# SNAKEMAKE AT ICM

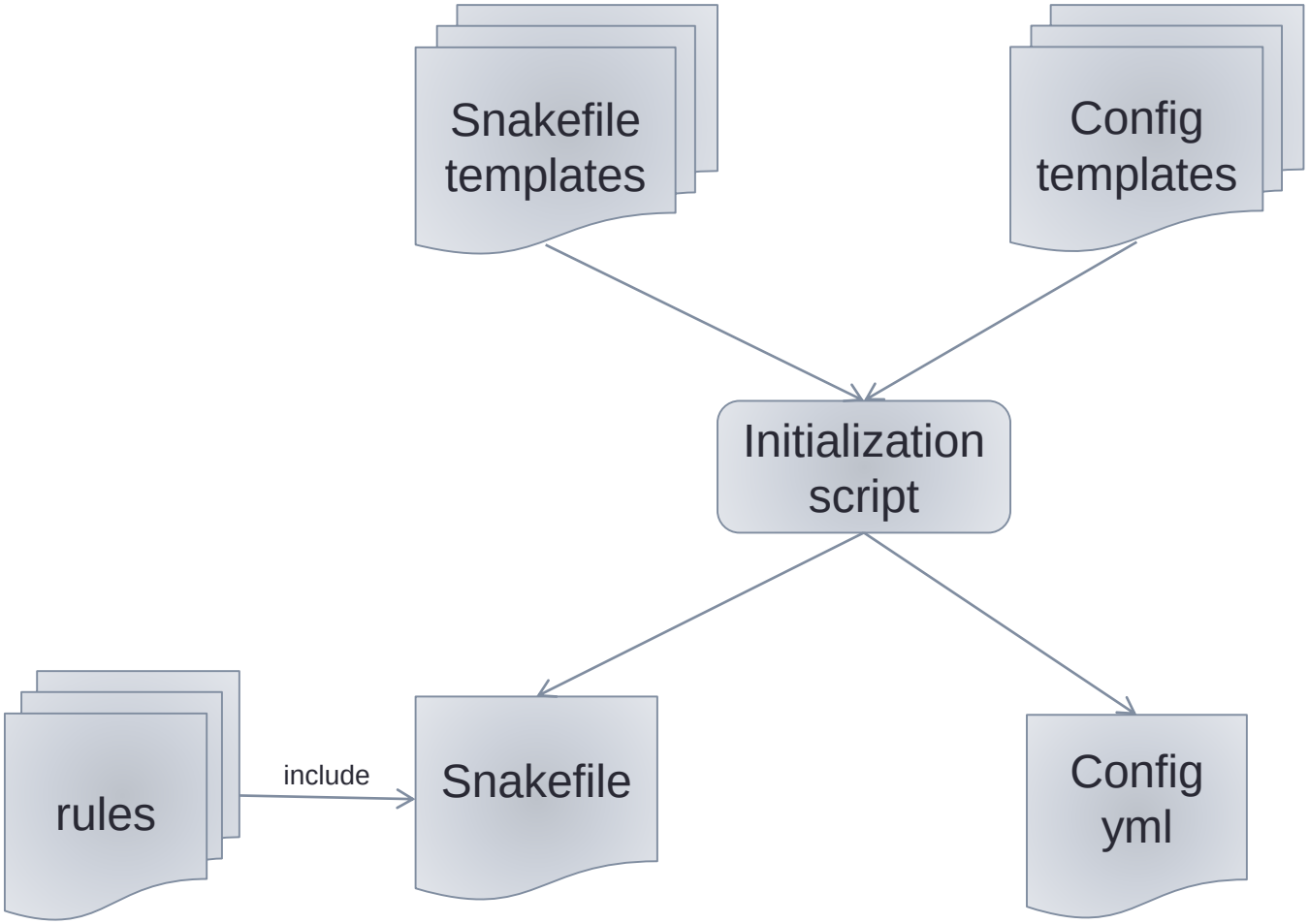
Vivien Deshaies

Justine Guegan

# Pipelines

- WES
- Target sequencing
- RNA-seq
- WGS

# Organization



# Config file (1/3)

```
samples:
    sample1:
        run1:
            fastq_read1: fastq/run1/sample1_1.fastq.gz
            fastq_read2: fastq/run1/sample1_2.fastq.gz
        run2:
            fastq_read1: fastq/run2/sample2_1.fastq.gz
            fastq_read2: fastq/run2/sample2_2.fastq.gz

include_path: ~/git/WGS_pipeline/include/
scripts_path: ~/git/WGS_pipeline/scripts/
```

# Config file (2/3)

```
gatk:
    jar: /tools/GenomeAnalysisTK.jar
    resources:
        low:
            xmx: 6g
            mem: 7
        high:
            xmx: 19g
            mem: 20
```

# Config file (3/3)

```
genome:
    fasta:
        /genomes/human_g1k_v37_decoy.fasta
    chromosomes:
        [1, 2, 3, 4, 5, 6, 7, …]
```

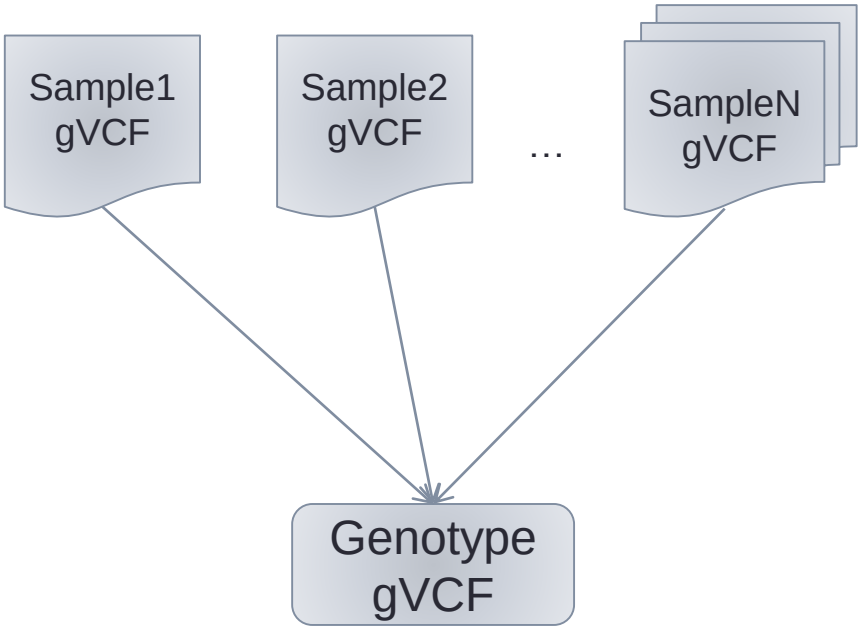
- WGS analysis → pipeline parallelization by chromosome

```
rule merge_clean_bam:
    input:
        bams = expand(
            "tmp_split/{{sample}}_{chrom}_sorted_markdup_recal.bam",
            chrom=config["genome"]["chromosomes"]
        )
```

# Rules

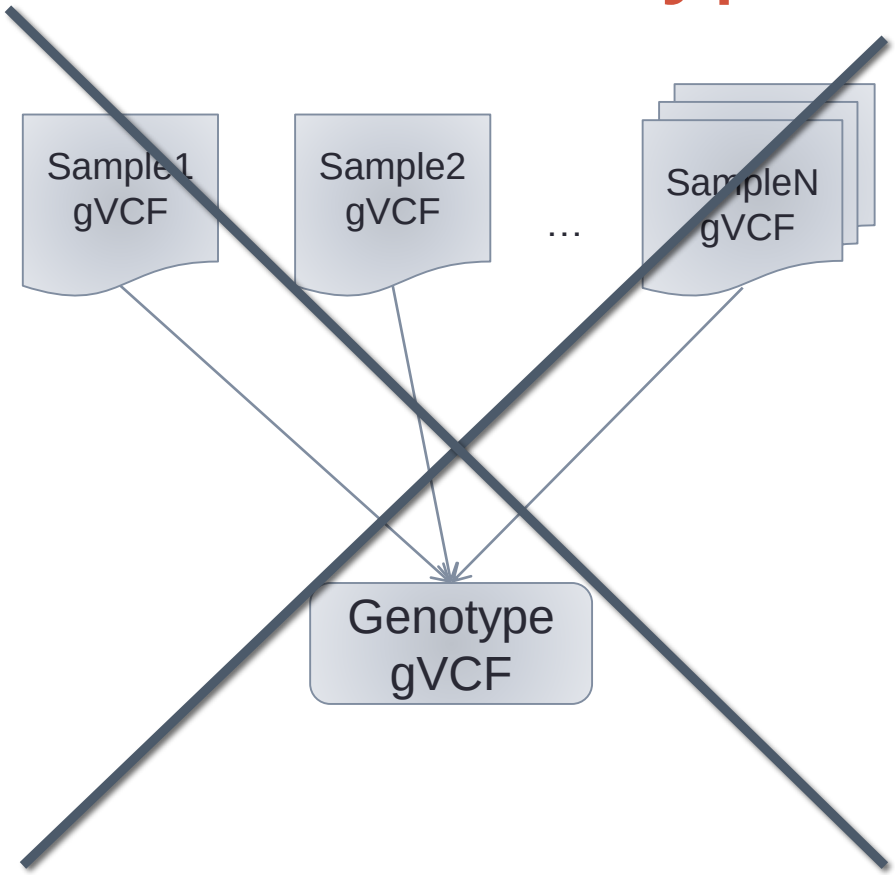
```
rule example:
    input: "merged_reads/{sample,[A-Za-z0-9\-]+}_merged.bam“
    output: "output"
    log:
        out = "logs/example.log",
        err = "logs/example.err"
    benchmark: "benchmarks/example.txt"
    params:
        gatk_jar = config["gatk"]["jar"],
        xmx = config["gatk"]["resources"]["high"]["xmx"],
        genome = config["genome"]["fasta"]
    threads: 1
    resources:
        mem = config["gatk"]["resources"]["high"]["mem"]
    shell:
        "commande"
        " > {log.out} 2> {log.err}"
```

# GATK GenotypeGVCF

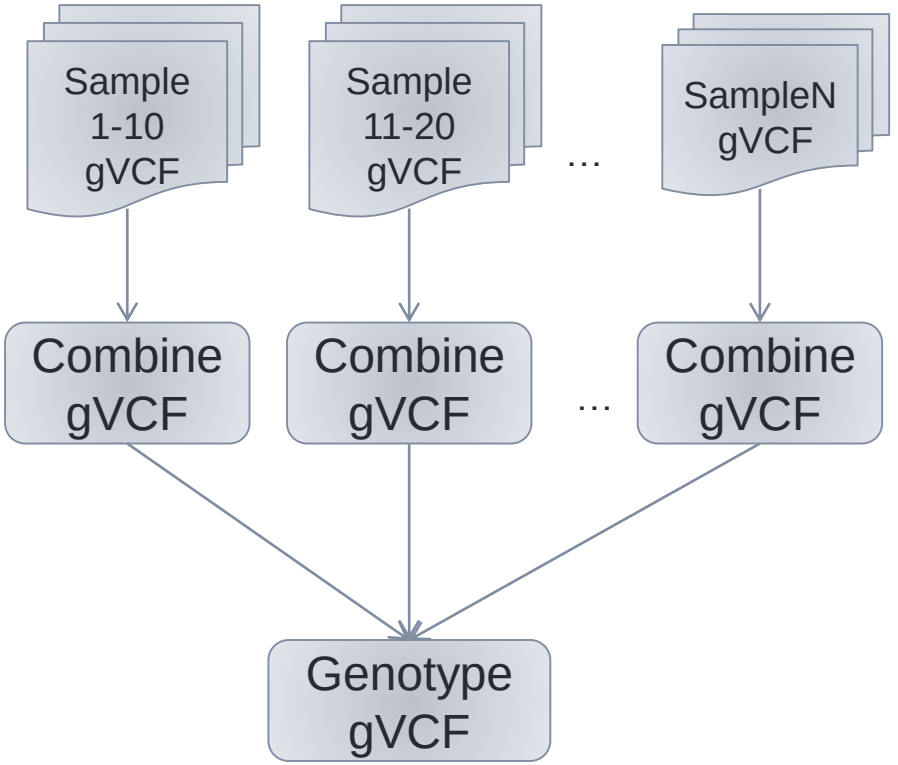



- With 70 samples ~ 100 GB

# GATK GenotypeGVCF




- With 70 samples ~ 100 GB




- With 70 samples ~ 20 GB

# Rule : merge_gvcf

```
sampleGroupList = get_sample_group_list(
                list(config["samples"].keys()),
                config["number_of_sample_by_gvcf"]
)

rule merge_gvcf:
    input:
        lambda wildcards: list(
            ["variants/{sample}.g.vcf.gz".format(sample = sample) for sample in
sampleGroupList[int(wildcards.n)]]
        )
    output: "variants/sample_group.{n}.g.vcf.gz"

rule genotype_call:
     input:
       expand("variants/sample_group.{n}.g.vcf.gz",n=range(0,len(sampleGroupList)))
```

# Snakemake launching

- Local :

```
snakemake -p --cores=20 --resources mem=60
```

- Slurm :

```
snakemake -j200 -p
        --cluster "sbatch --mem={resources.mem}000 -c {threads}"
```

# Rulegraph


bwa_map
bwa_samtools_sort
merge_bam
samtools_index
split_bam
get_unmapped_reads
markdup
generate_base_recalibration_table
apply_recal
call_variants
merge_clean_bam
merge_sample_gvcf
bam_qc
merge_gvcf
genotype_call
vep_annotate
fastqc
all

# Perspectives of evolution

- Scripts conservation
  - Bash scripts created by snakemake are currently automatically deleted
- Log slurm / snakemake
  - Cluster errors are in : slurm-{jobid}.out
  - Command errors are in snakemake output

# Thanks to


- Justine Guegan
- Ivan Moszer
- Romain Daveau
- Vincent Perlbarg
- Ludovic Prevost