

a

```

1 scattergather:
2     someprocess=8
3
4 rule scatter:
5     output:
6         scatter.someprocess("scattered/{scatteritem}.txt")
7
8 rule step2:
9     input:
10        "scattered/{scatteritem}.txt"
11    output:
12        "transformed/{scatteritem}.txt"
13
14 rule gather:
15     input:
16        gather.someprocess("transformed/{scatteritem}.txt")

```

b

```

1 rule step1:
2     output:
3         pipe("hello.txt")
4     shell:
5         "echo hello > {output}"
6
7 rule step2:
8     output:
9         pipe("world.txt")
10    shell:
11        "echo world > {output}"
12
13 rule step3:
14     input:
15        "hello.txt",
16        "world.txt"
17    output:
18        "hello-world.txt"
19    shell:
20        "cat {input} > {output}"

```

c

```

1 rule step:
2     input:
3         "data/{sample}.txt"
4     output:
5         "results/{sample}.txt"
6     params:
7         threshold=lambda w: config["threshold"][w.sample]
8     shell:
9         "some-tool -x {params.threshold} {input} > {output}"

```

d

```

1 pepfile: "pep/config.yaml"
2 pepschema: "schemas/pep.yaml"
3
4 rule all:
5     input:
6         expand(
7             "results/{sample}.somerresult.txt",
8             sample=pep.sample_table["sample_name"]
9         )

```

e

```

1 def get_results(wildcards):
2     with checkpoints.qc.get().output[0].open() as f:
3         qc = pd.read_csv(f, sep="\t")
4         return expand(
5             "results/processed/{sample}.txt",
6             sample=qc[qc["some-value"] > 90.0]["sample"]
7         )
8
9 rule all:
10    input:
11        get_results
12
13 checkpoint qc:
14    input:
15        expand("results/preprocessed/{sample}.txt", sample=samples)
16    output:
17        "results/qc.tsv"
18    shell:
19        "perform-qc {input} > {output}"
20
21 rule process:
22    input:
23        "results/preprocessed/{sample}.txt"
24    output:
25        "results/processed/{sample}.txt"
26    shell:
27        "process {input} > {output}"

```

f

```

1 rule step:
2     input:
3         "data/{sample}.txt"
4     output:
5         "results/{sample}.txt"
6     benchmark:
7         "benchmarks/some-tool/{sample}.txt"
8     shell:
9         "some-tool {input} > {output}"

```