

# RNA-seq 演習

高橋 弘喜

2019-03-25

## RNA-seq 演習

テストデータを用いて，RNA-seq 解析を実際にやってみる．テストデータとして，*Saccharomyces cerevisiae* を対象に取得されたデータを使用する<sup>1</sup>．

リードのマッピング (HISAT2<sup>2,3</sup>)，遺伝子発現量算出 (StringTie<sup>3</sup>) までを遺伝研のスパコン上で行う．その後の解析は，ローカル環境で統計言語 R<sup>4</sup> (ballgown<sup>5</sup>) を利用することで，各種統計量の可視化，ヒートマップなどの作成，有意差のあった遺伝子群の抽出などが可能である．

なお，今回の演習で紹介する方法以外にも，各遺伝子のリード数に基づいた解析も数多くなされている．ソフトウェアとしては，HTSeq<sup>6</sup>, DESeq<sup>7</sup>, DESeq2<sup>8</sup>, edgeR<sup>9</sup> などが挙げられる．

## データ準備@スパコン

通常は，取得したデータを遺伝研スパコン上で解析するために，スパコン上へデータ転送を行う．今回は，スパコン上でデータをダウンロードし，解凍して作業を進める．

## ファイル転送

遺伝研スパコンにデータを転送する．

1. FileZilla, WinScp などのファイル転送ソフトによって，データ転送を行う．
2. scp コマンドによるファイル転送

## 遺伝研スパコンへログイン

Windows の場合は，TeraTerm などを用いる．Mac, Linux の場合は，端末を起動して実行する．

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

サーバーログイン後

## #作業場所の確認

```
$ pwd
/home/hi-takah
```

## #データのダウンロード

```
curl -O http://bioinfo.pf.chiba-u.jp/rna-seq-nig/rna-seq_20190325.zip
```

## #ファイルリストの確認

\$ 1s

## #配布データの解凍

```
$ unzip rna-seq_20190325.zip
```

## #作業場所へ移動

```
$ cd rna-seq_20190325
```

## #場所の確認

```
$ pwd
/home/hi-takah/rna-seq_20190325
```

## #ファイルの確認

```
$ ls
annotation  fastq  r64_tran
```

## fastq の中身確認

[illegible]

## データの取得

用意したデータには下記のファイルが含まれている.

```
rna-seq_20190325/
├── annotation
│   └── genes.gtf
├── fastq
│   ├── SRR453566_1.fq
│   ├── SRR453566_2.fq
│   ├── SRR453567_1.fq
│   ├── SRR453567_2.fq
│   ├── SRR453569_1.fq
│   ├── SRR453569_2.fq
│   ├── SRR453570_1.fq
│   └── SRR453570_2.fq
└── r64_tran
    ├── genome_tran.1.ht2
    ├── genome_tran.2.ht2
    ├── genome_tran.3.ht2
    ├── genome_tran.4.ht2
    ├── genome_tran.5.ht2
    ├── genome_tran.6.ht2
    ├── genome_tran.7.ht2
    ├── genome_tran.8.ht2
    └── make_r64_tran.sh
```

ファイル名	condition	# of reads
SRR453566_1.fq	batch 1	100,000
SRR453566_2.fq		100,000
SRR453567_1.fq	batch 2	100,000
SRR453567_2.fq		100,000
SRR453569_1.fq	chemo 1	100,000
SRR453569_2.fq		100,000
SRR453570_1.fq	chemo 2	100,000
SRR453570_2.fq		100,000

## マッピング

今回は HISAT2 を用いて, RNA-seq リードをリファレンスゲノムにマッピングする. マッピングにおいて参照ゲノムと遺伝子情報ファイル (gtf ファイル) が必要となる.

用意すべきファイル	ファイル名	備考
fastq ファイル	***.fastq, ***.fq, ***.fastq.gz, ***.fq.gz	paired, single いずれかの RNA-seq データ
gtf ファイル	genes.gtf (Saccharomyces cerevisiae (Yeast) Ensembl R64-1-1)	アノテーションファイル（今回は iGenomes <sup>10</sup> より取得）
リファレンスゲノム のインデックスファ イル	HISAT2 index	ない場合は hisat2-build で作成する

主要なモデル生物に関しては、インデックスファイルが用意されている<sup>11</sup>。

種名	バージョン
H. sapiens	GRCh38 UCSC hg38 UCSC hg38 and Refseq gene
M. musculus	GRCm38
R. norvegicus	UCSC rn6
D. melanogaster	BDGP6
C. elegans	WBcel235
S. cerevisiae	UCSC sacCer3

## マッピング (HISAT2)

RNA-seq のマッピングに関しては、多くのソフトウェアが開発されている。いずれも真核生物を対象に実装されている。

- TopHat<sup>12</sup>
- TopHat2<sup>13</sup>
- STAR<sup>14</sup>

原核生物の場合は、ゲノムシーケンス同様 bowtie2<sup>15</sup> などを用いる。

## コマンドの確認

スパコンにインストール済みのソフトウェア一覧<sup>16</sup>を参照すると、HISAT2 2.0.0~2.1.0 までのバージョンが使用可能である。

名称	バージョン	PATH
HISAT2	2.1.0	/usr/local/biotools/h/hisat2:2.1.0-py36pl5.22.0_0
	2.0.5	/usr/local/biotools/h/hisat2:2.0.5-py36pl5.22.0_2
	2.0.4	/usr/local/biotools/h/hisat2:2.0.4-py35_0
	2.0.3beta	/usr/local/biotools/h/hisat2:2.0.3beta-py35_0
	2.0.2beta	/usr/local/biotools/h/hisat2:2.0.2beta-py35_0
	2.0.1beta	/usr/local/biotools/h/hisat2:2.0.1beta-py35_0
	2.0.0beta	/usr/local/biotools/h/hisat2:2.0.0beta-py35_0
StringTie	1.3.3	/usr/local/biotools/s/stringtie:1.3.3-py36_2.1
	1.3.0	/usr/local/biotools/s/stringtie:1.3.0-0
	1.2.4	/usr/local/biotools/s/stringtie:1.2.4-0
SAMtools	1.6	/usr/local/biotools/s/samtools:1.6-0.1
	1.5	/usr/local/biotools/s/samtools:1.5-2
	1.4.1	/usr/local/biotools/s/samtools:1.4.1-0

コマンドを確認してみる.

```
$ module load singularity
```

```
$ singularity exec /usr/local/biotools/h/hisat2\:2.1.0--py36pl5.22.0_0.1 hisat2 -h
```

```
HISAT2 version 2.1.0 by Daehwan Kim (infphilo@gmail.com, www.ccb.jhu.edu/people/infphilo)
Usage:
```

```
hisat2 [options]* -x <ht2-idx> {-1 <m1> -2 <m2> | -U <r> | --sra-acc <SRA accession number>} [-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).
```

```
<SRA accession number> Comma-separated list of SRA accession numbers, e.g. -sra-acc SRR353653,SRR353654.
```

```
<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'.

## ライブラリーについて

strand specific のサンプル調整キットを使用している場合は、“-rna-strandness” オプションを用いることで、リードの向きを考慮したマッピングが可能となる。次の StringTie においては、“-rf”、“-fr” オプションを指定すればよい。

### hisat2\_\_1.sh

SRR453566 データを HISAT2 でマッピングを行う。

```
#!/bin/bash
#$ -S /bin/bash
#$ -N hisat2
#$ -pe def_slot 4
#$ -cwd
```

```
module load singularity
```

```
f=("SRR453566")
```

```
singularity exec /usr/local/biotools/h/hisat2\:2.1.0--py36pl5.22.0_0.1 hisat2 \
  -p 4 -x r64_tran/genome_tran --dta \
  -1 fastq/${f}_1.fq \
  -2 fastq/${f}_2.fq \
  -S ${f}.sam
```

## 実行

```
$ qsub -l epyc -l s_vmem=1G -l mem_req=1G hisat2_1.sh
$ qstat
```

計算が終わると、SRR453566.sam が作成される。

### samtools\_\_1.sh

SRR453566.sam を bam ファイルへ変換する。

```
#!/bin/bash
#$ -S /bin/bash
#$ -N samtools
#$ -pe def_slot 4
```

```
#$ -cwd
```

```
module load singularity
```

```
f=("SRR453566")
```

```
singularity exec /usr/local/biotools/s/samtools\:1.6--0.1 samtools sort \  
-@ 4 -o ${f}.sort.bam ${f}.sam
```

実行

```
$ qsub -l epyc -l s_vmem=1G -l mem_req=1G samtools_1.sh  
$ qstat
```

**hisat2.sh**

残りの3つのデータについても HISAT2, samtools を実行する。複数サンプルの処理には, for ループを使用することができる。

```
#!/bin/bash
```

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

```
#$ -N hisat2
```

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

```
#$ -cwd
```

```
module load singularity
```

```
files=("SRR453567" "SRR453569" "SRR453570")
```

```
for f in ${files[@]}
```

```
do
```

```
## HISAT2
```

```
    singularity exec /usr/local/biotools/h/hisat2\:2.1.0--py36pl5.22.0_0.1 hisat2 \  
    -p 4 -x r64_tran/genome_tran --dta \  
    -1 fastq/${f}_1.fq \  
    -2 fastq/${f}_2.fq \  
    -S ${f}.sam
```

```
## samtools
```

```
    singularity exec /usr/local/biotools/s/samtools\:1.6--0.1 samtools sort \  
    -@ 4 -o ${f}.sort.bam ${f}.sam
```

done

実行

```
$ qsub -l epyc -l s_vmem=1G -l mem_req=1G hisat2.sh
$ qstat
```

## StringTie

HISAT2 で得られたマッピング結果に基づいて、各遺伝子の発現量算出を行う。

コマンドの確認

```
$ singularity exec /usr/local/biotools/s/stringtie\:1.3.3--py36_2.1 stringtie -
h
StringTie v1.3.3 usage:
  stringtie <input.bam ..> [-G <guide_gff>] [-l <label>] [-o <out_gtf>] [-
p <cpus>]
  [-v] [-a <min_anchor_len>] [-m <min_tlen>] [-j <min_anchor_cov>] [-
f <min_iso>]
  [-C <coverage_file_name>] [-c <min_bundle_cov>] [-g <bdist>] [-u]
  [-e] [-x <seqid,..>] [-A <gene_abund.out>] [-h] {-B | -b <dir_path>}
Assemble RNA-Seq alignments into potential transcripts.
Options:
--version : print just the version at stdout and exit
-G reference annotation to use for guiding the assembly process (GTF/GFF3)
```

## stringtie.sh

マッピング結果を用いて、StringTie による遺伝子発現量算出を行う。

```
#!/bin/bash
#$ -S /bin/bash
#$ -N stringtie
#$ -pe def_slot 4
#$ -cwd
```

```
module load singularity
```



```
files=("SRR453566" "SRR453567" "SRR453569" "SRR453570")
```

```
for f in ${files[@]}
```

```
do
```

```
    singularity exec /usr/local/biotools/s/stringtie\:1.3.3--py36_2.1 stringtie -e -B \
    -p 4 -G annotation/genes.gtf \
    -o ballgown/${f}/${f}.gtf ${f}.sort.bam
```

```
done
```

実行

```
$ qsub -l epyc -l s_vmem=1G -l mem_req=1G stringtie.sh
```

```
$ qstat
```

結果

下記の結果ファイルが出力される.

ballgown/

```
|—— SRR453566
|   |—— SRR453566.gtf
|   |—— e2t.ctab
|   |—— e_data.ctab
|   |—— i2t.ctab
|   |—— i_data.ctab
|   |—— t_data.ctab
|—— SRR453567
|   |—— SRR453567.gtf
|   |—— e2t.ctab
|   |—— e_data.ctab
|   |—— i2t.ctab
|   |—— i_data.ctab
|   |—— t_data.ctab
|—— SRR453569
|   |—— SRR453569.gtf
|   |—— e2t.ctab
|   |—— e_data.ctab
|   |—— i2t.ctab
|   |—— i_data.ctab
|   |—— t_data.ctab
```

```

└── SRR453570
    ├── SRR453570.gtf
    ├── e2t.ctab
    ├── e_data.ctab
    ├── i2t.ctab
    ├── i_data.ctab
    └── t_data.ctab

```

結果ファイルの確認.

```

$ less ballgown/SRR453566/SRR453566.gtf
# stringtie -e -B -p 4 -G annotation/genes.gtf -o ballgown/SRR453566/SRR453566.gtf SR
# StringTie version 1.3.3
I      StringTie      transcript      12046    12426    1000    +      .      gene_
B"; transcript_id "YAL064W-B"; ref_gene_name "YAL064W-B"; cov "0.061680"; FPKM "3.689
I      StringTie      exon      12046    12426    1000    +      .      gene_id "YALO
B"; transcript_id "YAL064W-B"; exon_number "1"; ref_gene_name "YAL064W-
B"; cov "0.061680";
I      StringTie      transcript      24000    27968    1000    -      .      gene_
I      StringTie      exon      24000    27968    1000    -      .      gene_id "YALO

```

次のステップとしては, R (ballgown) を用いることで可視化などが実現できる. ballgown ディレクトリそのものを ballgown の入力として使用する.

## その他

講習会のデータ作成に用いたツール.

1. seqtk<sup>17</sup>: リードのサンプリング
2. SRA Toolkit<sup>18</sup>: SRA から sra データの取得, fastq への変換

## 参考文献

1. Nookaew, I. *et al.* 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*. *Nucleic Acids Res* **40**, 10084–10097
2. Kim, D., Langmead, B. & Salzberg, S. L. HISAT: A fast spliced aligner with low memory requirements. *Nat Methods* **12**, 357–360
3. Pertea, M., Kim, D., Pertea, G. M., Leek, J. T. & Salzberg, S. L. Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown. *Nat Protoc* **11**, 1650–1667 (2016).

4. R Core Team. *R: A language and environment for statistical computing*. (R Foundation for Statistical Computing, 2018).
5. Fu, J., Frazee, A. C., Collado-Torres, L., Jaffe, A. E. & Leek, J. T. *Ballgown: Flexible, isoform-level differential expression analysis*. (2019).
6. Anders, S., Pyl, P. T. & Huber, W. HTSeq—a python framework to work with high-throughput sequencing data. *Bioinformatics* **31**, 166–169
7. Anders, S. & Huber, W. Differential expression analysis for sequence count data. *Genome Biol* **11**, R106 (2010).
8. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biol* **15**, 550 (2014).
9. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: A bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140
10. iGenomes. Available at: [http://jp.support.illumina.com/sequencing/sequencing\\_software/igenome.html](http://jp.support.illumina.com/sequencing/sequencing_software/igenome.html).
11. HISAT2. Available at: <https://ccb.jhu.edu/software/hisat2/index.shtml>.
12. Trapnell, C., Pachter, L. & Salzberg, S. L. TopHat: Discovering splice junctions with rna-seq. *Bioinformatics* **25**, 1105–1111
13. Kim, D. *et al.* TopHat2: Accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol* **14**, R36
14. Dobin, A. *et al.* STAR: Ultrafast universal rna-seq aligner. *Bioinformatics* **29**, 15–21
15. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with bowtie 2. *Nat Methods* **9**, 357–359
16. 利用可能オープンソースソフトウェア. Available at: <https://sc2.ddbj.nig.ac.jp/index.php/available-biotools>.
17. Seqtk. Available at: <https://github.com/lh3/seqtk>.
18. SRA toolkit. Available at: <https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>.