

# 遺伝研スパコンでの解析の実践

## RNA-seq 解析

解析手順構築  
データ準備  
資料作成  
その他

望月孝子さん（遺伝研）

代理発表

東光一（遺伝研）

# RNA-seq とは

RNA-Seqとは、次世代シーケンサーを用いて mRNAなどの配列情報を網羅的に読み、得られた配列情報から遺伝子の発現量を解析できる手法。

## 次世代シーケンサー

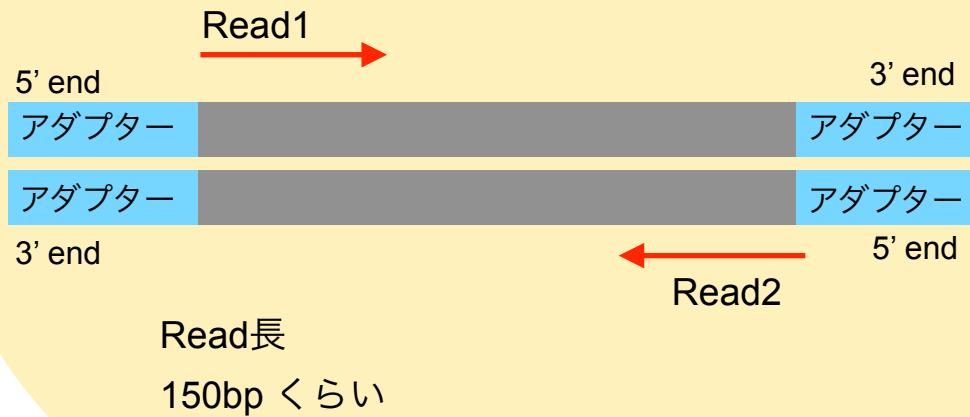
RNA-seq における代表機種



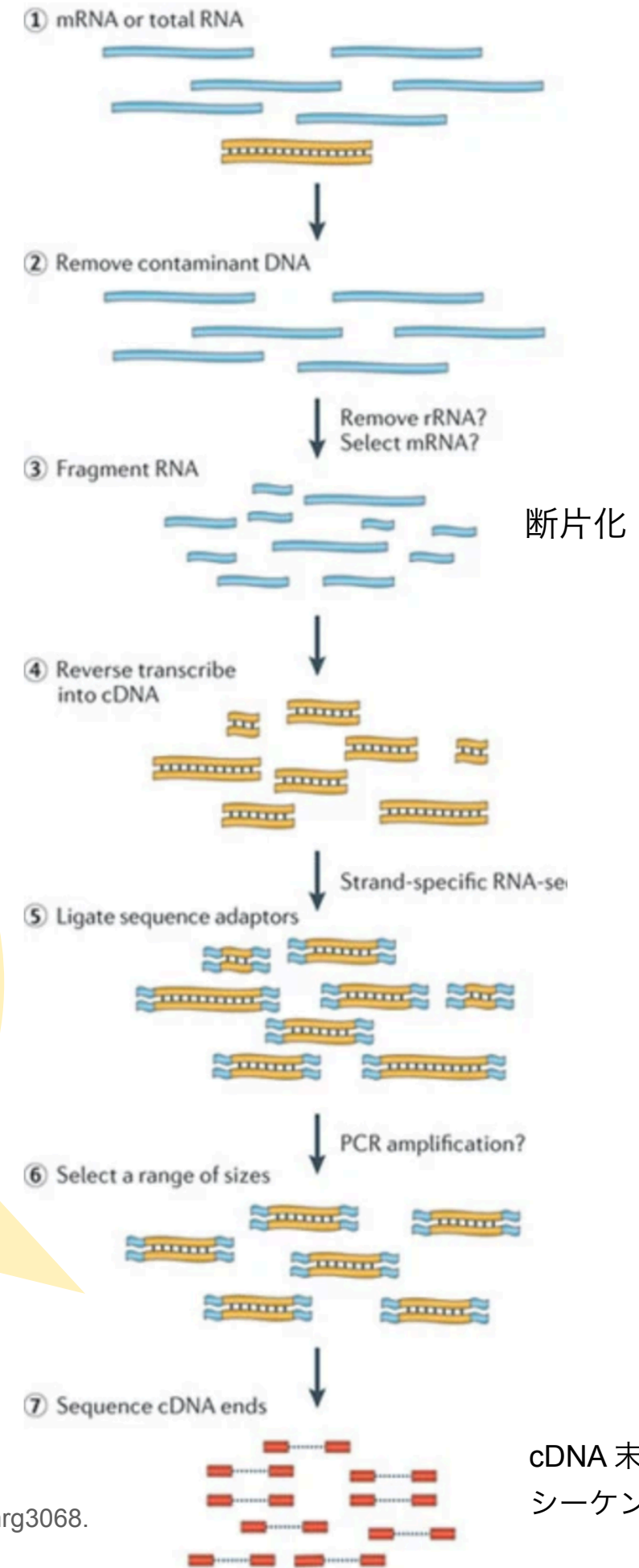
Illumina 社 NextSeq 550 システム

<https://jp.illumina.com/systems/sequencing-platforms.html>

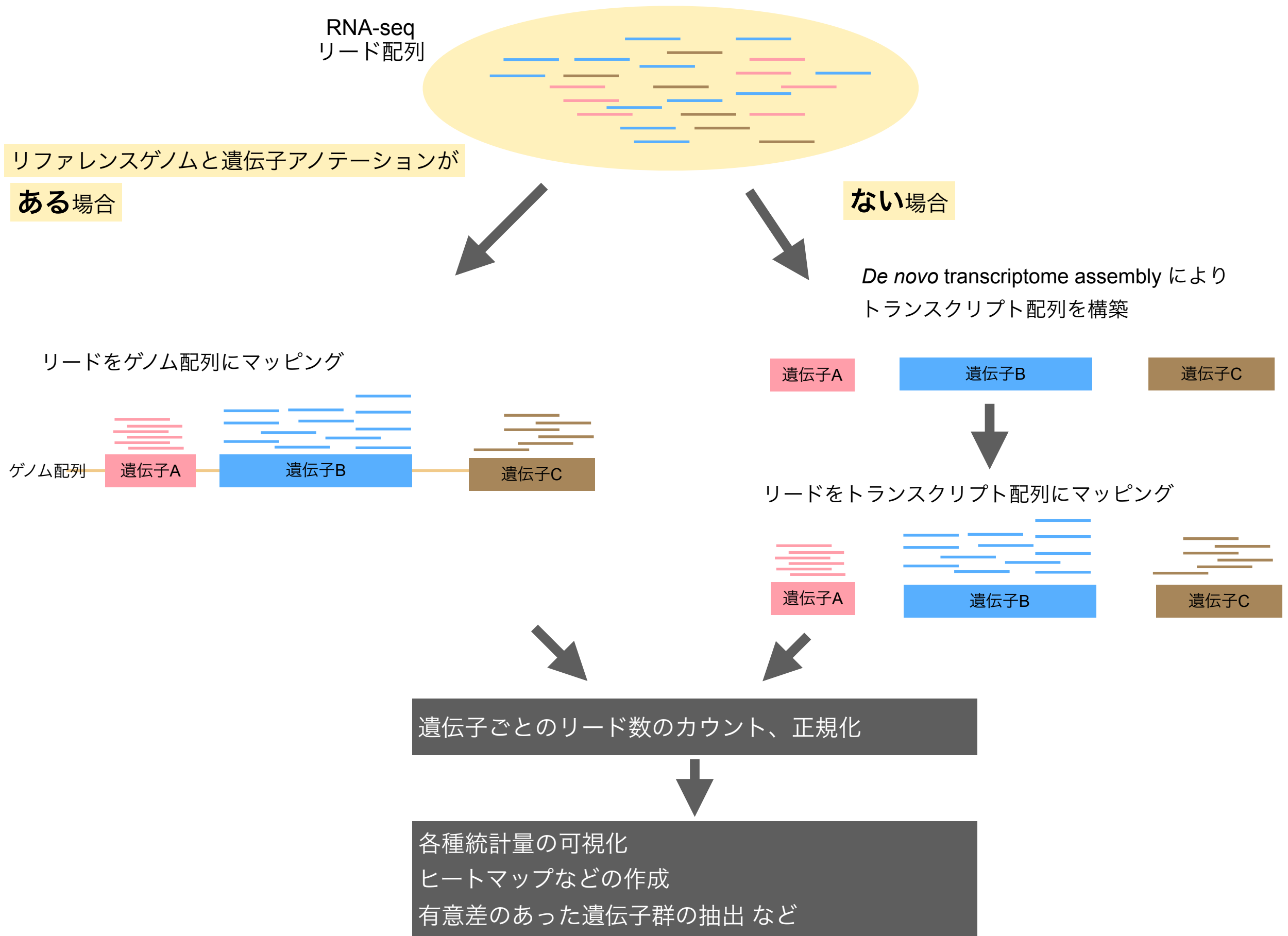
ペアードエンドリード



## RNA-seq 取得手順



# RNA-seq による発現量解析



# DDBJ Sequence Read Archive (SRA)

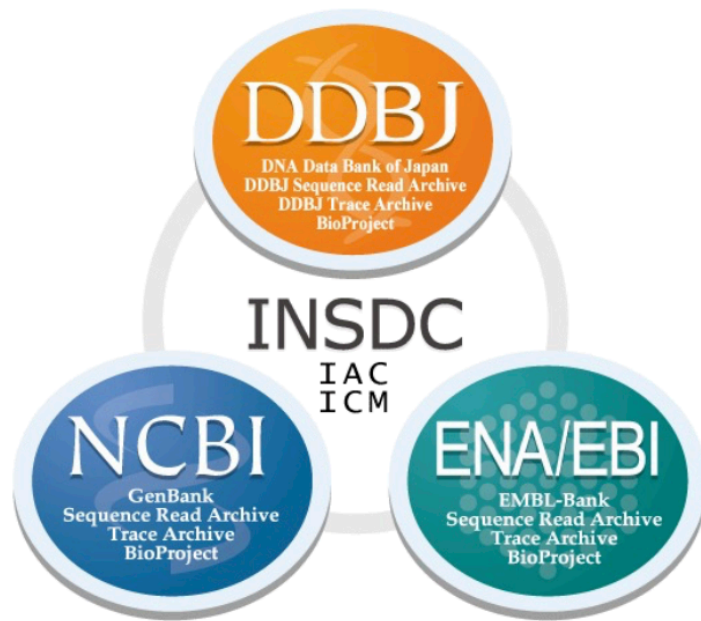


INSDC (International Nucleotide Sequence Database Collaboration) の一員として、塩基配列データを収集・公開。

NCBI (米) および ENA/EBI (欧) との間でデータの共有。

次世代シーケンサーからの出力データ (生データ) は、Sequence Read Archive に登録されています。

<https://www.ddbj.nig.ac.jp/dra/index.html>



## Sequence Read Archive

Home Overview ▼ FAQ Search Downloads ▼ About DRA

ホーム > dra > Sequence Read Archive

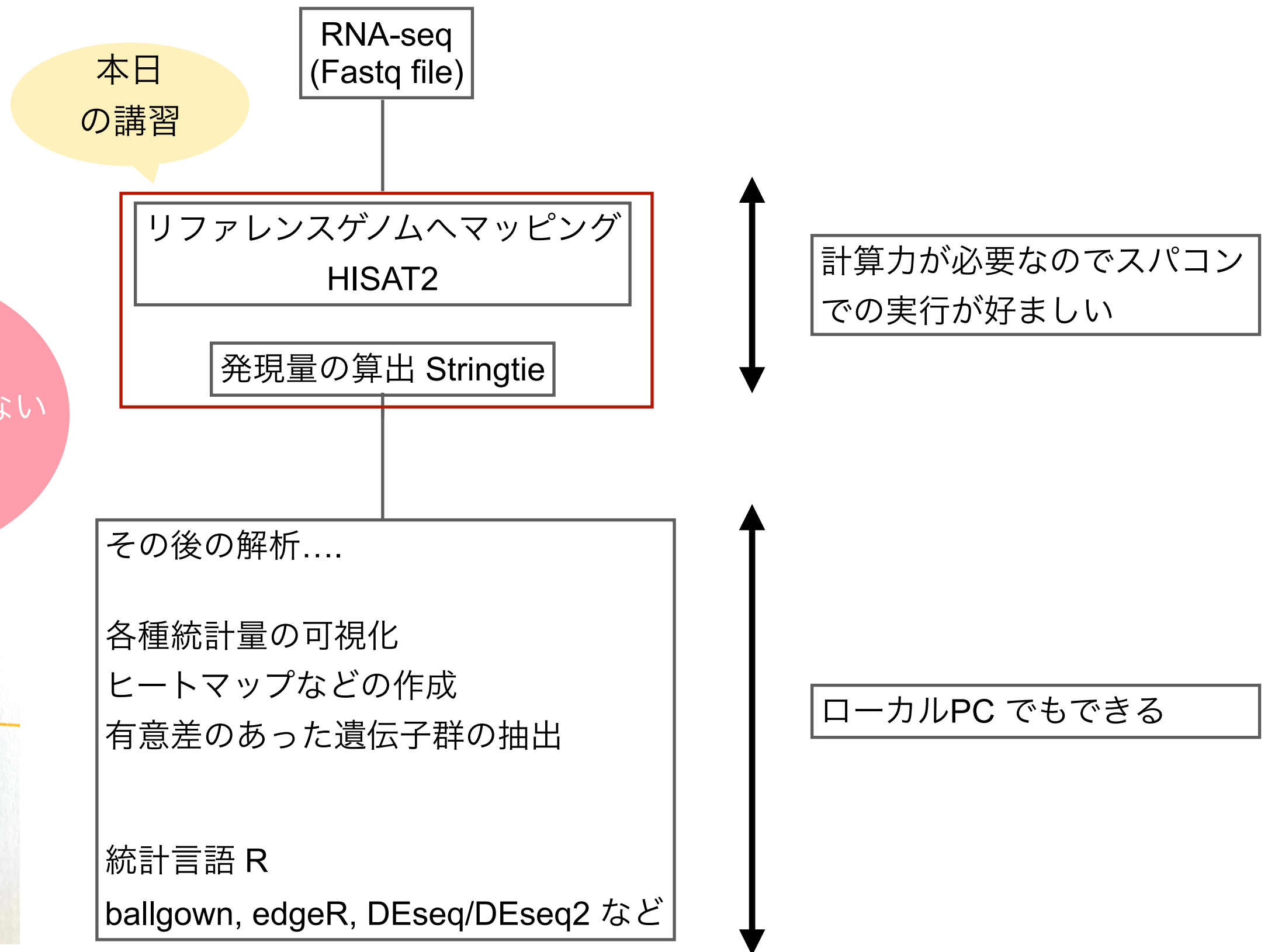
DDBJ Sequence Read Archive (DRA) は科学研究の再現性担保、及び、データ解析による新たな発見を支えるために生シーケンスデータとアライメント情報をアーカイブしています。DRA は [International Nucleotide Sequence Database Collaboration \(INSDC\)](#) のメンバーであり、[NCBI Sequence Read Archive \(SRA\)](#) [\[1\]](#) と [EBI Sequence Read Archive \(ERA\)](#) [\[2\]](#) との国際協力のもと、運営されています。



このスライドは、遺伝研 谷澤助教のスライドを参考にさせて頂きました。

実験をする前に検索すると、欲しいデータが見つかるかもしれません。

# 本日の講習 ゲノム配列を利用したRNA-seq 発現量解析



講習内では  
qsub は実行しないでください。



# 配布データのコピー

---

ホームディレクトリ に移動

```
$ cd
```

ホームディレクトリ に20231030 というファイル/ディレクトリがないかを確認。

```
$ ls 20231030
```

```
ls: 20231030 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
```

あれば、講習時間だけファイル/ディレクトリ 名前を一時的に変更してください。

```
$ mv 20231030 20231030_tmp
```

講習データをコピー

```
$ cp -r /usr/local/shared_data/lecture/20231030 .
```

本日の講習はこちらのディレクトリで

```
$ cd 20231030
```

# 配布データの確認 (1)

```
$ ls -al
```

```
合計 24
```

```
drwxr-xr-x  6 koshu3 koshu 4096 10月 16 15:30 .  
drwxr-x--- 13 koshu3 koshu 4096 10月 13 14:50 ..  
drwxr-xr-x  5 koshu3 koshu 4096 10月 16 15:28 outputs  
drwxr-xr-x  2 koshu3 koshu 4096 10月 16 10:53 reads  
drwxr-xr-x  2 koshu3 koshu 4096 10月 16 11:47 reference  
drwxr-xr-x  2 koshu3 koshu 4096 10月 17  2022 scripts
```

事前に実行した  
結果ファイル

講習用リード  
ファイル

リファレンス  
ファイル

実行スクリプト

それぞれのディレクトリ の中身は

```
$ ls outputs
```

```
$ ls outputs/hisat2_index
```

```
$ ls outputs/hisat2
```

```
$ ls outputs/stringtie
```

```
$ ls reads
```

```
$ ls reference
```

```
$ ls scripts
```

## 配布データの確認 (2)

---

配布データの構成

20231030

|\_\_\_\_ 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

|\_\_\_\_ hisat2\_index.sh # condaにて実装

|\_\_\_\_ stringtie.sh # condaにて実装

|\_\_\_\_ hisat2\_singularity.sh # singularityにて実装

|\_\_\_\_ hisat2\_index\_singularity.sh # singularityにて実装

|\_\_\_\_ stringtie\_singularity.sh # singularityにて実装



# Conda activate

---

```
$ conda activate pags_rnaseq
```

# 講習用 RNA-seq データ

## JOURNAL ARTICLE

### A comprehensive comparison of RNA-Seq-based transcriptome analysis from reads to differential gene expression and cross-comparison with microarrays: a case study in *Saccharomyces cerevisiae*

Intawat Nookaew, Marta Papini, Natapol Pornputtapong, Gionata Scalcinati, Linn Fagerberg, Matthias Uhlén, Jens Nielsen  [Author Notes](#)

*Nucleic Acids Research*, Volume 40, Issue 20, 1 November 2012, Pages 10084–10097, <https://doi.org/10.1093/nar/gks804>

**Published:** 08 September 2012 **Article history** ▼

*Saccharomyces cerevisiae* CEN.PK113-7D

バッチ培養

ケモスタット培養

SRR453566

SRR453569

SRR453567

SRR453570

SRR453568

SRR453571

Biological replicates 3回ずつ

```
$ ls reads/
```

```
SRR453566_1.fastq.gz SRR453566_2.fastq.gz
```

```
SRR453569_1.fastq.gz SRR453569_2.fastq.gz
```

Paired-end データ

Paired-end データ

# FASTQ フォーマット

4行で1配列の情報を表す。

```
$ zcat reads/SRR453566_1.fastq.gz | more
```

@SRR453566.1 HWI-ST167:4:1101:1597:1986/1

NAAAACTTTGGATGACTTCAACAACATTCTTCTGAAATCAACAAAATATCACCAACTTCCGCCAACACAAAGTCTTACAGTGCAACAACAAGTGATGTTG

+

[illegible]

1行目: @ の後ろにその配列のID

2行目: 配列

3行目: + を記載する。(配列のID を記載してもしなくてもよい)

4行目: その配列のクオリティ値

クオリティ値はアスキーコードで表示  
アスキー値 - 33 が クオリティ値



## クオリティ値 @ の場合

$$64 - 33 = 31$$

S - Sanger	Phred+33, raw reads typically (0, 40)
X - Solexa	Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+	Phred+64, raw reads typically (0, 40)
J - Illumina 1.5+	Phred+64, raw reads typically (3, 41)
	with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
	(Note: See discussion above).
L - Illumina 1.8+	Phred+33, raw reads typically (0, 41)
N - Nanopore	Phred+33, Duplex reads typically (0, 50)
P - PacBio	Phred+33, HiFi reads typically (0, 93)

## <- SRA

[https://en.wikipedia.org/wiki/FASTQ\\_format](https://en.wikipedia.org/wiki/FASTQ_format)

# 講習用 リファレンスファイル ゲノム配列 fasta

```
$ ls reference/  
s288c.fa s288c.gff
```

ゲノム配列

ファイルの中身を確認

```
$ more reference/s288c.fa  
>NC_001133.9 Saccharomyces cerevisiae S288C chromosome I, complete sequence  
CCACACCACACCCACACACCCACACACCACACACCACACCCACACACACATCCTAACA  
CTACCCTAACACAGCCCTAATCTAACCCTGGCCAACCTGTCTCTCAACTTACCCTCCATTACCCTGCCTC  
CACTCGTTACCCTGTCCCATTCAACCATAACCACTCCGAACCACCATCCATCCCTCTACTTACTACCACTC  
ACCCACCGTTACCCTCCAATTACCCATATCCAACCCACTGCCACTTACCCTACCATTACCCTA
```

染色体16本  
分の配列

FASTA ヘッダの出力

```
$ 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
```

# 講習用 リファレンスファイル アノテーションファイル

```
$ ls reference/  
s288c.fa s288c.gff
```

アノテ  
ーションファイル

ファイルの中身を確認

```
$ more reference/s288c.gff  
##gff-version 3  
#!gff-spec-version 1.21  
#!processor NCBI 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  
NC_001133.9 RefSeq region 1 230218 . + . ID=NC_001133.9:1..230218;Dbxref=taxon  
NC_001133.9 RefSeq telomere 1 801 . - . ID=id-NC_001133.9:1..801;Dbxref=SGD:S0  
NC_001133.9 RefSeq origin_of_replication 707 776 . + . ID=id-NC_001133.9:707..  
NC_001133.9 RefSeq gene 1807 2169 . - . ID=gene-YAL068C;Dbxref=GeneID:851229;M  
NC_001133.9 RefSeq mRNA 1807 2169 . - . ID=rna-NM_001180043.1;Parent=gene-YAL0  
NC_001133.9 RefSeq exon 1807 2169 . - . ID=exon-NM_001180043.1-1;Parent=rna-NM  
NC_001133.9 RefSeq CDS 1807 2169 . - 0 ID=cds-NP_009332.1;Parent=rna-NM_001180
```

# GFF フォーマット

## 遺伝子アノテーションのフォーマット

```
##gff-version 3
#!gff-spec-version 1.21
#!processor NCBI annotwriter
#!genome-build R64
#!genome-build-accession NCBI_Assembly:GCF_000146045.2
#!annotation-source SGD R64-2-1
##sequence-region NC_001133.9 1 230218
##species https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?id=559292
NC_001133.9 RefSeq gene 1807 2169 . - . ID=gene0;Dbxref=GeneID:851229;Name=PAU8;end_range=2169,.;gbkey=Gene;gene=PAU8;
NC_001133.9 RefSeq mRNA 1807 2169 . - . ID=rna0;Parent=gene0;Dbxref=GeneID:851229,Genbank:NM_001180043.1;Name=NM_001180043.1;gbkey=
NC_001133.9 RefSeq exon 1807 2169 . - . ID=id3;Parent=rna0;Dbxref=GeneID:851229,Genbank:NM_001180043.1;end_range=2169,.;gbkey=
NC_001133.9 RefSeq CDS 1807 2169 . - 0 ID=cds0;Parent=rna0;Dbxref=SGD:S000002142,GeneID:851229,Genbank:NP_009332.1;Name=NP_009332.1;gbkey=
NC_001133.9 RefSeq gene 2480 2707 . + . ID=gene1;Dbxref=GeneID:1466426;Name=YAL067W-A;end_range=2707,.;gbkey=Gene;gene=YAL067W-A;
NC_001133.9 RefSeq mRNA 2480 2707 . + . ID=rna1;Parent=gene1;Dbxref=GeneID:1466426,Genbank:NM_001184582.1;Name=NM_001184582.1;gbkey=
NC_001133.9 RefSeq exon 2480 2707 . + . ID=id4;Parent=rna1;Dbxref=GeneID:1466426,Genbank:NM_001184582.1;end_range=2707,.;gbkey=
NC_001133.9 RefSeq CDS 2480 2707 . + 0 ID=cds1;Parent=rna1;Dbxref=SGD:S000028593,GeneID:1466426,Genbank:NP_878038.1;Name=NP_878038.1;gbkey=
NC_001133.9 RefSeq gene 7235 9016 . - . ID=gene2;Dbxref=GeneID:851230;Name=SEO1;end_range=9016,.;gbkey=Gene;gene=SEO1;
NC_001133.9 RefSeq mRNA 7235 9016 . - . ID=rna2;Parent=gene2;Dbxref=GeneID:851230,Genbank:NM_001178208.1;Name=NM_001178208.1;gbkey=
NC_001133.9 RefSeq exon 7235 9016 . - . ID=id5;Parent=rna2;Dbxref=GeneID:851230,Genbank:NM_001178208.1;end_range=9016,.;gbkey=
NC_001133.9 RefSeq CDS 7235 9016 . - 0 ID=cds2;Parent=rna2;Dbxref=SGD:S000000062,GeneID:851230,Genbank:NP_009333.1;Name=NP_009333.1;gbkey=
```

タブ区切りフォーマット。値がない場合は、"." が設定される。

1. seqname : 染色体 or スキャフォールドの名前
2. source : アノテーションを生成したプログラムまたはデータソースの名前
3. feature : フィーチャータイプ (mRNA, gene, exon, CDS ....)
4. start : スタートポジション (1bp ~)
5. end : エンドポジション (1bp ~)
6. score : スコア
7. strand : +(forward)、-(reverse)または '.'
8. frame : 翻訳フレーム (0, 1, 2)
9. attribute : 追加情報。セミコロンで区切られたタグと値のペアのリスト。



# リファレンスゲノムへリードをマッピング

---

## ステップ

1. リファレンスゲノムのインデックスを作成

hisat2\_index.sh

2. リードをリファレンスゲノムへマッピング

hisat2.sh      2サンプル分をアレイジョブで同時実行

## スクリプト

```
$ ls scripts/hisat2*  
scripts/hisat2.sh  scripts/hisat2_index.sh  
scripts/hisat2_index_singularity.sh  scripts/hisat2_singularity.sh
```

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

```
conda activate pags_rnaseq
```

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

```
hisat2-build $GENOME $INDEX
```

conda 仮想環境  
の指定

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値。(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。



# マッピング用インデックスの作成

# オプションの確認

```
$ hisat2-build -h
```

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

Options:

-c	reference sequences given on cmd line (as <reference_in>)
--large-index	force generated index to be 'large', even if ref has fewer than 4 billion nucleotides
-a/--noauto	disable automatic -p/--bmax/--dcv memory-fitting
-p <int>	number of threads
--bmax <int>	max bucket sz for blockwise suffix-array builder
--bmaxdivn <int>	max bucket sz as divisor of ref len (default: 4)
--dcv <int>	diff-cover period for blockwise (default: 1024)
--nodc	disable diff-cover (algorithm becomes quadratic)
-r/--noref	don't build .3/.4.ht2 (packed reference) portion
-3/--justref	just build .3/.4.ht2 (packed reference) portion
-o/--offrate <int>	SA is sampled every 2^offRate BWT chars (default: 5)
-t/--ftabchars <int>	# of chars consumed in initial lookup (default: 10)
--localoffrate <int>	SA (local) is sampled every 2^offRate BWT chars (default: 3)
--localftabchars <int>	# of chars consumed in initial lookup in a local index (default: 6)
--snp <path>	SNP file name
--haplotype <path>	haplotype file name
--ss <path>	Splice site file name
--exon <path>	Exon file name
--repeat-ref <path>	Repeat reference file name
--repeat-info <path>	Repeat information file name
--repeat-snp <path>	Repeat snp file name
--repeat-haplotype <path>	Repeat haplotype file name
--seed <int>	seed for random number generator
-q/--quiet	disable verbose output (for debugging)
-h/--help	print detailed description of tool and its options
--usage	print this usage message
--version	print version information and quit

# マッピング用インデックスの作成

# 実行

## 実行

講習中はqsub を実行しないでください。

```
$ qsub scripts/hisat2_index.sh
Your job 24706258 ("hisat2_index.sh") has been submitted
```

## ステータスの確認

```
$ qstat
job-ID      prior    name          user      state submit/start at   queue                jclass slots ja-task-ID
-----
 24704406   0.25025 QLOGIN        koshu3    r       10/16/2023 10:42:12 login.q@at138         1
 24706258   0.00000 hisat2_ind    koshu3    qw      10/16/2023 11:47:09                2
```

実行  
待機

```
$ qstat
job-ID      prior    name          user      state submit/start at   queue                jclass slots ja-task-ID
-----
 24704406   0.25025 QLOGIN        koshu3    r       10/16/2023 10:42:12 login.q@at138         1
 24706258   0.25023 hisat2_ind    koshu3    r       10/16/2023 11:47:12 epyc.q@at147         2
```

実行  
中

```
$ qstat
job-ID      prior    name          user      state submit/start at   queue                jclass slots ja-task-ID
-----
 24704406   0.25026 QLOGIN        koshu3    r       10/16/2023 10:42:12 login.q@at138         1
```

ジョブが終了すると該当ジョブID が表示されなくなる。

# マッピング用インデックスの作成

# 実行結果の確認

## ログの確認

```
$ ls -al
合計 32
drwxr-xr-x  6 koshu3 koshu 4096 10月 16 11:58 .
drwxr-xr-x 13 koshu3 koshu 4096 10月 13 14:50 ..
-rw-r--r--  1 koshu3 koshu 2262 10月 16 11:47 hisat2_index.sh.e24706258
-rw-r--r--  1 koshu3 koshu 3919 10月 16 11:47 hisat2_index.sh.o24706258
-rw-r--r--  1 koshu3 koshu    0 10月 16 11:47 hisat2_index.sh.pe24706258
-rw-r--r--  1 koshu3 koshu    0 10月 16 11:47 hisat2_index.sh.po24706258
drwxr-xr-x  2 koshu3 koshu 4096 10月 16 11:58 outputs
drwxr-xr-x  2 koshu3 koshu 4096 10月 16 10:53 reads
drwxr-xr-x  2 koshu3 koshu 4096 10月 16 11:47 reference
drwxr-xr-x  2 koshu3 koshu 4096 10月 17  2022 scripts
```

```
$ more hisat2_index.sh.e24706258
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
  .
  .
  .
Total time for call to driver() for forward index: 00:00
```

```
$ more hisat2_index.sh.o24706258
Building DifferenceCoverSample
Building sPrime
Building sPrimeOrder
V-Sorting samples
  .
  .
  .
Returning block of 1908813 for bucket 7
```

## インデックスが作成された

```
$ ls -al reference/
合計 46228
drwxr-xr-x  2 koshu3 koshu    4096 10月 16 11:47 .
drwxr-xr-x  6 koshu3 koshu    4096 10月 16 11:58 ..
-rw-r--r--  1 koshu3 koshu 12245035 10月 17  2022 s288c.fa
-rw-r--r--  1 koshu3 koshu  8219756 10月 16 11:47 s288c.fa.1.ht2
-rw-r--r--  1 koshu3 koshu  3017836 10月 16 11:47 s288c.fa.2.ht2
-rw-r--r--  1 koshu3 koshu    152 10月 16 11:47 s288c.fa.3.ht2
-rw-r--r--  1 koshu3 koshu  3017832 10月 16 11:47 s288c.fa.4.ht2
-rw-r--r--  1 koshu3 koshu  5357645 10月 16 11:47 s288c.fa.5.ht2
-rw-r--r--  1 koshu3 koshu  3071004 10月 16 11:47 s288c.fa.6.ht2
-rw-r--r--  1 koshu3 koshu    12 10月 16 11:47 s288c.fa.7.ht2
-rw-r--r--  1 koshu3 koshu     8 10月 16 11:47 s288c.fa.8.ht2
-rw-r--r--  1 koshu3 koshu 12377219 10月 17  2022 s288c.gff
```

## 事前実行した結果で確認

```
$ ls outputs/hisat2_index/
$ more outputs/hisat2_index/hisat2_index.sh.e24706258
$ more outputs/hisat2_index/hisat2_index.sh.o24706258
```

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

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

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

```
#$ -cwd
```

```
#$ -t 1-2:1
```

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

アレイジョブを  
指定

```
conda activate pags_rnaseq
```

```
# Batch culture: SRR453566
```

```
# chemostat: SRR453569
```

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

```
no=`expr ${SGE_TASK_ID} - 1`
```

アレイジョブ  
のタスクID

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

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

```
# read file
```

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

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

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

—dta: reports alignments tailored  
for transcript assemblers

```
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
```

\$ **hisat2 --help**

HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com, [www.ccb.jhu.edu/people/infphilo](http://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'.

Options (defaults in parentheses):

Input:

```
-q           query input files are FASTQ .fq/.fastq (default)
--qseq      query input files are in Illumina's qseq format
-f          query input files are (multi-)FASTA .fa/.mfa
-r          query input files are raw one-sequence-per-line
-c          <m1>, <m2>, <r> are sequences themselves, not files
-s/--skip <int> skip the first <int> reads/pairs in the input (none)
-u/--upto <int> stop after first <int> reads/pairs (no limit)
-5/--trim5 <int> trim <int> bases from 5'/left end of reads (0)
-3/--trim3 <int> trim <int> bases from 3'/right end of reads (0)
--phred33   qualities are Phred+33 (default)
--phred64   qualities are Phred+64
--int-quals qualities encoded as space-delimited integers
```

Presets:

Same as:

```
--fast      --no-repeat-index
--sensitive --bowtie2-dp 1 -k 30 --score-min L,0,-0.5
--very-sensitive --bowtie2-dp 2 -k 50 --score-min L,0,-1
```

Alignment:

```
--bowtie2-dp <int> use Bowtie2's dynamic programming alignment algorithm (0) - 0: no dynamic programming, 1: conditional
dynamic programming, and 2: unconditional dynamic programming (slowest)
--n-ceil <func>    func for max # non-A/C/G/Ts permitted in aln (L,0,0.15)
--ignore-quals     treat all quality values as 30 on Phred scale (off)
--nofw            do not align forward (original) version of read (off)
--norc           do not align reverse-complement version of read (off)
--no-repeat-index do not use repeat index
```

```
$ 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 フォーマット/ BAM フォーマット

```
@HD VN:1.0 SO:unsorted
@SQ SN:NC_001133.9 LN:230218
@SQ SN:NC_001134.8 LN:813184
@SQ SN:NC_001135.5 LN:316620
@SQ SN:NC_001136.10 LN:1531933
@SQ SN:NC_001137.3 LN:576874
@SQ SN:NC_001138.5 LN:270161
@SQ SN:NC_001139.9 LN:1090940
@SQ SN:NC_001140.6 LN:562643
@SQ SN:NC_001141.2 LN:439888
@SQ SN:NC_001142.9 LN:745751
@SQ SN:NC_001143.9 LN:666816
@SQ SN:NC_001144.5 LN:1078177
@SQ SN:NC_001145.3 LN:924431
@SQ SN:NC_001146.8 LN:784333
@SQ SN:NC_001147.6 LN:1091291
@SQ SN:NC_001148.4 LN:948066
@SQ SN:NC_001224.1 LN:85779
```

@ヘッダ行

HD: ヘッダ行 SAMフォーマットのバージョンなど

SQ: リファレンスの情報

PG ツールの実行情報

```
@PG ID:hisat2 PN:hisat2 VN:2.2.1 CL:"/home/koshu3/miniconda3/envs/pags_rnaseq/bin/hisat2-align-s --wrapper basic-0 -p 4 -x referen
SRR453566.24 163 NC_001139.9 727518 60 69M = 727620 203 TTAATCAAG... =DFFFFHHHH... AS:i:0 XN:i:0 XM:i:0 XO:i:0 XG:i:0 NM:i:0 MD
SRR453566.22 99 NC_001142.9 509705 60 101M = 509740 136 CAAAGCGTA... CCCFFFFFFHG... AS:i:-6 XN:i:0 XM:i:1 XO:i:0 XG:i:0 NM
SRR453566.22 147 NC_001142.9 509740 60 101M = 509705 -136 GGTATATTT... @DA@:>>>@C... AS:i:-4 ZS:i:-7 XN:i:0 XM:i:1 XO
SRR453566.23 99 NC_001134.8 674240 60 101M = 674286 131 TTTTCTTCA... @BCFFFFFFHH... AS:i:-6 XN:i:0 XM:i:1 XO:i:0 XG:i:0 NM
SRR453566.23 147 NC_001134.8 674286 60 85M = 674240 -131 AACAAAAGC... ??>C>@5(@>... AS:i:-4 ZS:i:-10 XN:i:0 XM:i:1 XO
```

QNAME FLG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL optional fields

QNAME	リード名
FLG	アラインメント情報。参考 <a href="https://broadinstitute.github.io/picard/explain-flags.html">https://broadinstitute.github.io/picard/explain-flags.html</a>
RNAME	マップされたリファレンス名
POS	マップポジション
MAPQ	マッピングスコア
CIGAR	マッピングの状況 ex) M アライメントマッチ I リファレンスにインサクションあり など
RNEXT	ペアエンドの場合、ペアのリード名 (= QNAME)。
PNEXT	ペアエンドの場合、ペアのマップされた開始位置。
TLEN	ペアエンドのリード間の距離。
SEQ	FASTQ の塩基配列データ
QUAL	FASTQ のクオリティデータ。

BAM は SAM をバイナリ形式にしたファイル

# マッピング

# 実行

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

```
Your job-array 24707186.1-2:1 ("hisat2.sh") has been submitted
```

講習中はqsub を実行しないでください。

```
$ qstat
```

job-ID	prior	name	user	state	submit/start at	queue	jclass	slots	ja-task-ID
24704406	0.25081	QLOGIN	koshu3	r	10/16/2023 10:42:12	login.q@at138		1	
24707186	0.25068	hisat2.sh	koshu3	r	10/16/2023 14:16:24	intel.q@it008		4	1
24707186	0.25068	hisat2.sh	koshu3	qw	10/16/2023 14:15:32			4	2

```
$ ls -al
```

```
合計 8310040
```

```
drwxr-xr-x  6 koshu3 koshu          4096 10月 16 14:19 .
drwxr-x--- 13 koshu3 koshu          4096 10月 13 14:50 ..
-rw-r--r--  1 koshu3 koshu 4173908395 10月 16 14:17 SRR453566.sam
-rw-r--r--  1 koshu3 koshu  821523448 10月 16 14:18 SRR453566.sorted.bam
-rw-r--r--  1 koshu3 koshu 2922686962 10月 16 14:19 SRR453569.sam
-rw-r--r--  1 koshu3 koshu  591294623 10月 16 14:19 SRR453569.sorted.bam
-rw-r--r--  1 koshu3 koshu          693 10月 16 14:18 hisat2.sh.e24707186.1
-rw-r--r--  1 koshu3 koshu          688 10月 16 14:19 hisat2.sh.e24707186.2
-rw-r--r--  1 koshu3 koshu           0 10月 16 14:16 hisat2.sh.o24707186.1
-rw-r--r--  1 koshu3 koshu           0 10月 16 14:18 hisat2.sh.o24707186.2
-rw-r--r--  1 koshu3 koshu           0 10月 16 14:16 hisat2.sh.pe24707186.1
-rw-r--r--  1 koshu3 koshu           0 10月 16 14:18 hisat2.sh.pe24707186.2
-rw-r--r--  1 koshu3 koshu           0 10月 16 14:16 hisat2.sh.po24707186.1
-rw-r--r--  1 koshu3 koshu           0 10月 16 14:18 hisat2.sh.po24707186.2
-rw-r--r--  1 koshu3 koshu        2262 10月 16 11:47 hisat2_index.sh.e24706258
-rw-r--r--  1 koshu3 koshu       3919 10月 16 11:47 hisat2_index.sh.o24706258
-rw-r--r--  1 koshu3 koshu           0 10月 16 11:47 hisat2_index.sh.pe24706258
-rw-r--r--  1 koshu3 koshu           0 10月 16 11:47 hisat2_index.sh.po24706258
drwxr-xr-x  2 koshu3 koshu          4096 10月 16 11:58 outputs
drwxr-xr-x  2 koshu3 koshu          4096 10月 16 10:53 reads
drwxr-xr-x  2 koshu3 koshu          4096 10月 16 11:47 reference
drwxr-xr-x  2 koshu3 koshu          4096 10月 17  2022 scripts
```

事前実行した結果で確認

```
$ ls -al outputs/hisat2/
```



# マッピング 実行結果 ログの確認

```
$ more hisat2.sh.e24707186.1
```

```
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 1 files and 4 in-memory blocks...
```

事前実行した結果で確認

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

```
$ more hisat2.sh.e24707186.2
```

```
4032514 reads; of these:
```

```
4032514 (100.00%) were paired; of these:
```

```
975045 (24.18%) aligned concordantly 0 times
```

```
2882289 (71.48%) aligned concordantly exactly 1 time
```

```
175180 (4.34%) aligned concordantly >1 times
```

```
-----
```

```
975045 pairs aligned concordantly 0 times; of these:
```

```
89479 (9.18%) aligned discordantly 1 time
```

```
-----
```

```
885566 pairs aligned 0 times concordantly or discordantly; of these:
```

```
1771132 mates make up the pairs; of these:
```

```
1274482 (71.96%) aligned 0 times
```

```
459285 (25.93%) aligned exactly 1 time
```

```
37365 (2.11%) aligned >1 times
```

```
84.20% overall alignment rate
```

```
[bam_sort_core] merging from 0 files and 4 in-memory blocks...
```

事前実行した結果で確認

```
$ more outputs/hisat2/hisat2.sh.e16626062.2
```

## アライメントファイル (sam)の確認

[illegible]

## 事前実行した結果で確認

**\$ more outputs/hisat2/SRR453566.sam**

[illegible]

## 事前実行した結果で確認

**\$ more outputs/hisat2/SRR453569.sam**

# 発現量の算出

---

```
$ 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
```

- e only estimate the abundance of given reference transcripts (requires -G)
- B enable output of Ballgown table files which will be created in the same directory as the output GTF (requires -G, -o recommended)
- p number of threads (CPUs) to use (default: 1)
- G reference annotation to use for guiding the assembly process (GTF/GFF)
- o output path/file name for the assembled transcripts GTF (default: stdout)
- A gene abundance estimation output file

# 発現量の算出

```
$ stringtie -h
```

```
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.
```

```
Options:
```

```
--version : print just the version at stdout and exit  
--conservative : conservative transcript assembly, same as -t -c 1.5 -f 0.05  
--mix : both short and long read data alignments are provided  
        (long read alignments must be the 2nd BAM/CRAM input file)  
--rf : assume stranded library fr-firststrand  
--fr : assume stranded library fr-secondstrand  
-G reference annotation to use for guiding the assembly process (GTF/GFF)  
--ptf : load point-features from a given 4 column feature file <f_tab>  
-o output path/file name for the assembled transcripts GTF (default: stdout)  
-l name prefix for output transcripts (default: STRG)  
-f minimum isoform fraction (default: 0.01)  
-L long reads processing; also enforces -s 1.5 -g 0 (default:false)  
-R if long reads are provided, just clean and collapse the reads but  
  do not assemble  
-m minimum assembled transcript length (default: 200)  
-a minimum anchor length for junctions (default: 10)  
-j minimum junction coverage (default: 1)  
-t disable trimming of predicted transcripts based on coverage  
  (default: coverage trimming is enabled)  
-c minimum reads per bp coverage to consider for multi-exon transcript  
  (default: 1)  
-s minimum reads per bp coverage to consider for single-exon transcript  
  (default: 4.75)  
-v verbose (log bundle processing details)  
-g maximum gap allowed between read mappings (default: 50)  
-M fraction of bundle allowed to be covered by multi-hit reads (default:1)  
-p number of threads (CPUs) to use (default: 1)  
-A gene abundance estimation output file  
-E define window around possibly erroneous splice sites from long reads to  
  look out for correct splice sites (default: 25)  
-B enable output of Ballgown table files which will be created in the  
  same directory as the output GTF (requires -G, -o recommended)  
-b enable output of Ballgown table files but these files will be  
  created under the directory path given as <dir_path>  
-e only estimate the abundance of given reference transcripts (requires -G)  
--viral : only relevant for long reads from viral data where splice sites
```

# 発現量の算出

講習中はqsub を実行しないでください。

```
$ qsub scripts/stringtie.sh
Your job 24707314 ("stringtie.sh") has been submitted
```

```
$ 合計 8311004
drwxr-xr-x  7 koshu3 koshu      4096 10月 16 14:47 .
drwxr-x--- 13 koshu3 koshu      4096 10月 13 14:50 ..
-rw-r--r--  1 koshu3 koshu    490409 10月 16 14:46 SRR453566.gene_abund.tab
-rw-r--r--  1 koshu3 koshu 4173908395 10月 16 14:17 SRR453566.sam
-rw-r--r--  1 koshu3 koshu  821523448 10月 16 14:18 SRR453566.sorted.bam
-rw-r--r--  1 koshu3 koshu    491170 10月 16 14:47 SRR453569.gene_abund.tab
-rw-r--r--  1 koshu3 koshu 2922686962 10月 16 14:19 SRR453569.sam
-rw-r--r--  1 koshu3 koshu  591294623 10月 16 14:19 SRR453569.sorted.bam
drwxr-xr-x  4 koshu3 koshu      4096 10月 16 14:46 ballgown
-rw-r--r--  1 koshu3 koshu      693 10月 16 14:18 hisat2.sh.e24707186.1
-rw-r--r--  1 koshu3 koshu      688 10月 16 14:19 hisat2.sh.e24707186.2
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:16 hisat2.sh.o24707186.1
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:18 hisat2.sh.o24707186.2
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:16 hisat2.sh.pe24707186.1
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:18 hisat2.sh.pe24707186.2
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:16 hisat2.sh.po24707186.1
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:18 hisat2.sh.po24707186.2
-rw-r--r--  1 koshu3 koshu    2262 10月 16 11:47 hisat2_index.sh.e24706258
-rw-r--r--  1 koshu3 koshu    3919 10月 16 11:47 hisat2_index.sh.o24706258
-rw-r--r--  1 koshu3 koshu        0 10月 16 11:47 hisat2_index.sh.pe24706258
-rw-r--r--  1 koshu3 koshu        0 10月 16 11:47 hisat2_index.sh.po24706258
drwxr-xr-x  2 koshu3 koshu      4096 10月 16 11:58 outputs
drwxr-xr-x  2 koshu3 koshu      4096 10月 16 10:53 reads
drwxr-xr-x  2 koshu3 koshu      4096 10月 16 11:47 reference
drwxr-xr-x  2 koshu3 koshu      4096 10月 17  2022 scripts
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:46 stringtie.sh.e24707314
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:46 stringtie.sh.o24707314
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:46 stringtie.sh.pe24707314
-rw-r--r--  1 koshu3 koshu        0 10月 16 14:46 stringtie.sh.po24707314
```

事前実行した結果で確認

**\$ ls outputs/stringtie**

```
$ 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
```

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。

事前実行した結果で確認

**\$ ls outputs/stringtie/ballgown**



# 結果ファイルの確認

SRR453566.gene\_abund.tab ファイル

\$ more SRR453566.gene\_abund.tab

Gene ID	Gene Name	Reference	Strand	Start	End	Coverage	FPKM	TPM
gene-YAL068C	PAU8	NC_001133.9	-	1807	2169	1.011341	1.058091	1.143341
gene-YAL030W	SNC1	NC_001133.9	+	87286	87752	70.827682	74.101746	80.072098
gene-YAL029C	MY04	NC_001133.9	-	87855	92270	35.674591	37.323677	40.330833
gene-YAL028W	FRT2	NC_001133.9	+	92900	94486	5.316950	5.562730	6.010918
gene-YAL027W	SAW1	NC_001133.9	+	94687	95472	24.430025	25.559322	27.618629
gene-YAL026C	DRS2	NC_001133.9	-	95630	99697	27.811796	29.097420	31.441788
gene-YNCA0001W	HRA1	NC_001133.9	+	99305	99868	3.410652	3.568312	3.855810
gene-YAL025C	MAK16	NC_001133.9	-	100225	101145	146.168289	152.925034	165.246155
gene-YAL024C	LTE1	NC_001133.9	-	101565	105872	9.433844	9.869931	10.665148
gene-YAL023C	PMT2	NC_001133.9	-	106272	108551	185.183334	193.743591	209.353424
gene-YAL022C	FUN26	NC_001133.9	-	108877	110430	36.694981	38.391239	41.484402
gene-YAL021C	CCR4	NC_001133.9	-	110846	113359	35.354412	36.988697	39.968864
gene-YAL020C	ATS1	NC_001133.9	-	113614	114615	31.161676	32.602150	35.228893
gene-YAL019W	FUN30	NC_001133.9	+	114919	118314	30.554476	31.966883	34.542442
gene-YAL018C	LDS1	NC_001133.9	-	118564	119541	1.130879	1.183155	1.278481
gene-YAL017W	PSK1	NC_001133.9	+	120225	124295	18.278311	19.123240	20.663992

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)

# Singularity を利用したスクリプト

```
$ more scripts/hisat2_singularity.sh
#$ -S /bin/bash
#$ -pe def_slot 4
#$ -cwd
#$ -t 1-2:1
#$ -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`

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--h87f3376_4 \
  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:1.17--hd87286a_2 \
  samtools sort -@ ${NSLOTS} ${PREFIX}.sam -o ${PREFIX}.sorted.bam
```

ご受講ありがとうございました。





# GTF format とは

```
# stringtie -e -B -p 4 -G reference/s288c.gff -o ballgown/SRR453566/SRR453566.out.gtf -A SRR453566.gene_abund.tab SRR453566.sorted.bam
# StringTie version 2.2.1
NC_001133.9    StringTie    transcript    1807    21691000    -    .    gene_id "gene-YAL068C"; transcript_id "rna-NM_001180043.1"; ref_gene_name
NC_001133.9    StringTie    exon         1807    21691000    -    .    gene_id "gene-YAL068C"; transcript_id "rna-NM_001180043.1"; exon_number "
NC_001133.9    StringTie    transcript    87286    877521000    +    .    gene_id "gene-YAL030W"; transcript_id "rna-NM_001178175.1"; ref_gene_name
NC_001133.9    StringTie    exon         87286    873871000    +    .    gene_id "gene-YAL030W"; transcript_id "rna-NM_001178175.1"; exon_number
NC_001133.9    StringTie    exon         87501    877521000    +    .    gene_id "gene-YAL030W"; transcript_id "rna-NM_001178175.1"; exon_number
```

タブ区切りフォーマット。値がない場合は、"." が設定される。

1. seqname：染色体 or スキャフォールドの名前

2. source：アノテーションを生成したプログラムまたはデータソースの名前

3. feature：フィーチャータイプ（mRNA, gene, exon, CDS ....）

4. start：スタートポジション (1bp ～)

5. end：エンドポジション (1bp ～)

6. score：スコア

7. strand：+(forward)、-(reverse)または '.'

8. frame：翻訳フレーム（0, 1, 2）

9. attribute：追加情報。セミコロンで区切られたタグと値のペアのリスト。

**GFF では、gene\_id=XXXXXX; の形式に対して、GTF では、gene\_id “XXXXXX”; と記載していく。**