

AWS HealthOmics? Nf-core ????? ?????? ?? (rnaseq)

📄 📄 : <https://catalog.us-east-1.prod.workshops.aws/workshops/76d4a4ff-fe6f-436a-a1c2-f7ce44bc5d17/en-US>

📄 📄 📄 📄 📄 📄 . 📄 📄 📄 .

???? ??

nf-core repository??? ????? ??

```
cd ~  
git clone https://github.com/nf-core/rnaseq --branch 3.14.0 --single-branch
```

Docker Image Manifest? ??

```
cp ~/amazon-ecr-helper-for-aws-healthomics/lib/lambda/parse-image-uri/public_registry_properties.json  
namespace.config
```

```
inspect_nf.py 📄 📄 .
```

```
python3 amazon-omics-tutorials/utils/scripts/inspect_nf.py \  
--output-manifest-file rnaseq_3140_docker_images_manifest.json \  
-n namespace.config \  
--output-config-file omics.config \  
--region <region> \  
~/rnaseq/
```

📄 📄 📄 rnaseq_3140_docker_images_manifest.json 📄 omics.config 📄 .

```
rnaseq_3140_docker_images_manifest.json 📄 📄 📄 📄 📄 :
```

```
{
  "manifest": [
    "quay.io/biocontainers/bbmap:39.01--h5c4e2a8_0",
    "quay.io/biocontainers/bedtools:2.30.0--hc088bd4_0",
    "quay.io/biocontainers/bioconductor-dupradar:1.28.0--r42hdfd78af_0",
    "quay.io/biocontainers/bioconductor-summarizedexperiment:1.24.0--r41hdfd78af_0",
    "quay.io/biocontainers/bioconductor-tximeta:1.12.0--r41hdfd78af_0",
    "quay.io/biocontainers/fastp:0.23.4--h5f740d0_0",
    "quay.io/biocontainers/fastqc:0.12.1--hdfd78af_0",
    "quay.io/biocontainers/fq:0.9.1--h9ee0642_0",
    "quay.io/biocontainers/gffread:0.12.1--h8b12597_0",
    "quay.io/biocontainers/hisat2:2.2.1--h1b792b2_3",
    "quay.io/biocontainers/kallisto:0.48.0--h15996b6_2",
    "quay.io/biocontainers/mulled-v2-
1fa26d1ce03c295fe2fdcf85831a92fbcdb7e8c2:1df389393721fc66f3fd8778ad938ac711951107-0",
    "quay.io/biocontainers/mulled-v2-
1fa26d1ce03c295fe2fdcf85831a92fbcdb7e8c2:59cdd445419f14abac76b31dd0d71217994cbcc9-0",
    "quay.io/biocontainers/mulled-v2-
8849acf39a43cdd6c839a369a74c0adc823e2f91:ab110436faf952a33575c64dd74615a84011450b-0",
    "quay.io/biocontainers/mulled-v2-
a97e90b3b802d1da3d6958e0867610c718cb5eb1:2cdf6bf1e92acbeb9b2834b1c58754167173a410-0",
    "quay.io/biocontainers/mulled-v2-
cf0123ef83b3c38c13e3b0696a3f285d3f20f15b:64aad4a4e144878400649e71f42105311be7ed87-0",
    "quay.io/biocontainers/multiqc:1.19--pyhdfd78af_0",
    "quay.io/biocontainers/perl:5.26.2",
    "quay.io/biocontainers/picard:3.0.0--hdfd78af_1",
    "quay.io/biocontainers/preseq:3.1.2--h445547b_2",
    "quay.io/biocontainers/python:3.9--1",
    "quay.io/biocontainers/qualimap:2.3--hdfd78af_0",
    "quay.io/biocontainers/rseqc:5.0.3--py39hf95cd2a_0",
    "quay.io/biocontainers/salmon:1.10.1--h7e5ed60_0",
    "quay.io/biocontainers/samtools:1.16.1--h6899075_1",
    "quay.io/biocontainers/samtools:1.17--h00cdaf9_0",
    "quay.io/biocontainers/sortmerna:4.3.4--h9ee0642_0",
    "quay.io/biocontainers/stringtie:2.2.1--hecb563c_2",
    "quay.io/biocontainers/subread:2.0.1--hed695b0_0",
    "quay.io/biocontainers/trim-galore:0.6.7--hdfd78af_0",
    "quay.io/biocontainers/ucsc-bedclip:377--h0b8a92a_2",
    "quay.io/biocontainers/ucsc-bedgraphtobigwig:445--h954228d_0",
    "quay.io/biocontainers/umi_tools:1.1.4--py38hbff2b2d_1",
```

```
"quay.io/nf-core/ubuntu:20.04"
]
}
```

???? ??

```
aws stepfunctions start-execution \
--state-machine-arn arn:aws:states:<region>:<your-account-id>:stateMachine:omx-container-puller \
--input file://rnaseq_3140_docker_images_manifest.json
```

nf-core project ?? ????

```
mv omics.config rnaseq/conf
```

```
echo "includeConfig 'conf/omics.config'" >> rnaseq/nextflow.config
```

AWS HealthOmics ?????? ??

??1. AWS HealthOmics ???? ??

paramter-description.json 1 11 11 1 1111 .

1 11111 11 11111 111 1 1111 . 11 1111 111 111 SCRNA-Seq nf-core 11111 11111 . 1 1111 111 11 11 1 111111 11 11111 1111 111 .

NF-Core 11111 11 11111 1111 1111 11111 (1 : RNA-Seq 11111 11111 11).

```

{
  "input": {
    "description": "S3 URI to samplesheet.csv. Rows therein point to S3 URIs for fastq data",
    "optional": false
  },
  "genome": {
    "description": "Name of iGenomes reference. - e.g. GRCh38",
    "optional": true
  },
  "igenomes_base": {
    "description": "URI base for iGenomes references. (e.g. s3://ngi-igenomes/igenomes/)",
    "optional": true
  },
  "fasta": {
    "description": "Path to FASTA genome file. This parameter is mandatory if --genome is not specified. If you
don't have the appropriate alignment index available this will be generated for you automatically.",
    "optional": true
  },
  "gtf": {
    "description": "Path to GTF annotation file. This parameter is mandatory if --genome is not specified.",
    "optional": true
  },
  "gff": {
    "description": "Path to GFF3 annotation file. This parameter must be specified if `--genome` or `--gtf` are
not specified.",
    "optional": true
  },
  "pseudo_aligner": {
    "description": "Specifies the pseudo aligner to use - available options are 'salmon'. Runs in addition to `--
aligner`,
    "optional": true
  },
  "transcript_fasta": {
    "description": "Path to FASTA transcriptome file.",
    "optional": true
  },
  "additional_fasta": {
    "description": "FASTA file to concatenate to genome FASTA file e.g. containing spike-in sequences.",
    "optional": true
  },

```

```

"bbsplit_fasta_list": {
  "description": "Path to comma-separated file containing a list of reference genomes to filter reads against
with BBSplit. You have to also explicitly set `--skip_bbsplit` false if you want to use BBSplit.",
  "optional": true
},
"hisat2_index": {
  "description": "Path to directory or tar.gz archive for pre-built HISAT2 index.",
  "optional": true
},
"salmon_index": {
  "description": "Path to directory or tar.gz archive for pre-built Salmon index.",
  "optional": true
},
"rsem_index": {
  "description": "Path to directory or tar.gz archive for pre-built RSEM index",
  "optional": true
},
"skip_bbsplit": {
  "description": "Skip BBSplit for removal of non-reference genome reads.",
  "optional": true
},
"pseudo_aligner": {
  "description": "Specifies the pseudo aligner to use - available options are 'salmon'. Runs in addition to '--
aligner'.",
  "optional": true
},
"umitools_bc_pattern": {
  "description": "The UMI barcode pattern to use e.g. 'NNNNNN' indicates that the first 6 nucleotides of the
read are from the UMI.",
  "optional": true
}
}

```

??2. ????? ?????

```
zip -r rnaseq-workflow.zip rnaseq
```

```
aws s3 cp rnaseq-workflow.zip s3://<yourbucket>/workshop/rnaseq-workflow.zip
```

```
aws omics create-workflow \
```

```
--name rnaseq-v3140 \  
--definition-uri s3://<yourbucket>//workshop/rnaseq-workflow.zip \  
--parameter-template file://parameter-description.json \  
--engine NEXTFLOW
```

??3. ????? ?? ??

```
aws omics list-workflows --name rnaseq-v3140
```

????? ??????

???? ??

samplesheet_full

?? ? ???? ?_? ???? ? ? ???? .

???? ? ???? ?

nf-core/rnaseq repository ? ? `igenomes.config` ? ???? ? GRCh37 ? ? ?
? healthomics ? ? ? ? ? ? ? ? .

```
aws s3 sync s3://ngi-igenomes/igenomes/Homo_sapiens/  
s3://{mybucketname}/workshop/igenomes/Homo_sapiens/
```

???? ? ? ? ?

```
wget https://raw.githubusercontent.com/nf-core/test-datasets/rnaseq/samplesheet/v3.10/samplesheet_full.csv
```

`parameter-description.json` ? ? ? ? ? ? `input_rnaseq.json` ? ? ? ? . ?
???? ? ? ? S3 ? ? ? ? ? .

`samplesheet_full.csv` ? ? ? ? . ? HealthOmics ? ? ? ? s3 bucket ? FASTQ
???? ? ? ? . (aws s3 sync ? aws s3 cp ? ? ? ? ?)

?)

```
aws s3 sync s3://ngi-igenomes/test-data/rnaseq/ s3://{mybucketname}/workshop/
```

samplesheet_full.csv

```
sample,fastq_1,fastq_2,strandedness
GM12878_REP1,s3://{mybucketname}/test-data/rnaseq/SRX1603629_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX1603629_T1_2.fastq.gz,reverse
GM12878_REP2,s3://{mybucketname}/test-data/rnaseq/SRX1603630_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX1603630_T1_2.fastq.gz,reverse
K562_REP1,s3://{mybucketname}/test-data/rnaseq/SRX1603392_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX1603392_T1_2.fastq.gz,reverse
K562_REP2,s3://{mybucketname}/test-data/rnaseq/SRX1603393_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX1603393_T1_2.fastq.gz,reverse
MCF7_REP1,s3://{mybucketname}/test-data/rnaseq/SRX2370490_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX2370490_T1_2.fastq.gz,reverse
MCF7_REP2,s3://{mybucketname}/test-data/rnaseq/SRX2370491_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX2370491_T1_2.fastq.gz,reverse
H1_REP1,s3://{mybucketname}/test-data/rnaseq/SRX2370468_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX2370468_T1_2.fastq.gz,reverse
H1_REP2,s3://{mybucketname}/test-data/rnaseq/SRX2370469_T1_1.fastq.gz,s3://{mybucketname}/test-
data/rnaseq/SRX2370469_T1_2.fastq.gz,reverse
```

samplesheet

```
aws s3 mv samplesheet_full.csv s3://{mybucket}/workshop/
```

input_rnaseq_full.json

```
{
  "input": "s3://{mybucket}/workshop/samplesheet_full.csv",
  "genome": "GRCh37",
  "igenomes_base": "s3://{mybucket}/workshop/igenomes/",
  "pseudo_aligner": "salmon"
}
```

samplesheet_test

samplesheet_test

samplesheet_full.csv

```
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/genome.fasta
```

```
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/genes_with_empty_tid.gtf.gz
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/genes.gff.gz
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/transcriptome.fasta
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/gfp.fa.gz
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/bbsplit_fasta_list.txt
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/hisat2.tar.gz
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/salmon.tar.gz
wget https://raw.githubusercontent.com/nf-core/test-
datasets/7f1614baeb0ddf66e60be78c3d9fa55440465ac8/reference/rsem.tar.gz
```

|||| || |||| s3 bucket || |||

```
aws s3 sync . s3://{mybucket}/workshop/reference/
```

|||| || ||| ||||

```
wget https://raw.githubusercontent.com/nf-core/test-datasets/rnaseq/samplesheet/v3.10/samplesheet_test.csv
```

samplesheet_test.csv ||

```
sample,fastq_1,fastq_2,strandedness
WT_REP1,s3://{mybucketname}/workshop/test_fastq/SRR6357070_1.fastq.gz,s3://{mybucketname}/workshop/t
est_fastq/SRR6357070_2.fastq.gz,auto
WT_REP1,s3://{mybucketname}/workshop/test_fastq/SRR6357071_1.fastq.gz,s3://{mybucketname}/workshop/t
est_fastq/SRR6357071_2.fastq.gz,auto
WT_REP2,s3://{mybucketname}/workshop/test_fastq/SRR6357072_1.fastq.gz,s3://{mybucketname}/workshop/t
est_fastq/SRR6357072_2.fastq.gz,reverse
RAP1_UNINDUCED_REP1,s3://{mybucketname}/workshop/test_fastq/SRR6357073_1.fastq.gz,,reverse
RAP1_UNINDUCED_REP2,s3://{mybucketname}/workshop/test_fastq/SRR6357074_1.fastq.gz,,reverse
RAP1_UNINDUCED_REP2,s3://{mybucketname}/workshop/test_fastq/SRR6357075_1.fastq.gz,,reverse
RAP1_IAA_30M_REP1,s3://{mybucketname}/workshop/test_fastq/SRR6357076_1.fastq.gz,s3://{mybucketname}/
workshop/test_fastq/SRR6357076_2.fastq.gz,reverse
```

|||| **samplesheet** || |||| ||


```
aws s3 mv samplesheet_test.csv s3://{mybucket}/workshop/
```

```
{
  "input": "s3://{mybucketname}/workshop/samplesheet_test.csv",
  "fasta": "s3://{mybucketname}/workshop/reference/genome.fasta",
  "gtf": "s3://{mybucketname}/workshop/reference/genes_with_empty_tid.gtf.gz",
  "gff": "s3://{mybucketname}/workshop/reference/genes.gff.gz",
  "transcript_fasta": "s3://{mybucketname}/workshop/reference/transcriptome.fasta",
  "additional_fasta": "s3://{mybucketname}/workshop/reference/gfp.fa.gz",
  "bbsplit_fasta_list": "s3://{mybucketname}/workshop/reference/bbsplit_fasta_list.txt",
  "hisat2_index": "s3://{mybucketname}/workshop/reference/hisat2.tar.gz",
  "salmon_index": "s3://{mybucketname}/workshop/reference/salmon.tar.gz",
  "rsem_index": "s3://{mybucketname}/workshop/reference/rsem.tar.gz",
  "skip_bbsplit"      : false,
  "pseudo_aligner"    : "salmon",
  "umitools_bc_pattern" : "NNNN"
}
```

```

bbsplit_fasta_list.txt  s3 bucket

```

s3://{mybucket}/workshop/reference/bbsplit_fasta_list.txt

```
sarscov2,s3://{mybucket}/workshop/reference/GCA_009858895.3_ASM985889v3_genomic.200409.fna
human,s3://{mybucket}/workshop/reference/chr22_23800000-23980000.fa
```

S3 policy ?? ???? ?

1. 1990년대 초반부터 시작된 '신용보증기금'의 설립은 중소기업의 자금 조달을 지원하기 위한 정책의 일환이었다.

s3:GetObject, s3:ListBucket 

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject"
    ],
    "Resource": [
      "arn:aws:s3:::*/",
      "arn:aws:s3:::*/",
      "arn:aws:s3:::ngi-igenomes/*",
      "arn:aws:s3:::brandon-us-east-1-gym/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::",
      "arn:aws:s3:::",
      "arn:aws:s3:::ngi-igenomes",
      "arn:aws:s3:::brandon-us-east-1-gym"
    ]
  }
],

```

s3:PutObject

```

25     },
26     {
27       "Effect": "Allow",
28       "Action": [
29         "s3:PutObject"
30       ],
31       "Resource": [
32         "arn:aws:s3:::*/",
33         "arn:aws:s3:::brandon-us-east-1-gym/*"
34       ]
35     },

```

Policy ??

Prepare IAM service role to run AWS HealthOmics workflow

omics_workflow_policy.json

```

#
export yourbucket="your-bucket-name"
export your_account_id="your-account-id"
export region="your-region"

# JSON
cat << EOF > omics_workflow_policy.json
{
  "Version": "2012-10-17",
  "Statement": [

```

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject"
  ],
  "Resource": [
    "arn:aws:s3:::${yourbucket}/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket"
  ],
  "Resource": [
    "arn:aws:s3:::${yourbucket}"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::${yourbucket}/*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "logs:DescribeLogStreams",
    "logs:CreateLogStream",
    "logs:PutLogEvents"
  ],
  "Resource": [
    "arn:aws:logs:${region}:${your_account_id}:log-group:/aws/omics/WorkflowLog:log-stream:*"
  ]
},
{
  "Effect": "Allow",
```

```

    "Action": [
        "logs:CreateLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:${region}:${your_account_id}:log-group:/aws/omics/WorkflowLog:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchCheckLayerAvailability"
    ],
    "Resource": [
        "arn:aws:ecr:${region}:${your_account_id}:repository/*"
    ]
}
]
}
EOF

echo "omics_workflow_policy.json" > omics_workflow_policy.json

```

trust_policy.json

```

# Set variables
export your_account_id="your-account-id"
export region="your-region" # Default: us-east-1

# JSON content
cat << EOF > trust_policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "omics.amazonaws.com"
            },

```

```

    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "${your_account_id}"
      },
      "ArnLike": {
        "aws:SourceArn": "arn:aws:omics:${region}:${your_account_id}:run/*"
      }
    }
  }
}
}
EOF

echo "trust_policy.json 信任策略文档"

```

IAM Role ??

```
aws iam create-role --role-name omics_start_run_role_v1 --assume-role-policy-document file://trust_policy.json
```

Policy document 信任策略文档

```
aws iam put-role-policy --role-name omics_start_run_role_v1 --policy-name OmicsWorkflowV1 --policy-document file://omics_workflow_policy.json
```

????? ??

```
--storage-type 静态存储桶 静态存储桶 STATIC 存储桶 . (桶名)
```

```

桶名 桶名 (DYNAMIC) 桶名 桶名 (STATIC) 桶名 桶名 /桶名 桶名 桶名
桶名 . 桶名 桶名 桶名 桶名 桶名 桶名 /桶名 桶名 桶名 桶名
桶名 . 桶名 桶名 桶名 桶名 桶名 桶名 桶名 DYNAMIC 桶名 桶名 桶名
桶名 桶名 STATIC 桶名 桶名 桶名 桶名 .

```

```
aws omics start-run \  
  --name rnaseq_workshop_run_1 \  
  --role-arn arn:aws:iam::<your-account-id>:role/omics_start_run_role_v1 \  
  --workflow-id <your-workflow-id> \  
  --parameters file://input_rnaseq.json \  
  --output-uri s3://<yourbucket>/output/ \  
  --storage-type DYNAMIC
```

```
aws omics start-run \  
  --name rnaseq_workshop_run_1 \  
  --role-arn arn:aws:iam::<your-account-id>:role/omics_start_run_role_v1 \  
  --workflow-id <your-workflow-id> \  
  --parameters file://input_rnaseq.json \  
  --output-uri s3://<yourbucket>/output/ \  
  --storage-type STATIC \  
  --storage-capacity 2048
```

test ☐ ☐ ☐

```
aws omics start-run \  
  --name rnaseq_workshop_test_run_1 \  
  --role-arn arn:aws:iam::<your-account-id>:role/omics_start_run_role_v1 \  
  --workflow-id <your-workflow-id> \  
  --parameters file://input_rnaseq_test.json \  
  --output-uri s3://<yourbucket>/output/
```

☐ ☐ ☐ ☐

- <https://github.com/aws-samples/amazon-omics-tutorials/tree/main/example-workflows/nf-core/workflows/rnaseq>

Revision #15

Created 19 August 2024 08:09:03 by Hyunmin Kim

Updated 23 August 2024 15:41:58 by Hyunmin Kim