

# RNA-seq 演習

国立遺伝学研究所 望月孝子

2022/10/20

## 本日の講習

遺伝研スパコンで RNA-seq 解析を行うことでスパコンの操作に慣れることを目的とする。テストデータとして *Saccharomyces cerevisiae* のバッチ培養とケモスタット培養のデータを使用する。

## 遺伝研スパコンへログイン

```
$ ssh user@gw.ddbj.nig.ac.jp  
$ qlogin
```

## 講習用データの確認

ホームディレクトリ に 20221020 というファイル/ディレクトリがないかを確認。

```
$ ls 20221020
```

ls: 20221020 にアクセスできません: そのようなファイルやディレクトリはありません

→ 20221020 があれば、講習中だけ他の名前に変更してください。

```
$ mv 20221020 20221020_tmp  
など、、、、。
```

ホームディレクトリへコピー

```
cp -r /home/ddbjshare/public/lecture/20221020 .
```

## 作業場所へ移動

```
$ cd 20221020
```

## 場所の確認

```
$ pwd  
/home/xxxx/20221020
```

## ファイルの確認

```
$ ls  
outputs  reads  reference  scripts  
$ ls outputs  
hisat2  hisat2_index  stringtie  
$ ls outputs/hisat2  
SRR453566.sam  SRR453569.sam  hisat2.sh.e16626062.1  
hisat2.sh.o16626062.1  hisat2.sh.pe16626062.1  
hisat2.sh.po16626062.1  
SRR453566.sorted.bam  SRR453569.sorted.bam  hisat2.sh.e16626062.2  
hisat2.sh.o16626062.2  hisat2.sh.pe16626062.2  
hisat2.sh.po16626062.2
```

## データの中身の構成

```
20201020  
|___ outputs  解析結果  
|   |___ hisat2  
|   |___ hisat2_index  
|   |___ stringtie  
|---- reads      リードファイル格納用  
|   |___ SRR453566_1.fastq.gz  
|   |___ SRR453566_2.fastq.gz  
|   |___ SRR453569_1.fastq.gz  
|   |___ SRR453569_2.fastq.gz  
|---- reference  リファレンスファイル  
|   |___ s288c.fa  
|   |___ s288c.gff  
|---- scripts    スクリプト  
|   |___ hisat2.sh      #conda にて実装
```

```
$ grep ">" reference/s288c.fa
```

>NC\_001133.9 *Saccharomyces cerevisiae* S288C chromosome I, complete sequence  
>NC\_001134.8 *Saccharomyces cerevisiae* S288C chromosome II, complete sequence  
>NC\_001135.5 *Saccharomyces cerevisiae* S288C chromosome III, complete sequence  
>NC\_001136.10 *Saccharomyces cerevisiae* S288C chromosome IV, complete sequence  
>NC\_001137.3 *Saccharomyces cerevisiae* S288C chromosome V, complete sequence  
>NC\_001138.5 *Saccharomyces cerevisiae* S288C chromosome VI, complete sequence  
>NC\_001139.9 *Saccharomyces cerevisiae* S288C chromosome VII, complete sequence  
>NC\_001140.6 *Saccharomyces cerevisiae* S288C chromosome VIII, complete sequence  
>NC\_001141.2 *Saccharomyces cerevisiae* S288C chromosome IX, complete sequence  
>NC\_001142.9 *Saccharomyces cerevisiae* S288C chromosome X, complete sequence  
>NC\_001143.9 *Saccharomyces cerevisiae* S288C chromosome XI, complete sequence  
>NC\_001144.5 *Saccharomyces cerevisiae* S288C chromosome XII, complete sequence  
>NC\_001145.3 *Saccharomyces cerevisiae* S288C chromosome XIII, complete sequence  
>NC\_001146.8 *Saccharomyces cerevisiae* S288C chromosome XIV, complete sequence  
>NC\_001147.6 *Saccharomyces cerevisiae* S288C chromosome XV, complete sequence  
>NC\_001148.4 *Saccharomyces cerevisiae* S288C chromosome XVI, complete sequence

アノテーションファイル (GFF) の確認

\$ more reference/s288c.gff

##gff-version 3

```
#!/gff-spec-version 1.21
#!/processor NC11BI annotwriter
#!/genome-build R64
#!/genome-build-accession NCBI_Assembly:GCF_000146045.2
#!/annotation-source SGD R64-3-1
##sequence-region NC_001133.9 1 230218
##species
https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=559292
```

### スクリプトの確認

```
ls scripts/hisat2*
hisat2.sh hisat2_index.sh
```

### インデックスの作成

```
$ more scripts/hisat2_index.sh
#$ -S /bin/bash
#$ -pe def_slot 2
#$ -cwd
#$ -l mem_req=10G,s_vmem=10G
```

```
conda activate pags_rnaseq ← conda 仮想環境の指定
```

```
GENOME=./reference/s288c.fa
```

```
INDEX=./reference/s288c.fa
```

```
hisat2-build $GENOME $INDEX ← リファレンスゲノムのインデックス化
```

### qsub コマンドのオプション

- S 使用するインタプリタのパス
- pe def\_slot 1 ジョブスロット数
- cwd ホームディレクトリではなく、qsubコマンド実行時のディレクトリで  
ジョブを実行。標準出力 / 標準エラー出力ファイルは、qsubコマンド実行時の  
ディレクトリに出力。
- l 主にキューの選択、メモリ利用上限の変更に使う
  - mem\_req: 使用するメモリの量を宣言する。(ジョブ管理システムUGEの  
ジョブリソース管理に対する宣言)
  - s\_vmem: ジョブが使用可能な仮想メモリの上限值。(OS に対する宣言)

キューの指定: Thin ノードへの投入は、キューの指定は不要。

### コマンドの確認

```
$ hisat2-build --help
HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com,
http://www.ccb.jhu.edu/people/infphilo)
Usage: hisat2-build [options]* <reference_in> <ht2_index_base>
      reference_in          comma-separated list of files with ref
sequences
      hisat2_index_base     write ht2 data to files with this
dir/basename
```

### 実行

```
$ qsub scripts/hisat2_index.sh
```

```
$ ls reference
```

計算が終わると `reference` ディレクトリ内に以下のファイルが作成される。

```
s288c.fa.1.ht2 s288c.fa.2.ht2 s288c.fa.3.ht2 s288c.fa.4.ht2
s288c.fa.5.ht2 s288c.fa.6.ht2 s288c.fa.7.ht2 s288c.fa.8.ht2
```

※ ジョブが終わらなかった場合は、事前実行した以下で確認

```
$ ls outputs/hisat2_index/
```

### 実行ログの確認

```
$ more hisat2_index.sh.e16626048
```

Settings:

Output files: `"/reference/s288c.fa.*.ht2"`

Line rate: 6 (line is 64 bytes)

Lines per side: 1 (side is 64 bytes)

Offset rate: 4 (one in 16)

FTable chars: 10

```
$ more hisat2_index.sh.o16626048
```

Building DifferenceCoverSample

Building sPrime

Building sPrimeOrder

V-Sorting samples

※ ジョブが終わらなかった場合

```
$ more outputs/hisat2_index/hisat2_index.sh.e16626048
```

```
$ more outputs/hisat2_index/hisat2_index.sh.o16626048
```

マッピング

```
$ more scripts/hisat2.sh
```

```
#$ -S /bin/bash
```

```
#$ -pe def_slot 4
```

```
#$ -cwd
```

```
#$ -t 1-2:1 <- アレイジョブの指定
```

```
#$ -l mem_req=8G,s_vmem=8G
```

```
conda activate pags_rnaseq
```

```
# Batch culture: SRR453566
```

```
# chemostat: SRR453569
```

```
ACCESSIONS=(453566 453569)
```

```
no=`expr ${SGE_TASK_ID} - 1` <-${SGE_TASK_ID}アレイジョブのタスク ID 指定
```

```
NUM=${ACCESSIONS[${no}]}
```

```
PREFIX=SRR${NUM}
```

```
# read file
```

```
DIR=./reads/
```

```
QUERY1_1=${DIR}${PREFIX}"_1.fastq.gz"
```

```
QUERY1_2=${DIR}${PREFIX}"_2.fastq.gz"
```

```
hisat2 -p ${NSLOTS} -x reference/s288c.fa --dta ¥
```

```
-1 ${QUERY1_1} -2 ${QUERY1_2} ¥
```

```
-S ${PREFIX}.sam
```

```
# convert sam to bam
```

```
# sort by position
```

```
samtools sort -@ ${NSLOTS} ${PREFIX}.sam -o ${PREFIX}.sorted.bam
```

## コマンドの確認

```
$ hisat2 --help
```

HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com,  
www.ccb.jhu.edu/people/infphilo)

### Usage:

```
hisat2 [options]* -x <ht2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S  
<sam>]
```

<ht2-idx> Index filename prefix (minus trailing .X.ht2).

<m1> Files with #1 mates, paired with files in <m2>.

Could be gzip'ed (extension: .gz) or bzip2'ed

(extension: .bz2).

<m2> Files with #2 mates, paired with files in <m1>.

Could be gzip'ed (extension: .gz) or bzip2'ed

(extension: .bz2).

<r> Files with unpaired reads.

Could be gzip'ed (extension: .gz) or bzip2'ed

(extension: .bz2).

<sam> File for SAM output (default: stdout)

<m1>, <m2>, <r> can be comma-separated lists (no whitespace) and  
can be

specified many times. E.g. '-U file1.fq,file2.fq -U file3.fq'.

```
$ samtools sort --help
```

sort: unrecognized option '--help'

Usage: samtools sort [options...] [in.bam]

### Options:

-l INT Set compression level, from 0 (uncompressed) to 9  
(best)

-u Output uncompressed data (equivalent to -l 0)

-m INT Set maximum memory per thread; suffix K/M/G recognized  
[768M]

-M Use minimiser for clustering unaligned/unplaced reads

-K INT Kmer size to use for minimiser [20]

-n Sort by read name (not compatible with samtools index)



command)

-t TAG      Sort by value of TAG. Uses position as secondary index  
(or read name if -n is set)

-o FILE     Write final output to FILE rather than standard output

-T PREFIX   Write temporary files to PREFIX.nnnn.bam

    --no-PG  
            Do not add a PG line

    --template-coordinate  
            Sort by template-coordinate

    --input-fmt-option OPT[=VAL]  
            Specify a single input file format option in the form  
            of OPTION or OPTION=VALUE

-O, --output-fmt FORMAT[,OPT[=VAL]]...

            Specify output format (SAM, BAM, CRAM)

    --output-fmt-option OPT[=VAL]  
            Specify a single output file format option in the form  
            of OPTION or OPTION=VALUE

    --reference FILE  
            Reference sequence FASTA FILE [null]

-@, --threads INT  
            Number of additional threads to use [0]

    --write-index  
            Automatically index the output files [off]

    --verbosity INT  
            Set level of verbosity

## **SAM / BAM** ファイル

[https://en.wikipedia.org/wiki/SAM\\_\(file\\_format\)](https://en.wikipedia.org/wiki/SAM_(file_format))

## 実行

```
$ qsub scripts/hisat2.sh
```

```
$ ls
```

計算が終わるとカレントディレクトリに以下のファイルが作成される。

```
hisat2.sh.e16626062.1 hisat2.sh.e16626062.2
```

```
hisat2.sh.o16626062.1 hisat2.sh.o16626062.2
```

```
hisat2.sh.pe16626062.1 hisat2.sh.pe16626062.2
```

```
hisat2.sh.po16626062.1 hisat2.sh.po16626062.2
```

```
SRR453566.sam SRR453566.sorted.bam
```

```
SRR453569.sam SRR453569.sorted.bam
```

※ ジョブが終わらなかった場合

```
$ ls outputs/hisat2/
```

## 結果ファイルの確認

```
$ more hisat2.sh.e16626062.1
```

```
5725730 reads; of these:
```

```
5725730 (100.00%) were paired; of these:
```

```
1222756 (21.36%) aligned concordantly 0 times
```

```
4258780 (74.38%) aligned concordantly exactly 1 time
```

```
244194 (4.26%) aligned concordantly >1 times
```

```
----
```

```
1222756 pairs aligned concordantly 0 times; of these:
```

```
128491 (10.51%) aligned discordantly 1 time
```

```
----
```

```
1094265 pairs aligned 0 times concordantly or discordantly; of  
these:
```

```
2188530 mates make up the pairs; of these:
```

```
1470694 (67.20%) aligned 0 times
```

```
662896 (30.29%) aligned exactly 1 time
```

```
54940 (2.51%) aligned >1 times
```

```
87.16% overall alignment rate
```

```
[bam_sort_core] merging from 4 files and 4 in-memory blocks...
```

```
$ more SRR453566.sam
```

---ファイルの中身はターミナル上でご確認ください。----

※ ジョブが終わらなかった場合

```
$ more outputs/hisat2/hisat2.sh.e16626062.1
```

```
$ more outputs/hisat2/SRR453566.sam
```

## 発現量の算出 (stringtie)

```
$ more scripts/stringtie.sh
```

```
#$ -S /bin/bash
```

```

#$ -pe def_slot 4
#$ -cwd
#$ -l mem_req=8G,s_vmem=8G

conda activate pags_rnaseq

ACCESSIONS=(453566 453569)
for NUM in ${ACCESSIONS[@]}
do
    PREFIX=SRR${NUM}
    BAM=${PREFIX}"/sorted.bam"
    stringtie -e -B -p ${NSLOTS} ¥
        -G reference/s288c.gff -o
ballgown/${PREFIX}/${PREFIX}.out.gtf -A ${PREFIX}.gene_abund.tab $BAM
done

```

### コマンドの確認

```
$ stringtie --help
```

StringTie v2.2.1 usage:

```

stringtie <in.bam ..> [-G <guide_gff>] [-l <prefix>] [-o <out.gtf>]
[-p <cpus>]
[-v] [-a <min_anchor_len>] [-m <min_len>] [-j <min_anchor_cov>] [-f
<min_iso>]
[-c <min_bundle_cov>] [-g <bdist>] [-u] [-L] [-e] [--viral] [-E
<err_margin>]
[--ptf <f_tab>] [-x <seqid,..>] [-A <gene_abund.out>] [-h] {-B|-b
<dir_path>}
[--mix] [--conservative] [--rf] [--fr]
Assemble RNA-Seq alignments into potential transcripts.

```

### 実行

```
$ qsub scripts/stringtie.sh
```

計算が終わるとカレントディレクトリ内に以下のファイルが作成される

```
$ ls
```

```
ballgown
SRR453566.gene_abund.tab
SRR453569.gene_abund.tab
stringtie.sh.e16626115
stringtie.sh.o16626115
stringtie.sh.pe16626115
stringtie.sh.po16626115
ls ballgown/*
ballgown/SRR453566:
SRR453566.out.gtf e2t.ctab e_data.ctab i2t.ctab i_data.ctab
t_data.ctab

ballgown/SRR453569:
SRR453569.out.gtf e2t.ctab e_data.ctab i2t.ctab i_data.ctab
t_data.ctab
```

※ジョブが終わらなかった場合

```
$ ls outputs/stringtie/
$ ls outputs/stringtie/ballgown/*
```

### 結果ファイルの確認

```
$ more SRR453566.gene_abund.tab
```

---ファイルの中身はターミナル上でご確認ください。---

発現量のノーマライズ

FPKM: Fragments Per Kilobase of exon per Million reads mapped

TPM: Transcripts Per kilobase Milion

FPKMもTPMも以下の二つで補正するが、補正する順番が異なる。

(1) 総リード数での補正 (総リード数 100万)

(2) 遺伝子長での補正 (遺伝子長 1000b)

FPKM (1) -> (2)

TPM (2) -> (1)

※ジョブが終わらなかった場合

```
$ more outputs/stringtie/SRR453566.gene_abund.tab
```

次のステップとして、R などを用いることで可視化などができる。  
ballgown を使う場合は、このballgown ディレクトリを  
そのまま入力データとして使用できる。

## Singularity を利用したスクリプトの確認

```
$ more scripts/hisat2_singularity.sh
#$ -S /bin/bash
#$ -pe def_slot 4
#$ -cwd
#$ -t 1-2:1
#$ -l mem_req=8G,s_vmem=8G
```

```
# Batch culture: SRR453566
# chemostat: SRR453569
ACCESSIONS=(453566 453569)
no=`expr ${SGE_TASK_ID} - 1`
```

```
NUM=${ACCESSIONS[${no}]}
PREFIX=SRR${NUM}
```

```
# read file
DIR=./reads/
QUERY1_1=${DIR}${PREFIX}"_1.fastq.gz"
QUERY1_2=${DIR}${PREFIX}"_2.fastq.gz"
```

```
singularity exec /usr/local/biotools/h/hisat2:2.2.1--py38he1b5a44_0
```

```
hisat2 -p ${NSLOTS} -x reference/s288c.fa --dta ¥
    -1 ${QUERY1_1} -2 ${QUERY1_2} ¥
    -S ${PREFIX}.sam
```

```
# convert sam to bam
```

```
# sort by position
```

```
singularity exec /usr/local/biotools/s/samtools:1.16.1--h6899075_0
```

```
samtools sort -@ ${NSLOTS} ${PREFIX}.sam -o ${PREFIX}.sorted.bam
```