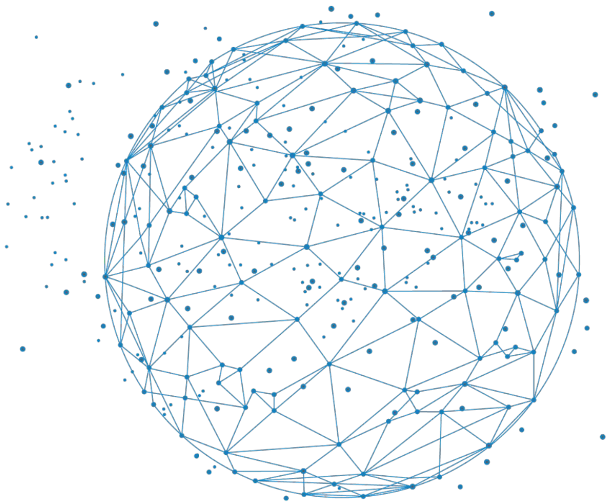
2025 June 6th



What’s a Workflow? From Chaos to Automation

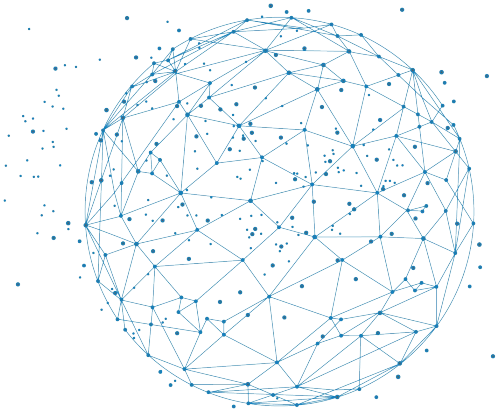
Rousseau Baptiste

IFB INSTITUT FRANÇAIS DE BIOINFORMATIQUE

elixir FRANCE

cnrs INRAE Inserm cea

What’s a Workflow?



What’s a Workflow?

A workflow is **a set of steps to transform input data into output data**

What's a Workflow?

A workflow is **a set of steps to transform input data into output data**

input data

What's a Workflow?

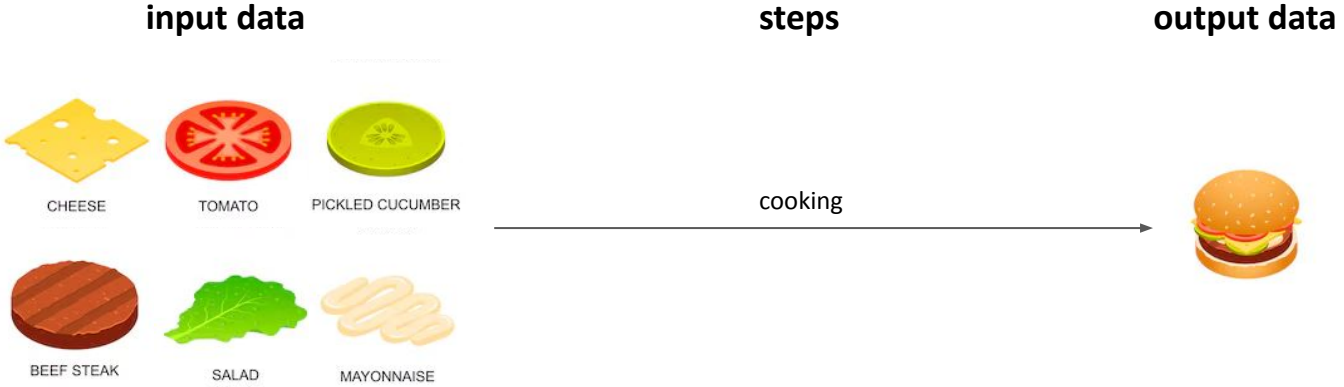
A workflow is **a set of steps to transform input data into output data**

input data

output data

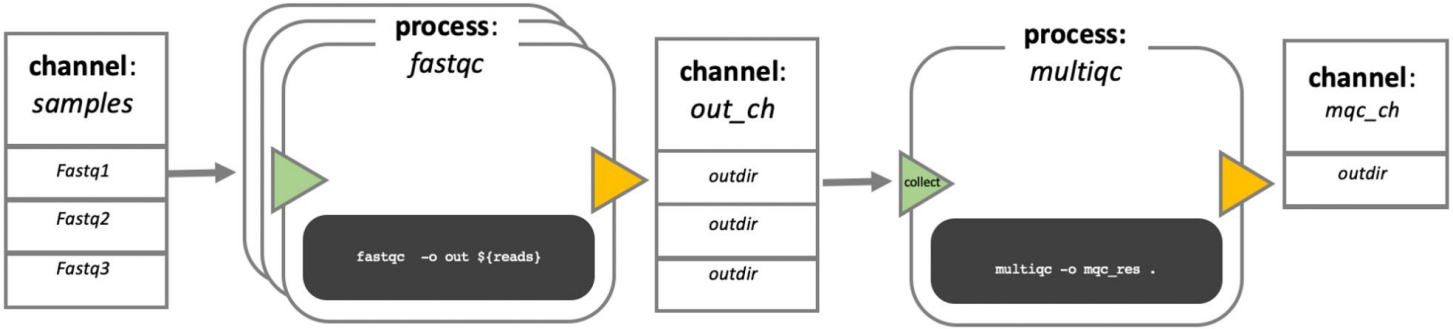
What's a Workflow?

A workflow is **a set of steps to transform input data into output data**



What's a Workflow?

A workflow is **a set of steps to transform input data into output data**



Why Workflows matter?

When planning an analysis, we need to consider and keep track of:

- The format of the data at all stages
- Any pre-processing steps
- Tools/software used, including which release/version
- Parameters we select when using tools

Why Workflows matter?

Why not a tracking table?

- Error-prone
- Non-reproducible
- No automation
- Not collaborative

Why Workflows matter?

Why not a script?

- Hardcoded paths
- No error handling
- Not reproducible
- Not scalable

rna_seq_workflow.sh

```
#!/bin/bash

# Set input and output files
RAW_FASTQ="sample.fastq.gz"
TRIMMED_FASTQ="sample_trimmed.fastq.gz"
QC_DIR="qc_reports"
REFERENCE_GENOME="genome_index"
SAM_FILE="aligned.sam"

# 1. Run FastQC
echo "Running FastQC..."
mkdir -p $QC_DIR
fastqc -o $QC_DIR $RAW_FASTQ

# 2. Trim reads
echo "Trimming reads..."
trimmomatic SE -phred33 \
    $RAW_FASTQ $TRIMMED_FASTQ \
    LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

# 3. Align reads
echo "Aligning reads with HISAT2..."
hisat2 -x $REFERENCE_GENOME -U $TRIMMED_FASTQ -S $SAM_FILE

echo "Workflow completed!"
```

Why Workflows matter?

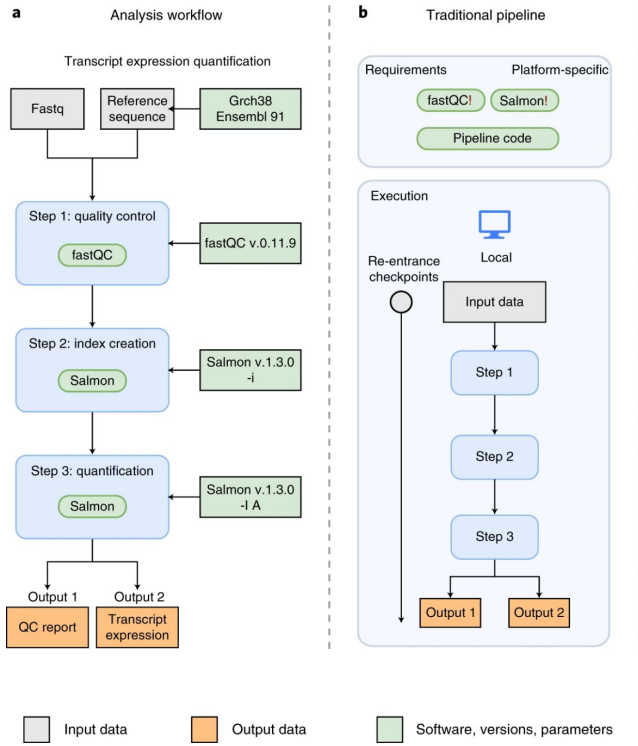
Why a workflow

- Automation
- Run time management
- Software management
- Portability & Interoperability
- Reproducibility

Why Workflows matter?

Why a workflow

- Automation
- Run time management
- Software management
- Portability & Interoperability
- Reproducibility

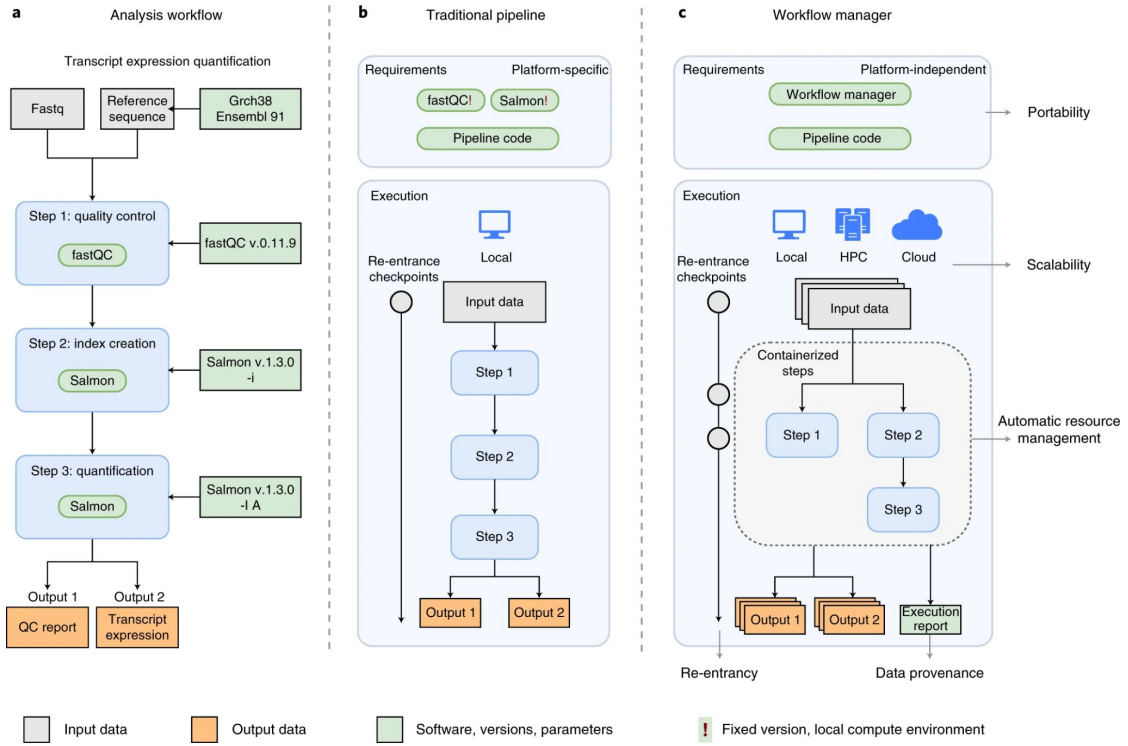


https://doi.org/10.1038/s41592-021-01254-9

Why Workflows matter?

Why a workflow **manager**

- Automation
- Run time management
- Software management
- Portability & Interoperability
- Reproducibility
- **Scalability (local, cluster, cloud)**
- **Resuming of failed runs or steps**
- **Modularity**
- **Easy configuration**

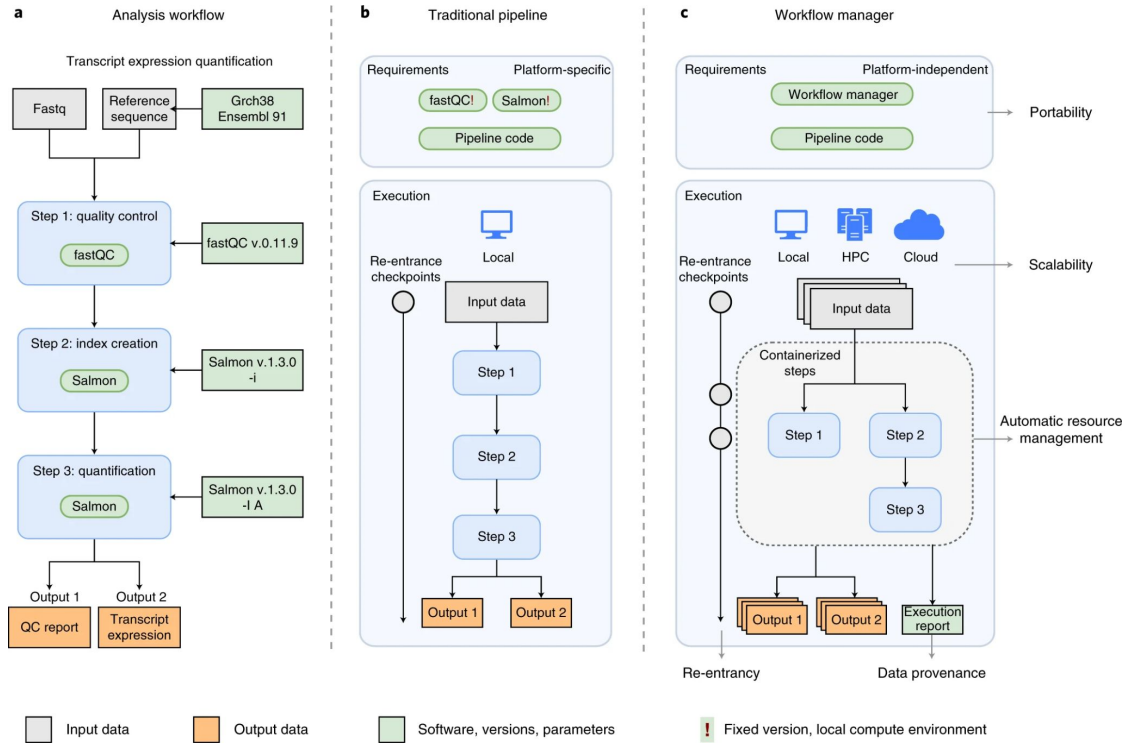


https://doi.org/10.1038/s41592-021-01254-9

Why Workflows matter?

Why a workflow manager

- Automation
- Run time management
- Software management
- Portability & Interoperability
- Reproducibility
- Scalability (local, cluster, cloud)
- Resuming of failed runs or steps
- Modularity
- Easy configuration



https://doi.org/10.1038/s41592-021-01254-9

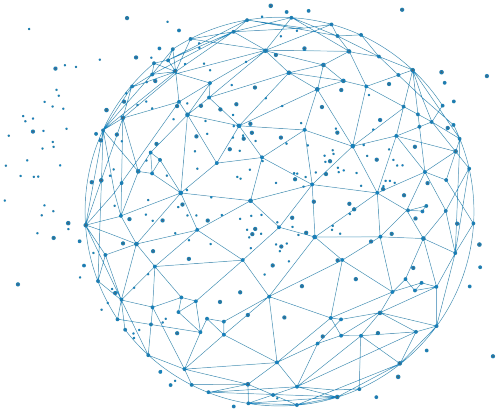
Workflow Managers

- Web-based, no coding required
- Point-and-click interface
- Strong in community/shared workflows

- Python-inspired
- “Makefile for bioinformatics”
- Brief syntax sample
- Emphasize: rule-based, simple to start

- Based on Groovy (scripting language for Java)
- Good for cloud/scalability
- Popular in nf-core community

Anatomy of a Workflow



General Structure

Key Concepts:

- A workflow is made of **steps** (aka *rules* in Snakemake, *processes* in Nextflow).
- Steps are connected by **data flow**: the output of one step becomes the input of the next.
- Each step may:
  - Run a tool or script
  - Process multiple inputs
  - Produce one or more outputs

General Structure

BEEF STEAK

Prepare the patties

Cook the patties

CHEESE

TOMATO

PICKLED CUCUMBER

Assemble the burger

SALAD

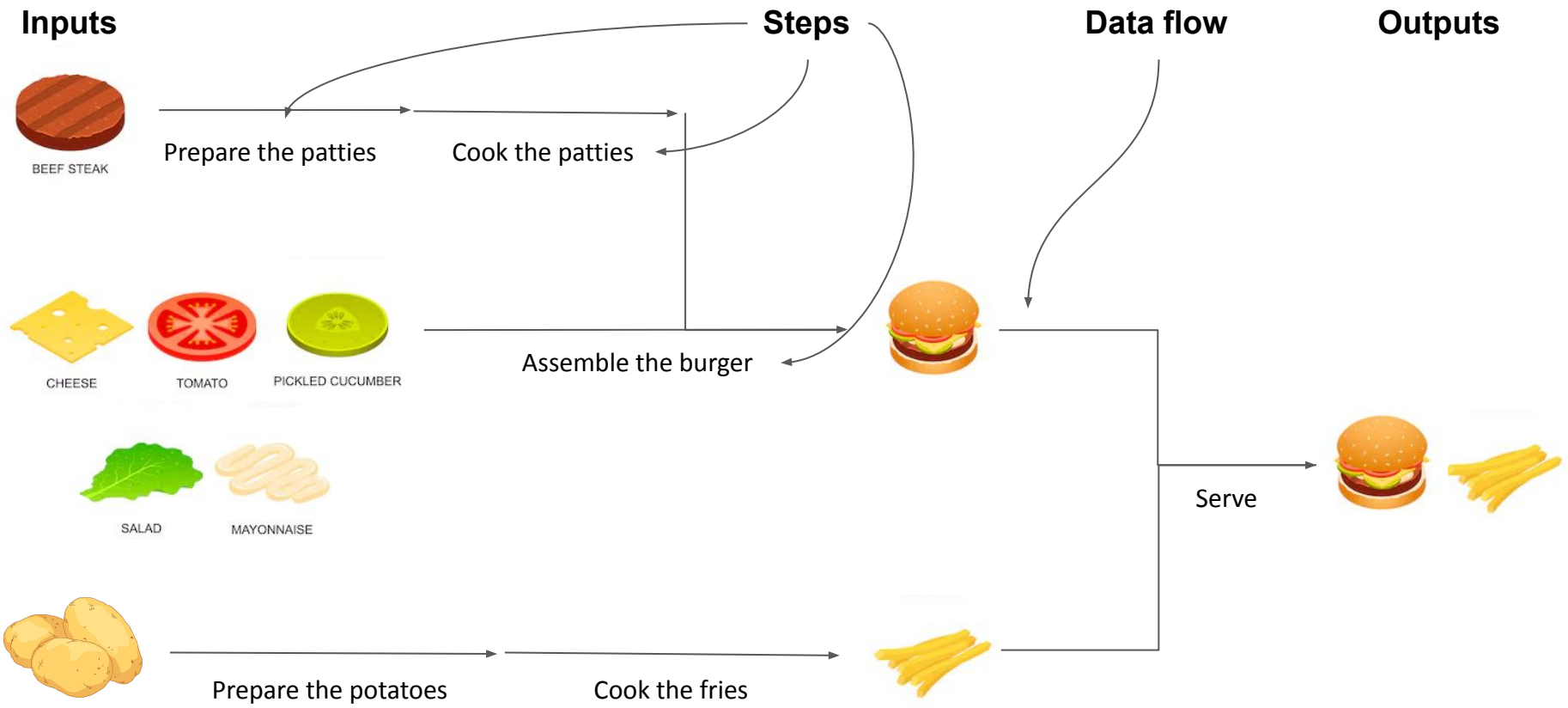
MAYONNAISE

Serve

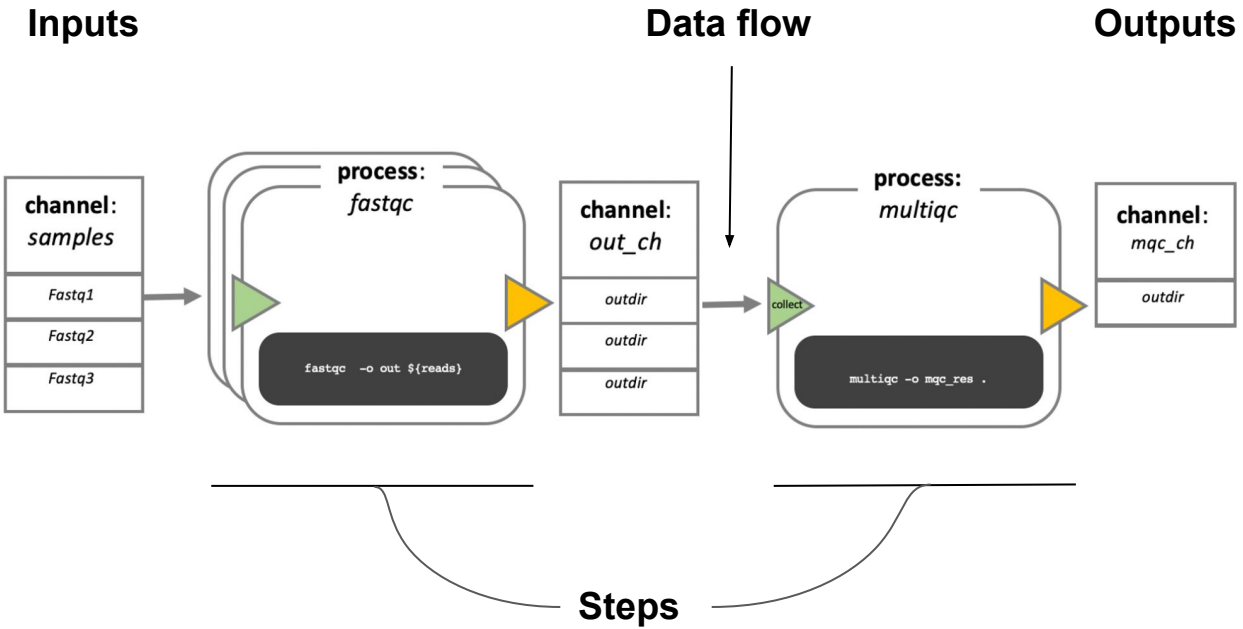
Prepare the potatoes

Cook the fries

General Structure



General Structure



General Structure

Steps: the building blocks of workflows.

- A step is a self-contained unit of work.
- It declares what it needs (**inputs**), what it produces (**outputs**), and how to do it (**command**).

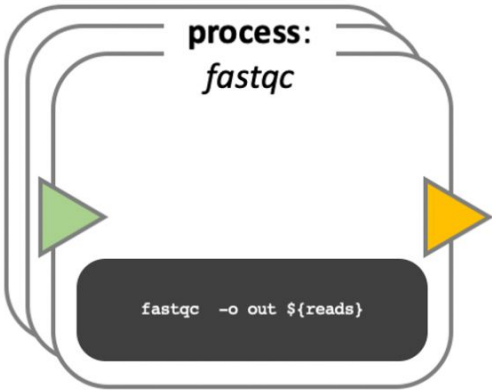
- In Snakemake → called a **rule**.
- In Nextflow → called a **process**.
- In Galaxy → each **tool** is like a process, connected via a GUI.

General Structure

Steps: the building blocks of workflows.

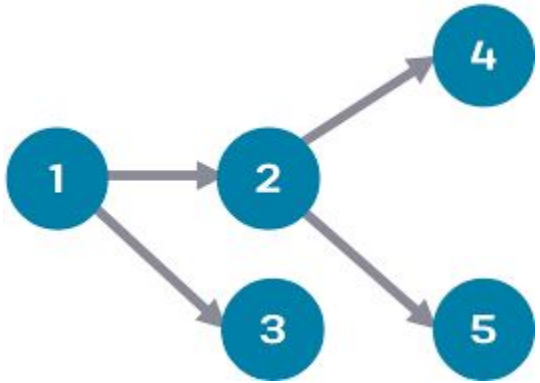
- A step is a self-contained unit of work.
- It declares what it needs (**inputs**), what it produces (**outputs**), and how to do it (**command**).

- In Snakemake → called a **rule**.
- In Nextflow → called a **process**.
- In Galaxy → each **tool** is like a process, connected via a GUI.



The **data flow**:

- In Nextflow, **channels** are the way data flows between processes.
- In Snakemake, **dependencies** are defined by files - a rule runs when its inputs are ready.

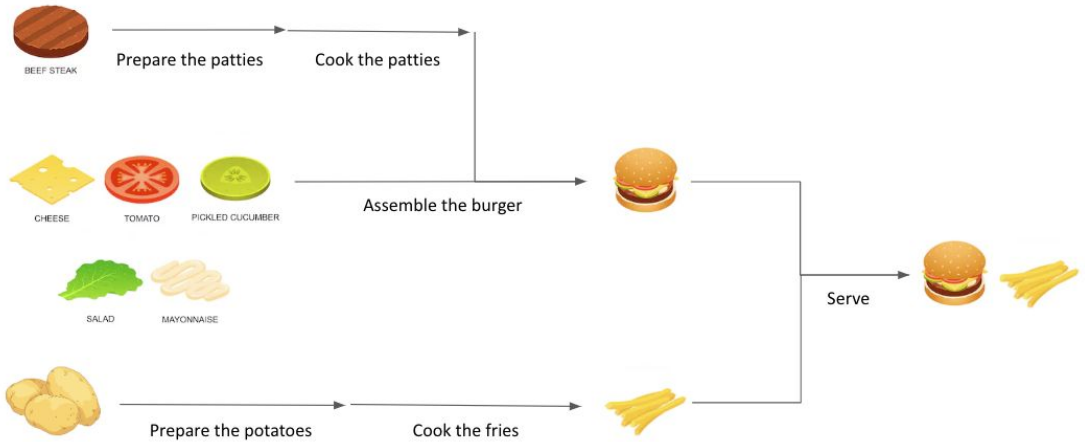


A simple Directed Acyclic Graph (DAG)

An example workflow

Steps:

1. Prepare the potatoes → cut_potatoes
2. Cook the fries → fry_fries (after 1)
3. Prepare the patties → shape_patties
4. Cook the patties → cook_patties (after 3)
5. Assemble the burgers → assemble_burgers (after 4)
6. serve → serve (after 2 and 5)



An example workflow with snakemake

```
rule name:
    input:
        <input files or functions>

    output:
        <output files>

    params:
        <extra parameters, e.g. tool options>  # optional

    threads:
        <number of threads to use>             # optional

    resources:
        <resource limits like mem_mb, time>    # optional

    conda:
        "<path/to/env.yaml>"                   # optional


    shell:
        """
        <command to run>
        """
```

An example workflow with snakemake

```
rule name:
    input:
        <input files or functions>

    output:
        <output files>

    params:
        <extra parameters, e.g. tool options>  # optional

    threads:
        <number of threads to use>             # optional

    resources:
        <resource limits like mem_mb, time>    # optional

    conda:
        "<path/to/env.yaml>"                   # optional


    shell:
        """
        <command to run>
        """
```

```
# Snakefile

SAMPLES = ["order1", "order2", "order3"]

rule all:
    input:
        expand("served/{sample}.txt", sample=SAMPLES)

rule cut_potatoes:
    output:
        "fries/cut_{sample}.txt"
    shell:
        "echo 'Pommes de terre coupées pour {wildcards.sample}' > {output}"

rule fry_fries:
    input:
        "fries/cut_{sample}.txt"
    output:
        "fries/fried_{sample}.txt"
    shell:
        "echo 'Frites cuites pour {wildcards.sample}' > {output}"

rule shape_patties:
    output:
        "burgers/raw_patty_{sample}.txt"
    shell:
        "echo 'Steak haché formé pour {wildcards.sample}' > {output}"
```

An example workflow with nextflow

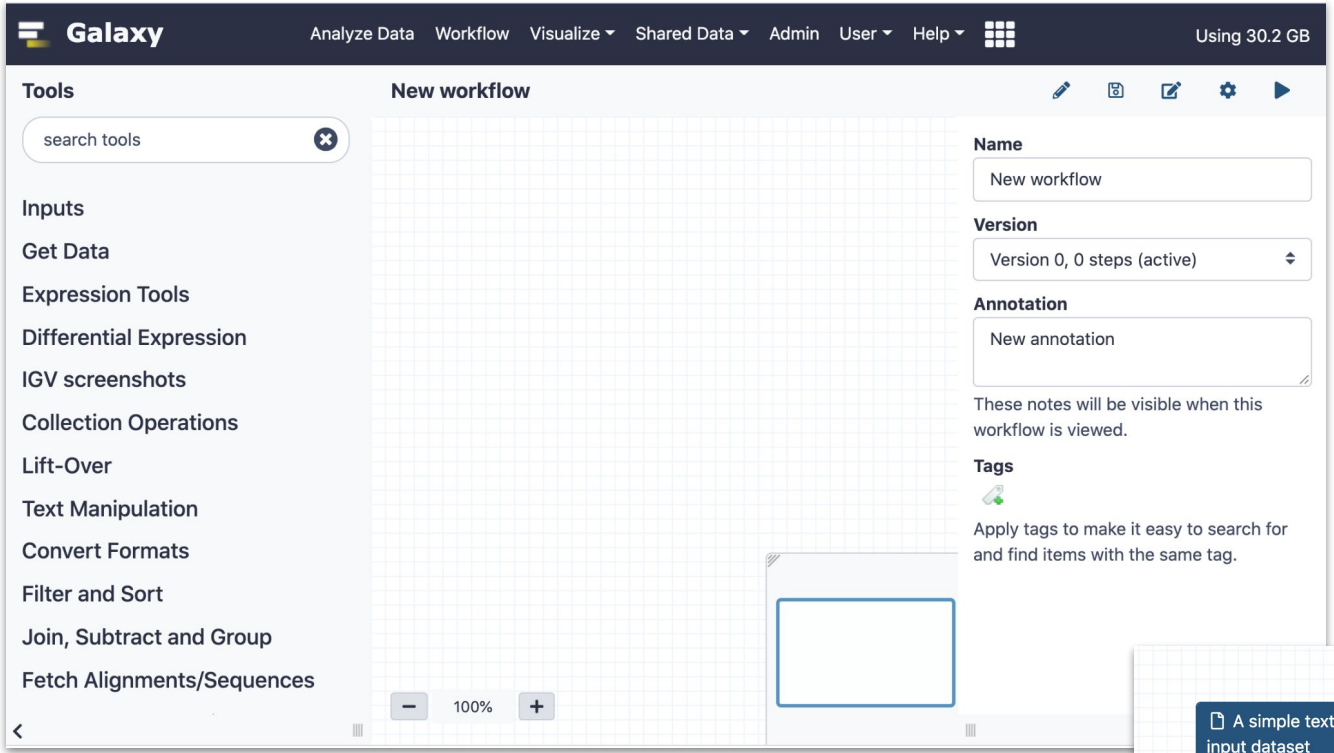
```
process name {
    tag "<short description for logs>"

    publishDir "results/", mode: 'copy'       // optional

    cpus 4                                    // optional
    memory '8 GB'                             // optional
    time '2h'                                 // optional
    errorStrategy 'retry'                     // optional

    container 'biocontainers/tool:version'    // optional

    input:
        <process input>

    output:
        <process output>

    when:
        <condition>                           // optional

    script:
    """
    <command to run>
    """
}
```

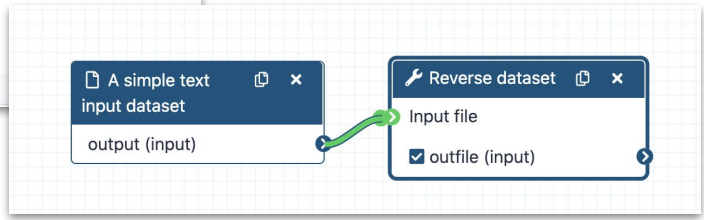
An example workflow with nextflow

```
process name {
    tag "<short description for logs>"

    publishDir "results/", mode: 'copy'       // optional

    cpus 4                                    // optional
    memory '8 GB'                             // optional
    time '2h'                                 // optional
    errorStrategy 'retry'                     // optional

    container 'biocontainers/tool:version'    // optional

    input:
        <process input>

    output:
        <process output>

    when:
        <condition>                           // optional

    script:
    """
    <command to run>
    """
}
```

```
// main.nf (Nextflow DSL2 version)

nextflow.enable.dsl = 2

params.orders = ['order1', 'order2', 'order3']
Channel.fromList(params.orders).set { orders }

// Step 1: Cut potatoes
process CutPotatoes {
  tag "\$order"
  input:
    val order from orders
  output:
    path "fries/cut_\${order}.txt" into cut_fries_ch

  script:
    """
    mkdir -p fries
    echo 'Pommes de terre coupées pour \${order}' > fries/cut_\${order}.txt
    """
}

// Step 2: Fry fries
process FryFries {
  tag "\$order"
  input:
    path cut_file from cut_fries_ch.map { file ->
      def name = file.getBaseName().replaceFirst(/^cut_/, '')
      [name, file]
    }
  output:
    path "fries/fried_\${order}.txt" into fried_fries_ch

  script:
    """
    order_name=\$(basename \${cut_file} | sed 's/^cut_//' | sed 's/\.txt//')
    mkdir -p fries
    echo 'Frites cuites pour \${order_name}' > fries/fried_\${order_name}.txt
    """
}
```

An example workflow with Galaxy





Deep dive into more workflow concepts

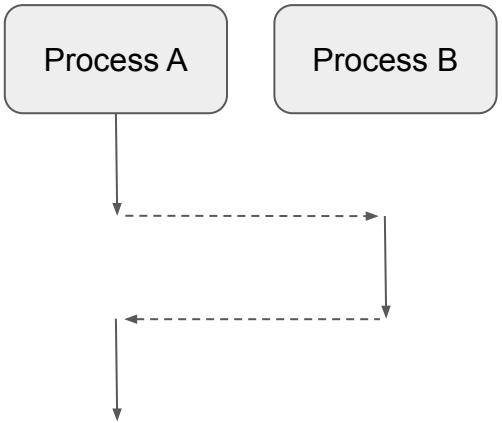
Parallelisation:

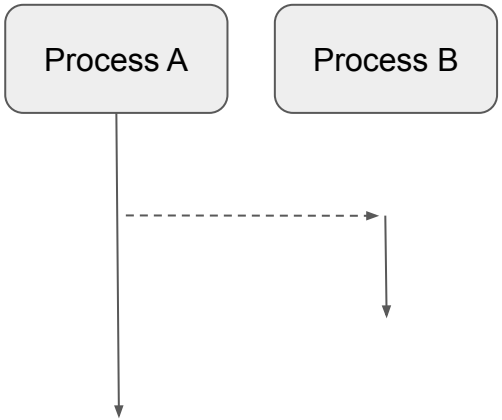
Parallelisation:

- When two steps can be applied simultaneously.

Parallelisation:

- When two steps can be applied simultaneously.





Deep dive into more workflow concepts

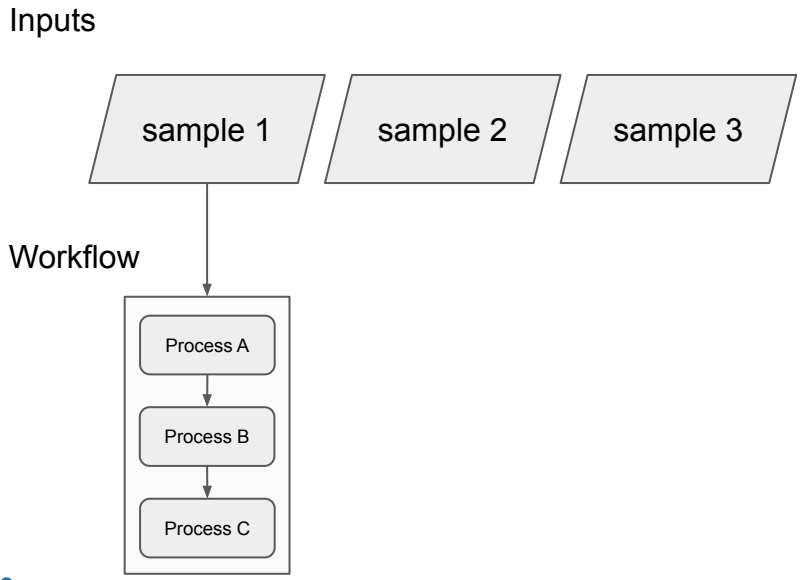
Parallelisation:

- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples

Deep dive into more workflow concepts

Parallelisation:

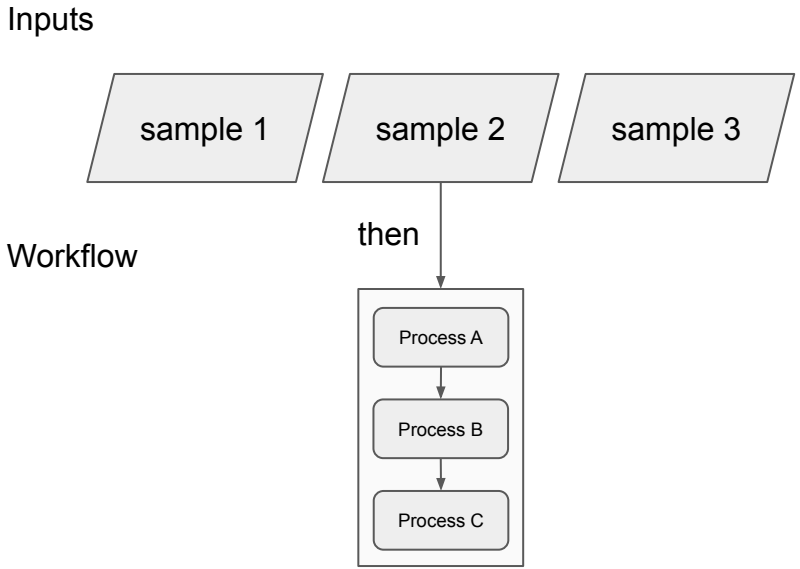
- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples



Deep dive into more workflow concepts

Parallelisation:

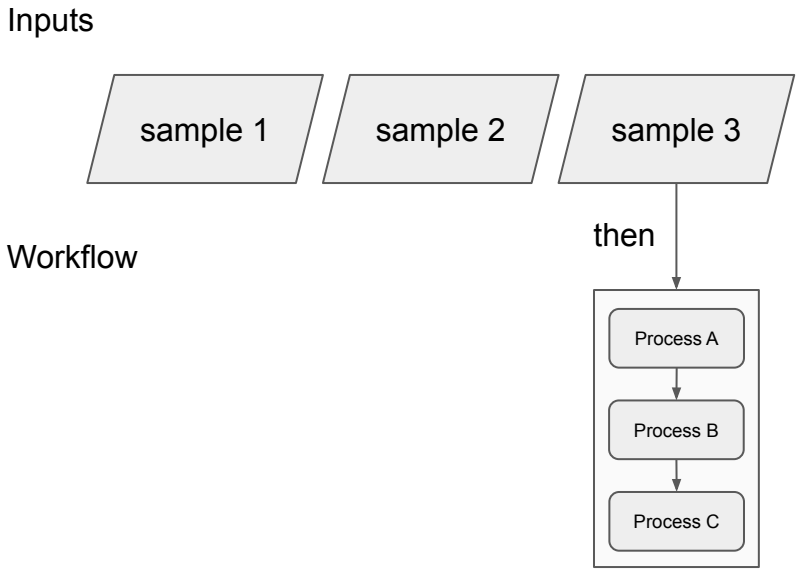
- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples



Deep dive into more workflow concepts

Parallelisation:

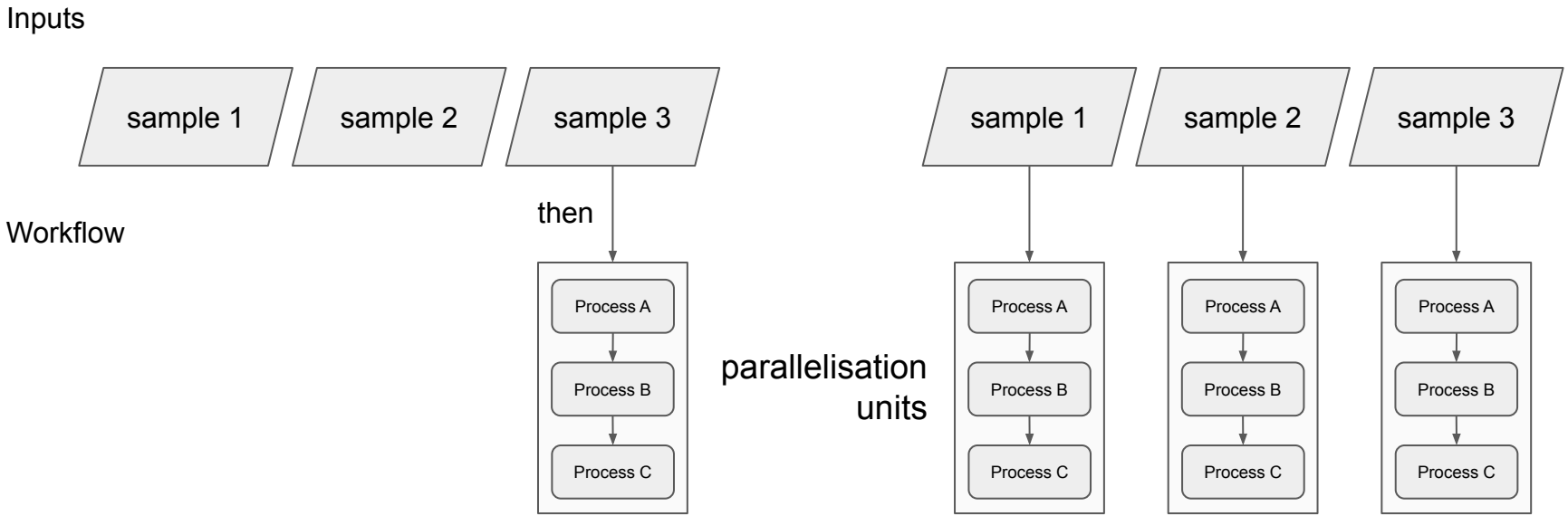
- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples



Deep dive into more workflow concepts

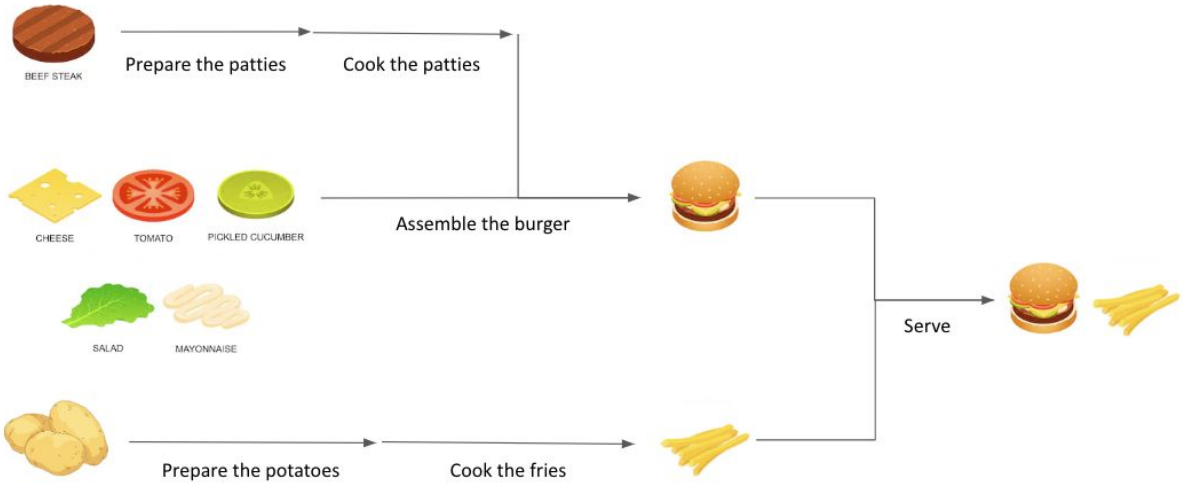
Parallelisation:

- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples



Parallelisation:

- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples



Deep dive into more workflow concepts

Parallelisation:

- When two steps can be applied simultaneously.
- When the same step can be applied to multiple samples

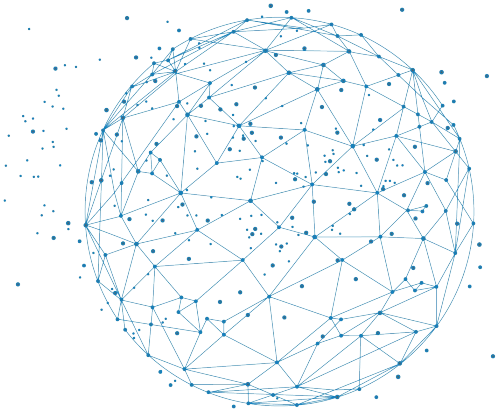
- Workflow engines know what can be done at the same time, they optimize execution.
- You don’t need to write loops, the workflow engine handles it.

Deep dive into more workflow concepts

Operators (Nextflow):

- They allow dynamic and reactive workflows
- Operators let you transform channels
  - Filter (filter, first, unique)
  - Combine (map, groupTuple, collect, flatten)
  - Process text (splitSV, splitFasta)
  - Fork (multiMap, branch)
  - Math (count, min, max)

Takeaways



Key takeaways

What is a Workflow?

- A **workflow** is a formal way to chain multiple processing steps.
- It ensures **automation**, **reproducibility**, and **scalability**.

Why Use a Workflow System?

- No more handwritten tracking
- Each step runs only when needed
- Automatic error detection and resuming
- Tracks software versions, parameters, and inputs/outputs
- Parallel execution = faster processing of many samples

When Should You Consider One**?**

- When your analysis involves **different steps**
- When you need to **reuse**, **share**, or **scale** your work
- When you want to **run the same analysis on multiple datasets**
- When you're collaborating with others or aiming for **FAIR principles**

Key takeaways

Which Workflow manager?

- Galaxy
  - non-coders
  - web interface
  - community workflows
- Snakemake
  - simple, readable workflows
  - great for Python users
- Nextflow
  - scalable and cloud-ready
  - strong for pipelines & HPC

Resources

Trainings:

- IFB WF4bioinfo training - https://moodle.france-bioinformatique.fr/course/view.php?id=29
- Galaxy - https://training.galaxyproject.org/
- Snakemake - https://snakemake.readthedocs.io/en/stable/tutorial/tutorial.html
- Nextflow - https://training.nextflow.io/2.1.7/fr/

Communities:

- WorkflowHub - https://workflowhub.eu/
- Galaxy - https://galaxyproject.org/community/
- nf-core (Nextflow) - https://nf-co.re/

Tools:

- Seqera AI (Snakemake) - https://seqera.io/ask-ai/chat
- GENIAC (Nextflow) - https://geniac.readthedocs.io/en/latest/intro.html

Thank you for your attention !

IFB
INSTITUT FRANÇAIS DE BIOINFORMATIQUE

elixir FRANCE

cnrs INRAE Inserm cea INVESTIR L'AVENIR