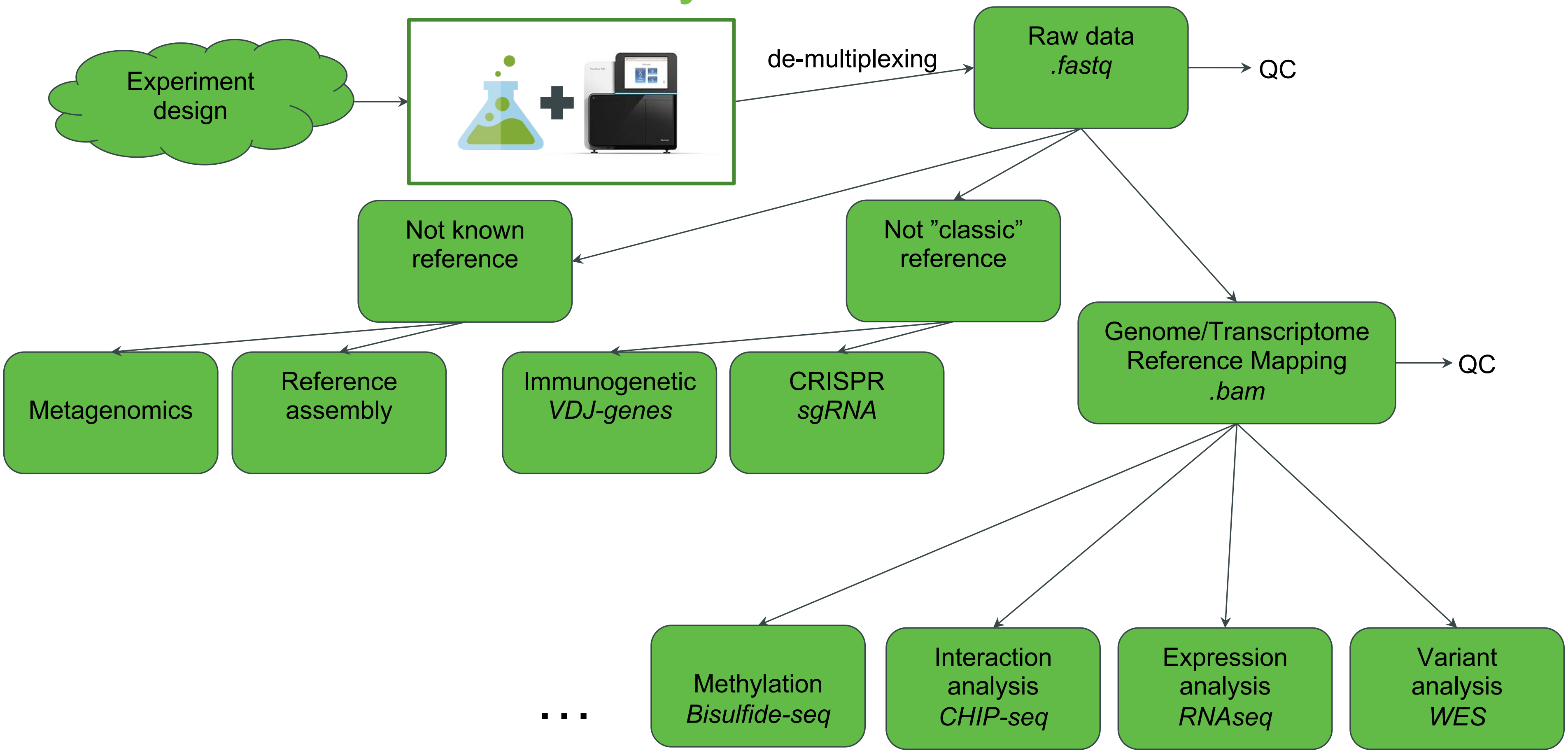
CEITEC

Central European Institute of Technology
BRNO | CZECH REPUBLIC

Bioinformatics workflow management tools

Vojta Bystry
vojtech.bystry@ceitec.muni.cz

NGS data analysis



Experiment design

de-multiplexing

Raw data
.fastq

QC

Not known reference

Not ”classic” reference

Genome/Transcriptome Reference Mapping
.bam

QC

Metagenomics

Reference assembly

Immunogenetic
VDJ-genes

CRISPR
sgRNA

. . .

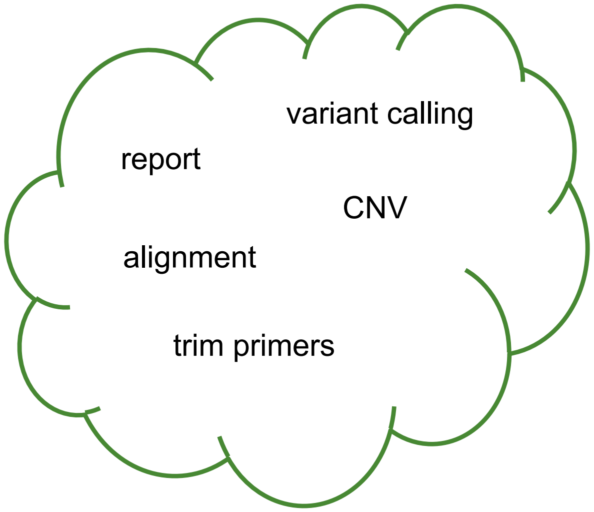
Methylation
Bisulfide-seq

Interaction analysis
CHIP-seq

Expression analysis
RNAseq

Variant analysis
WES

Bioinformatics workflow (pipeline)



variant calling

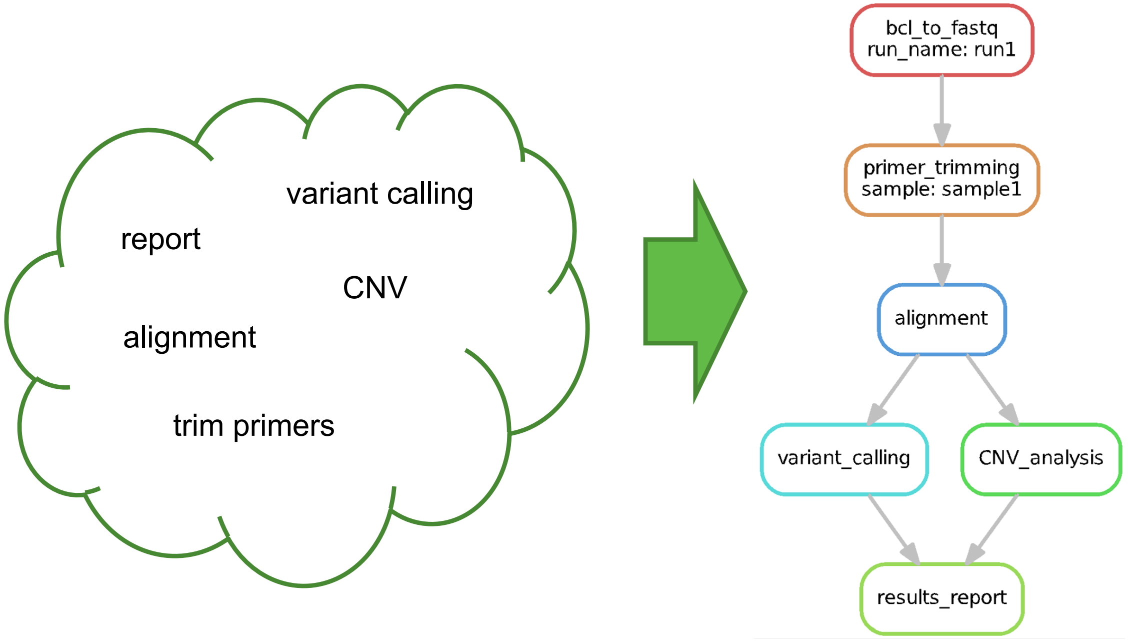
report

CNV

alignment

trim primers

Bioinformatics workflow (pipeline)



variant calling

report

CNV

alignment

trim primers

bcl_to_fastq
run_name: run1

primer_trimming
sample: sample1

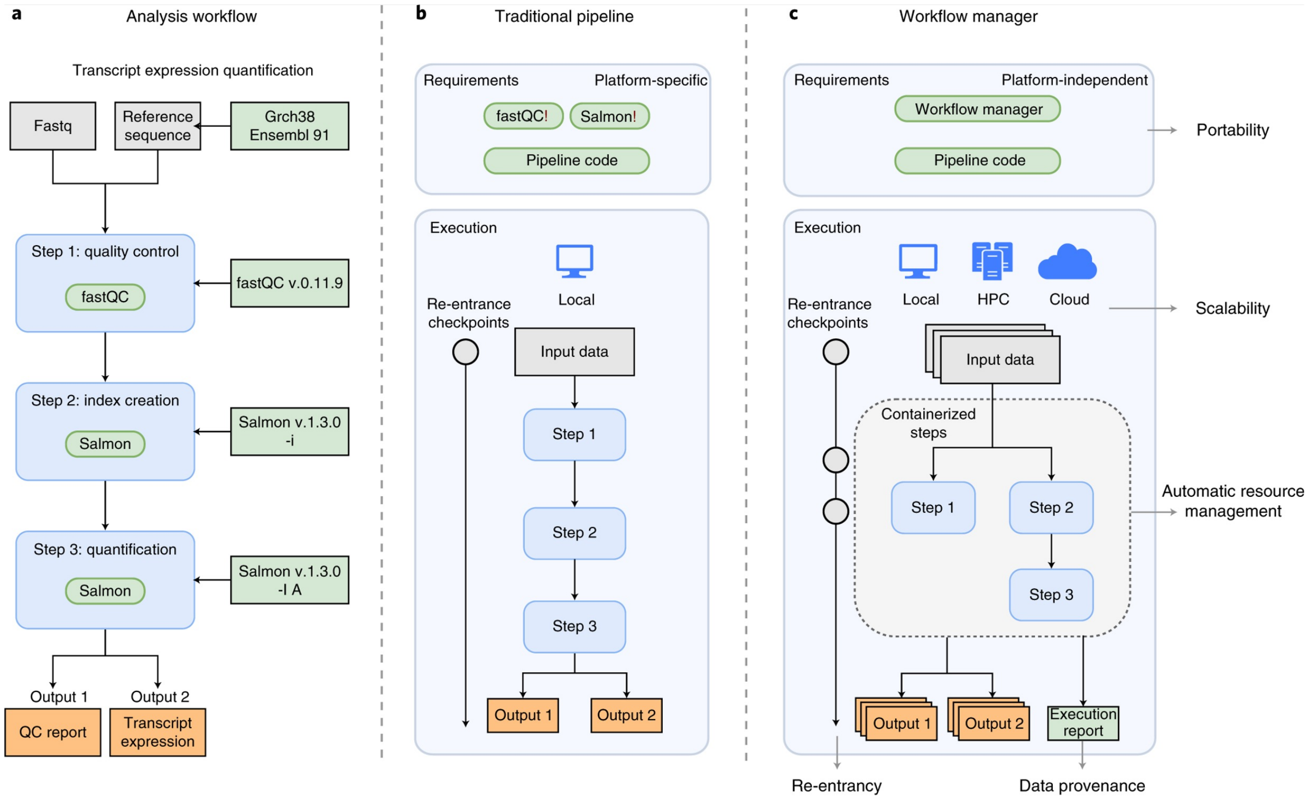
alignment

variant_calling

CNV_analysis

results_report

Bioinformatic workflow management

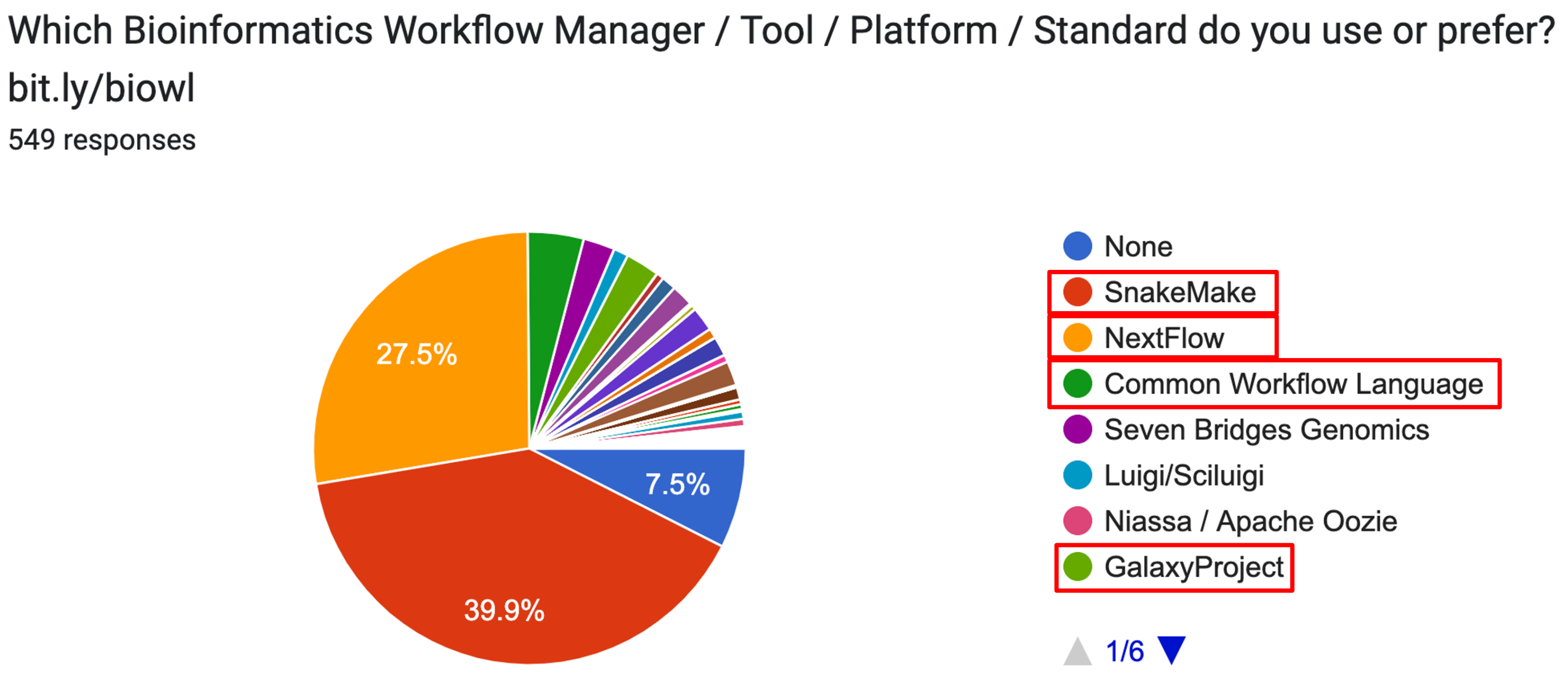


⬜ Input data 🟧 Output data 🟩 Software, versions, parameters ! Fixed version, local compute environment

Bioinformatic workflow management

- Reusability and Reproducibility
- Parallelization and Scale
- Error solving / debuging

Bioinformatic workflow managers



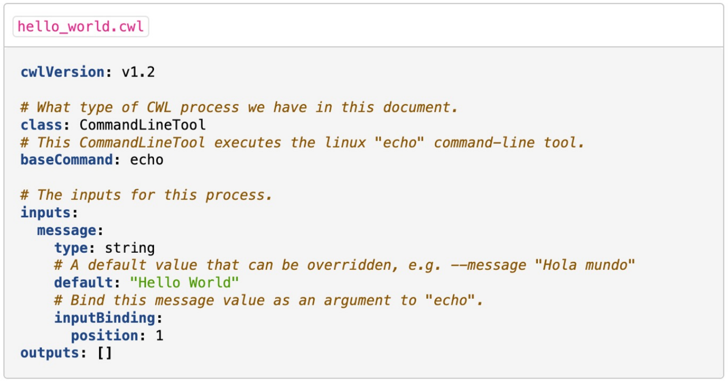
Which Bioinformatics Workflow Manager / Tool / Platform / Standard do you use or prefer?

bit.ly/biowl

549 responses

Common Workflow Language (CWL)

- Pushed by EU projects
- Not big grassroots community
- Scripts in .yaml format

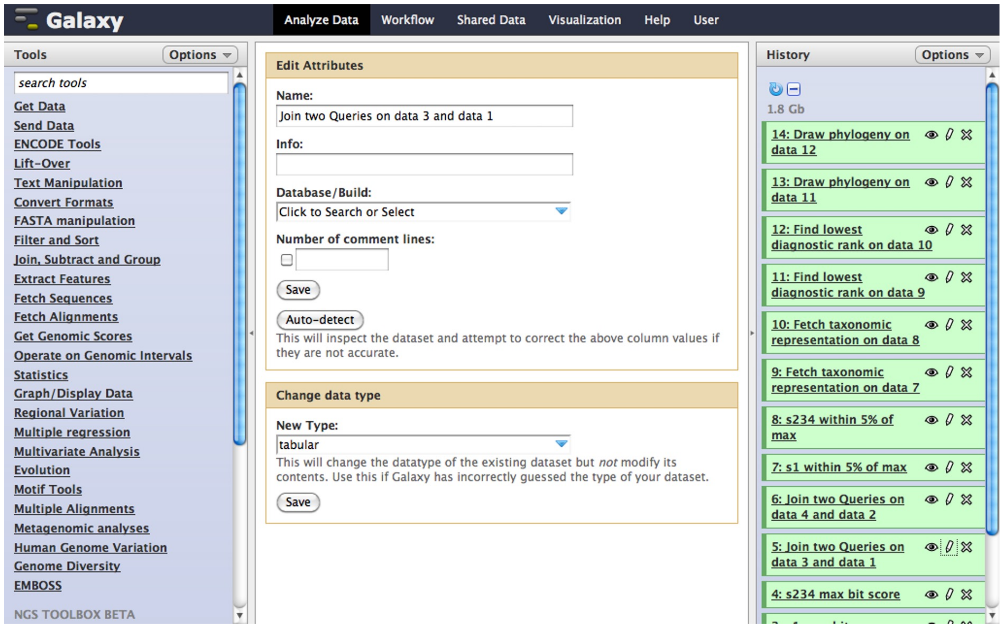


hello_world.cwl

```
cwlVersion: v1.2

# What type of CWL process we have in this document.
class: CommandLineTool
# This CommandLineTool executes the linux "echo" command-line tool.
baseCommand: echo

# The inputs for this process.
inputs:
  message:
    type: string
    # A default value that can be overridden, e.g. --message "Hola mundo"
    default: "Hello World"
    # Bind this message value as an argument to "echo".
    inputBinding:
      position: 1
outputs: []
```

Galaxy project

- Workflow manager with GUI
- Biologists can do their own analysis ???
- It can work - EMBL



Edit Attributes

Name: Join two Queries on data 3 and data 1

Info:

Database/Build: Click to Search or Select

Number of comment lines: □

This will inspect the dataset and attempt to correct the above column values if they are not accurate.

Change data type

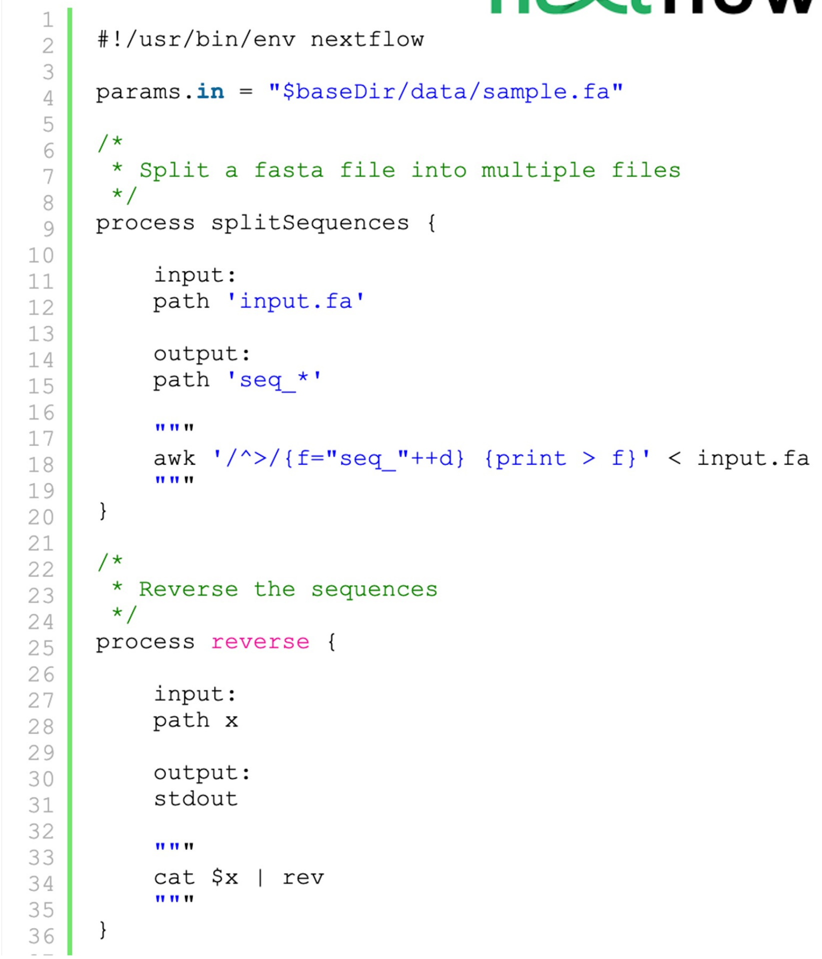
New Type: tabular

This will change the datatype of the existing dataset but *not* modify its contents. Use this if Galaxy has incorrectly guessed the type of your dataset.

Save
History
Options
1.8 Gb
14: Draw phylogeny on data 12
13: Draw phylogeny on data 11
12: Find lowest diagnostic rank on data 10
11: Find lowest diagnostic rank on data 9
10: Fetch taxonomic representation on data 8
9: Fetch taxonomic representation on data 7
8: s234 within 5% of max
7: s1 within 5% of max
6: Join two Queries on data 4 and data 2
5: Join two Queries on data 3 and data 1
4: s234 max bit score

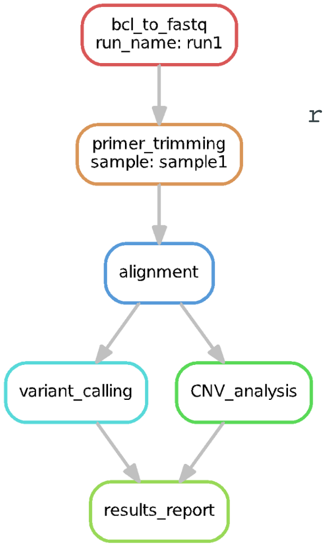
Nextflow

- Great deployability
- Great existing workflow repository



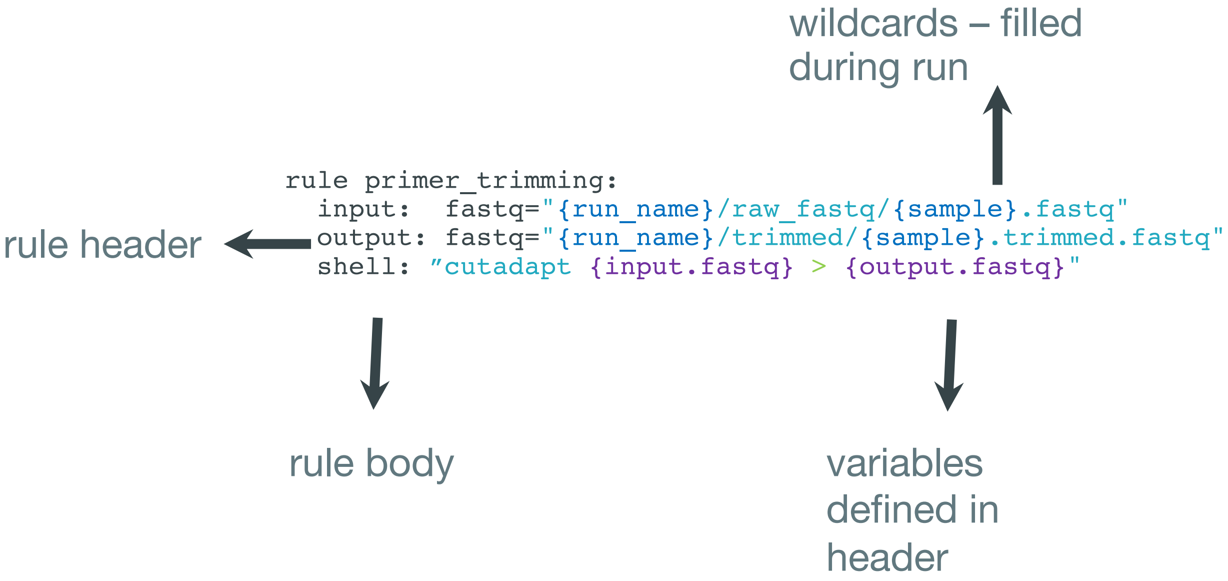
```
#!/usr/bin/env nextflow

params.in = "$baseDir/data/sample.fa"

/*
 * Split a fasta file into multiple files
 */
process splitSequences {

    input:
    path 'input.fa'

    output:
    path 'seq_*'

    """
    awk '/^>/{f="seq_"++d} {print > f}' < input.fa
    """
}

/*
 * Reverse the sequences
 */
process reverse {

    input:
    path x

    output:
    stdout

    """
    cat $x | rev
    """
}
```

Snakemake



```
rule bcl_to_fastq:

rule primer_trimming:
  input:  fastq="{run_name}/raw_fastq/{sample}.fastq"
  output: fastq="{run_name}/trimmed/{sample}.trimmed.fastq"
  shell: ”cutadapt {input.fastq} > {output.fastq}"

rule alignment:

rule VC:    rule CNV:

rule results_report:
```

Snakemake



wildcards – filled during run

```
rule primer_trimming:
  input:  fastq="{run_name}/raw_fastq/{sample}.fastq"
  output: fastq="{run_name}/trimmed/{sample}.trimmed.fastq"
  shell: ”cutadapt {input.fastq} > {output.fastq}"
```

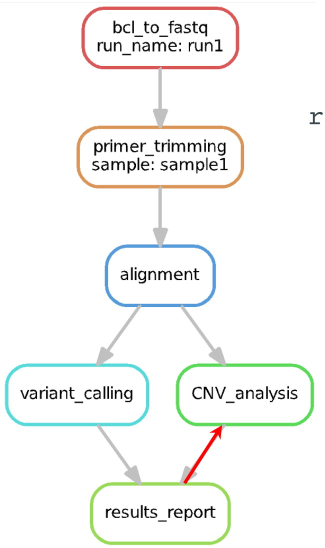
rule header

rule body

variables defined in header

Snakemake

snakemake



```
 rule bcl_to_fastq:

rule primer_trimming:
  input:  fastq="{run_name}/raw_fastq/{sample}.fastq"
  output: fastq="{run_name}/trimmed/{sample}.trimmed.fastq"
  shell: "cutadapt {input.fastq} > {output.fastq}"

 rule alignment:

 rule VC:    rule CNV:
               output: "{run_name}/CNV/{sample}.CNV.tsv"

 rule results_report:
   input: vcf = "{run_name}/VC/{sample}.vcf"
          cnv = "{run_name}/CNV/{sample}.CNV.tsv"
```

config_file.json

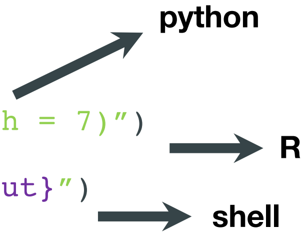
Snakemake

- Simple shell script

```
shell: ”mv —R {input} {output}”
```

- Combine languages

```
run:
if {params.cluster} is TRUE:
  R(”cutree(hclust({input}),h = 7)”)
else:
  shell(”mv —R {input} {output}”)
```



python

R

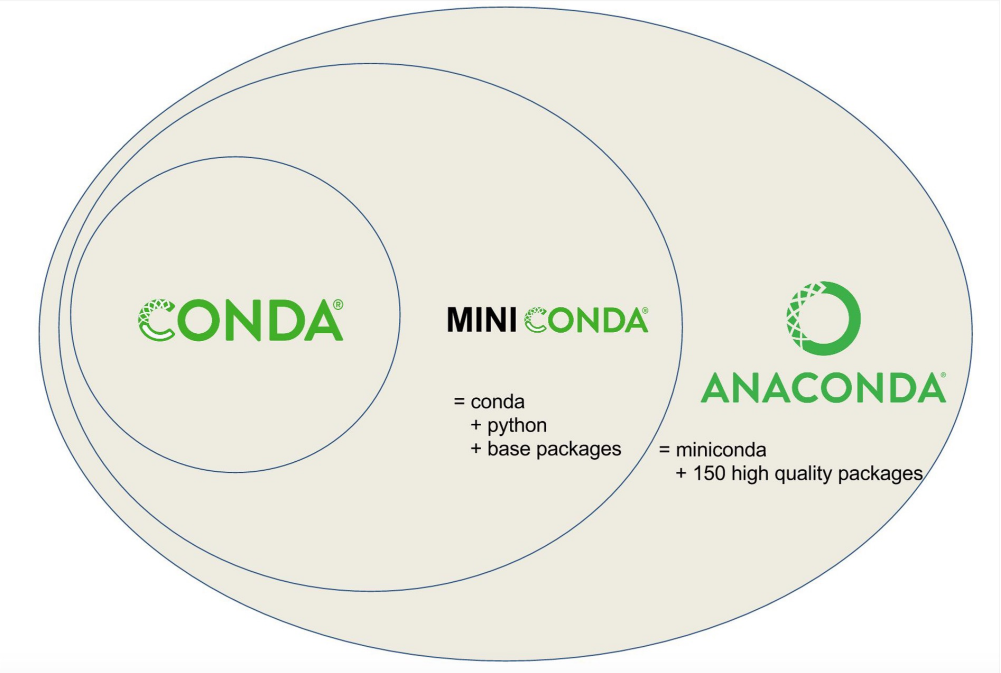
shell

- Wrap it in separate script

```
script: “my_script.py”
```

- Separation of logic and functionality
  - Organization
  - Re-usability

Conda / Anaconda / Bioconda



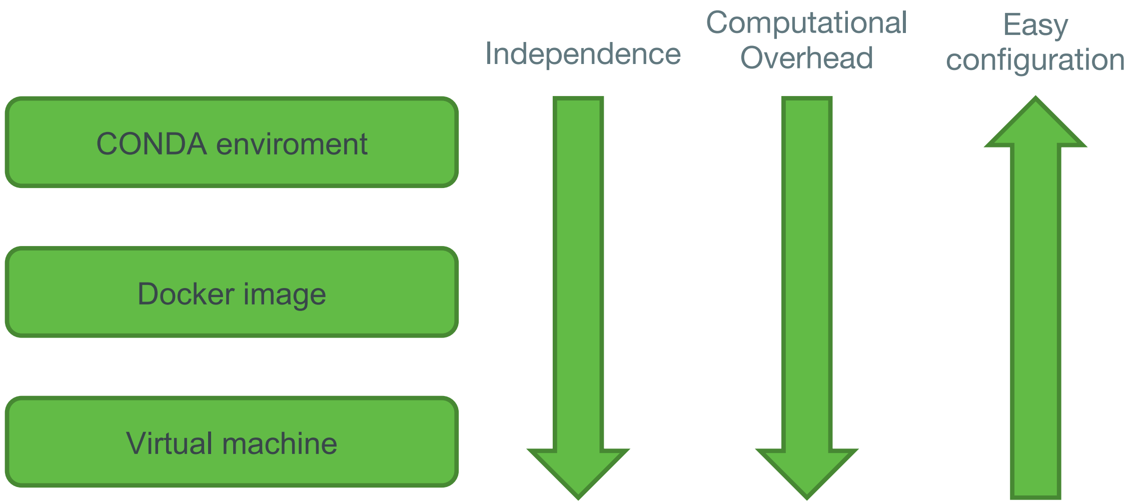
= conda
+ python
+ base packages

= miniconda
+ 150 high quality packages

BIOCONDA

Bioconda is a distribution of bioinformatics software realized as a channel for the versatile Conda package manager.

Conda



Independence

Computational Overhead

Easy configuration

CONDA enviroment

Docker image

Virtual machine

Conda

- Easy installation and management
- Instalation recepies:

```
conda install vardict
conda update vardict
conda remove vardict
conda env create -f myenv.yaml -n myenv
```

- Isolated environments:

```
channels:
 - conda-forge
 - defaults
dependencies:
 - pandas ==0.20.3
 - statsmodels ==0.8.0
 - r-dplyr ==0.7.0
 - r-base ==3.4.1
```

Conda

- Cheat sheet
  - https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf
- Google it
  - conda [bioinformatics tool name]

Computational resources and execution

- Snakemake is quite flexible in cluster execution
  - https://snakemake.readthedocs.io/en/stable/executing/cloud.html
  - ! Nothing works as advertise ☺