Build bioinformatics pipelines with Snakemake

2016.09.20 김가경

Snakemake References

1. 공식 documents
   https://bitbucket.org/snakemake/snakemake/wiki/Documentation

2. Johannes Köster and Sven Rahmann. *Snakemake--a scalable bioinformatics workflow engine.* Bioinformatics 2012, 28:2520-2522.Pubmed | PMC | Journal
3. Simple examples https://slowkow.com/notes/snakemake-tutorial/

4. 이인구님이 정리한 tutorial https://github.com/dlsrnsi/snakemake_tutorial

5. 고성능 병렬서버에서 snakemake 사용법
   https://hpc.nih.gov/apps/snakemake.html
6. [BioPython 스터디 특강] Snakemake와 Joblib 소개
7. 장혜린 학생 관련 발표자료

Basic rules

rule NAME:
  input: "path/to/inputfile", "path/to/other/inputfile"
   output: "path/to/outputfile", "path/to/another/outputfile"
  params: param1=param1, pram2=param2
  shell: "command --option1 {params.param1} --option2 {params.param2} {input} {output}"

shell 대신 python3 script를 실행할때는 “run:” 을 사용

Command line options

Option	Effect
-s FILE	실행시키는 파일 specify
-n	dryrun: execution plan만 확인
-p	print all shell commands
-F	input file 존재 여부와 상관없이 모든 rule 실행
-j [N]	사용 가능한 최대 core개수 설정 (default=1)
--config [KEY=VALUE [KEY=VALUE ...]]	workflow 내에 config 값 변환 가능
-d DIR	working directory 설정
-c CMD	주어진 submit command 규칙을 따라 실행한다 (예: qsub)

Wild cards

```
$ snakemake -s EXAMPLE mapped_reads/A.bam
$ snakemake -s EXAMPLE mapped_reads/{A,B}.bam
```

Exercise following reference #3 (1)

Snakefile

```
SAMPLES = ['3C-629-2-1', '3C-629-2-2']
rule all:
    input:
        expand('{sample}.txt', sample=SAMPLES)

rule quantify_genes:
    input:
        genome = 'genome.fa',
        r1 = 'fastq/{sample}_R1.fastq.gz',
        r2 = 'fastq/{sample}_R2.fastq.gz'
    output:
        '{sample}.txt'
    shell:
        'echo {input.genome} {input.r1} {input.r2} > {output}'
```

Exercise following reference #3 (2)

```
[kkkim@comet-ln2 snakemake-example]$ snakemake
Provided cores: 1
Rules claiming more threads will be scaled down.
Job counts:
        count   jobs
        1       all
        2       quantify_genes
        3
rule quantify_genes:
        input: genome.fa, fastq/3C-629-2-2_R2.fastq.gz, fastq/3C-629-2-2_R1.fastq.gz
        output: 3C-629-2-2.txt
        wildcards: sample=3C-629-2-2
1 of 3 steps (33%) done
rule quantify_genes:
        input: genome.fa, fastq/3C-629-2-1_R2.fastq.gz, fastq/3C-629-2-1_R1.fastq.gz
        output: 3C-629-2-1.txt
        wildcards: sample=3C-629-2-1
2 of 3 steps (67%) done
localrule all:
        input: 3C-629-2-1.txt, 3C-629-2-2.txt
3 of 3 steps (100%) done
[kkkim@comet-ln2 snakemake-example]$ cat 3C-629-2-1.txt
genome.fa fastq/3C-629-2-1_R1.fastq.gz fastq/3C-629-2-1_R2.fastq.gz
```

Exercise following reference #3 (3) - Not working~! ㅠㅠ

$ snakemake --forceall --dag | dot -Tpng > dag1.png

Format: "png" not recognized. Use one of:

Exception ignored in: <_io.TextIOWrapper name='<stdout>' mode='w' encoding='ISO-8859-15'>

BrokenPipeError: [Errno 32] Broken pipe

$ snakemake –dag > dag1.dot

dag1.dot

```
digraph snakemake_dag {
    graph[bgcolor=white, margin=0];
    node[shape=box, style=rounded, fontname=sans, fontsize=10, penwidth=2];
    edge[penwidth=2, color=grey];
    0[label = "all", color = "0.00 0.6 0.85", style="rounded,dashed"];
    1[label = "quantify_genes\nsample: 3C-629-2-1", color = "0.33 0.6 0.85", style="rounded,dashed"];
    2[label = "quantify_genes\nsample: 3C-629-2-2", color = "0.33 0.6 0.85", style="rounded,dashed"];
    2 -> 0
    1 -> 0
}
```

RNA-seq analysis

Expression (Gene, Transcript)

Alternative splicing

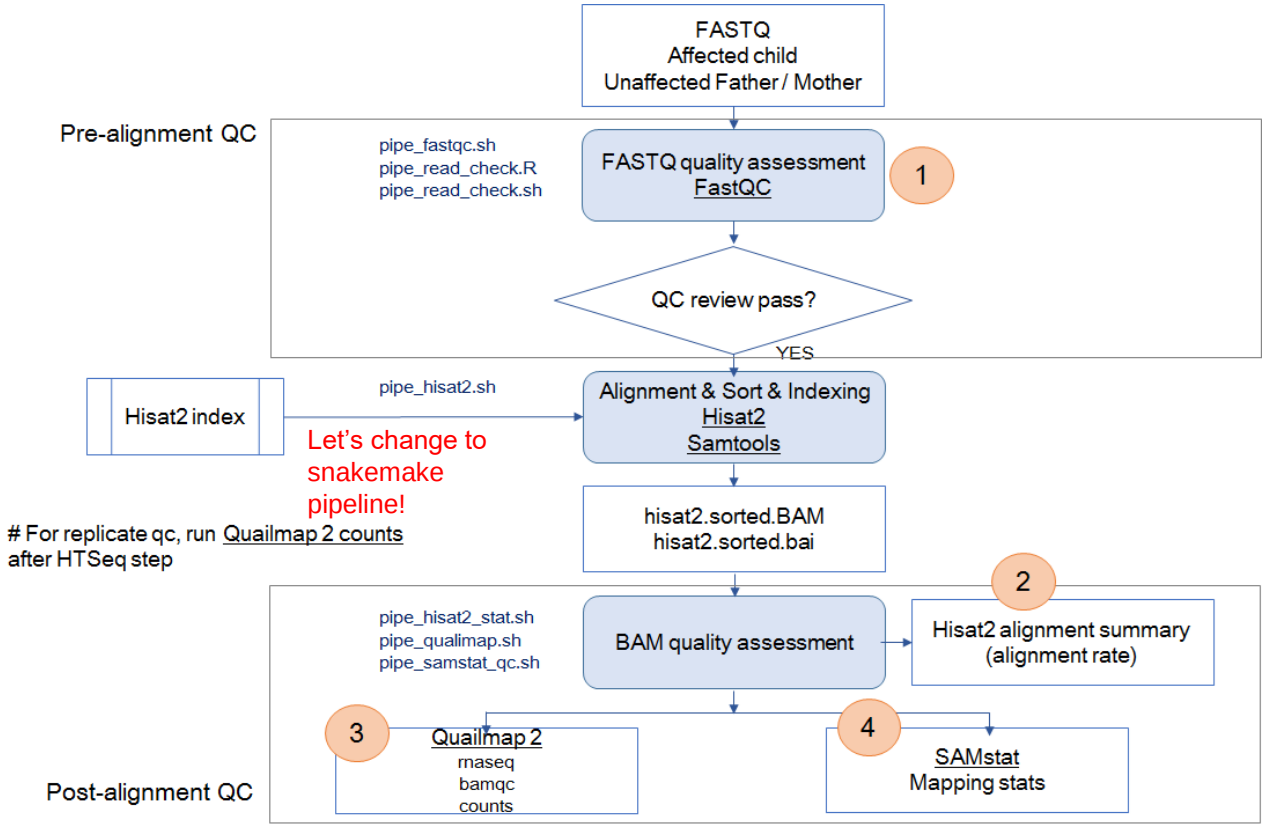
Fusion gene

Variant calling (SNPs, Indels)

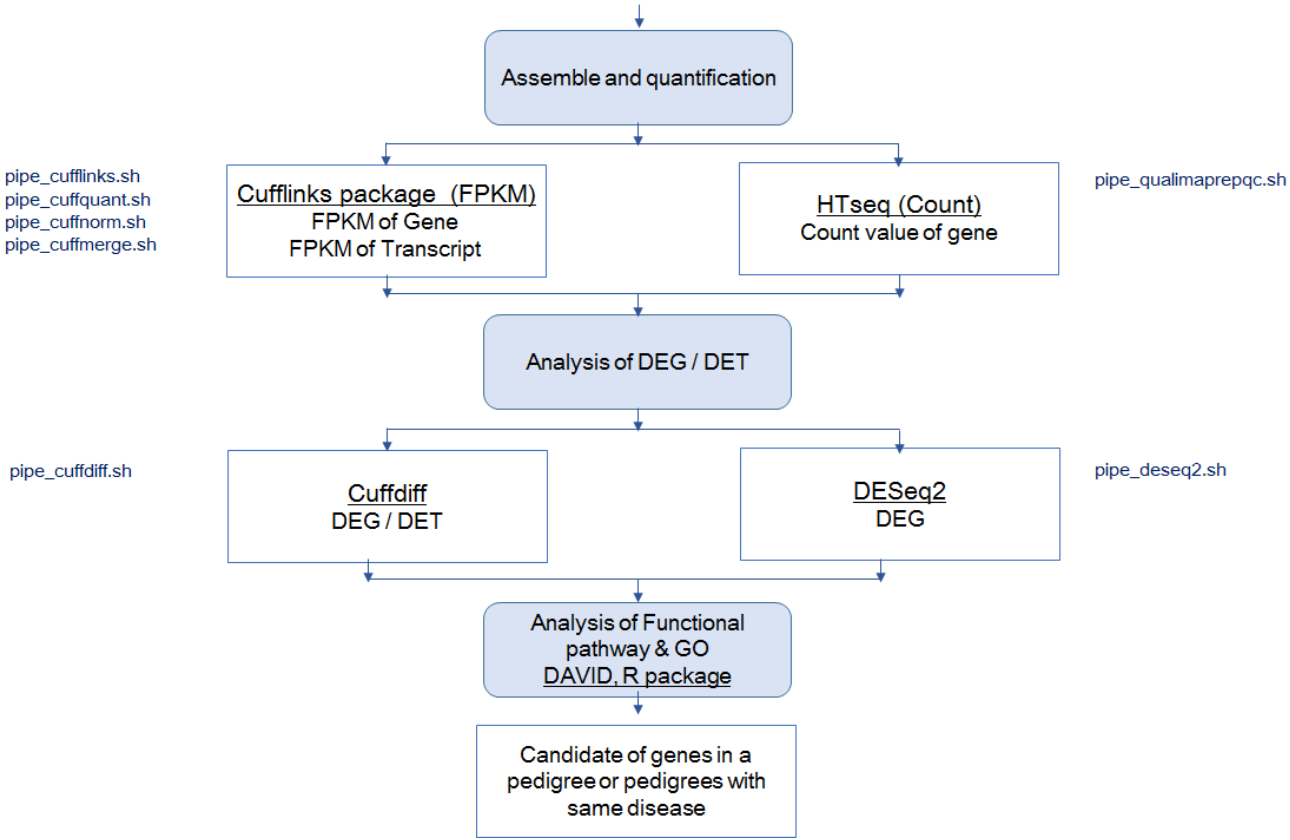
Allele specific expression

..etc..

RNA-seq expression pipeline (1)



RNA-seq expression pipeline (2)



hisat2 alignment pipeline (1)

```
WDIR  = "/home/kkkim/project/201604_RNAseq/"
workdir: WDIR
 rule hisat2:
      input: fwd="1.raw/{sample}_1.fastq.gz", rev="1.raw/{sample}_2.fastq.gz"
      output: "3.aligned/hisat2/{sample}.sam"
      threads: 32
      params: GFF=GFF, Hisat2Index=Hisat2Index
      shell: """
              mkdir -p 3.aligned/hisat2
              hisat2 -t -p {threads} --dta-cufflinks --rna-strandness RF \
              --met-file 3.aligned/hisat2/{wildcards.sample}.metrics \
              -x {params.Hisat2Index}  -1 {input.fwd} -2 {input.rev} \
              -S 3.aligned/hisat2/{wildcards.sample}.sam \
             2> 3.aligned/hisat2/{wildcards.sample}.stat
             """
```

hisat2 alignment pipeline (2)

```
rule sam2bam:
    input: "3.aligned/hisat2/{sample}.sam"
    output: "3.aligned/hisat2/{sample}.bam"
     threads: 32
     shell: samtools view -b -h -S -@ {threads} -O bam -o {output} {input}

 rule sort_bam:
     input: "3.aligned/hisat2/{sample}.sam"
     output: "3.aligned/hisat2/{sample}.sorted.bam"
     threads: 32
     shell: "samtools sort -@ {threads} -O bam -o {output} -T {output}.tmp {input}"
```

hisat2 alignment pipeline (3)

```
rule index_bam:
   input: "3.aligned/hisat2/{sample}.sorted.bam"
   output: "3.aligned/hisat2/{sample}.sorted.bam.bai"
    shell: """
           samtools sort -@ 32 -O bam -o {output} -T {input}.tmp
           samtools index 3.aligned/hisat2/{sample}.sorted.bam
           rm 3.aligned/hisat2/{sample}.sam
           rm 3.aligned/hisat2/{sample}.bam
   """
```

장점

input, output 파일 형태를 파악하기 용이
전체 workflow의 visualization 가능
rule의 재활용이 용이
rule 속에 shell, python3, R 을 필요에 따라 직접 사용 가능
이미 input 파일이 만들어져 있는 경우, 다시 만들지 않고 그 파일 그대로 사용
단, input 파일이 output 파일보다 만들어진 시간이 나중일 경우에는
    다시 rule을 execute: update된 input file이 있으면 그 단계부터 전개
project file을 configuration파일에서 설정: 최대한 pipeline자체를 변경하는 일이 없도록 함
configuration 파일을 변경하는 것 만으로 personalization 가능
final output을 만들기 위한 파일만 생성 :여러 개의 pipeline을 통합하여 운영가능
cluster environment에서 운영되도록 설계됨: 자동으로 어떤 rule들을 같이 실행시킬 수 있는지 파악, 다수의 core에서 동시에 실행가능

출처 : 장혜린 학생 발표자료