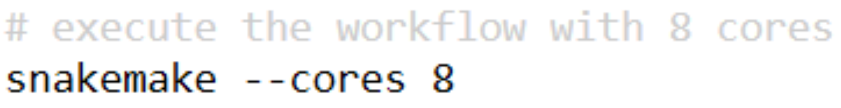
Job 병렬화

Disjoint paths in the DAG of jobs can be executed in parallel.

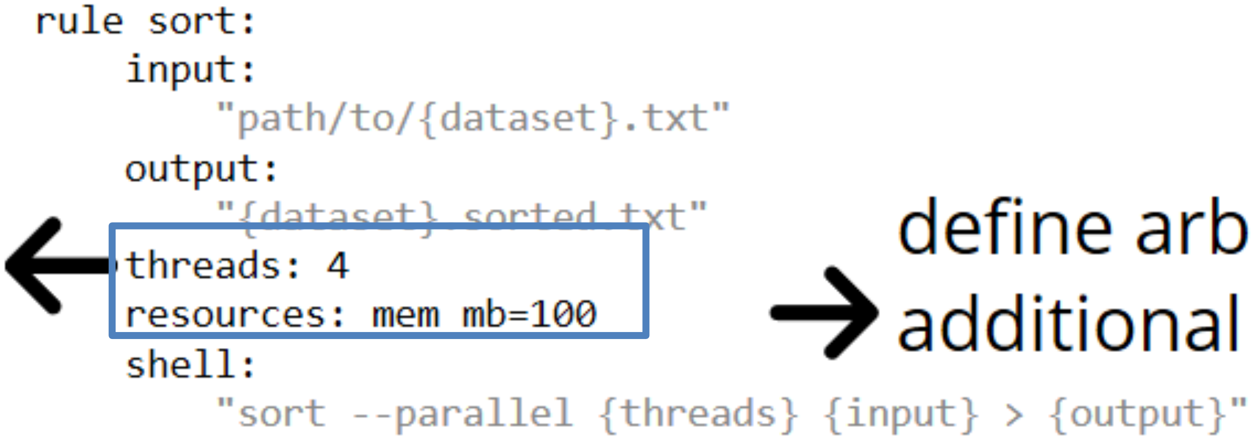
execute 8 jobs in parallel?

↑



```
# execute the workflow with 8 cores
snakemake --cores 8
```

Job 병렬화



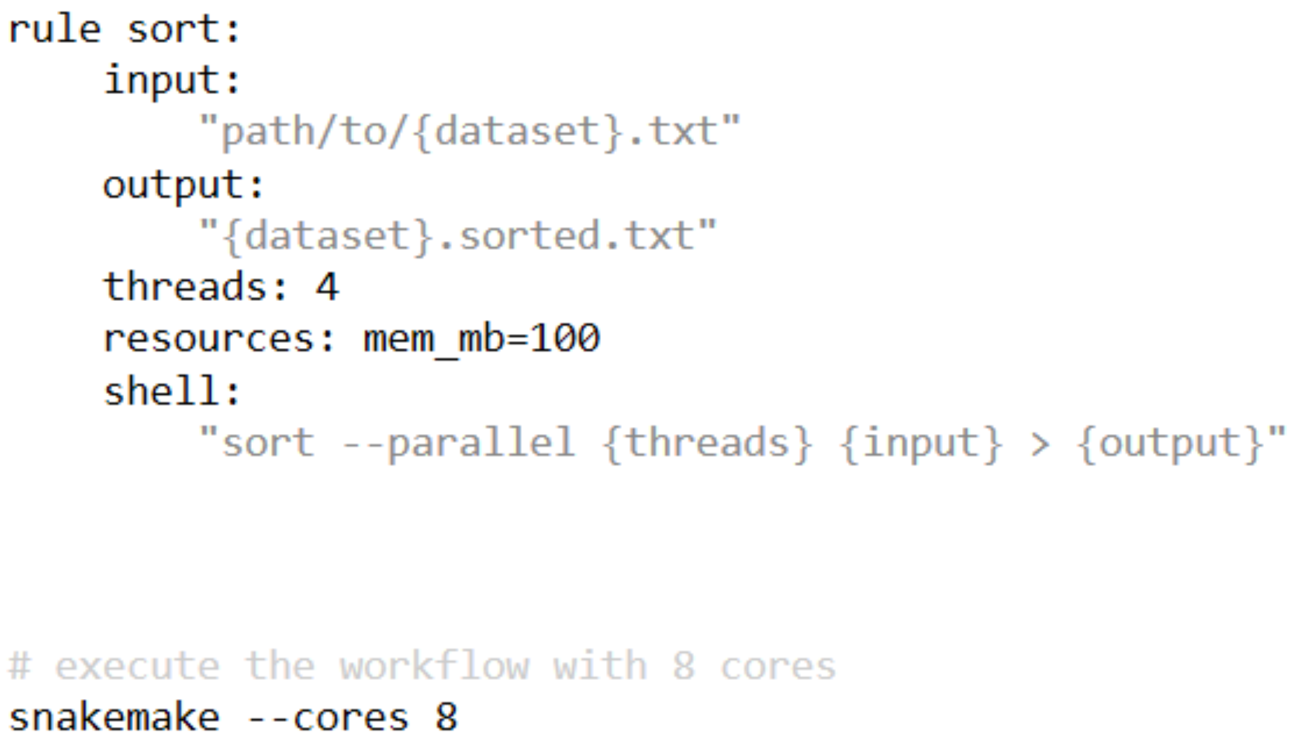
```
rule sort:
    input:
        "path/to/{dataset}.txt"
    output:
        "{dataset} sorted.txt"
    threads: 4
    resources: mem mb=100
    shell:
        "sort --parallel {threads} {input} > {output}"
```

define used threads

define arbitrary additional resources

refer to defined thread number

Job 병렬화



```
rule sort:
    input:
        "path/to/{dataset}.txt"
    output:
        "{dataset}.sorted.txt"
    threads: 4
    resources: mem_mb=100
    shell:
        "sort --parallel {threads} {input} > {output}"
```

```
# execute the workflow with 8 cores
snakemake --cores 8
```

↓

can execute 2 sort jobs in parallel

Defining resources

```
rule sort:
    input:
        "path/to/{dataset}.txt"
    output:
        "{dataset}.sorted.txt"
    threads: 4
    resources: mem_mb=100
    shell:
        "sort --parallel {threads} {input} > {output}"
```

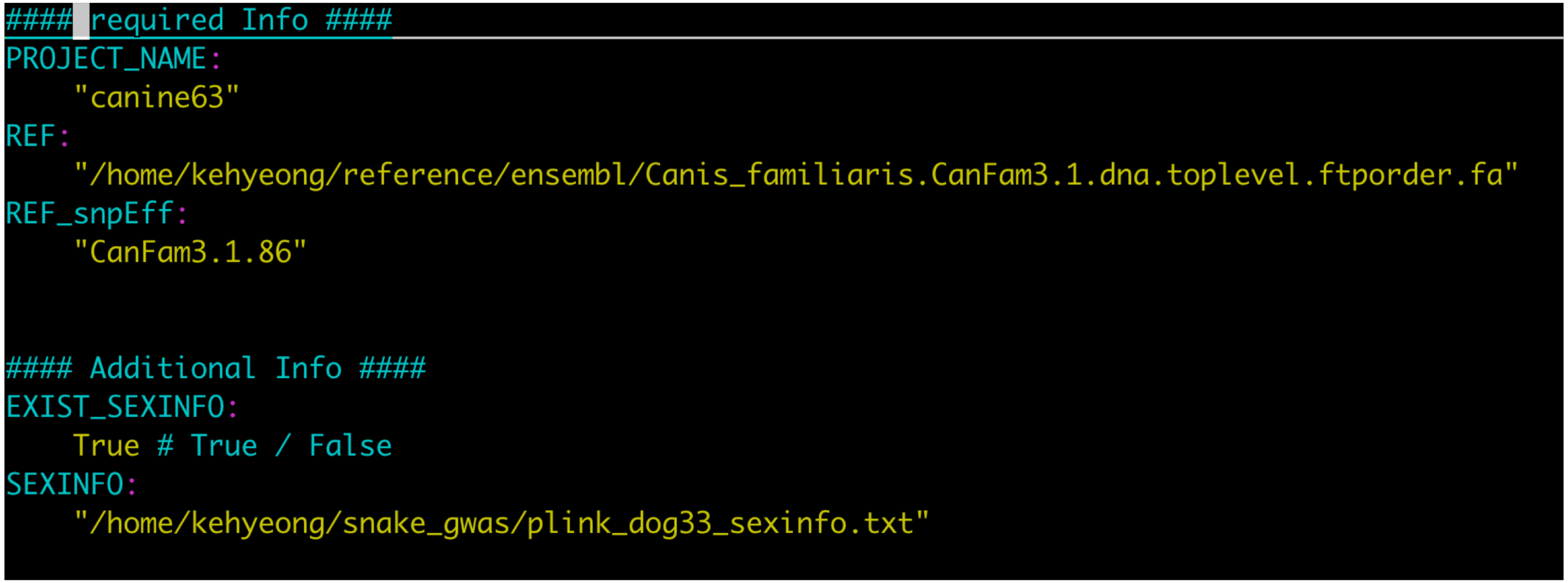
```
# execute the workflow with 10 cores
snakemake --cores 10
```

↓

can execute 2 sort jobs in parallel

Config file 사용

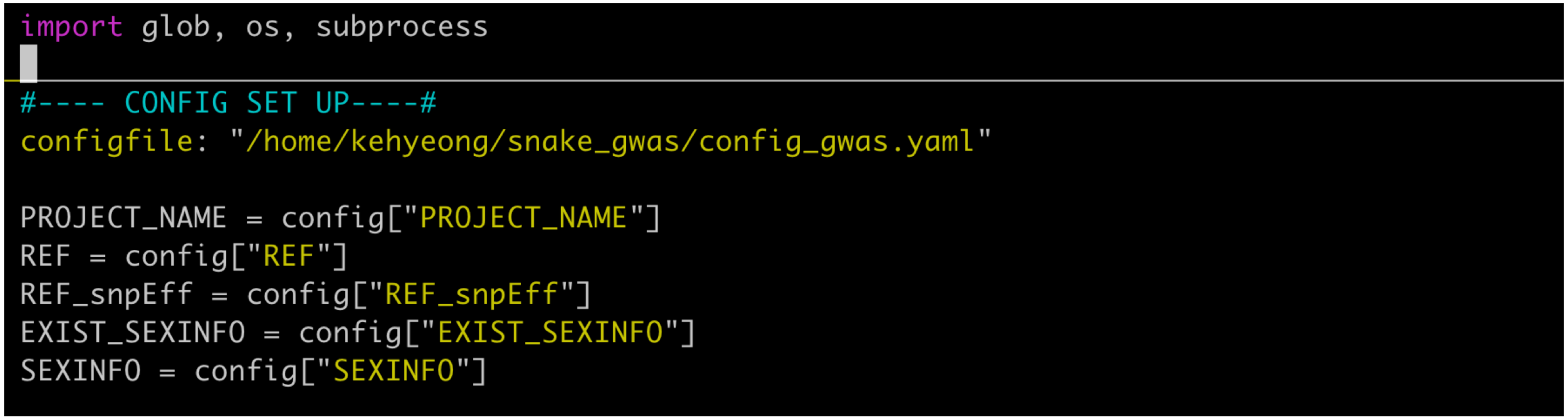
- ✓ config_gwas.yaml



```
#### required Info ####
PROJECT_NAME:
    "canine63"
REF:
    "/home/kehyeong/reference/ensembl/Canis_familiaris.CanFam3.1.dna.toplevel.ftporder.fa"
REF_snpEff:
    "CanFam3.1.86"


#### Additional Info ####
EXIST_SEXINFO:
    True # True / False
SEXINFO:
    "/home/kehyeong/snake_gwas/plink_dog33_sexinfo.txt"
```

- ✓ Snakefile

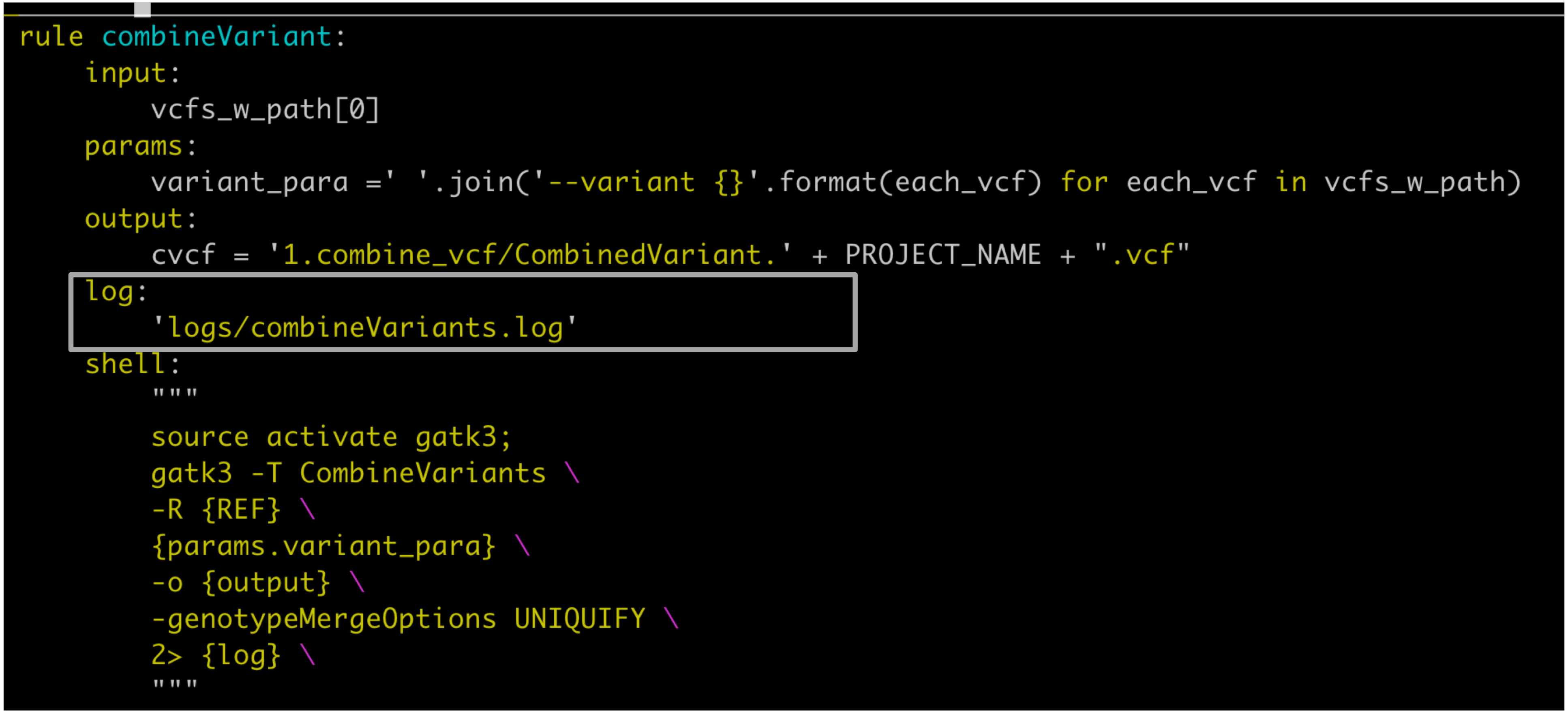


```
import glob, os, subprocess

#---- CONFIG SET UP----#
configfile: "/home/kehyeong/snake_gwas/config_gwas.yaml"

PROJECT_NAME = config["PROJECT_NAME"]
REF = config["REF"]
REF_snpEff = config["REF_snpEff"]
EXIST_SEXINFO = config["EXIST_SEXINFO"]
SEXINFO = config["SEXINFO"]
```

Logging



```
rule combineVariant:
    input:
        vcfs_w_path[0]
    params:
        variant_para =' '.join('--variant {}'.format(each_vcf) for each_vcf in vcfs_w_path)
    output:
        cvcf = '1.combine_vcf/CombinedVariant.' + PROJECT_NAME + ".vcf"
    log:
        'logs/combineVariants.log'
    shell:
        """
        source activate gatk3;
        gatk3 -T CombineVariants \
        -R {REF} \
        {params.variant_para} \
        -o {output} \
        -genotypeMergeOptions UNIQUIFY \
        2> {log} \
        """
```

External scripts

```
rule sort:
    input:
        a="path/to/{dataset}.txt"
    output:
        b="{dataset}.sorted.txt"
    script:
        "scripts/myscript.R"
```

R scripts:

@로 접근 가능

```
data <- read.table(snakemake@input[["a"]])
data <- data[order(data$id),]
write.table(data, file = snakemake@output[["b"]])
```

Python scripts:

```
import pandas as pd

def sort(infile, outfile):
    data = pd.read_table(infile)
    data = data.sort_values("id")
    data.to_csv(outfile, sep="\t")

if __name__ == "__main__":
    sort(snakemake.input.a, snakemake.output.b)
```

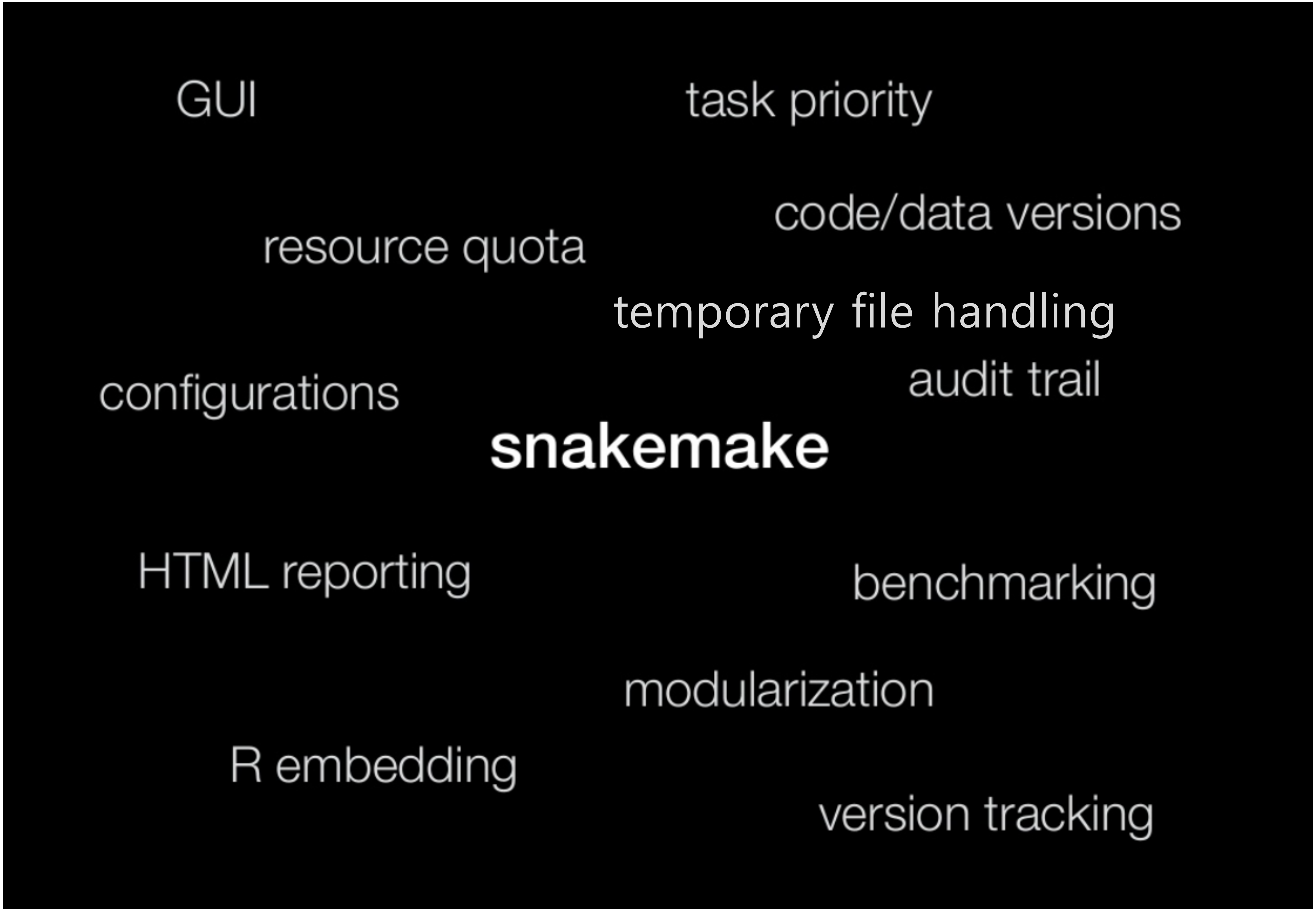
객체 처럼 접근

Include (pipeline 상속)

✓ Another Snakefile

```
include: “../snake_gwas/gwas.snakefile”


rule all:
        ‘another.rule.outfile.txt’
```



GUI

task priority

code/data versions

resource quota

temporary file handling

audit trail

configurations

snakemake

HTML reporting

benchmarking

modularization

R embedding

version tracking