RNA-seq解析パイプライン：Transcript-based

Shuji Shigenobu
重信　秀治

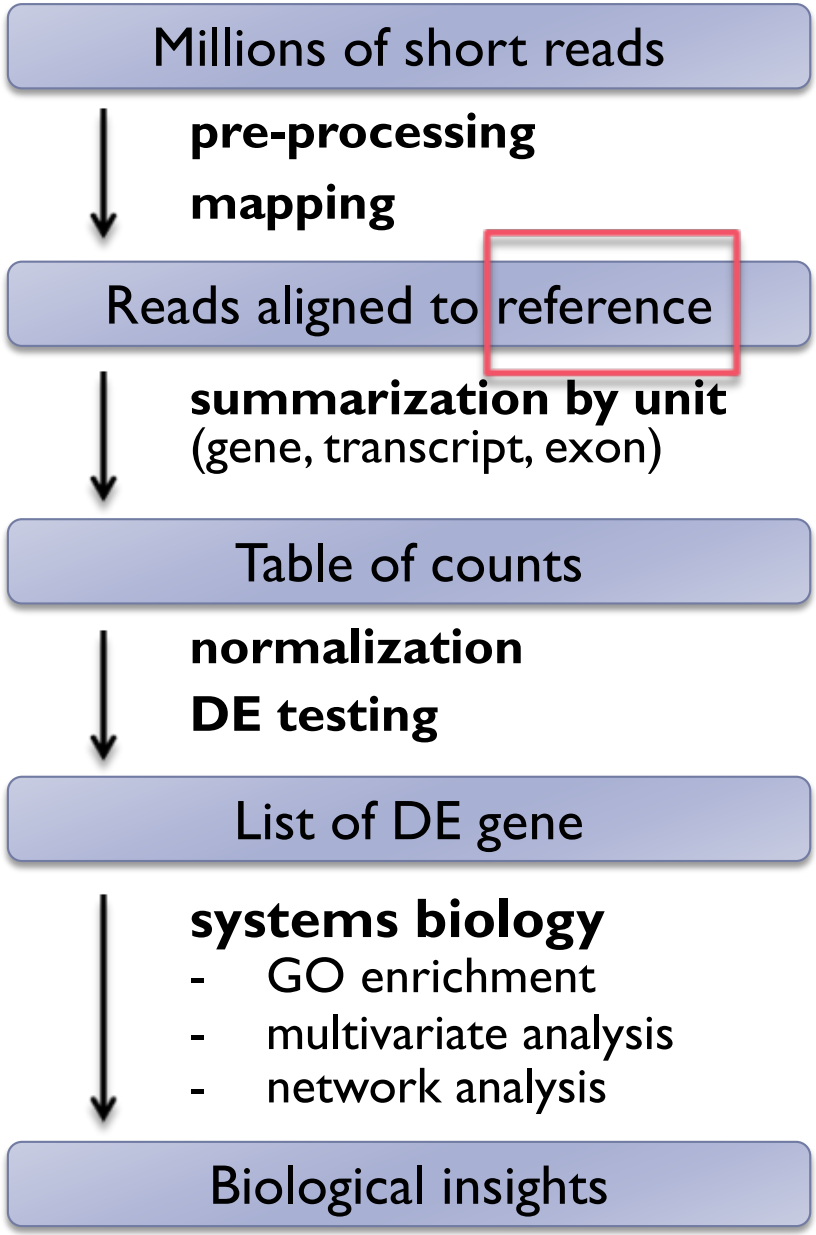
基礎生物学研究所
生物機能解析センター

Two Basic Pipelines

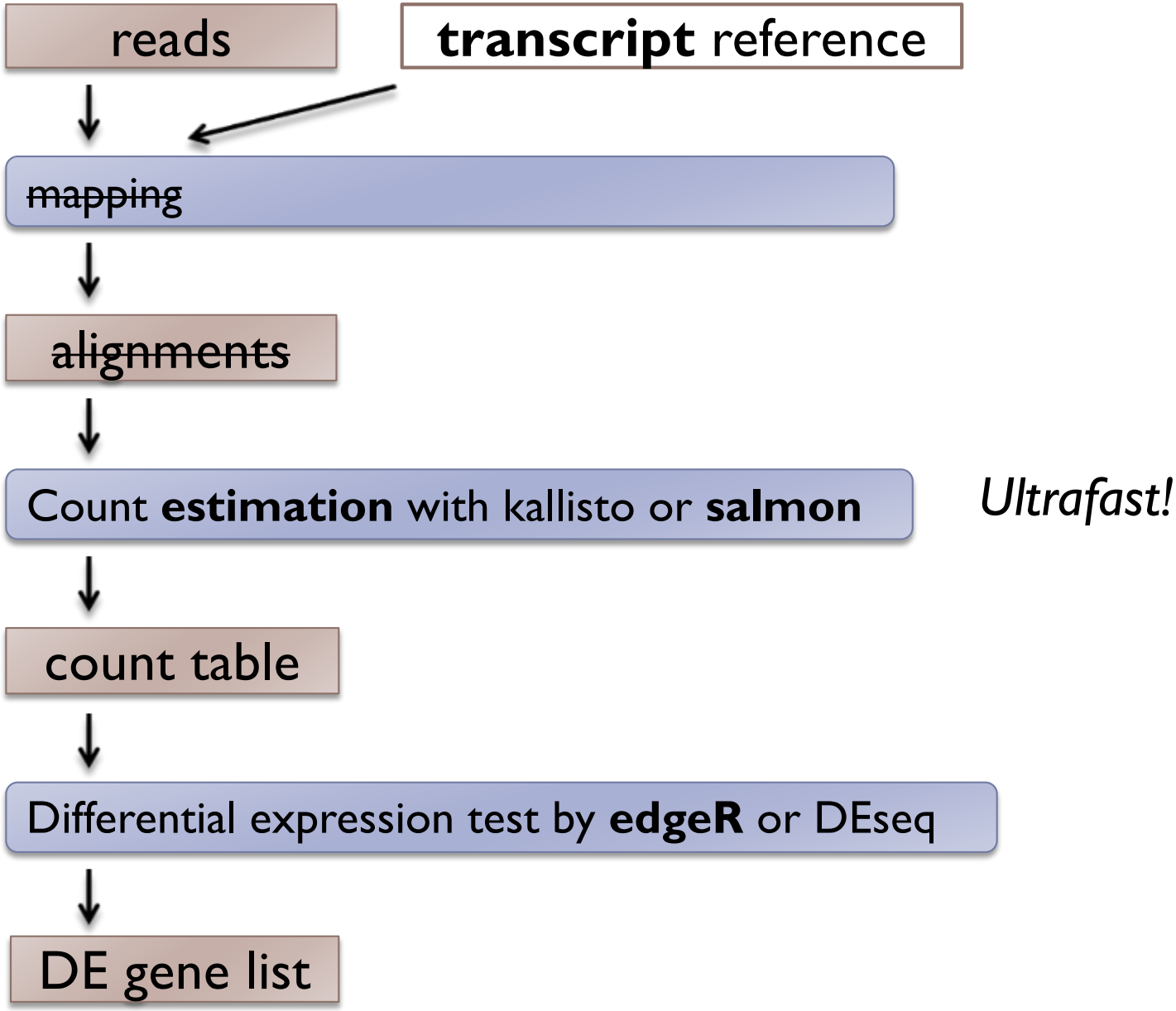
- Choice of reference

  - **Genome** – standard for genome-known species

  - **Transcript** – the only way for genome-unknown species
    -- can be used for genome-known species

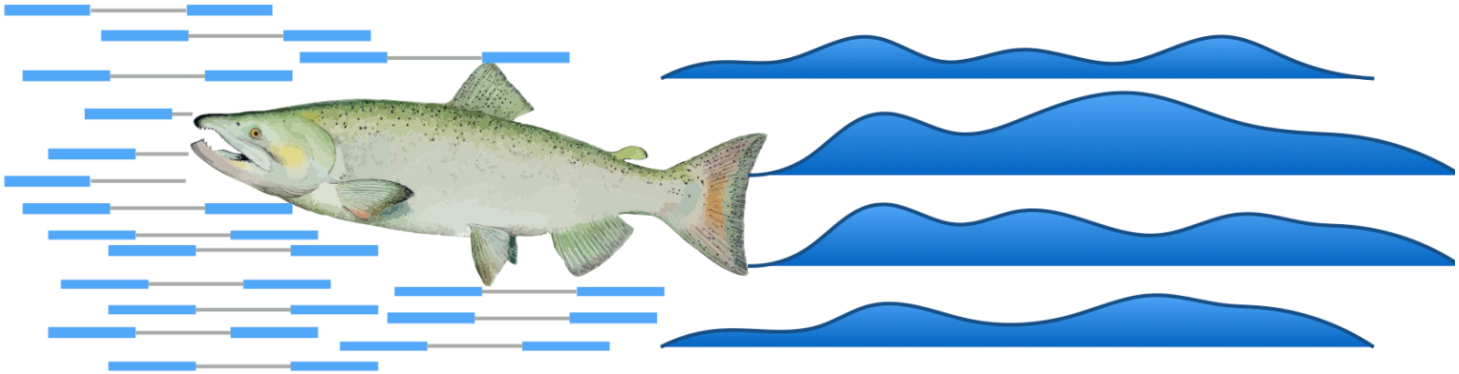

Millions of short reads
pre-processing
mapping
Reads aligned to reference
summarization by unit
(gene, transcript, exon)
Table of counts
normalization
DE testing
List of DE gene
systems biology
- GO enrichment
- multivariate analysis
- network analysis
Biological insights

A transcript-based pipeline (alignment-free method)



Alignment-free RNAseq quantification

- Software
  - Salmon
  - Kallisto
  - Sailfish
- Concept
  - Precise alignments are not required to assign reads to their origins.
  - => “psudo-alignment” using a de bruijn graph information (kallisto), a k-mer approach (Sailfish old ver.) or a “quasi-mapping” (Salmon)
- Benefit
  - Ultrafast
  - Computationally cheap
  - Accuracy: similar or better than mapping-based methods



Salmon —*Don't count . . . quantify!*

Overview

Salmon is a tool for quantifying the expression of transcripts using RNA-seq data. Salmon uses new algorithms (specifically, coupling the concept of *quasi-mapping* with a two-phase inference procedure) to provide accurate expression estimates very quickly (i.e. *wicked-fast*) and while using little memory. Salmon performs its inference using an expressive and realistic model of RNA-seq data that takes into account experimental attributes and biases commonly observed in *real* RNA-seq data.

Citing Salmon

If you find Salmon useful, or have suggestions for improv https://combine-lab.github.io/salmon/
Salmon in your work, please cite the Salmon paper:

Salmon

Salmon is a tool for quantifying the expression of transcripts using RNA-seq data. Salmon uses new algorithms to provide accurate expression estimates very quickly.

```
(example)

$salmon index …    # step 1. build index
$salmon quant …    # step 2. quantification
```

- Input
  - reference (fasta) and reads (fastq)
- Output
  - Count estimation table: **`quant.sf`**

Let's Try **Salmon**

ex401

Map 75-bp Illumina reads to a transcript reference and quantify the abundance.

Prepare reads and reference genome

Sequences for this exercise are stored in **~/gitc/data/SS/**.

```
IlluminaReads1.fq : Illumina reads in fastq format
minimouse_mRNA.fa : a set of transcript sequnences
```

Build index of reference sequence

```
$salmon index -t minimouse_mRNA.fa ¥
  -i minimouse_mRNA.fa.salmon.idx -k 31
```

Quantification

```
$salmon quant -i minimouse_mRNA.fa.salmon.idx ¥
  -l A -o salmon_out -r IlluminaReads1.fq
```

Salmon outputs

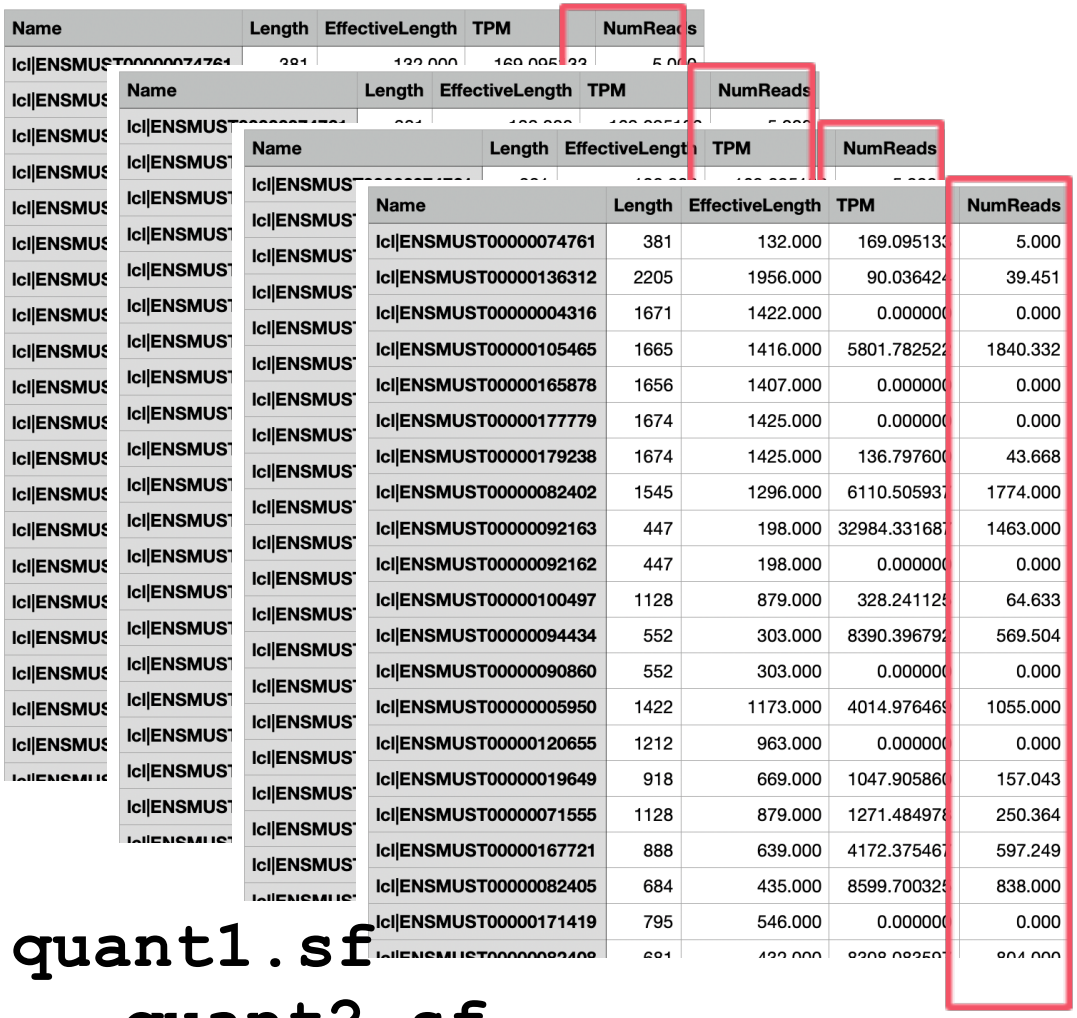
NumReads => edgeR

quant.sf

Name	Length	EffectiveLength	TPM	NumReads
lcl|ENSMUST00000074761	381	132.000	169.095133	5.000
lcl|ENSMUST00000136312	2205	1956.000	90.036424	39.451
lcl|ENSMUST00000004316	1671	1422.000	0.000000	0.000
lcl|ENSMUST00000105465	1665	1416.000	5801.782522	1840.332
lcl|ENSMUST00000165878	1656	1407.000	0.000000	0.000
lcl|ENSMUST00000177779	1674	1425.000	0.000000	0.000
lcl|ENSMUST00000179238	1674	1425.000	136.797600	43.668
lcl|ENSMUST00000082402	1545	1296.000	6110.505937	1774.000
lcl|ENSMUST00000092163	447	198.000	32984.331687	1463.000
lcl|ENSMUST00000092162	447	198.000	0.000000	0.000
lcl|ENSMUST00000100497	1128	879.000	328.241125	64.633
lcl|ENSMUST00000094434	552	303.000	8390.396792	569.504
lcl|ENSMUST00000090860	552	303.000	0.000000	0.000
lcl|ENSMUST00000005950	1422	1173.000	4014.976469	1055.000
lcl|ENSMUST00000120655	1212	963.000	0.000000	0.000
lcl|ENSMUST00000019649	918	669.000	1047.905860	157.043
lcl|ENSMUST00000071555	1128	879.000	1271.484978	250.364
lcl|ENSMUST00000167721	888	639.000	4172.375467	597.249
lcl|ENSMUST00000082405	684	435.000	8599.700325	838.000
lcl|ENSMUST00000171419	795	546.000	0.000000	0.000
lcl|ENSMUST00000082408	681	432.000	8208.082597	804.000

Salmon to edgeR

Notes:
Count matrix 作成するのに一手間かける必要があることに言及。最近のsalmonではsalmon merge でそれができるようになった。前回までお手製rubyスクリプトを使っていたが、今回からsalmon mergeにswitch



Name	Length	EffectiveLength	TPM	NumReads
lcl|ENSMUST00000074761	381	132.000	169.095133	5.000
lcl|ENSMUST00000136312	2205	1956.000	90.036424	39.451
lcl|ENSMUST00000004316	1671	1422.000	0.000000	0.000
lcl|ENSMUST00000105465	1665	1416.000	5801.782522	1840.332
lcl|ENSMUST00000165878	1656	1407.000	0.000000	0.000
lcl|ENSMUST00000177779	1674	1425.000	0.000000	0.000
lcl|ENSMUST00000179238	1674	1425.000	136.797600	43.668
lcl|ENSMUST00000082402	1545	1296.000	6110.505937	1774.000
lcl|ENSMUST00000092163	447	198.000	32984.331687	1463.000
lcl|ENSMUST00000092162	447	198.000	0.000000	0.000
lcl|ENSMUST00000100497	1128	879.000	328.241125	64.633
lcl|ENSMUST00000094434	552	303.000	8390.396792	569.504
lcl|ENSMUST00000090860	552	303.000	0.000000	0.000
lcl|ENSMUST00000005950	1422	1173.000	4014.976469	1055.000
lcl|ENSMUST00000120655	1212	963.000	0.000000	0.000
lcl|ENSMUST00000019649	918	669.000	1047.905860	157.043
lcl|ENSMUST00000071555	1128	879.000	1271.484978	250.364
lcl|ENSMUST00000167721	888	639.000	4172.375467	597.249
lcl|ENSMUST00000082405	684	435.000	8599.700325	838.000
lcl|ENSMUST00000171419	795	546.000	0.000000	0.000

quant1.sf
quant2.sf
quant3.sf
quant4.sf

Name	Lib-1|NumReads	Lib-2|NumReads	Lib-3|NumReads	Lib-4|NumReads
lcl|ENSMUST00000074761	5.000	27.957	15.037	230.000
lcl|ENSMUST00000136312	39.451	674.000	61.696	42.809
lcl|ENSMUST00000004316	0.000	528.689	0.000	156.235
lcl|ENSMUST00000105465	1840.332	521.304	111.396	0.000
lcl|ENSMUST00000165878	0.000	148.549	0.000	218.000
lcl|ENSMUST00000177779	0.000	470.496	348.508	215.000
lcl|ENSMUST00000179238	43.668	0.000	0.000	29.956
lcl|ENSMUST00000082402	1774.000	195.495	1.534	192.288
lcl|ENSMUST00000092163	1463.000	499.000	260.604	159.365
lcl|ENSMUST00000092162	0.000	454.000	104.191	198.000
lcl|ENSMUST00000100497	64.633	447.000	0.000	0.000
lcl|ENSMUST00000094434	569.504	0.000	0.000	159.985
lcl|ENSMUST00000090860	0.000	410.000	1.608	0.000
lcl|ENSMUST00000005950	1055.000	332.068	135.311	1.002
lcl|ENSMUST00000120655	0.000	266.106	239.000	192.000
lcl|ENSMUST00000019649	157.043	406.000	0.000	6.017
lcl|ENSMUST00000071555	250.364	0.000	264.000	0.000
lcl|ENSMUST00000167721	597.249	4.998	89.000	135.272

count matrix

```
$salmon quantmerge …
```

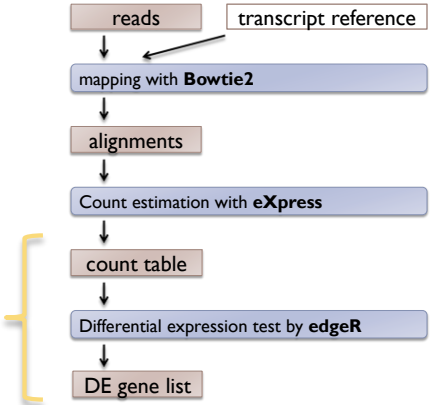


Table of counts

data import

diagnostics

normalization

DE testing

evaluation

List of DE gene

edgeR

- A Bioconductor package for differential expression analysis of digital gene expression data
- **Model**: An over dispersed Poisson model, negative binomial (NB) model, is used
- **Normalization**: TMM method (trimmed mean of M values) to deal with composition effects
- **DE test**: exact test and generalized linear models (GLM)

edgeR (classic)

- input: **count data** (not RPKM or TPM)
- output: gene table with DE significance statistics (FDR)

```
(example)

$ R
> library(edgeR)                          #load edgeR library
> dat <- read.delim("count_data.txt", …) #import count table to R
> group <- c(rep("M", 3), rep("H", 3))    #assign groups
> D <- DGEList(dat, group=group)          #import data to edgeR
> D <- calcNormFactors(D)                 #normalization (TMM)
> D <- estimateCommonDisp(D)              #estimate common disbpersion
> D <- estimateTagwiseDisp(D)             #estimate tagwise dispersion
> de <- exactTest(D, pair=c("M", "H")) #DE test
> topTags(de)
Comparison of groups: H-M
            logConc    logFC      P.Value          FDR
AT5G48430 -15.36821 6.255498 9.919041e-12 2.600872e-07
AT5G31702 -15.88641 5.662522 3.637593e-10 4.083773e-06
AT3G55150 -17.01537 5.870635 4.672331e-10 4.083773e-06
…
```

Advanced

edgeR (GLM)

- input: **count data** (not RPKM or TPM)
- output: gene table with DE significance statistics (FDR)

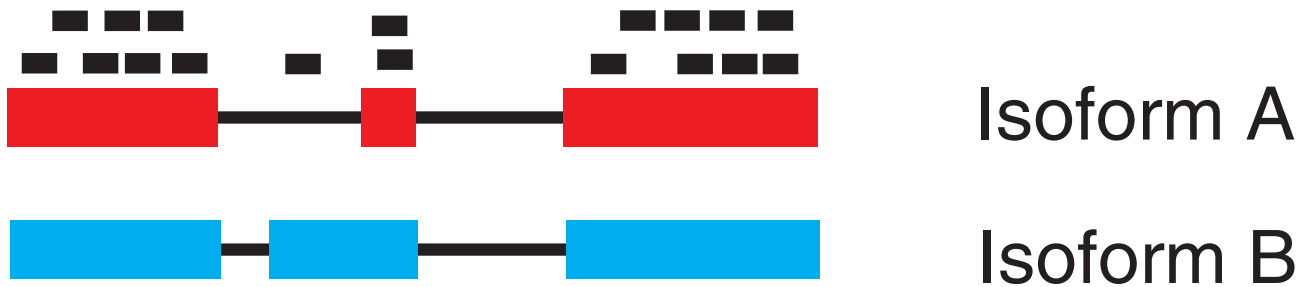
```
(example)

$ R
> library(edgeR)                          #load edgeR library
> dat <- read.delim("count_data.txt", …) #import count table to R

> treat <- factor(c("M", "M", "M", "H", "H", "H"))
> treat <- relevel(treat, ref="M)
> design <- model.matrix(~treat)
> rownames(design) <- colnames(y)

> D <- DGEList(dat, group=treat)      #import data to edgeR
> D <- calcNormFactors(D, method="TMM")   #normalization (TMM)
> D <- estimateDisp(D, design)        #estimate dispersion
> fit <- glmFit(D, design)            #fitting to model
> lrt <- glmLRTt(D, coef=2))          #DE test
> topTags(lrt)
> …
```

Let's try edgeR

- edgeR classic
  - ex402: Differential expression analysis with edgeR (pairwise)
- edgeR linear model [advanced]
  - ex403-1: Differential expression analysis with edgeR (GLM)
  - ex403-2: Differential expression analysis with edgeR (GLM; considering batch effect)

Estimate Abundance

- **Multimapping issues**
  - Isoforms
  - Very similar paralogs
  - Repetitive sequences
  - => cannot align reads uniquely
- Mapping ambiguity should be taken into consideration.



- Critical for RNA-seq de novo analysis
- Software: RSEM and eXpress (EM algorithm)