Microbiome omics data analysis pipeline

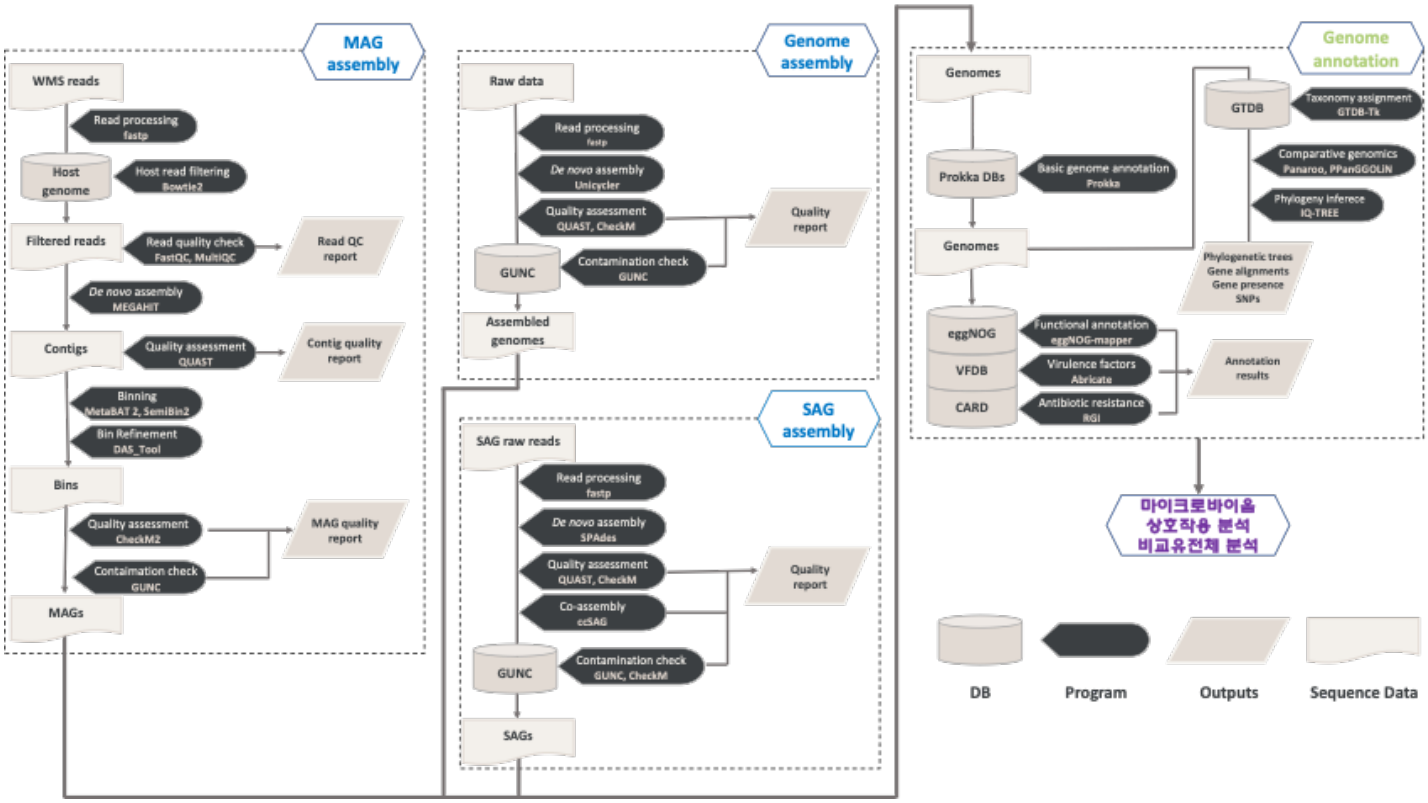
목차

1. 파이프라인 소개

본 파이프라인은 미생물 유전체 분석 파이프라인으로 Microbial isolate genome, single amplified genome (SAG), metagenome 데이터 분석을 지원합니다. 이 파이프라인은 사용자가 개별 프로그램 사용에 익숙하지 않아도 분석을 진행할 수 있도록 다수의 데이터에 최적화된 파라미터 값들을 설정 해놓았으며, 최소한의 명령어 입력을 통해 결과를 얻을 수 있도록 자동화하였습니다. Raw sequence data와 분석하고자 하는 파이프라인을 지정하여 명령어를 입력하면, 자동화된 일련의 분석을 통해 결과를 얻을 수 있습니다. Raw sequence data부터 분석이 필요 없을 경우, genome sequence를 넣은 후 annotation 단계부터 분석이 가능합니다. 또한 modular approach 를 통해 7단계로 구성하였으며, 목

차 5의 빠른 실행을 참고하여 커맨드라인 3~4 줄 이내로 원하는 분석을 수행할 수 있습니다.

이 파이프라인은 KISTI 에서 제공하는 K-BDS 분석 인프라에 맞춰 최적화하였으며 이를 이용하여 실행할 수 있습니다.



Input files 요약

- **Microbial isolate genome : Illumina short paired-end read sequence 또는 PacBio long read sequence**
- **SAG : Illumina short paired-end read sequence**
- **Metagenome : Illumina short paired-end read sequence**
- **Metadata : Sample에 관한 정보가 담긴 파일**

metadata는 ,(콤마)로 구분된 csv 파일이어야 하며, 맨 위 첫행은 각 항목의 변수 이름, 맨 첫 열은 sample ID로 구성되어야 합니다.

예시)

metadata의 accession 을 fastq sequence의 공통된 이름과 같은 sampleID 로 지정해주면 하위 분석이 수월함. (metadata 권장사항)

```
ERR1018185_fastp_hg38_1.fastq.gz  ERR1018187_fastp_hg38_1.fastq.gz
ERR1018185_fastp_hg38_2.fastq.gz  ERR1018187_fastp_hg38_2.fastq.gz
ERR1018186_fastp_hg38_1.fastq.gz  ERR1018188_fastp_hg38_1.fastq.gz
ERR1018186_fastp_hg38_2.fastq.gz  ERR1018188_fastp_hg38_2.fastq.gz
```

accession	artificial_metadata1	artificial_metadata2
ERR1018185	Asia	Dog
ERR1018186	Asia	Cat
ERR1018187	Europe	Cat
ERR1018188	Europe	Tiger

---

2. 파이프라인 모듈

이 파이프라인은 총 **7개의 모듈**로 구성되어 있으며, 분석에 다양한 모듈을 활용할 수 있습니다.

- **: 모듈 - [분석 대상]**

A : GENOME_ASSEMBLY - [Isolate genome or SAG]

B : RAWREAD_QC - [Metagenome]

C : ASSEMBLY_BINNING - [Metagenome]

D : BIN_ASSESSMENT - [Isolate genome, SAG, MAG]

E : COMPARATIVE_ANNOTATION - [Isolate genome, SAG, MAG]

F : WMS_TAXONOMY - [Metagenome]

G : WMS_FUNCTION - [Metagenome]

대표 분석 예시 - 대표 분석 예시에 대해서는 5의 빠른실행에서 예시 참조 가능

Microbial isolate genome 분석

- **A => D => E**
    - Single isolate genome sequencing raw 리드 필터링 후 어셈블리 (A) + 조립된 유전체 품질 평가 및 분류군 할당(D) + 비교유전체 분석을 위한 기능 주해

SAG 분석

- **A => D => E**
    - SAG raw 리드 필터링 후 어셈블리 (A) + 조립된 유전체 품질 평가 및 분류군 할당(D) + 비교유전체 분석을 위한 기능 주해

MAG 분석

- **B => C => D => E**
    - RAW read 퀄리티 컨트롤 (B) + assembly와 binning (C) + 조립된 유전체 품질 평가 및 분류군 할당 (D) +비교 유전체 분석을 위한 기능 주해 (E)

마이크로바이옴 상호작용 분석

- **B => F => G**
    - RAW read 퀄리티 컨트롤 (B) + read based WMS taxonomy 통계분석 (F) + read based WMS function 할당 (G)

각 분석을 수행하기 위해 일련의 분석 module들을 엮어서 분석을 수행할 수 있으며, **메타데이터 파일**과 **메타데이터에서 분석할 항목의 열 (column) 번호** , **input 파일의 종류**에 대해 지정해주면 쉬운 분석수행이 가능합니다.

---

3. 분석환경 설정

파이프라인 이용을 위해 **아래의 command line**을 입력하여 분석 환경설정을 진행함

```
sh /scratch/tools/microbiome_analysis/env_setup_files/env_setup_for_analysis.sh
# 환경설정을 적용하거나 K-BDS 분석인프라에 재접속
source ~/.bashrc
conda init
```

```
k113a01@07.02.2024 bdata-login01:~: 10> sh /scratch/tools/microbiome_analysis
/env_setup_files/env_setup_for_analysis.sh
미생물 유전체 데이터 분석을 위해 설계된 분석 파이프라인입니다.

상업적 목적에는 이용이 제한될 수 있습니다.

KEGG API는 아카데미 기관 소속 사용자의 아카데믹 사용에 한해 가능합니다.

비아카데미 소속 사용자는 KEGG annotation 결과 이용에 license 제한이 있습니다.

설치 후 접속이 종료됩니다. 다시 K-BDS에  접속해야합니다.

계속하시겠습니까? (y/N):
 y
```

아카데미 기관 소속 사용자가 아닌 경우 license 관련 문제가 발생할 수 있습니다. 일부 프로그램들은 상업적 이용에 적합하지 않을 수 있습니다.

---

4. 분석파이프라인 실행방법

4.1 [A] : GENOME_ASSEMBLY

GENOME_ASSEMBLY : Isolate genome, SAG sequence genome 조립

Input data : seqeuncing file and parameter(short, long, SAG)

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/GENOME_ASSEMBLY_execution.sh**
- **필수 설정 parameter**
  - *-i* : raw fastq 파일이 들어있는 폴더. 절대경로로 지정. 한 폴더안에 input fastq.gz 데이터도 가능
  - *-t* : 3가지 중 하나는 지정하여 실행 **short, long, SAG**
- **optional parameter & parameter 기본 설정 값** :
  - *-o* : $HOME/results/genome/GENOME_ASSEMBLY/{short,long,SAG}
  - *-L* : --pacbio-hifi, pacbio-hifi가 기본 값으로 설정되어 있음. hifi 가 아닌 기존의 pacbio-raw 리드를 조립할 경우 원할 시 -L --pacbio--raw 를 추가

    ex : --longReadType --pacbio-raw
  - *-o* : output directory를 지정하지 않는 경우에 다음 위치에 결과가 생성됨$HOME/results/genome/GENOME_ASSEMBLY
- **실행 코드** : *sbatch /scratch/tools/microbiome_analysis/sbatch_execution/GENOME_ASSEMBLY_execution.sh -i $PWD -t short -o result*

```
# 실제 실행 예시
# input directory 는 절대경로로 입력 : -i
(base) ID@dd:mm:yyyy bdata-login01: ~/assembly/short_read: > sbatch
/scratch/tools/microbiome_analysis/sbatch_execution/GENOME_ASSEMBLY_execution.sh -i $PWD -t short -o
result
# 현재 폴더의 result에 결과 폴더가 생성됨.
# -o option을 사용하지 않는 경우 $HOME/results/genome/GENOME_ASSEMBLY에 결과가 생성됨.

# manual 실행을 위한 code
(base) ID@dd:mm:yyyy bdata-login01:  sbatch -p cpu32_only wrap="nextflow run assembly_pipeline.nf --
inputDir /path/to/input --inputReadType long --longReadType --pacbio-hifi"
```

결과 파일 : [A] GENOME_ASSEMBLY

- **$HOME/results/genome/GENOME_ASSEMBLY** 폴더에 long, short, SAG 디렉토리가 생성됨
  - short read : fastp result folder, assembled.fa 파일들, *assembly_info.csv*
  - long read : (PacBio read는 전처리 과정이 없음), assembled.fa 파일들, *assembly_info.csv*
  - SAG : fastp result folder, assembled.fa 파일들, *assembly_info.csv*

결과물을 BIN_ASSESSMENT and ANNOTATION에 이용

4.2 [B] : RAWREAD_QC

RAWREAD_QC : Metagenome sequence 품질 전처리

Input data : paired-end short read metagenome sequence

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh**
- **필수 설정 parameter**
  - *-i* : paired end read 가 있는 input directory 절대경로
- **optional parameter & parameter 기본 설정 값** :
  - *-o* : *output directory* : $HOME/results/metagenome/RAWREAD_QC - **fastqc_raw**, **fastqc_filtered**, **multiqc** , **read_filtered** 결과가 생성됨

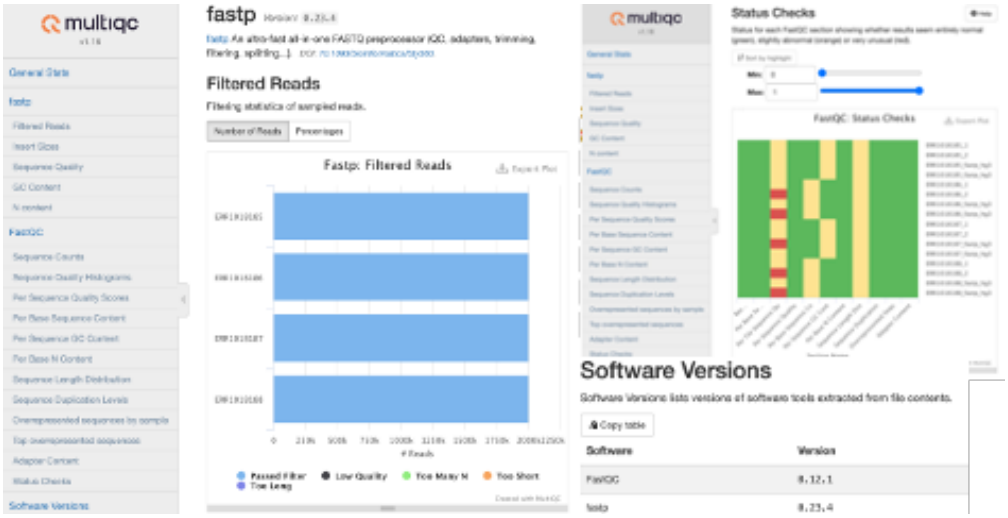
```
# 실제 실행 예시
# input directory 는 절대경로로 입력 : -i
(base) ID@dd:mm:yyyy bdata-login01: ~/nextflow_run/read_filtered: 1120> sh
/scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh -i $PWD

# read quliaty filter, huamn read filter 후 결과 read file 을 생성함
# results at directory
$HOME/results/metagenome/RAWREAD_QC
# read quality 결과를 눈으로 보고싶은경우, multiqc 디렉토리를 다운로드 후 html파일을 보거나 VScode 등으로 확인
# go  to result : $ cd $HOME/results/metagenome/RAWREAD_QC
```

**결과 파일 : [B] RAWREAD_QC

- **$HOME/results/metagenome/RAWREAD_QC** 하위에 4개 디렉토리가 생성됨

  : **fastqc_raw**, **fastqc_filtered**, **multiqc** , **read_filtered**
  - **fastqc_raw** : metagenomic sequence raw data의 read stat 결과
  - **fastqc_filtered** : read quality trimming, human read filtering out 후의 read stat 결과
  - **multiqc** : read quality contorl 결과 html파일
  - **read_filtered** : 하위 분석에 사용하는 결과 fastq 파일



multiQC html파일 : read filtering 전, 후의 결과 그림으로 파악

4.3 [C] : ASSEMBLY_BINNING

MAG (metagenome-assembled genome) 조립

Input data : quality controlled paired-end short read metagenome sequence

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh**
- **필수 설정 parameter**
  - 없음
- **optional parameter & parameter 기본 설정 값** :
  - *-i* :$HOME/results/metagenome/RAWREAD_QC/read_filtered : quality controlled paired end read 가 있는 input directory
  - *-o* : $HOME/results/metagenome/ASSEMBLY_BINNING - **assembled_contigs**, **final_bins**, **metabat2_bins**, **semibin2_bins** 디렉토리가 생성됨

```
# 실제 실행 예시
# input, output directory 는 절대경로로 입력 : -i,-o optional argument 로 넣지 않아도 됨.
# 예시의 실행폴더 위치 : ~/test_data/test_As_Bi
(base)  ID@dd:mm:yyyy bdata-login01: ~/test_data/test_As_Bi: 100> sh
/scratch/tools/microbiome_analysis/sbatch_execution/ASSEMBLY_BINNING._exeuction.sh  # +  -i
${absolute_PATH_of_input} -o ${absolute_PATH_of_output_directory}
3741
# 실행한 디렉토리에서 실행 log 정보가 저장되며 현재 진행상황 확인 가능.
(base)  ID@dd:mm:yyyy bdata-login01: ~/test_data/test_As_Bi: 1001> vi slurm-3741.out
# 만약 bin이 생성되지 않는 샘플이 있는경우 프로세스가 종료됨.

# results at directory
$HOME/results/metagenome/ASSEMBLY_BINNING
```

```
(base)  ID@dd:mm:yyyy bdata-login01: $HOME/results/metagenome/ASSEMBLY_BINNING: 1002> ls
#
assembled_contigs  dastool_bins  metabat2_bins  semibin2_bins


# options ?
# manual run code # If you are using K-BDS, you should do it with sbatch
nextflow run /scratch/tools/microbiome_analysis/nf_scripts/ASSEMBLY_BINNING.nf
# nextflow parameter를 이용하고 싶은 경우 --parmaname parameter로 변경하여 이용가능
```

결과 파일 : [C] ASSEMBLY_BINNING

- **$HOME/results/metagenome/ASSEMBLY_BINNING** 하위에 4개 디렉토리가 생성됨

  : **assembled_contigs**, **final_bins**, **metabat2_bins**, **semibin2_bins**

  - **assembled_contigs** : MEGAHIT 을 이용하여 *de novo assembly* 된 contig fasta file
  - **dastool_bins** : MetaBAT2와 SemiBin2의 정보를 합쳐 향상된 MAG이 들어있는 디렉토리
    하위 분석에 활용되는 MAG
  - **metabat2_bins** : MetaBAT2 를 이용하여 나온 MAG들이 있는 디렉토리
  - **semibin2_bins** : SemiBin2 를 이용하여 나온 MAG들이 있는 디렉토리

4.4 [D] : BIN_ASSESSMENT

GENOME 품질 평가 및 분류군 할당

Input data : 유전체 파일

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh**
- **필수 설정 parameter**
  - 없음
- **optional parameter & parameter 기본 설정 값** :
  - *-i* : $HOME/results/metagenome/RAWREAD_QC/read_filtered : quality controlled paired end read 가 있는 input directory
  - *-o* : $HOME/results/metagenome/ASSEMBLY_BINNING - **assembled_contigs**, **final_bins**, **metabat2_bins**, **semibin2_bins** 디렉토리가 생성됨
- **연계 분석** :
  - 생성 유전체 평가. **4 - A** : SAG, Isolate assembled Genome , **4 - C** : MAG 데이터를 활용하여 결과분석 가능. GTDB 결과 등을 살펴 유전체의 신규성 확인 가능
  - 비교유전체 분석 및 메타유전체 연계 분석에 활용 : **4 - E** , **4 - F** 의 하위 분석에 활용 가능. 특정 품질, 분류군을 충족하는 유전체만 가지고 메타데이터를 이용한 비교분석 및 메타유전체의 kraken을 활용한 분류군 분석 결과와 결합하여 데이터 해석 및 비교유전체 분석이 가능

```
# 실제 실행 예시
# input, output directory 는 절대경로로 입력 : -i,-o optional argument 로 넣지 않아도 됨.
# 예시의 실행폴더 위치 :~/test_data/test_BA
(base)  ID@dd:mm:yyyy bdata-login01: ~/test_data/test_BA: 1> sh
/scratch/tools/microbiome_analysis/sbatch_execution/BIN_ASSESSMENT_execution.sh
Total bin number: 3
Number of batches: 1
Submitted job for $HOME/results/metagenome/ASSEMBLY_BINNING/dastool_bins/subdir_1 with job ID 3810
All subdir jobs submitted. Dependencies: 3810
Submitted batch job 3811
wrapping up job submitted : 3811

# optional argument 를 이용하기 원할 시 다음 명령어를 추가하여  진행.
# +  -i ${absolute_PATH_of_input} -o ${absolute_PATH_of_output_directory}

# 실행한 디렉토리에서 실행 log 정보가 저장되며 현재 진행상황 확인 가능.
# wrapping up job 을 확인하여   성공적으로 실행될 시 , BIN_ASSESSMENT is finished successfully 문구가 생성됨
(base)  ID@dd:mm:yyyy bdata-login01: ~/test_data/test_BA: 2> vi slurm-3811.out

# 최종 결과 파일은 MIMAG 기준, (Completeness 50% 이상, contamination 10% 미만) medium_quality.pass 이상을 기준
으로 사용하도록 설정되어 있음
(base) ID@dd:mm:yyyy bdata-login01: ~/results/metagenome/BIN_ASSESSMENT/final_report: 3> ls

gtdb_summary_combined_final.tsv  quality_taxonomy_combined_final.csv
# quality_taxonomy_combined_final.csv 디렉토리에 유전체의 품질 정보 및 GTDB 분류군 정보가 있음
# Completeness, Contamination 은 Checkm2,GUNC 품질 기준 통과 여부가 나타나 있음
# 각 program 에 대한 상세정보는 각 폴더안에서 확인 가능함
(base) ID@dd:mm:yyyy bdata-login01: ~/results/metagenome/BIN_ASSESSMENT/final_report: 3> head -n1
quality_taxonomy_combined_final.csv

# 1~12  columns
Genome,Completeness,Contamination,medium_quality.pass,near_complete.pass,medium_quality_gunc.pass,ne
ar_complete_gunc.pass,QS50,QS50.pass,pass.GUNC,QS50_gunc.pass,classification
```

결과 파일 : [D] BIN_ASSESSMENT

- **$HOME/results/metagenome/BIN_ASSESSMENT** 하위에 4개 디렉토리가 생성됨. 만약 genome의 개수가 많은 경우에는 1000개 단위로 새로운 디렉토리를 합친 결과가 생성됨. 다양한 디렉토리와 파일이 생성됨. MIMAG 기준 medium quality 를 통과한 유전체는 bins_quality_passed_$(random ID) 디렉토리에 존재.

  : **checkm2_${randomID}**, **gtdb_outdir_${randomID}**, **gunc_${randomID}**, **bins_quality_passedFinal**,**final_report**

  - **checkm2_${randomID}** : 생성된 전체 유전체의 CheckM2 실행 결과가 있는 디렉토리
  - **gunc_${randomID}** : 분석에 이용한 전체 유전체의 GUNC 실행 결과가 있는 디렉토리
  - **gtdb_outdir** : medium_quality 이상의 품질을 가진 유전체의 GTDB taxonomy 할당 결과 폴더
  - **bins_quality_passedFinal** : medium quality 이상의 유전체가 있는 디렉토리

- **final_report** :
  - *gtdb_summary_combined_final.tsv* 전체 유전체의 GTDB 결과 파일
  - *quality_taxonomy_combined_final.csv* - 전체 유전체의 CheckM2, GUNC, GTDB의 분류군 할당정보가 합쳐진 파일

4.5 [E] : COMPARATIVE_ANNOTATION

유전체 비교 분석과 대사회로 및 특이 유전자 분석

Input data : 유전체 파일

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh**
- **필수 설정 parameter**
  - 없음
- **optional parameter & parameter 기본 설정 값** :
  - -i : $HOME/results/metagenome/BIN_ASSESSMENT/bins_quality_passedFinal : 일정 품질기준을 통과한 유전체가 있는 폴더. 혹은 유저가 직접 선별한 유전체가 있는 디렉토리
  - -o : $HOME/results/metagenome/COMPARATIVE_ANNOTATION : 분석 결과가 저장되는 디렉토리
  - -g : $HOME/results/metagenome/BIN_ASSESSMENT/final_report/quality_taxonomy_combined_final.csv : **4 - D** 의 결과 파일. **4 - D**를 실행했다면 자동으로 잡도록 설정됨
  - -m : 메타유전체 데이터 이용시 활용한 메타데이터.csv. 첫 줄은 **파일이름 기준** accession이어야함
  - -n : metadata column numer - 숫자로 **분석할 metadata column** 을 선택 (ex : -n 3)
  - -u : user 가 주는 메타데이터 (MAG 생성이 아닐경우 선택), 첫번째 column은 **4 -E**의 quality_taxonomy_combined_final.csv 파일의 것과 같아야함 (genome file 의 base name)
- **연계 분석** :
  - 생성 유전체 평가. **4 - A** : SAG, Isolate assembled genome , **4 - C** : MAG 데이터를 활용하여 결과분석 가능. GTDB 결과 등을 살펴 유전체의 신규성 확인 가능
  - 비교유전체 분석 및 메타유전체 연계 분석에 활용 : **4 - E** , **4 - F** 의 하위 분석에 활용 가능. 특정 품질, 분류군을 충족하는 유전체만 가지고 메타데이터를 이용한 비교분석 및 메타유전체의 kraken을 활용한 분류군 분석 결과와 결합하여 데이터 해석 및 비교유전체 분석이 가능

```
# 실제 실행 예시
# input, output directory 는 절대경로로 입력 : -i,-o optional argument 로 넣지 않아도 됨.
# 4-D 를 돌렸을 경우 옵션 선택 없이 기본 옵션으로 실행가능
# 분석을 위해서는 metadata 파일 제공을 권장.
# case1 :  metagenome으로부터 MAG 을 생성한 경우, metagenome의 metadata를 제공하면 script 실행 가능.
# case1 :  metagenome의 메타데이터의 첫번째 column은 input paired end reads 의 identifier이어야 함
# case1 :  _fastq_{1,2}.gz 를 제외한 부분
# case1 : ex) AKB3_fastq_1.gz AKB3_fastq_2.gz 의 identifier 는 AKB3

# case1 : 메타데이터 3번째 컬럼 정보를 가지고 visualization 등 분석에 활용.
```

```
(base) ID@dd:mm:yyyy bdata-login01: ~/test 1> sh
/scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh  -m
metagenome_metadata.csv  -n 3
# $HOME/results/metagenome/COMPARATIVE_ANNOTATION/ 에 각종 annotation 결과가 있음.
(base) ID@dd:mm:yyyy bdata-login01: ~/test 2> ls ~/results/metagenome/COMPARATIVE_ANNOTATION/
CARD  kofamscan  panaroo_result  prokka  VFDB
# 상업적 이용 혹은 아카데미 소속이 아닌 사용자는 KEGG license 를 확인해야함.

# case 2 : user가 특정 유전체를 BIN_ASSESSMENT를 진행한 후 annotation 진행을 원할 경우
# case 2 : -u : user metadata, -n metadata column
(base) ID@dd:mm:yyyy bdata-login01: ~/test 3> sh
/scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh  -u
user_metadata.csv  -n 3
# 분석 결과가 생성됨
# 상업적 이용 혹은 아카데미 소속이 아닌 사용자는 KEGG license 를 확인해야함.
```

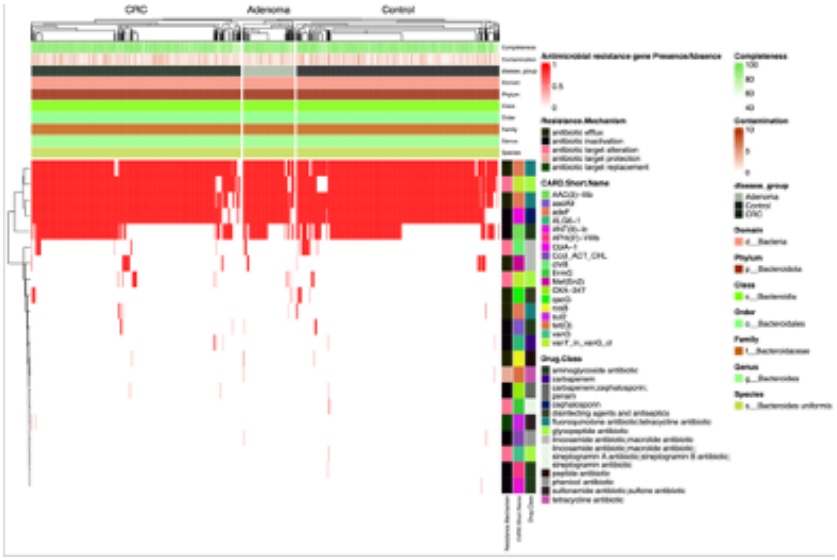
결과 파일 : [E] COMPARATIVE_ANNOTATION

- **$HOME/results/metagenome/COMPARATIVE_ANNOTATION/** 하위에 5개 디렉토리가 생성됨

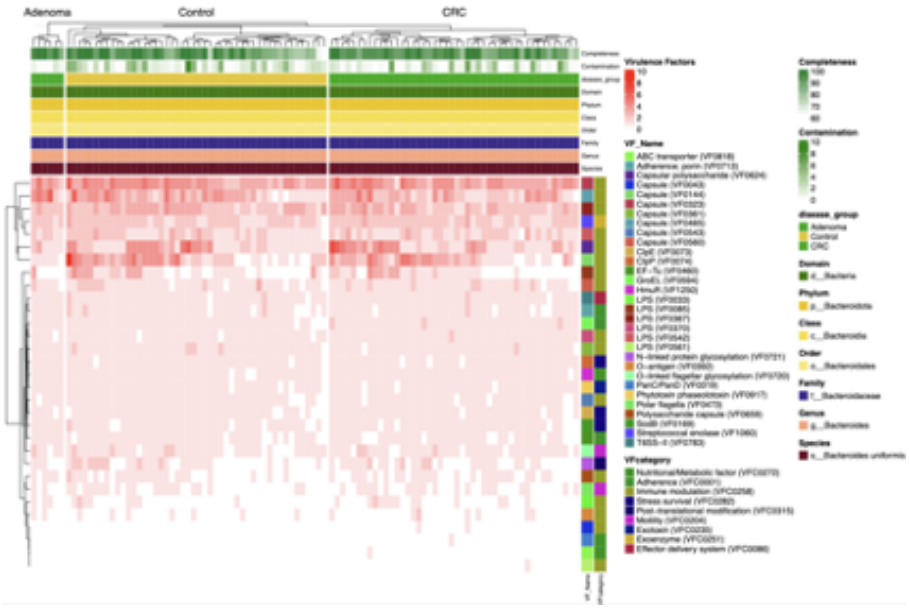
  : **CARD, kofamscan, panaroo_result, prokka, VFDB,**

  - **prokka** : 분석 파이프라인 **4-D**의 일정 품질 이상의 유전체에 대해 CDS, rRNA annotation 을 진행한 결과가 있는 디렉토리
  - **panaroo_result** : 비교 유전체 분석 및 빠른 annotation 을 위한 결과 파일들이 들어있는 폴더. pangene 등 각종 metric을 구하는데 활용됨
  - **CARD** : 분석 대상 유전체의 pangenome으로 antimirobial resistance gene 정보와 유전체의 각종 메타데이터가 시각화로 나타나있는 결과와, CARD annotation table이 있는 디렉토리
  - **VFDB** : CARD와 마찬가지 파일로, virulence factor에 대한 정보가 담겨있는 디렉토리
  - **kofamscan** :
  -
    - *KEGG_module_completeness.csv* : 각 유전체의 KEGG module completeness 정보가 있는 파일
    - *ko_matrix.csv* : pangenome KEGG_orthology annotation 정보가 있는 테이블
    - **KEGG_module_visualization_shiny** : interactive KEGG module visualization 이 들어있는 폴더.

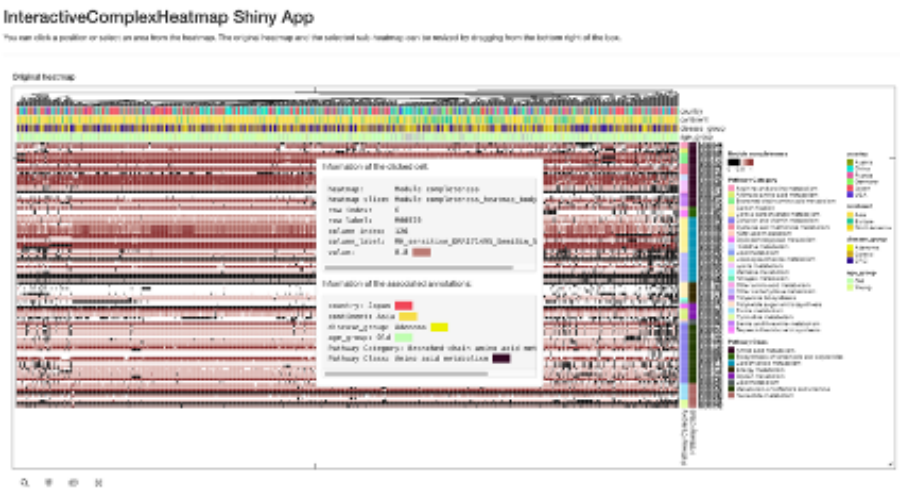
      visual studio code를 이용할 시, R432_environment 를 활성화 한 후 shiny.sh 를 실행하면 확인할 수 있다.



- CARD 의 시각화 결과 예시 : 대장암, Adenoma, 건강인의 장 메타유전체에서 조립된 *Bacteroides uniformis* 1380 MAGs 의 CARD annotation 결과. Metadata column을 지정할 시, 해당 메타데이터로 그룹이 나뉘어 시각화 결과가 생성됨. 상단의 heatmap에는 유전체의 완성도, 오염도, Taxonomy, metadata 단일 컬럼에 대한 정보가 표기되어 있음. 세로의 한 열은 각 antimicrobial resistance gene 을 나타냄. **rgi6.0.3**을 이용하여 CARD **3.2.8** 데이터베이스로 annotation을 진행함. **include_nudge option, strict, complete** 결과만이 시각화 결과에 표시됨. MAG, SAG 등의 불완전한 유전체 품질로 인한 정보상실을 include_nudge로 완화하였음



VFDB 의 시각화 결과 : 대장암, Adenoma, 건강인의 장 메타유전체에서 조립된 *Bacteroides uniformis* 1380 MAGs 의 VFDB annotation 결과. 상단의 heatmap 에는 CARD와 같은 정보가 나타나 있으며, metadata 제공 시 자동으로 구분하여 시각화 진행함. 우측의 heatmap 은 VFDB의 작용기전 (Resistance Mechanism, CARD Short name, 저항성 가진 Drug Class) 정보가 나타나 있음. Raw data table 에서 자세한 결과 확인이 가능함. Diamond 를 사용하여 **VFDB setB** protein database (2023 Aug) 에 identity 50%, subject_cover 80%, evalue 1e-10을 기준으로 병독성 유전자 확인을 진행함



KEGG module 시각화 결과, 각 줄은 KEGG module 을 나타내며 , module 의 완성도에 따라 cell 의 색이 변함. KEGG module이 존재하는 기준은 50% 설정되어 있으며, Rscript 에서 이값을 변화시켜서 적용가능함. 대규모 유전체를 분석하는 경우를 위해 interactive visualization이 가능하게 하였으며, default option 을 이용하였을 경우 Visual Studio Code 등에서 shiny 결과 폴더 (**$HOME/results/metagenome/COMPARATIVE_ANNOTATION/KEGG_module_visualization_shiny**)로 이동 후 sh shiny.sh를 실행 시 나오는 localhost 정보를 인터넷 창에 입력하여 KEGG 모듈 분석 결과의 interactive visualization 이 가능함. KEGG 정보를 이용 시 아카데미 유형의 사용자가 아닌경우 사용에 license 를 참고해야 함

```
# 해당 conda 환경을 활성화하고, Visual Studio Code 를 실행 시, localhost로 연결되며 시각화 결과를 모든 cell마다 확인 가능
(R432_environment)
$HOME/results/metagenome/COMPARATIVE_ANNOTATION/KEGG_module_visualization_shiny: 14> sh htShiny.sh
```

4.6 [F] : WMS_TAXONOMY

Whole metagenome shotgun sequence 분류군 분석

Input data : short read metagenome data (fastq.gz) , metadata table (csv)

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/WMS_TAXANOMY_execution.sh**
- **필수 설정 parameter**
  - -m : 메타데이터 csv 파일
  - -a : 분석할 메타데이터의 column ID : categorical column을 선택하여 제공해야 함.
- **optional parameter & parameter 기본 설정 값** :
  - -i : $HOME/results/metagenome/RAWREAD_QC/read_filtered : quality control 을 진행한 metagenome 데이터
  - -c : metadata.csv 의 sample accession coumn ID
  - -o : $HOME/results/metagenome/WMS_TAXONOMY : 분석 결과가 저장되는 디렉토리

- **연계 분석** :
    - 메타유전체 품질 평가인 **4 - C** 를 활용하여 진행. 분석 결과를 활용하여 특정 관심그룹을 가지고 비교유전체 분석 **4 - E** 및 많은 분석에 활용가능함

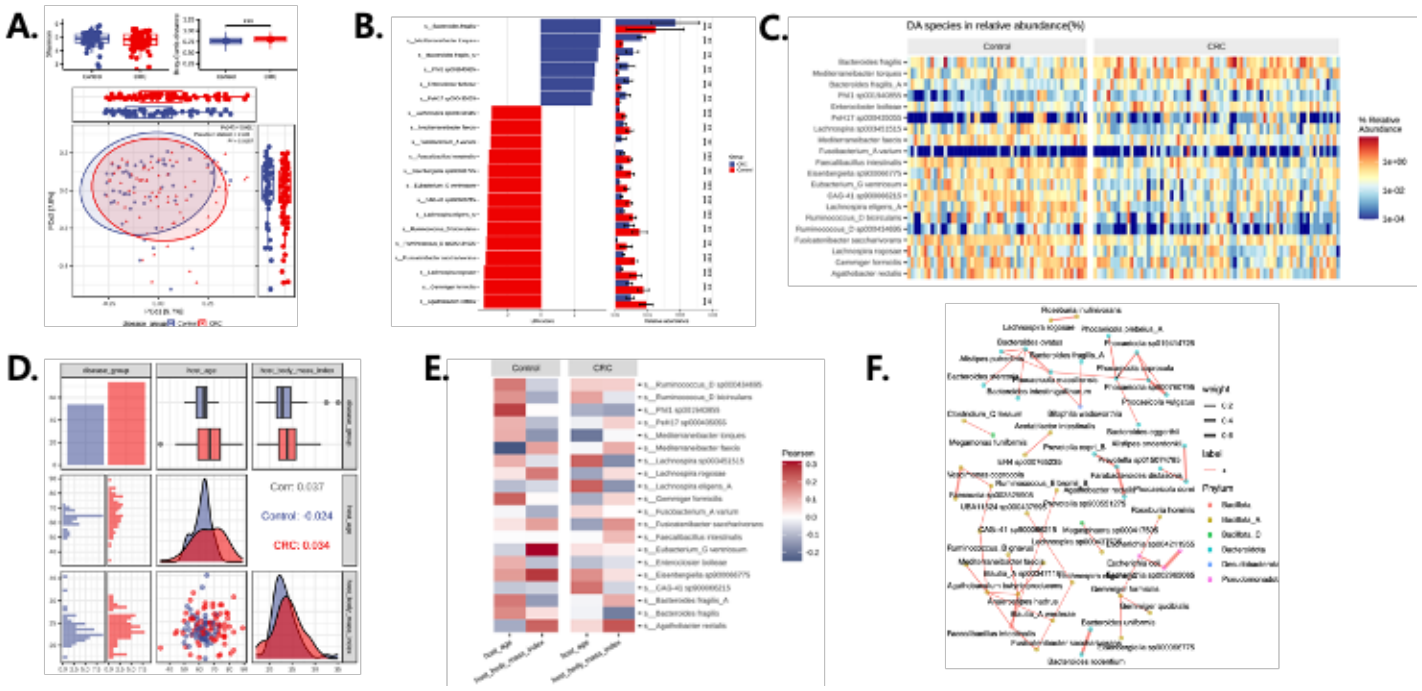
```
# WMS taxonomy
#해당 분석은 cpu64-only 노드에서만 분석이 진행 가능함.
(base) k113a02@07.02.2024 bdata-login01:~/test_data: 1 > sbatch -p cpu64-only
/scratch/tools/microbiome_analysis/sbatch_execution/WMS_TAXANOMY_execution.sh  -m metadata4.csv -a 3


# 직접 실행을 원할 경우.
(base) k113a02@07.02.2024 bdata-login01:~/test_data: > sh
/scratch/tools/microbiome_analysis/sbatch_execution/WMS_TAXANOMY_execution.sh  -m metadata4.csv -a 3
```

결과 파일 : [F] WMS_TAXONOMY

- **$HOME/results/metagenome/WMS_TAXONOMY/** 하위에 4개 디렉토리가 생성됨

  : bracken, kraken2, phyloseq, WMS_taxonomy_analysis
    - **kraken2** : GTDBr214 종 대표 유전체를 이용해 제작한 kraken2 데이터베이스를 이용한 kraken2 실행 결과 디렉토리
    - **bracken** : kraken2의 데이터를 정량하여 보다 정확한 relative abundance data를 추정한 결과 디렉토리
    - **phyloseq** : kraken2와 메타유전체를 합쳐, 통계분석, 시각화 분석에 활용할 데이터가 있는 디렉토리
    - **WMS_taxonomy_analysis** : 메타유전체의 분류군 정보를 이용하여 메타데이터와 각종 통계 분석을 수행한 최종 결과 테이블 및 시각화 결과가 있는 디렉토리



해당 그림은 128개의 대장암 - CRC 데이터세트를 이용하여 4 - F 를 실행하여 얻은 결과. 다양한 그림과 그에 필요한 원본 데이터 테이블들이 생성됨. 통계적 분석이 가능할 경우(수가 어느 정도 있을 경우) 해당 그림들은 자동으로 생성됨

A : Metadata에 따른 군집분포의 양상을 살펴보기 위한 동일한 그룹 사이의 Shannon diveristy 와 Bray-curtis dissimilarity 의 차이와 유의성을 나타내는 boxplot. pCoA plot은 마찬가지로 메타데이터가 얼마나 그룹간의 차이를 설명할 수 있는지 시각적으로 확인하기 편하며, Pseudo-F statistic과 R^2로 설명력을 확인할 수 있음.

B : 대장암에 유의하게 연관되어 있는 taxa를 밝히기 위한 분석과 그 시각화 결과. Lefse를 활용하여 질병 상태에 유의하게 연관되어 있는 분류군(종 수준)들을 표지 하였으며 오른쪽에는 relative abundnace의 boxplot으로 유의한 차이를 보이는 것이 *로 표시되어 있음

C : B 그림에 해당하는 유의하게 분포가 차이나는 , 상위 20개 taxa의 relative abundance heatmap

D : 메타유전체의 메타데이터 사이의 상관관계를 분석한 plot. 수치형 변수가 표시되며, 수치형 변수의 개수에 따라 plot이 늘어남

E : 수치형 변수와 metadata를 활용하여 lefse에서 나온 최대 20 개 taxa와의 correlation heatmap.

F : Species correlation Network 로 Sparcc, SpiecEasi를 활용하여 종간의 연결성을 확인할 수 있다. Node 는 Phylum 수준에서 색이 다르며, edge 는 weight 정도 label 의 방향에 따라 크기와 색이 다르게 나타남

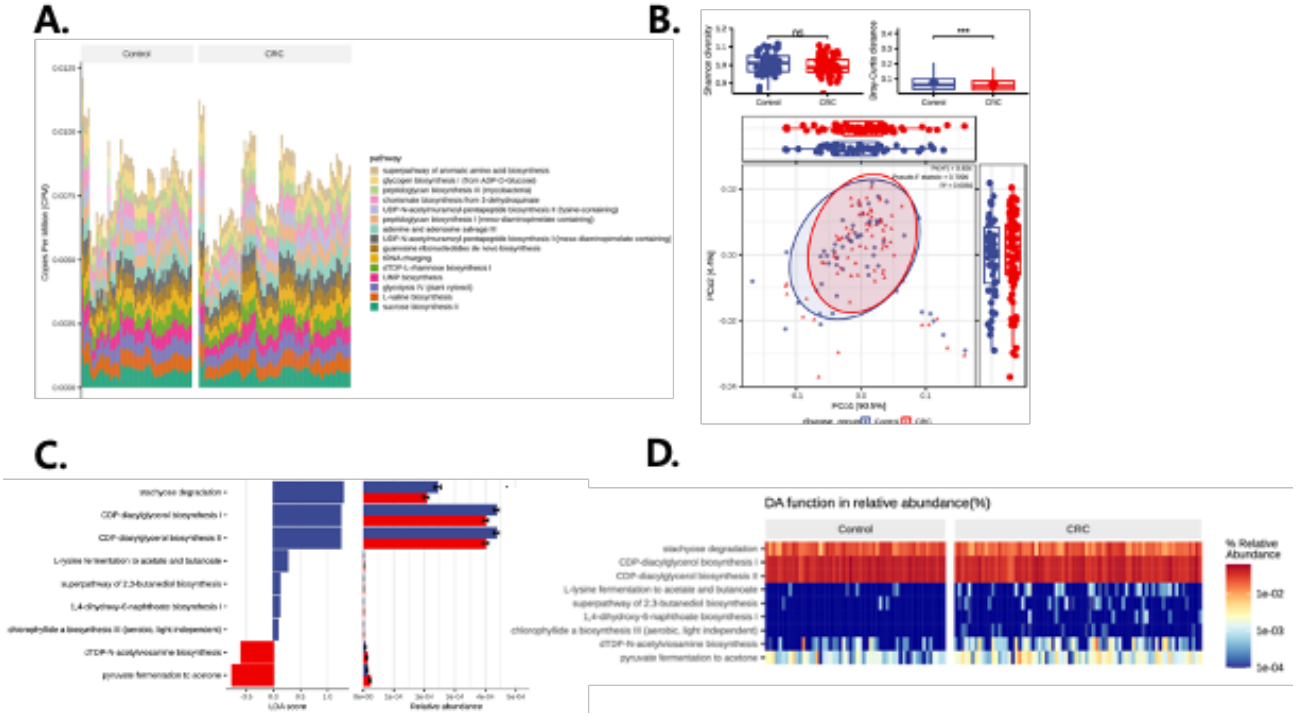
4 - G : WMS_FUNCTION

Whole metagenome shotgun sequence 기능 분석

Input data : short read metagenome data (fastq.gz) , metadata table (csv)

- **실행 스크립트**
  - **/scratch/tools/microbiome_analysis/sbatch_execution/WMS_TAXANOMY_execution.sh**
- **필수 설정 parameter**
  - -m : 메타데이터 csv 파일
  - -a : 분석할 메타데이터의 column ID : categorical column을 선택하여 제공해야 함.
- **optional parameter & parameter 기본 설정 값** :
  - -i : $HOME/results/metagenome/RAWREAD_QC/read_filtered
  - -o : $HOME/results/metagenome/WMS_FUNCTION
  - -c : metadata.csv 의 sample ID 로 사용할 column (default is 1)

```
# 실제 실행 예시
(base) k113a02@07.02.2024 bdata-login01:~/test_data: 1 >  sbatch -p cpu32-only
/scratch/tools/microbiome_analysis/sbatch_execution/WMS_FUNCTION_execution.sh -m metada
ta4.csv  -a 2
# humann3 를 돌리고 , 분석 object를 생선한 후 R에서 통계분석이 진행된다.
```



해당 그림은 128개의 대장암 - CRC 데이터세트를 이용하여 4 - G 를 실행하여 얻은 결과.

A : HUMAnN을 사용하여 metacyc pathway 의 분포를 집단별로 확인할 수 있는 bar plot 그림. 상위 15개 대사경로에 대해서 가시화 함

B : 대장암에 유의하게 연관되어 있는 기능을 밝히기 위한 분석과 그 시각화 결과. Lefse를 활용하여 질병 상태에 유의하게 연관되어 있는기능을 표지 하였으며 오른쪽에는 relative abundnace의 boxplot으로 유의한 차이를 보이는 것이 *로 표시되어 있음

C : 메타데이터에 따라 유의하게 차이나는 기능군 분류

D : C수치형 변수와 metadata를 활용하여 lefse에서 나온 최대 20 개 taxa와의 correlation heatmap.

---

5. 빠른실행

Microbial isolate genome 분석

- **A => D => E**
- Single isolate genome sequencing raw 리드 필터링 후 어셈블리 (A) + 조립된 유전체 품질 평가 및 분류군 할당(D) + 비교유전체 분석을 위한 기능 주해
- 준비물 : fastq.gz 파일 (short, SAG) 또는 long read fasta 파일, 분석에 사용할 메타데이터 파일

Isolate genome 분석 : short paired-end read 빠른실행

```
$ conda activate base
# Isolate genome short quality filtering and assembly: A
# read type 지정 필수, input directory 지정 필수  : -t {short, long, SAG}, -i {intput_dir}
$ sbatch /scratch/tools/microbiome_analysis/sbatch_execution/GENOME_ASSEMBLY_execution.sh -i $PWD -t
short -o result_A
# BIN ASSESSMENT : D
# input directory 지정 필요 : -i {intput_dir}
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/BIN_ASSESSMENT_execution.sh -i result_A
# COMPARATIVE ANNOTATION : E
# metadata.csv 파일과 , annotation visualization 에 활용할 column ID (숫자 : 몇번째인지) 필요.
# -u metadata.csv -n 3 (3th column is used for visualization )
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh  -u
metagenome_metadata.csv  -n 3
```

Isolate genome 분석 : long read 빠른실행

```
$ conda activate base
# Isolate genome short quality filtering and assembly: A
# read type 지정, input directory 지정
$ sbatch /scratch/tools/microbiome_analysis/sbatch_execution/GENOME_ASSEMBLY_execution.sh -i $PWD -t
long -o result_A
# BIN ASSESSMENT : D
# input directory 지정 필요
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/BIN_ASSESSMENT_execution.sh -i result_A
# COMPARATIVE ANNOTATION : E
# metadata.csv 파일과 , annotation visualization 에 활용할 column ID (숫자 : 몇번째인지) 필요.
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh  -u
metagenome_metadata.csv  -n 3
```

SAG 분석 : 빠른실행

- **A => D => E**

  SAG raw 리드 필터링 후 어셈블리 (A) + 조립된 유전체 품질 평가 및 분류군 할당(D) + 비교유전체 분석을 위한 기능 주해

```
$ conda activate base
# Isolate genome short quality filtering and assembly: A
# read type 지정, input directory 지정
$ sbatch /scratch/tools/microbiome_analysis/sbatch_execution/GENOME_ASSEMBLY_execution.sh -i $PWD -t
SAG -o result_A
# BIN ASSESSMENT : D
# input directory 지정 필요
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/BIN_ASSESSMENT_execution.sh -i result_A
# COMPARATIVE ANNOTATION : E
# metadata.csv 파일과 , annotation visualization 에 활용할 column ID (숫자 : 몇번째인지) 필요.
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh  -u
metagenome_metadata.csv  -n 3
```

MAG 분석 : 빠른실행

- **B => C => D => E**

RAW read 퀄리티 컨트롤 (B) + assembly와 binning (C) + 조립된 유전체 품질 평가 및 분류군 할당 (D) +비교 유전체 분석을 위한 기능 주해 (E)

```
#B  :  qulity control
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh -i ${inputDir}
#C  : 조립 및 binning
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/ASSEMBLY_BINNING._exeuction.sh
#D  : 유전체 평가 분류군 할당
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/BIN_ASSESSMENT_execution.sh
#E  : Metagenome의 metadata가 중요.  제공여부가 결과에 큰 차이를 줌
$ sh /scratch/tools/microbiome_analysis/sbatch_execution/COMPARATIVE_ANNOTATION_execution.sh  -m
metagenome_metadata.csv  -n 3
```

마이크로바이옴 상호작용 분석

- **B => F | G**

RAW read 퀄리티 컨트롤 (B) + read based WMS taxonomy 통계분석 (F) | read based WMS function 할당 (G)

B ==> F

```
# B  :  raw read QC
$ /scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh -i ${inputDir}
# F : Taxonomy annotation of WMS and statistical analysis
$ sbatch -p cpu64-only
/scratch/tools/microbiome_analysis/sbatch_execution/WMS_TAXANOMY_execution.sh  -m metadata4.csv -a 3
```

B ==> G

```
# B  :  raw read QC
$ /scratch/tools/microbiome_analysis/sbatch_execution/RAW_READQC_execution.sh -i ${inputDir}
# G : Functional annotation of WMS and statistical analysis
$ sbatch -p cpu32-only
/scratch/tools/microbiome_analysis/sbatch_execution/WMS_FUNCTION_execution.sh -m metada
ta4.csv  -a 2
```

6. 라이센스

Program	License type
nextflow	free of user choice
Diamond	GNU General Public License v3.0
BLAST	United States Government Work
HMMER	BSD-3-Clause license
Miniforge	BSD-3-Clause license
FastQC	GNU General Public License v3.0
MultiQC	GNU General Public License v3.0
fastp	MIT License
Bowtie2	GNU General Public License v3.0
bwa	GNU General Public License v3.0
MEGAHIT	GNU General Public License v3.0
Samtools	MIT/Expat License
MetaBAT2	MetaBAT Custom License
Das_Tool	Das_Tool Custom License
SemiBin2	MIT License
CheckM2	GNU General Public License v3.0
GUNC	GNU General Public License v3.0
gtdbtk	GNU General Public License v3.0

GTDB	CC BY-SA 4.0 DEED
kraken2	MIT License
bracken	GNU General Public License v3.0
prokka	GNU General Public License v3.0
panaroo	MIT License
RGI	Custom License
ARO	Custom License
CARD	Custom License
AMRFinderPlus	United States Government Work
Flye	BSD-3-Clause license
EMBOSS	GNU General Public License v3.0

7. FAQ

Q : nextflow 실행 오류가 납니다.

A : work 디렉토리 삭제후 재실행하면 하면 에러가 해결될 수 있습니다.

Q : conda 관련 명령어(activate 찾을 수 없다)가 에러가 납니다.

A : base conda 환경을 활성화하고 실행해보세요.

Q : code정보는 어디에 있나요?
A : K-BDS의 /scratch/tools/microbiome_analysis 폴더에 코드 및 데이터베이스가 있습니다.

문의처 : hyeongown.tg@gmail.com