基礎生物学研究所　ゲノムインフォマティクス・トレーニングコース　2019年 3月

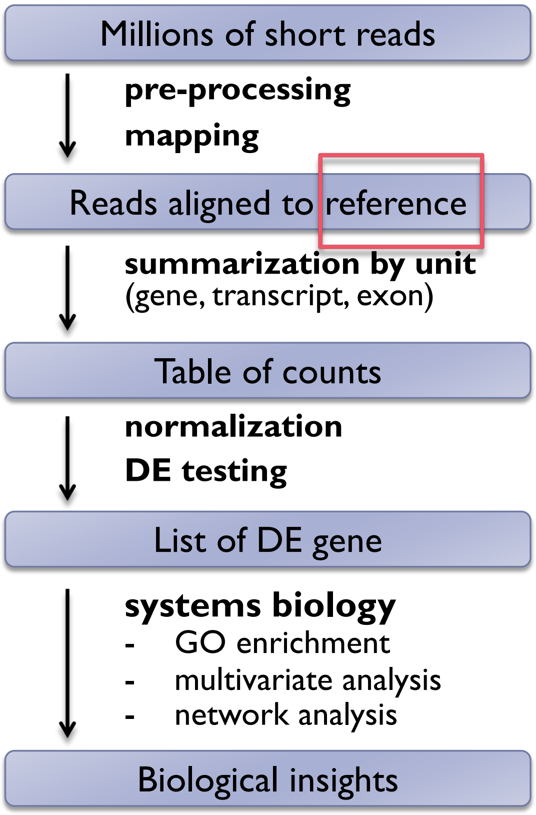
RNA-seq解析パイプライン：Transcript-based

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

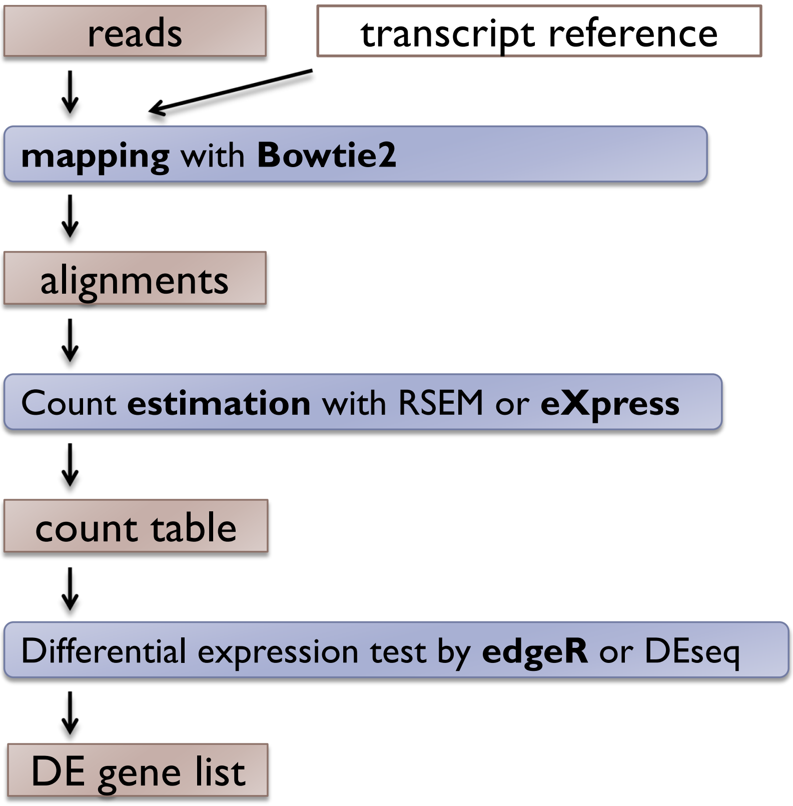
List of DE gene

systems biology

- GO enrichment
- multivariate analysis
- network analysis

Biological insights

A Pipeline: Transcript-based



reads

transcript reference

mapping with **Bowtie2**

alignments

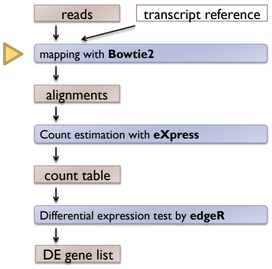
Count **estimation** with RSEM or **eXpress**

count table

Differential expression test by **edgeR** or DEseq

DE gene list

Mapping – alignment software



- For mapping reads onto transcript reference

  short read mapper (unspliced read aligner) is used

  - **Bowtie2**

    http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

bowtie2

Bowtie is an ultrafast, memory-efficient short read aligner.

http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

```
(example)

$ bowtie2 –x transcript.fa –U reads.fq –a –S out.sam
```

- Input
  - Reads (fastq) and reference (bowtie2-db)
- Output
  - Alignment in SAM format : **`out.sam`**

Let’s Try Bowtie2

Align 75-bp Illumina reads with a transcript reference using Bowtie2.

Prepare reads and reference genome

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

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

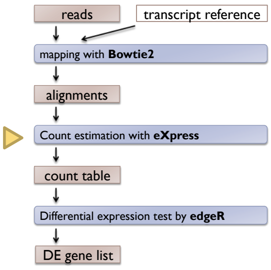
Build index of reference sequence

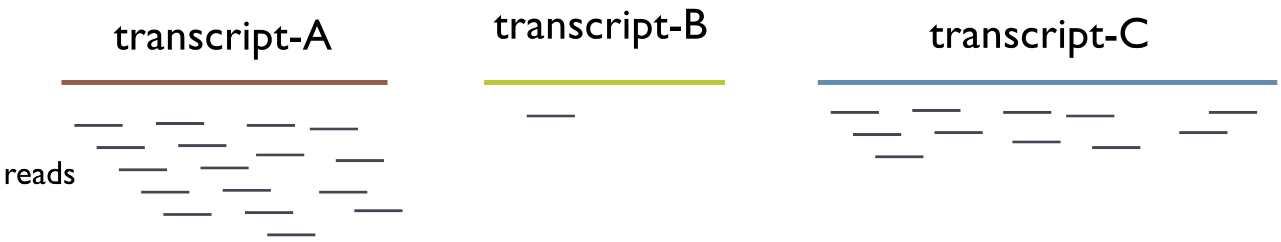
```
$bowtie2-build minimouse_mRNA.fa myref
```

Align reads with reference

```
$bowtie2 –x myref –U IlluminaReads1.fq –a –S out.sam
```

Count Reads by Transcript/gene





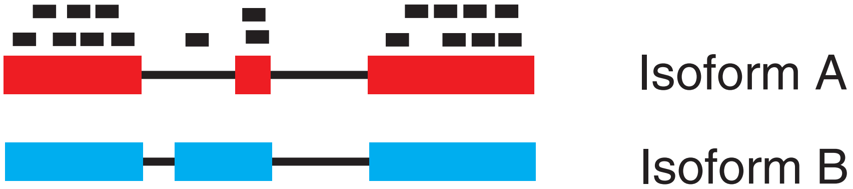
- The simplest way: just count reads by contig.

But…

- Mapping ambiguity should be taken into consideration.

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)

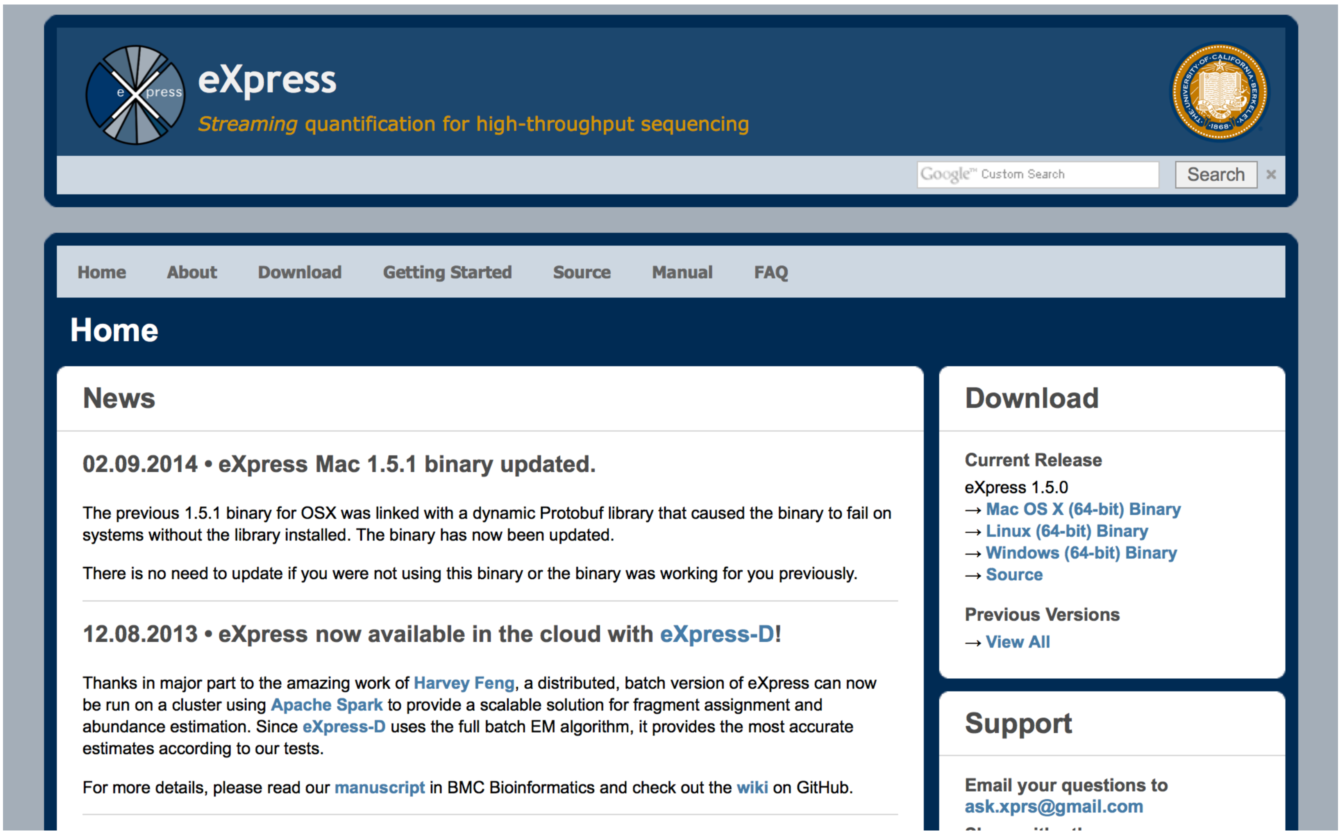
eXpress

eXpress is a streaming tool for quantifying the abundances of a set of target sequences from sampled subsequences.

http://bio.math.berkeley.edu/eXpress/

```
(example)

$ express transcripts.fasta hits.bam
```

- Input
  - alignment (bam|sam) and reference (fasta)
- Output
  - Count estimation table: **`results.xprs`**



Home

News

02.09.2014 • eXpress Mac 1.5.1 binary updated.

The previous 1.5.1 binary for OSX was linked with a dynamic Protobuf library that caused the binary to fail on systems without the library installed. The binary has now been updated.

There is no need to update if you were not using this binary or the binary was working for you previously.

12.08.2013 • eXpress now available in the cloud with eXpress-D!

Thanks in major part to the amazing work of Harvey Feng, a distributed, batch version of eXpress can now be run on a cluster using Apache Spark to provide a scalable solution for fragment assignment and abundance estimation. Since eXpress-D uses the full batch EM algorithm, it provides the most accurate estimates according to our tests.

For more details, please read our manuscript in BMC Bioinformatics and check out the wiki on GitHub.

Download

Current Release

eXpress 1.5.0
→ Mac OS X (64-bit) Binary
→ Linux (64-bit) Binary
→ Windows (64-bit) Binary
→ Source

Previous Versions

→ View All

Support

Email your questions to
ask.xprs@gmail.com

https://pachterlab.github.io/eXpress/index.html

Let's Try eXpress

Prepare alignments and reference genome

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

```
IlluminaReads1.fq – Illumina reads in fastq format
out.sam – this file should be generated in the previous bowtie practice
```

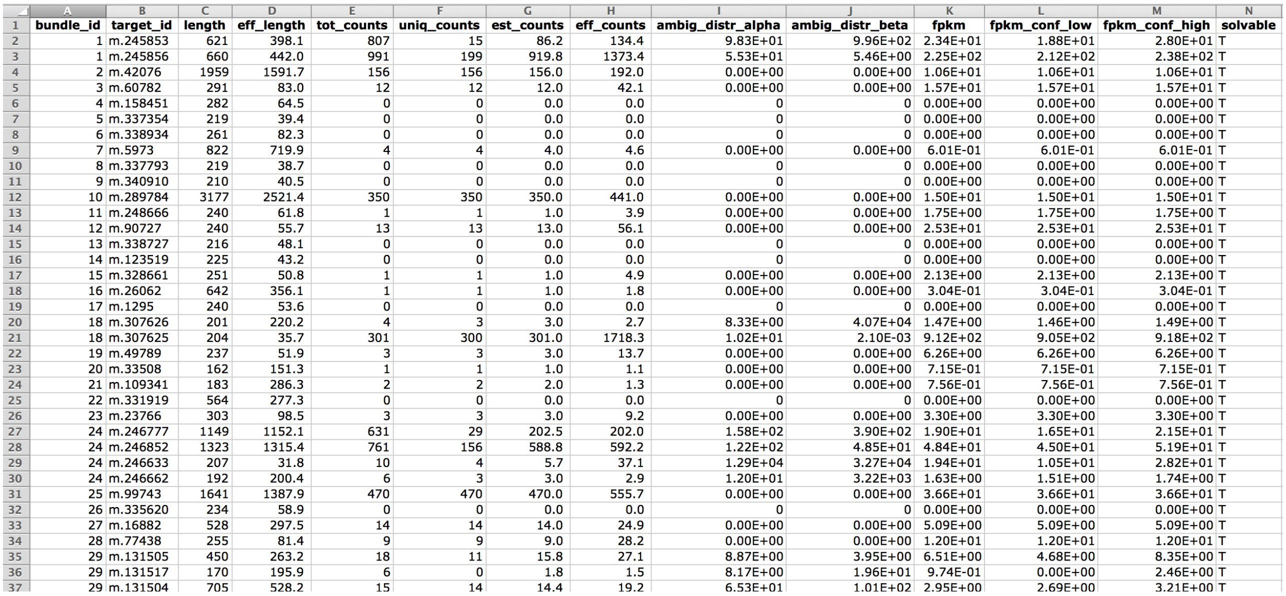
Run eXpress

```
$ express minimouse_mRNA.fa out.sam
```

```
Output : results.xprs, params.xprs
```

eXpress: output

`results.xprs`



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	bundle_id	target_id	length	eff_length	tot_counts	uniq_counts	est_counts	eff_counts	ambig_distr_alpha	ambig_distr_beta	fpkm	fpkm_conf_low	fpkm_conf_high	solvable
2	1	m.245853	621	398.1	807	15	86.2	134.4	9.83E+01	9.96E+02	2.34E+01	1.88E+01	2.80E+01	T
3	1	m.245856	660	442.0	991	199	919.8	1373.4	5.53E+01	5.46E+00	2.25E+02	2.12E+02	2.38E+02	T
4	2	m.42076	1959	1591.7	156	156	156.0	192.0	0.00E+00	0.00E+00	1.06E+01	1.06E+01	1.06E+01	T
5	3	m.60782	291	83.0	12	12	12.0	42.1	0.00E+00	0.00E+00	1.57E+01	1.57E+01	1.57E+01	T
6	4	m.158451	282	64.5	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
7	5	m.337354	219	39.4	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
8	6	m.338934	261	82.3	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
9	7	m.5973	822	719.9	4	4	4.0	4.6	0.00E+00	0.00E+00	6.01E-01	6.01E-01	6.01E-01	T
10	8	m.337793	219	38.7	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
11	9	m.340910	210	40.5	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
12	10	m.289784	3177	2521.4	350	350	350.0	441.0	0.00E+00	0.00E+00	1.50E+01	1.50E+01	1.50E+01	T
13	11	m.248666	240	61.8	1	1	1.0	3.9	0.00E+00	0.00E+00	1.75E+00	1.75E+00	1.75E+00	T
14	12	m.90727	240	55.7	13	13	13.0	56.1	0.00E+00	0.00E+00	2.53E+01	2.53E+01	2.53E+01	T
15	13	m.338727	216	48.1	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
16	14	m.123519	225	43.2	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
17	15	m.328661	251	50.8	1	1	1.0	4.9	0.00E+00	0.00E+00	2.13E+00	2.13E+00	2.13E+00	T
18	16	m.26062	642	356.1	1	1	1.0	1.8	0.00E+00	0.00E+00	3.04E-01	3.04E-01	3.04E-01	T
19	17	m.1295	240	53.6	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
20	18	m.307626	201	220.2	4	3	3.0	2.7	8.33E+00	4.07E+04	1.47E+00	1.46E+00	1.49E+00	T
21	18	m.307625	204	35.7	301	300	301.0	1718.3	1.02E+01	2.10E-03	9.12E+02	9.05E+02	9.18E+02	T
22	19	m.49789	237	51.9	3	3	3.0	13.7	0.00E+00	0.00E+00	6.26E+00	6.26E+00	6.26E+00	T
23	20	m.33508	162	151.3	1	1	1.0	1.1	0.00E+00	0.00E+00	7.15E-01	7.15E-01	7.15E-01	T
24	21	m.109341	183	286.3	2	2	2.0	1.3	0.00E+00	0.00E+00	7.56E-01	7.56E-01	7.56E-01	T
25	22	m.331919	564	277.3	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
26	23	m.23766	303	98.5	3	3	3.0	9.2	0.00E+00	0.00E+00	3.30E+00	3.30E+00	3.30E+00	T
27	24	m.246777	1149	1152.1	631	29	202.5	202.0	1.58E+02	3.90E+02	1.90E+01	1.65E+01	2.15E+01	T
28	24	m.246852	1323	1315.4	761	156	588.8	592.2	1.22E+02	4.85E+01	4.84E+01	4.50E+01	5.19E+01	T
29	24	m.246633	207	31.8	10	4	5.7	37.1	1.29E+04	3.27E+04	1.94E+01	1.05E+01	2.82E+01	T
30	24	m.246662	192	200.4	6	3	3.0	2.9	1.20E+01	3.22E+03	1.63E+00	1.51E+00	1.74E+00	T
31	25	m.99743	1641	1387.9	470	470	470.0	555.7	0.00E+00	0.00E+00	3.66E+01	3.66E+01	3.66E+01	T
32	26	m.335620	234	58.9	0	0	0.0	0.0	0	0	0.00E+00	0.00E+00	0.00E+00	T
33	27	m.16882	528	297.5	14	14	14.0	24.9	0.00E+00	0.00E+00	5.09E+00	5.09E+00	5.09E+00	T
34	28	m.77438	255	81.4	9	9	9.0	28.2	0.00E+00	0.00E+00	1.20E+01	1.20E+01	1.20E+01	T
35	29	m.131505	450	263.2	18	11	15.8	27.1	8.87E+00	3.95E+00	6.51E+00	4.68E+00	8.35E+00	T
36	29	m.131517	170	195.9	6	0	1.8	1.5	8.17E+00	1.96E+01	9.74E-01	0.00E+00	2.46E+00	T
37	29	m.131504	705	528.2	15	14	14.4	19.2	6.53E+01	1.01E+02	2.95E+00	2.69E+00	3.21E+00	T

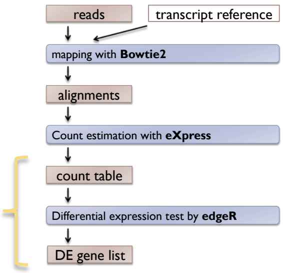


Table of counts

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 dispersion
> 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
…
```

Let’s try edgeR

- edgeR classic
  - ex5: Differential expression analysis with edgeR (pairwise)

Advanced

edgeR (GLM)

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

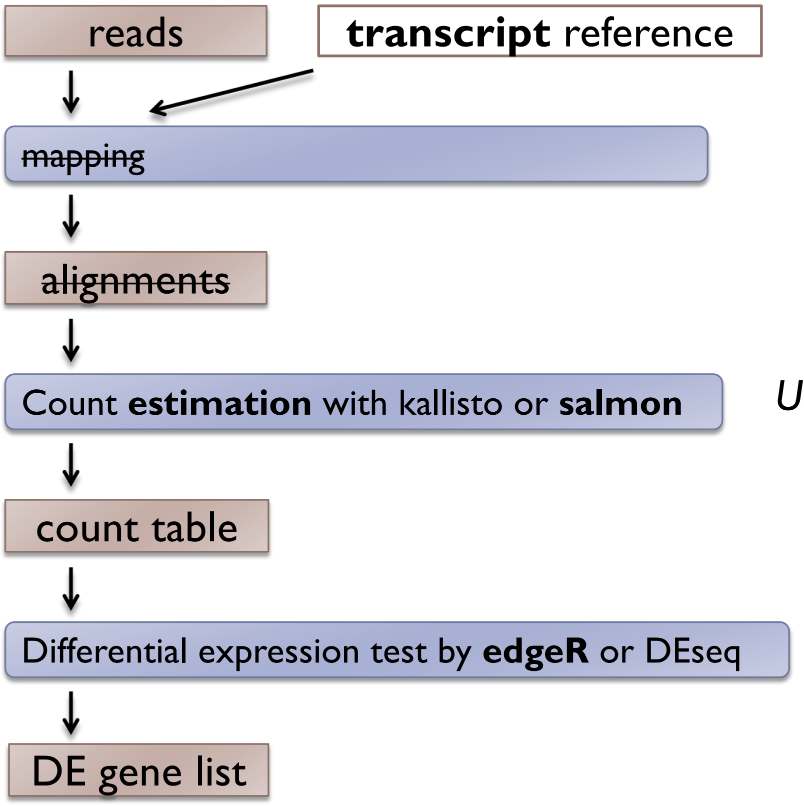
```
(example)

$ 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 (GLM)

- edgeR linear model [advanced]
  - ex12-1: Differential expression analysis with edgeR (GLM)
  - ex12-2: Differential expression analysis with edgeR (GLM; considering batch effect)

new

A transcript-based pipeline (alignment-free method)


reads
transcript reference
~~mapping~~
~~alignments~~
Count **estimation** with kallisto or **salmon**
count table
Differential expression test by **edgeR** or DEseq
DE gene list

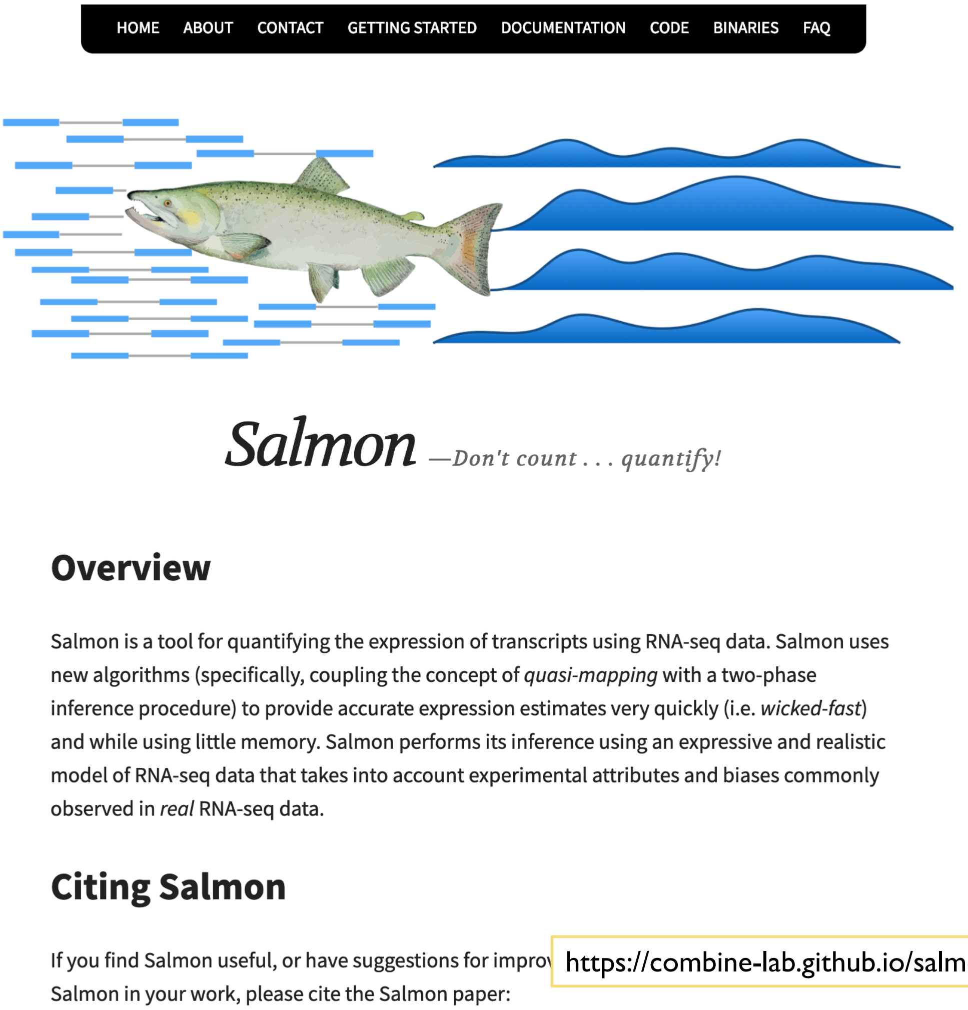
Ultrafast!

new

Alignment-free RNAseq quantification

- Software
  - Salmon
  - Kallisto
  - Sailfish
- Motivation / key 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

new



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

Salmon in your work, please cite the Salmon paper:

https://combine-lab.github.io/salmon/

new

Salmon

```
(two-step procedure)

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

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

ex14

new

Let’s Try **Salmon**

Align 75-bp Illumina reads with a transcript reference using Bowtie2.

Prepare reads and reference genome

Sequences for this exercise are stored in **`~/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_quasi.idx --type quasi -k 31
```

Quantification

```
$salmon quant -i minimouse_mRNA.fa.salmon_quasi.idx \
  -l U -o salmon_out -r IlluminaReads1.fq
```