

Quantification

Normalization

Advanced bash commands

(for, while, if, md5sum)

Introduction to R

Mapping이 잘 되었는지 확인

```
$ cd ~/day1/star/example1
```

```
$ ls
```

```
example1Chimeric.out.sam
```

```
example1Chimeric.out.junction
```

```
example1Aligned.out.bam
```

```
example1SJ.out.tab
```

```
example1Log.progress.out
```

```
example1Log.out
```

```
example1Log.final.out
```

```
$ samtools -h view example1Aligned.out.bam | less
```

HD = header, VN = format version

SQ = sequence dictionary, SN = reference sequence name, LN = Reference sequence length

```
kjyi@bnode1:~/Projects/workshop/day1/star/example1
@HD      VN:1.4
@SQ      SN:1      LN:249250621
@SQ      SN:2      LN:243199373
@SQ      SN:3      LN:198022430
@SQ      SN:4      LN:191154276
@SQ      SN:5      LN:180915260
@SQ      SN:6      LN:171115067
@SQ      SN:7      LN:159138663
@SQ      SN:8      LN:146364022
@SQ      SN:9      LN:141213431
@SQ      SN:10     LN:135534747
@SQ      SN:11     LN:135006516
@SQ      SN:12     LN:133851895
@SQ      SN:13     LN:115169878
@SQ      SN:14     LN:107349540
@SQ      SN:15     LN:102531392
@SQ      SN:16     LN:90354753
@SQ      SN:17     LN:81195210
@SQ      SN:18     LN:78077248
@SQ      SN:19     LN:59128983
@SQ      SN:20     LN:63025520
@SQ      SN:21     LN:48129895
@SQ      SN:22     LN:51304566
@SQ      SN:X      LN:155270560
@SQ      SN:Y      LN:59373566
:
```

```
kji@bnode1:~/Projects/workshop/day1/star/example1
@SQ SN:GL000195.1 LN:182896
@SQ SN:GL000212.1 LN:186858
@SQ SN:GL000222.1 LN:186861
@SQ SN:G 35
@SQ SN:G 89
@SQ SN:GL000194.1 LN:191469
@SQ SN:GL000225.1 LN:211173
@SQ SN:GL000192.1 LN:547496
@PG ID:STAR PN:STAR VN:STAR_2.5.4b CL:STAR --runMode alignReads --genomeDir /
home/users/kji/ref/hg19/star_index --readFilesIn ./fastq/R1.fastq ./fastq/R2.fast
q --outFileNamePrefix star/example1/example1 --outSAMtype BAM Unsorted
@CO user command line: STAR --runMode alignReads --outSAMtype BAM Unsorted --genom
eDir /home/users/kji/ref/hg19/star_index --outFileNamePrefix star/example1/example1 -
-readFilesIn ./fastq/R1.fastq ./fastq/R2.fastq
NB502049:59:HVK22AFX:1:11101:22327:1059 99 17 7578435 255 120M75
7N31M = 7578462 935 CTGCTTGTAGATGACCATGGCGCGGACGCGGGTGCCGGGCGGGGGTGTGGAA
TCAACCCACAGCTGCACAGGGCAGGTCTTGCCAGTTGGCAAACATCTTGTTGAGGGCAGGGGAGTACGTGCAAGTCACAGACTT
GGCTGTCCAGAA AAAAAEEE6EAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
HI:i:1 AS:i:302 nM:i:1
NB502049:59:HVK22AFX:1:11101:22327:1059 147 17 7578462 255 93M757
7578435 GCGGGGCGGGGGTGTGGAATCAACCCACAGCTGCACAGGGCAGGTC
ATCTTGTT ACCTGCAAGTCACAGACTTGGCTGTCCAGAAATGCAAGAAGCCCAG
E/AAAAEEEEEEEE/E<AA<A/ EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
HI:i:1 A :1
NB502049:59 4352:1059 99 4 187541398 255 151M
= 72 CCCGTTTCTGCCACTGTCTGTCTACAGCAGTGACATAGCGAATGACAT
GGCCACCTCAGTGTCCACTTTAACAACGGCGTAGGGGAAGGTTGACAAACACGGCGCATTATCATTTTGGTCTTCTACAATG
:
```

template 이름

FLAG

Chromosome (contig)

Mapping quality

position

Mate의 chromosome

CIGAR string

Template length

Mate의 position

PG = 프로그램 (bam 파일을 생성한)

CO = 코멘트 (STAR가 생성함)

여기부터 alignment section

<https://samtools.github.io/hts-specs/SAMv1.pdf>

FLAG: Combination of bitwise FLAGS.⁷ Each bit is explained in the following table:

Bit	Description
1	0x1 template having multiple segments in sequencing
2	0x2 each segment properly aligned according to the aligner
4	0x4 segment unmapped
8	0x8 next segment in the template unmapped
16	0x10 SEQ being reverse complemented
32	0x20 SEQ of the next segment in the template being reverse complemented
64	0x40 the first segment in the template
128	0x80 the last segment in the template
256	0x100 secondary alignment
512	0x200 not passing filters, such as platform/vendor quality controls
1024	0x400 PCR or optical duplicate
2048	0x800 supplementary alignment

Paired-end sequencing이면서 (0x1 = 1) (0x2 = 2)
두 pair가 proper mapping이 되고 (방향과 거리가 STAR에서 설정한 대로 적절함)
둘다 primary alignment이면서, 이 read가 First in pair인 경우 FLAG는? $1+2+64=67$
(0x40 = $4*16 = 64$),

Mate의 FLAG는? $0x1 + 0x2 + 0x80 = 131$

Sort, index


```
$ samtools sort example1Aligned.out.bam > example1Aligned.sort.bam
```

```
$ samtools index example1Aligned.sort.bam
```

(sort된 bam file 만 indexing할 수 있습니다.)

실습을 위해 example1Aligned.sort.bam file and example1Aligned.sort.bam.bai 파일을 다운로드하세요

https://software.broadinstitute.org/software/igv/download



Integrative Genomics Viewer

Home

Downloads

Documents

Hosted Genomes

FAQ

IGV User Guide

File Formats

Release Notes

Credits

Contact

Search website

search

© 2013-2018
Broad Institute, and
the Regents of the
University of California

Home » Downloads

Downloads

Integrative Genomics Viewer - IGV 2.4

Install IGV

NOTE: IGV 2.4.x requires [Java 8](#). Java versions 9 and above are currently not supported.

Use one of the following 4 options to install and run the current version of IGV.

- Download and unzip the **Mac App Archive**, then double-click the IGV application to run it. The application can be moved to the *Applications* folder, or anywhere else.
- Download and unzip the **Windows Zip Archive**, then double-click the *igv.bat* file to start IGV. A black console window will appear, followed by the IGV application. **Note:** Windows users with **high resolution screens** should use this version – it includes a modified Java executable for use with high-resolution screens.
- Download and unzip the **Binary Distribution archive**. IGV is launched from a command prompt – follow the instructions in the *readme* file. To launch IGV on Mac or Linux use the shell script *igv.sh*. On Windows use *igv.bat*.
- Click on one of the *Launch* buttons below to download a .jnlp file and execute the file using **Java Web Start (JWS)**.
 - Mac users:** If you are notified of security errors that prevent launching IGV, try the following:
 - Right-click on the downloaded .jnlp file; select *Open With > Java Web Start*; dismiss the warnings.
 - After IGV has been run this way at least once from the .jnlp file, you can double-click on the file to launch.
 - Windows users:** To run with more than 1.2 GB of memory on Windows you must install 64-bit Java. **Most Windows installs do not include 64-bit Java by default, even if the operating system is 64-bit.** Attempting to use the 2GB or greater launch options with 32-bit Java will result in the error *"could not create virtual machine"*.

Launch

Launch with 750 MB

Launch

Launch with 1.2 GB
(Max usable memory for
Windows with 32-bit Java)

Launch

Launch with 2 GB
(Max usable memory for
32-bit MacOS)

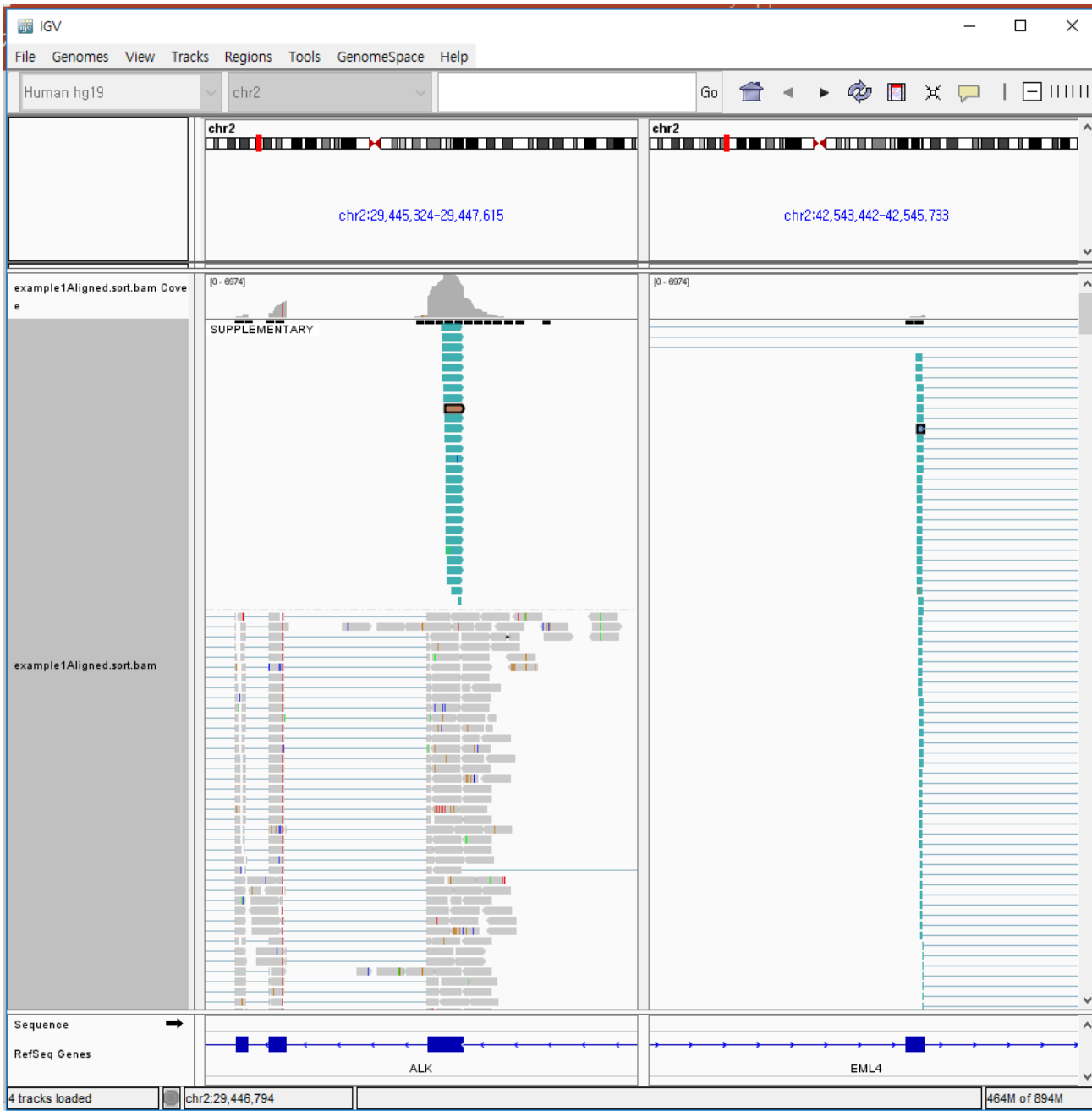
Launch

Launch with 10 GB
(Only for large memory
machines with 64-bit Java)

맥사용자

윈도우 사용자

리눅스, 기타 사용자



1. open bam file (sorted, indexed)
2. Go to the region
chr2:29,445,324-29,447,615
3. right click alignment tract ->
group by supplementary flag
4. Right click one of improperly mapped reads,
Select View mate region in split screen

Read name = NB502049:59:HVK22AFX:2:21111:8238:10792

Read length = 94bp

Mapping = Supplementary @ MAPQ 255

Reference span = chr2:29,446,301-29,446,394 (+) = 94bp

Cigar = 94M57H

Clipping = Right 57 hard

Mate is mapped = yes

Mate start = chr2:42544573 (+)

Insert size = 0

First in pair

Pair orientation = F1F2

SupplementaryAlignments

2:42,552,638-42,552,695 (-) = 57bp @MAPQ 255 NM0

NH = 1

HI = 1

NM = 0

nM = 0

AS = 92

Hidden tags: SA

Location = chr2:29,446,355

Base = A @ QV 36

example1Aligned.sort.bam

Rename Track...

Copy read details to clipboard

Link supplementary alignments

Group alignments by >

Sort alignments by >

Color alignments by >

Re-pack alignments

☒ Shade base by quality

☒ Show mismatched bases

Show all bases

Quick consensus mode

View as pairs

Go to mate

View mate region in split screen

Set insert size options ...

☒ Collapsed

Expanded

Quantification

Normalization

Advanced bash commands

(for, while, if, md5sum)

Introduction to R

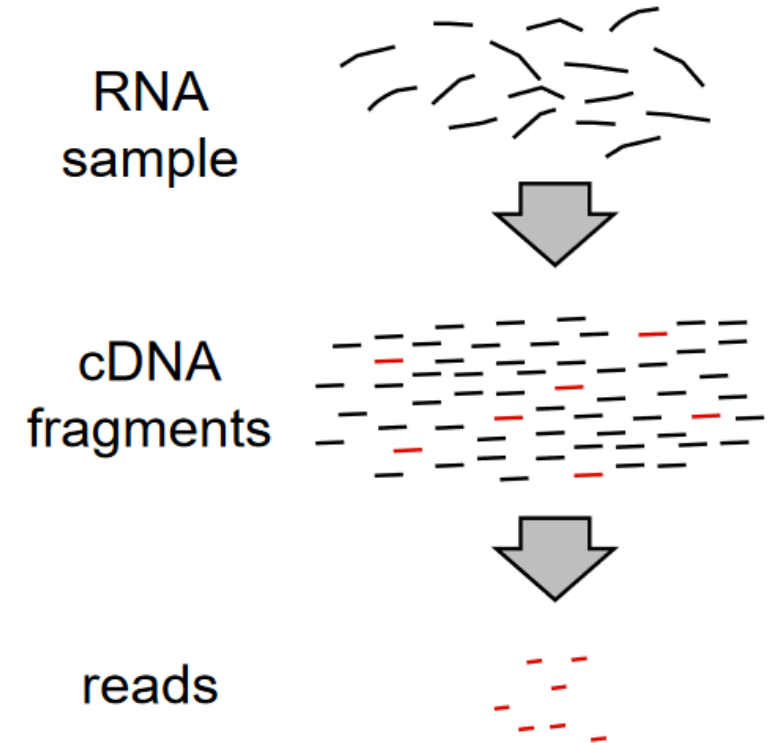
Quantification of Gene Expression

- Estimate relative abundance of transcripts
- Count reads, fetch depth/coverage
- Differential expression
 - 두 조건에서 얻은 gene expression profile을 비교하여, 어떤 transcript가 두 조건 사이에서 발현에 차이를 보이는지를 찾는 것

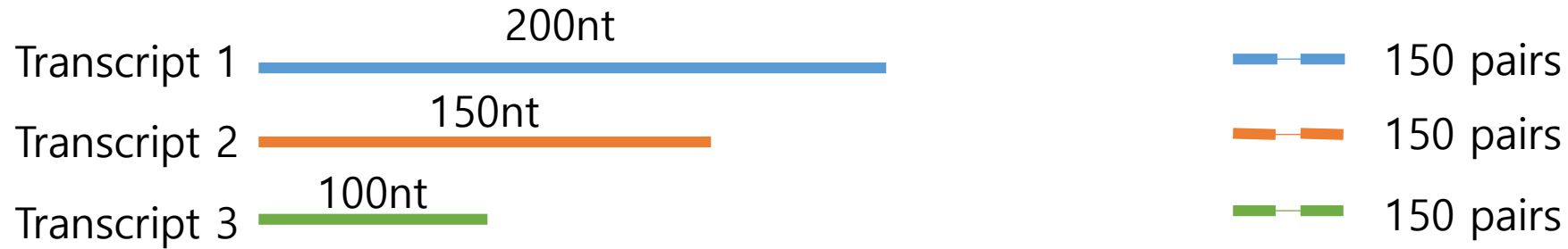
Quantification of Gene Expression (2)

고려해야 할 것들

- Absolute quantification
- Sequencing throughput (depth)
- Gene length
- Transcript variant (different exon usage)



- Gene length
- Sequencing throughput



$$\hat{f} \propto \frac{\frac{150}{450}}{200} = \frac{1}{600}$$

Relative abundance of Transcript 1 in sample 1

Number of reads mapped in transcript 1

Total number of reads

Length of transcript 1

RPKM – Reads Per Kilobase per Million mapped reads

1. 샘플의 total read에 1,000,000을 나눈다. 이것이 per million scaling factor이다.
2. Read counts를 per million scaling factor로 나눈다. 이것이 reads per million (RPM)이다.
3. RPM을 유전자(transcript)의 (평균)길이(kb 단위)로 나눈다. 이것이 RPKM이다.

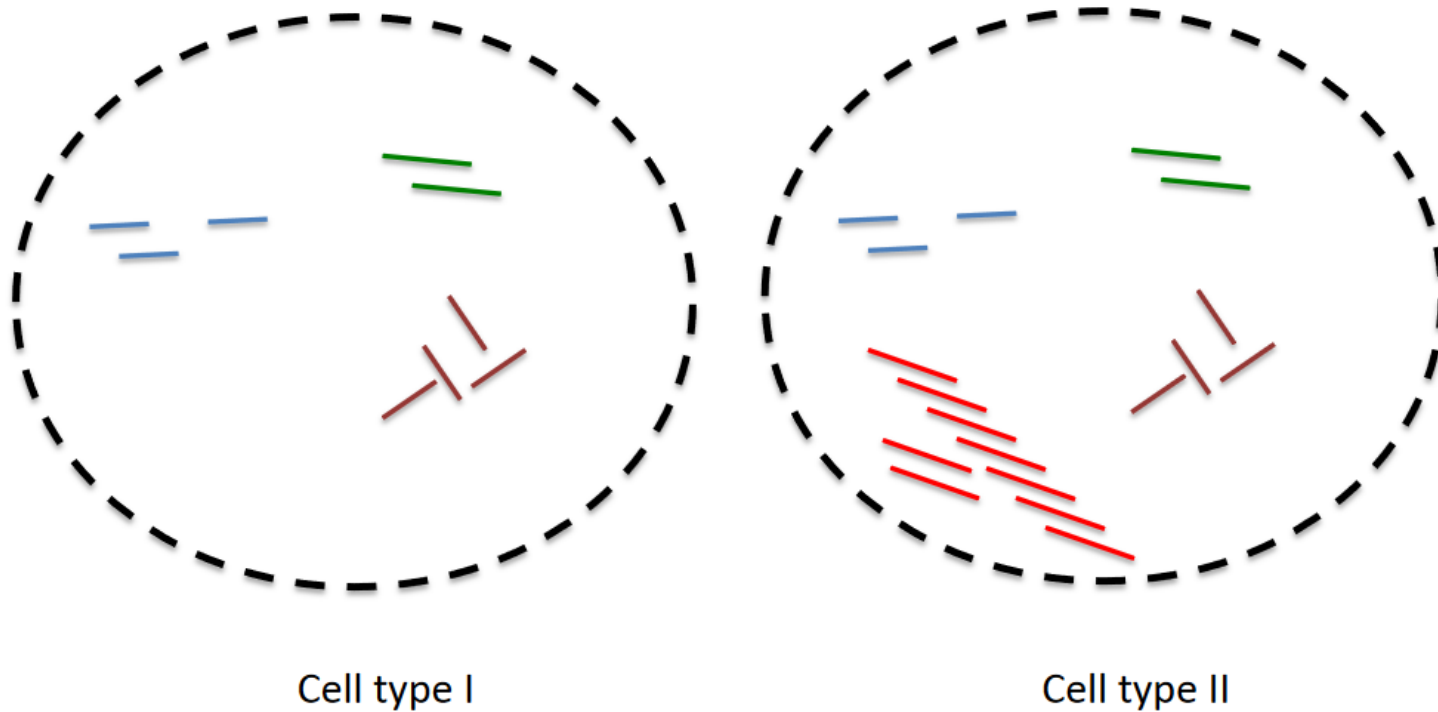
FPKM은 read pair 수를 이용하면 된다. Pair 중 하나만 mapping이 된 경우, 1로 취급한다.

TPM – transcript per million

1. Read count를 각 유전자의 길이(kb 단위)로 나눈다. 이것이 reads per kilobase(RPK)이다.
2. 한 샘플의 모든 transcript의 RPK를 다 더해서, 1,000,000로 나눈다. 이것이 per million scaling factor이다.
3. RPK를 per million scaling factor로 나눈다. 이것이 TPM이다.

Normalization for comparing a gene across samples

1. Technical spike-in
2. Cross-sample normalization

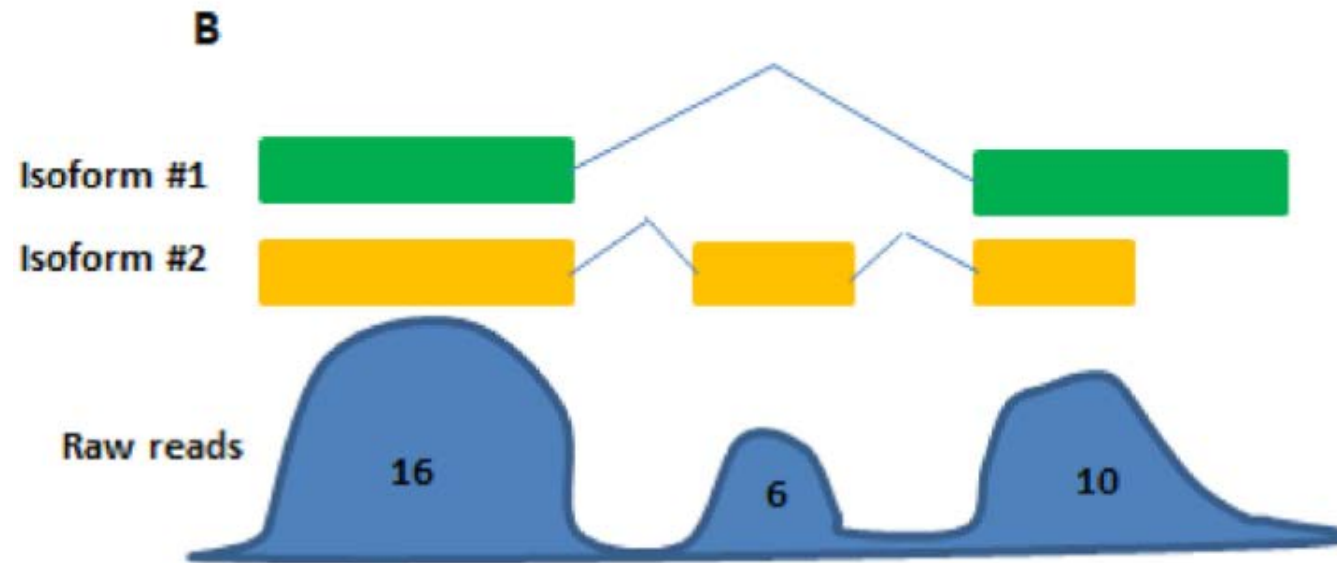


$$\hat{s}_j = \text{median}_i \frac{k_{ij}}{\left(\prod_{v=1}^m k_{iv}\right)^{1/m}}$$

Gene i 의 여러 샘플에서의 read count의 geometric mean

Size factor of sample j

$$\hat{q}_{i\rho} = \frac{1}{m_\rho} \sum_{j:\rho(j)=\rho} \frac{k_{ij}}{\hat{s}_j}$$

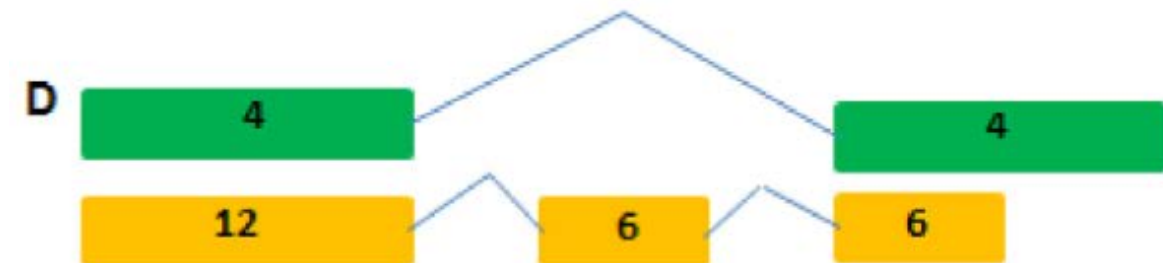


union-exon based approach

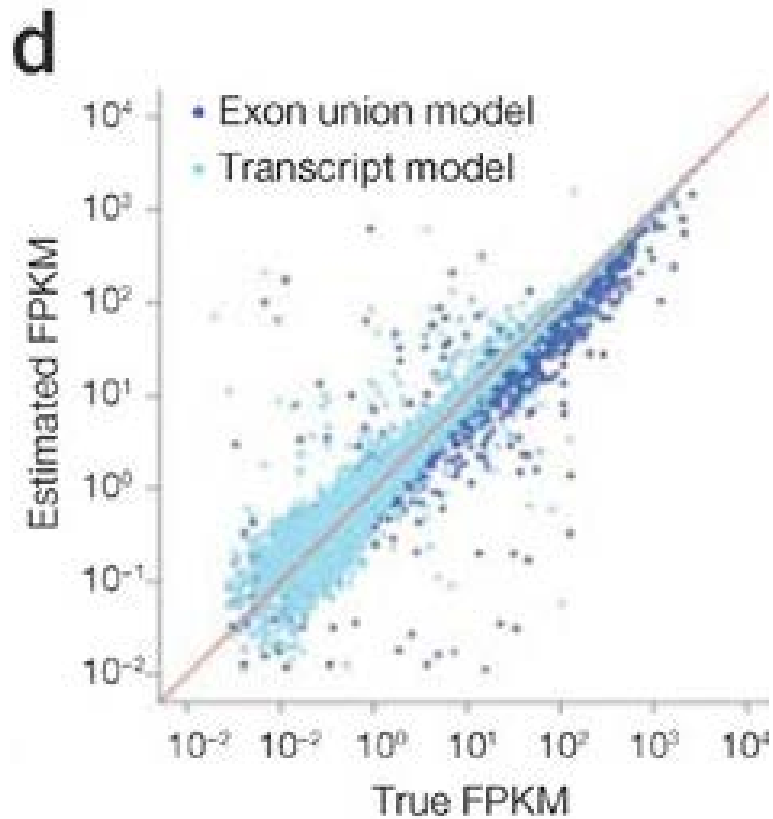
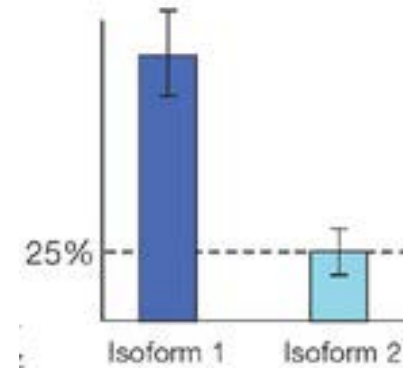
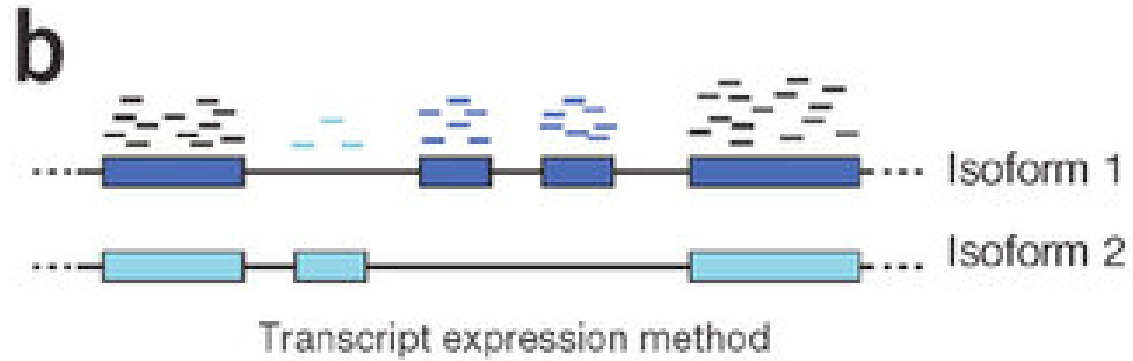


Total gene length after exon flattening: 5kb
 Total reads: 32
 RPKM for gene: 6.4 (=32/5)

Transcript based approach



Relative isoform abundance (#1/#2): 25% / 75%
 RPKM for isoform #1 and #2: 2 and 6
 RPKM for gene (=sum of isoforms): 8 (=2+6)



Exon union model underestimate gene expression

Transcript model을 이용하는 counting을 해야 한다

Quantification

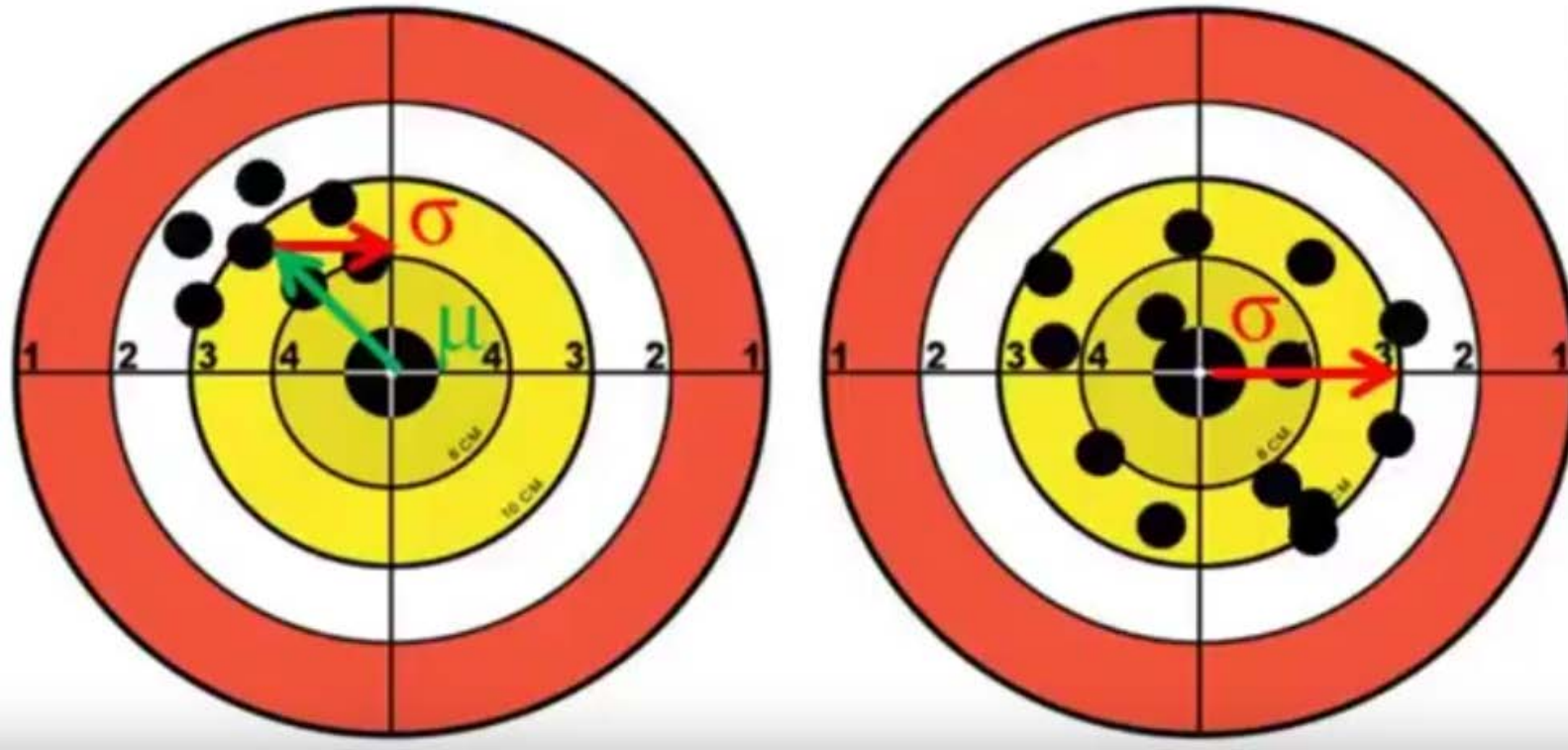
Normalization

Advanced bash commands

(for, while, if, md5sum)

Introduction to R

Normalization



Systematic measurement의 error를 보정하는 목적

Mean and Standard deviation

Z-score normalization

Mean

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

Standard Deviation

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

Z-score normalization

mean

sd

$(x - \text{mean}) / \text{sd}$

2	4	4
5	4	14
4	6	8
3	5	8
3	3	9

3.33	1.15
7.67	5.51
6	2
5.33	2.52
5	3.46

-1.2	0.6	0.6
-0.48	-0.67	1.15
-1	0	1
-0.93	-0.13	1.05
-0.58	-0.58	1.15

Quantile normalization example

Original

2	4	4
5	4	14
4	6	8
3	5	8
3	3	9

Ranked

2	3	4
3	4	8
3	4	8
4	5	9
5	6	14

Averaged

3	3	3
5	5	5
5	5	5
6	6	6
8	8	8

Re-ordered

3	5	3
8	5	8
6	8	5
5	6	5
5	3	6

Quantile normalization 방법 자세히 소개 (쉬운 강의)

https://www.youtube.com/watch?v=v0j4guy_z30

다양한 normalization 방법을 자세히 소개 (어려움)

Anders, Simon, and Wolfgang Huber. "Differential expression analysis for sequence count data." *Genome biology* 11.10 (2010): R106.

Upper quartile normalization (TCGA method)

<https://bioinformatics.stackexchange.com/questions/2586/how-to-apply-upperquartile-normalization-on-rsem-expected-counts>