a

```
scattergather:
    someprocess=8

rule scatter:
    output:
        scatter.someprocess("scattered/{scatteritem}.txt")

rule step2:
    input:
        "scattered/{scatteritem}.txt"
    output:
        "transformed/{scatteritem}.txt"

rule gather:
    input:
        gather.someprocess("transformed/{scatteritem}.txt")
```

b

```
rule step1:
    output:
        pipe("hello.txt")
    shell:
        "echo hello > {output}"

rule step2:
    output:
        pipe("world.txt")
    shell:
        "echo world > {output}"

rule step2:
    input:
        "hello.txt",
        "world.txt"
    output:
        "hello-world.txt"
    shell:
        "cat {input} > {output}"
```

c

```
pepfile: "pep/config.yaml"
pepschema: "schemas/pep.yaml"

rule all:
    input:
        expand(
            "results/{sample}.someresult.txt",
            sample=pep.sample_table["sample_name"]
        )
```

d

```
def get_results(wildcards):
    with checkpoints.qc.get().output[0].open() as f:
        qc = pd.read_csv(f, sep="\t")
        return expand(
            "results/processed/{sample}.txt",
            sample=qc[qc["some-value"] > 90.0]["sample"]
        )

rule all:
    input:
        get_results

checkpoint qc:
    input:
        expand("results/preprocessed/{sample}.txt", sample=samples)
    output:
        "results/qc.tsv"
    shell:
        "perfom-qc {input} > {output}"

rule process:
    input:
        "results/preprocessed/{sample}.txt"
    output:
       "results/processed/{sample}.txt"
    shell:
       "process {input} > {output}"
```