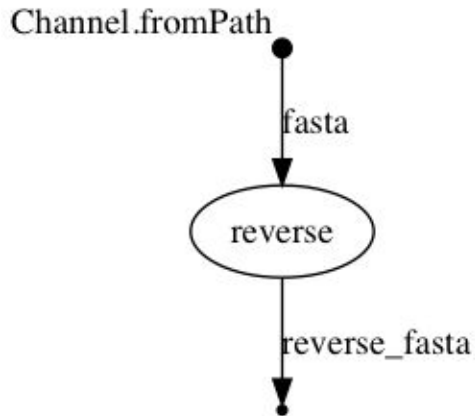
params.fasta = "$baseDir/data/*.fasta"
params.results = "$baseDir/results"

fasta = Channel.fromPath(params.fasta)

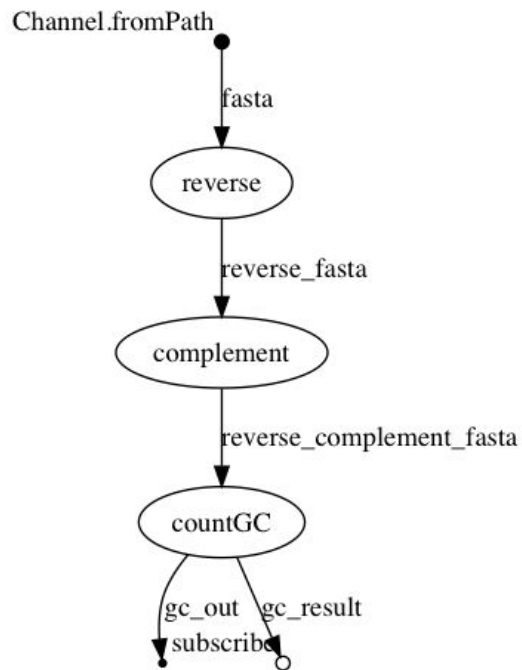
process reverse {
    input:
        file(fa) from fasta

    output:
        file("*.reversed.fa") into reverse_fasta

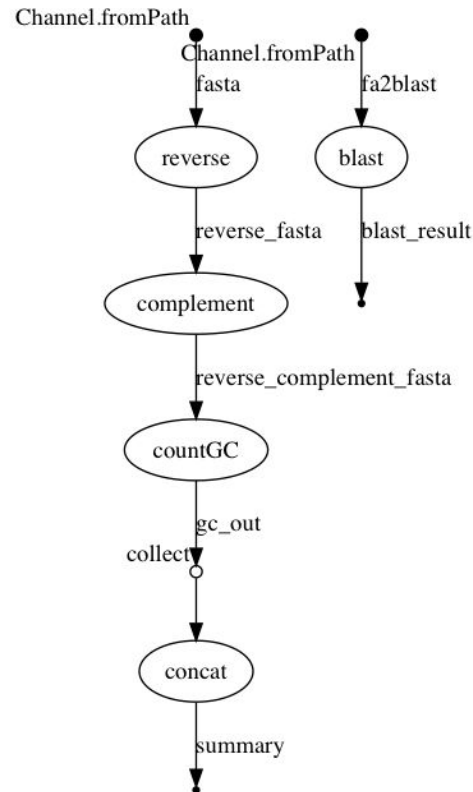
    script:
        """
        reverse.py $fa
        """
}
```



Other simple Nextflow examples

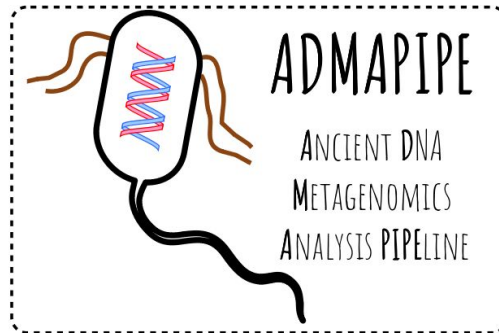
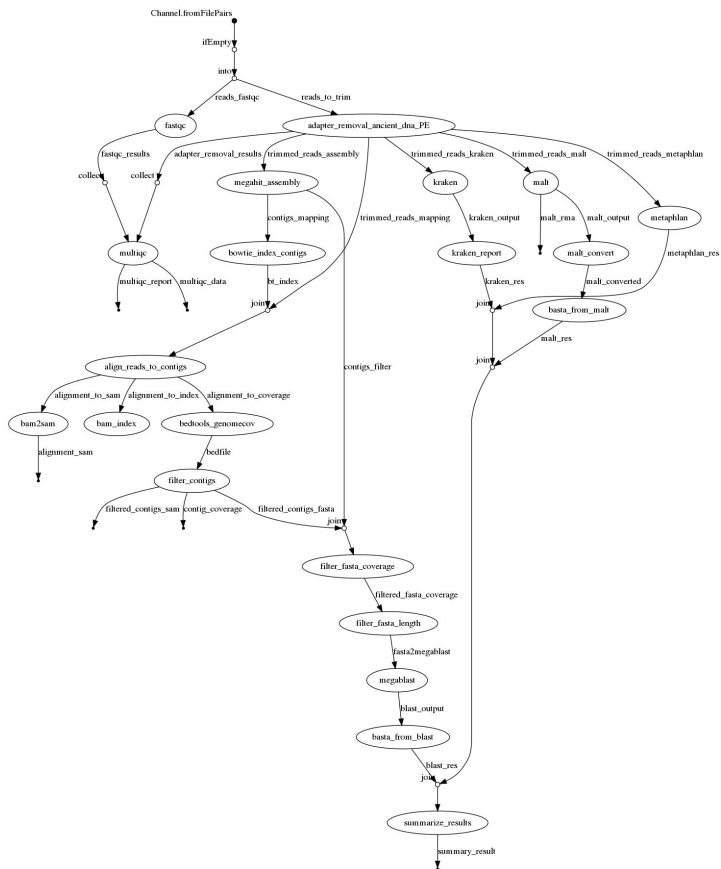


nf-example/example1.nf



nf-example/example2.nf

And more complicated pipelines...



github.com/maxibor/admapipe

Reproducibility

Can be used to run each step :

- Conda environment
- Singularity images
- Docker images

Nextflow Conda

Conda is a package manager that lets you easily install programs into isolated environments

```
conda 'conda_channel::package'
```

```
process blast {  
    conda 'bioconda::blast'  
  
    input:  
    file(fa) from fa  
  
    output:  
    file("*.out") into blast_result  
  
    script:  
    """  
    blastn -db ${params.blastdb} -query $fa -out blast.out -outfmt 6  
    """  
}
```

Example of Singularity recipe file

```
Bootstrap: docker
From: ubuntu:16.04
```

} header (always needed)

```
%files
    ~/genome.fasta /home
```

```
%post
    apt -y update
    apt -y install wget zlib1g bzip2 build-essential
    cd /home
    wget https://github.com/samtools/samtools/
    releases/download/1.3/samtools-1.3.tar.bz2

    tar -xjf samtools-1.3.tar.bz2
    cd samtools-1.3
    ./configure
    make
    make install
```

} sections (optional)

Nextflow Singularity

In code

```
process blast {  
    container '/path/to/fasta.simg'    local image  
  
    input:  
    file(fa) from fa  
  
    output:  
    file("*.out") into blast_result  
  
    script:  
    """  
    blastn -db ${params.blastdb} -query $fa  
    -out blast.out -outfmt 6  
    """  
}
```

need to add `-with-singularity` to the command line

Limitations and benefits

Disk space usage (workdir)

Full potential used on computing clusters

No automated arguments/parameters check (yet ?)

Fast evolving (syntax change)

Now it's up to you !

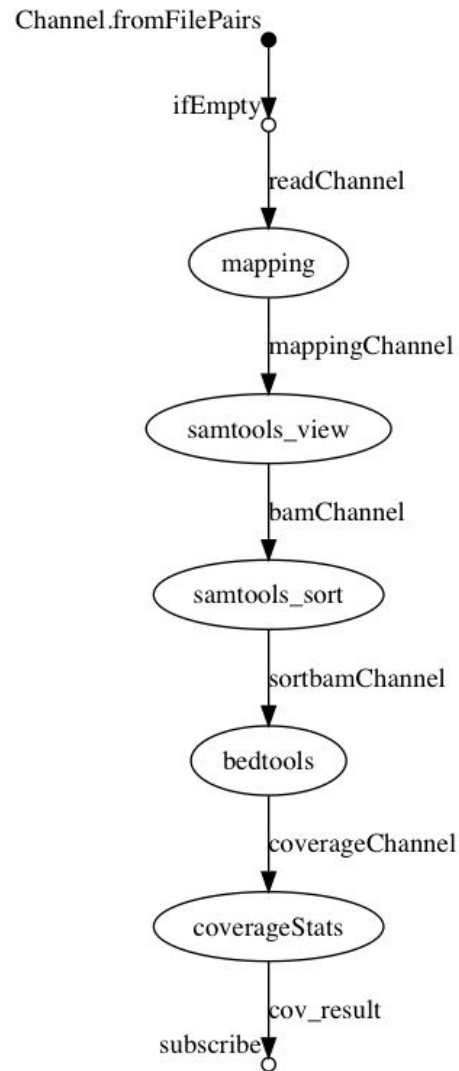
github.com/ParisDiderot-Bioinfo/nextflow-intro

Get Help:

Nextflow Documentation

maxime.borry@gmail.com

quentin.ltn92@gmail.com



First install Nextflow

With conda !

1- Install conda (if not already installed) : conda.io/miniconda

2- `conda install -c bioconda nextflow`