

NGSワークショップ2015@電農館

2015年12月8日-10日

RNA-Seq法による トランスクリプトーム解析の基礎

国立研究開発法人 農業生物資源研究所

川原 善浩

川原 善浩 (Yoshihiro KAWAHARA)



y.kawahara@affrc.go.jp



@YoshiKawahara

国立研究開発法人 農業生物資源研究所 農業生物先端ゲノム研究センター
ゲノムインフォマティクスユニット 主任研究員

2007年東京都立大学博士課程修了・博士（理学）

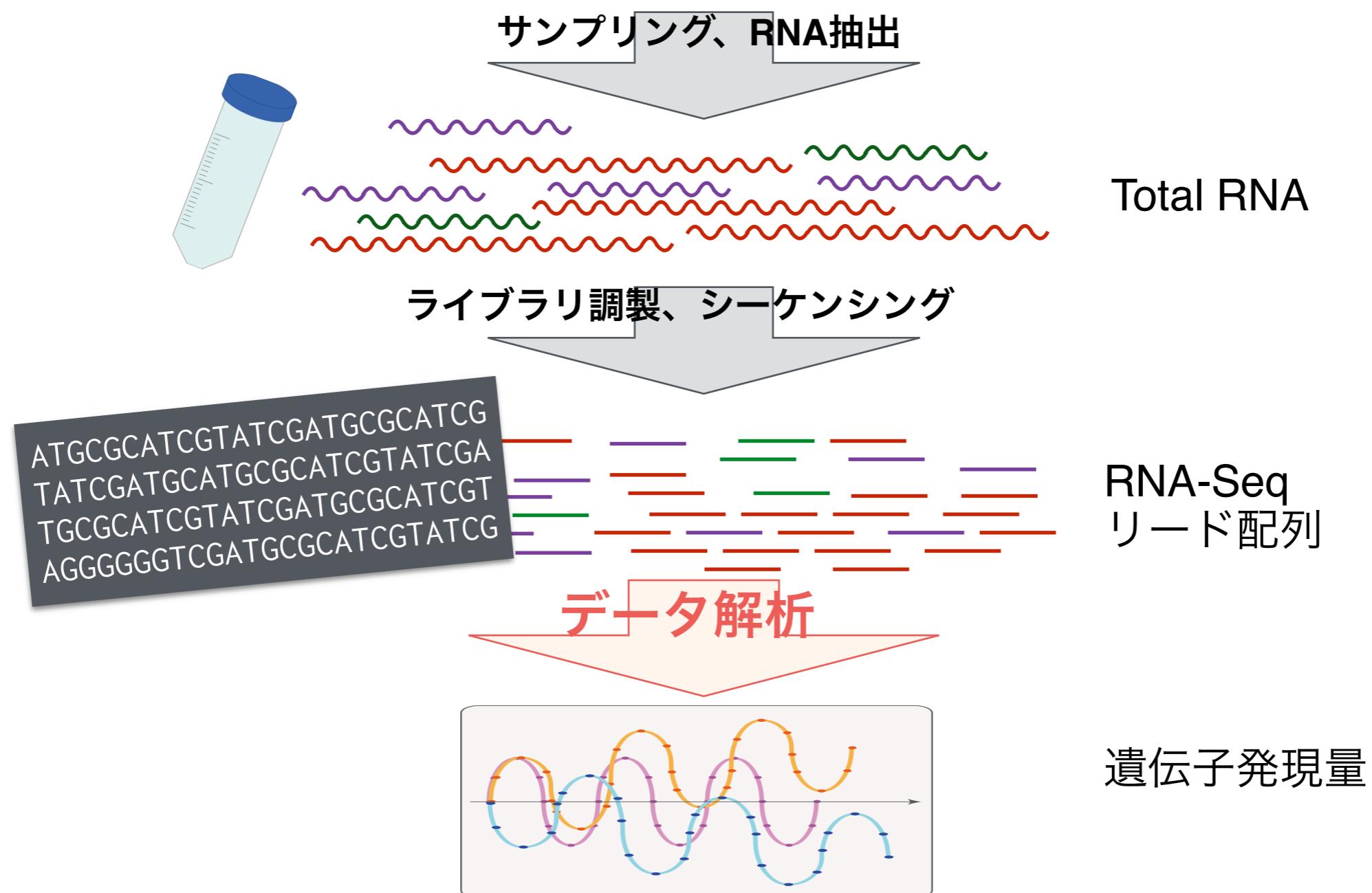
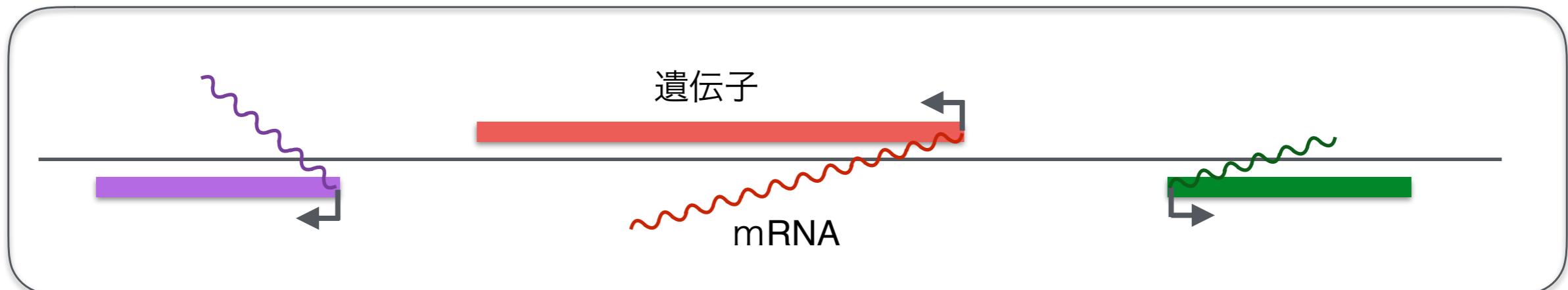
バイオインフォマティクス歴：18年

修士まではウェット&ドライな研究者でしたが、その後は完全なドライに。ショウジョウバエ、酵母、脊椎動物と様々な生物種の比較ゲノム、分子進化研究を行ってきました。8年前に生物研に来てからNGS解析を始め、以来、イネを中心とした植物のゲノム、トランскriプトーム研究に従事。様々なストレスや病気感染時の応答、品種間比較解析等を行っています。

- 主な研究業績 -

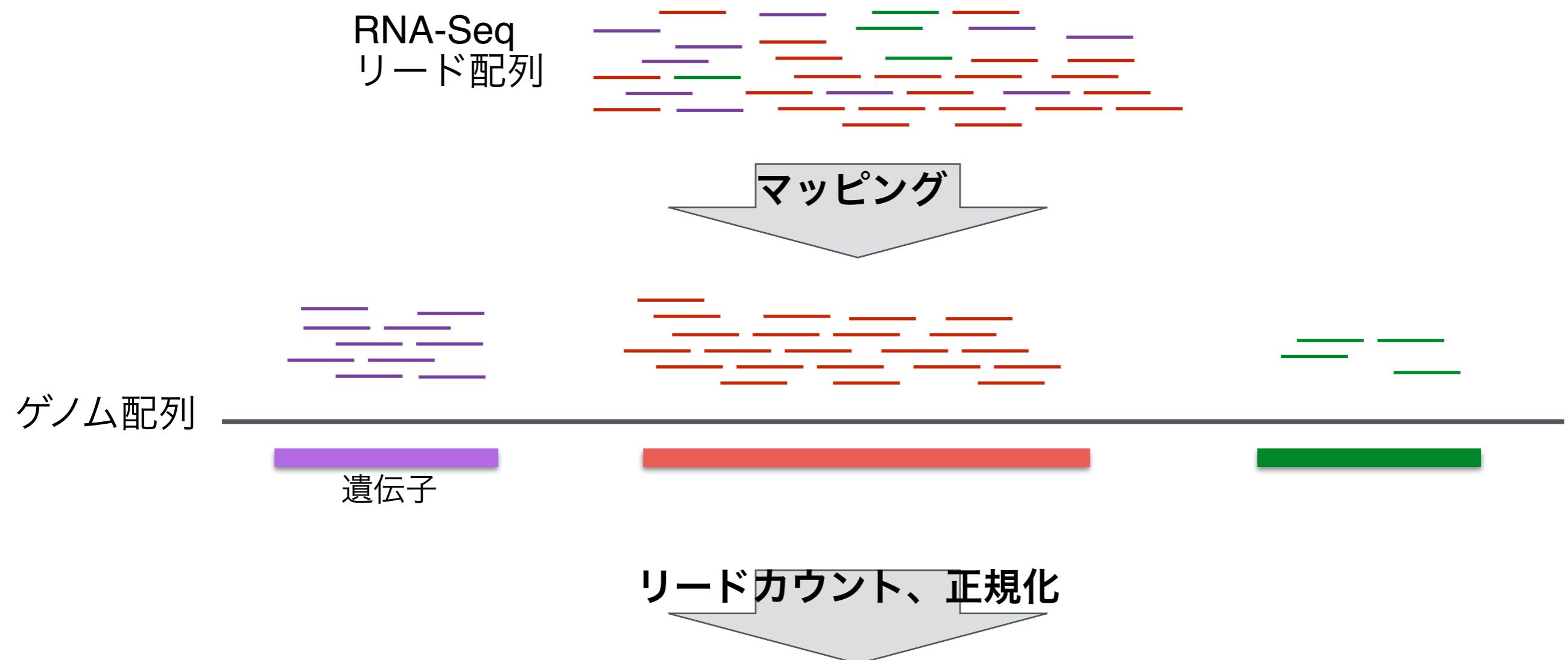
- Kawahara Y. et al. "TENOR: Database for Comprehensive mRNA-Seq Experiments in Rice." **Plant Cell Physiol.** (2015)
- Kawahara Y. et al. "Simultaneous RNA-seq analysis of a mixed transcriptome of rice and blast fungus interaction" **PLoS ONE** (2012)
- Kawahara Y. et al. "Improvement of the *Oryza sativa* Nipponbare reference genome using next generation sequence and optical map data" **Rice** (2013)

RNA-Seq法によるトランскriプトーム研究



戦略1：リファレンスゲノム配列やアノテーションを利用する

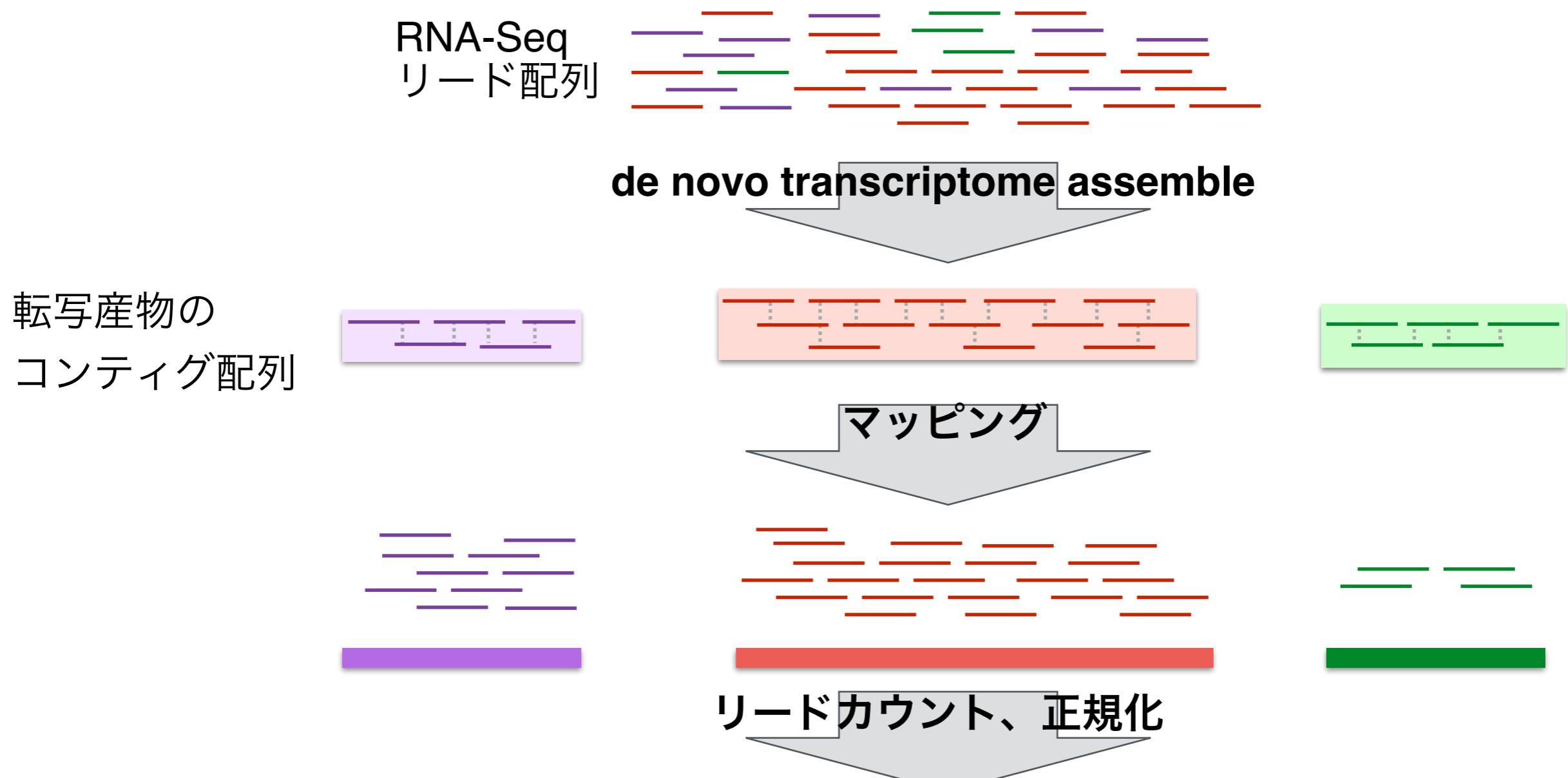
ゲノム配列にマッピング（アラインメント）し、遺伝子ごとにマップされたリードを数え、遺伝子発現を定量する。



戦略2：de novo assemble法を用いた遺伝子発現解析

バラバラになった転写産物配列をアセンブルし、転写産物構造を再構築する。

アセンブルされた転写産物配列上にRNA-Seqリードをマッピング、カウントすることにより遺伝子発現を定量する。



遺伝子発現量：

10

11

4

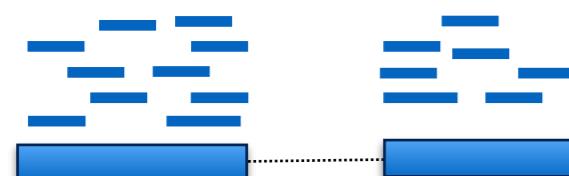
目的に応じたシークエンシング、ライブラリ調製法を選択する

遺伝子発現解析

ゲノムや遺伝子情報は充分。
とにかく発現量が知りたい。

とにかく安価に多くの
リードを読む。

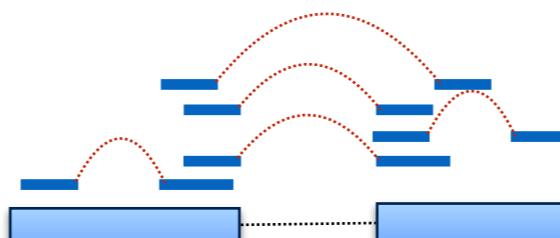
HiSeq, single-read, 50bp



遺伝子構造も発現量も
どちらも！

リード数と転写構造の情
報をバランスよく得る。

HiSeq, paired-end, 100bp

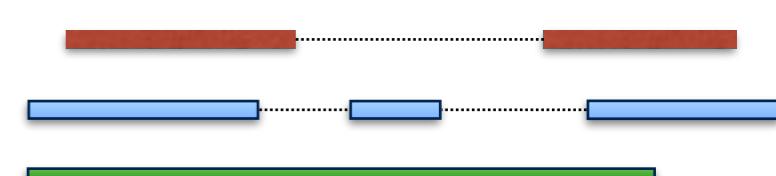


遺伝子カタログ

発現量よりもまずはどん
な遺伝子が発現している
か知りたい。

リード数は少なくとも、
とにかく長いリードで転
写産物の構造を得る。

MiSeq, PacBio RSII



本実習でおこなう解析

1. リファレンスゲノム配列を用いた発現解析
2. 発現解析の結果の可視化
3. *de novo assemble*法による発現解析
4. RNA-Seqデータによる変異解析

リファレンスゲノム配列 を用いた発現解析

でもその前に・・・

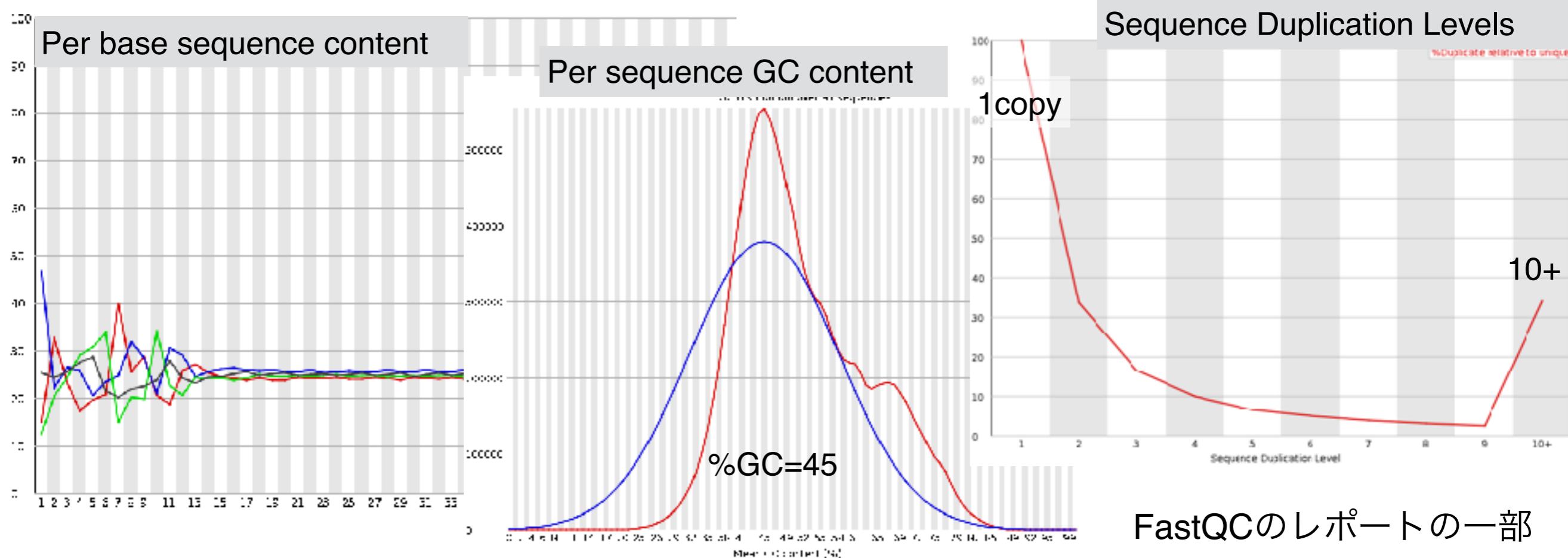
まずはどのようなデータかを知る - QC -

FastQC等のツールを使い、リード数、解読塩基のクオリティやバイアス、配列長などをチェック。

問題があるデータを解析しても
正しい結果は得られないです

RNA-Seqデータの特徴

- ・先頭部分の塩基頻度のバイアス
- ・GCバイアス
- ・高発現遺伝子によると思われるDuplication levelの高さ



FastQCのレポートの一部

解析の前に配列データを綺麗にする - 前処理 -

1. 低クオリティ塩基とアダプター配列の除去

TrimmomaticやFASTX-Toolkitなどのツールを用いる。

(例) Trimmomaticで 「ILLUMINACLIP:[adapter file]:2:30:10 LEADING:15
TRAILING:15 SLIDINGWINDOW:4:15 MINLEN:30」 などと指定して実行する。

2. 実験で除ききれなかったrRNA由来のリードの除去

a. 既知のrRNA配列ライブラリに対してBowtie2等でアラインメントし、そのアンマップリード（つまり、rRNA由来ではないリード）を以降の解析で用いる。

(例) bowtie2 -x [rRNA bowti2 index file] --un-conc-gz [unmap_read.fastq]
-1 [read1.fastq] -2 [read2.fastq] -S [mapped_read.sam]

b. 解析対象外とするゲノム領域（rRNA領域）の位置情報をGFF/GTF形式で用意しておき、発現量計算の際にマスク領域として利用する。

(例) CufflinksやCuffquantの際に、--mask-fileオプションで指定する。

QCや前処理で分かるサンプルの問題点と対応策

1. 低クオリティ塩基が多い。

シークエンシング時の問題であることが多いので読み直す。

2. アダプター配列の混入率が高い。

RNAの濃度が薄い、解読リード長に対してインサート長が短いなど。ライブラリ調製からやり直す。

3. rRNAの混入率が高い。

ポリA精製やrRNAの除去がうまくいっていない。ライブラリ調製からやり直す。

4. ゲノム、転写産物配列へのマップ率が低い。

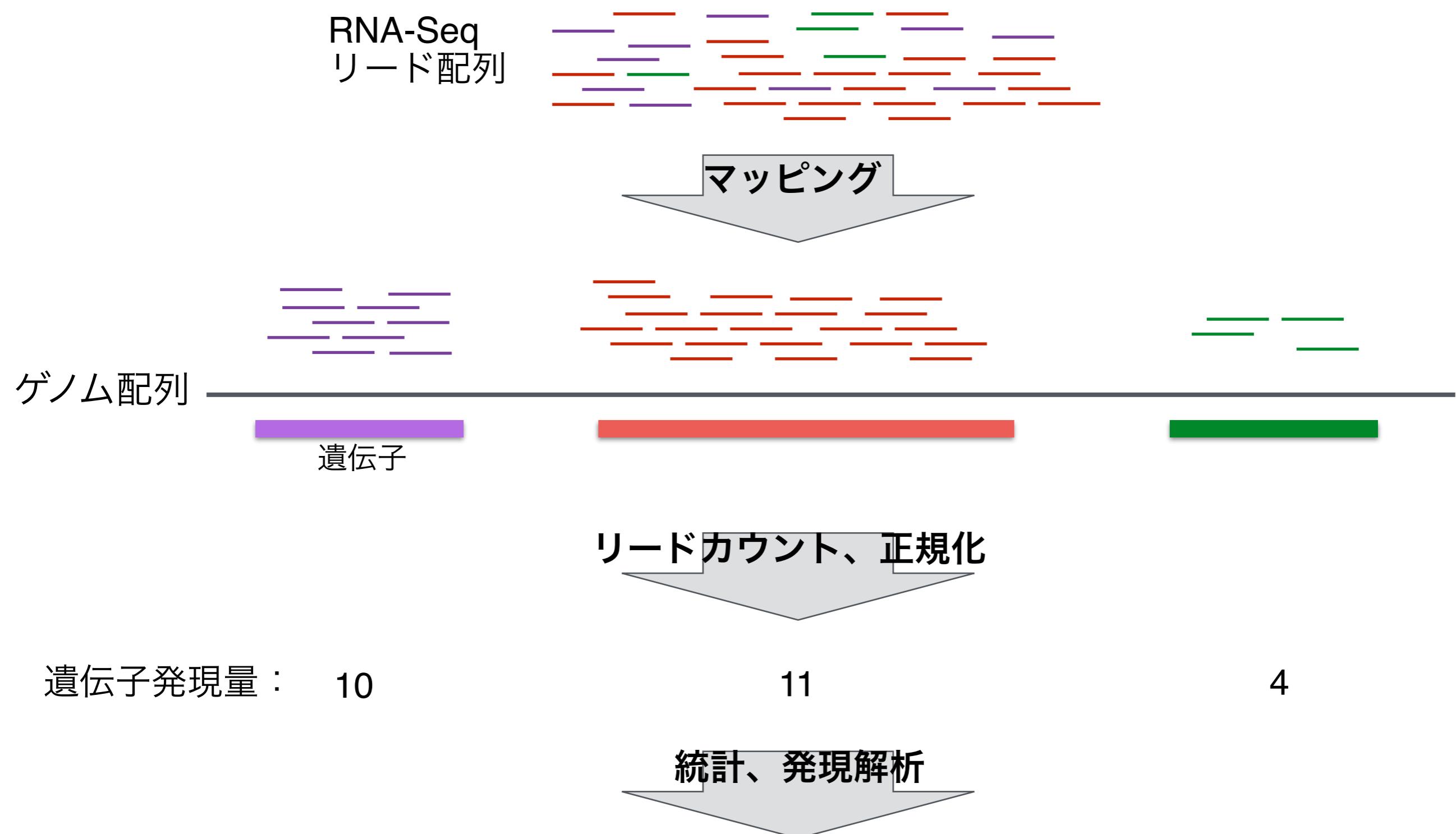
コンタミ等が疑われる。サンプリングからやり直した方が良い場合もある。

5. 有効なリード数が少ない。

シークエンシングを追加する。生物種や対象組織、目的等によっても必要なリード数は異なるが、個人的にはイネの葉であれば2千万（20 million）リードもあれば十分だと思う。

リファレンスゲノム配列 を用いた発現解析

リファレンスゲノム配列を用いた発現解析の流れ



サンプル間比較によるDifferentially expressed gene (DEG)の検出、解析結果の可視化など

RNA-Seq解析の業界標準（？）Tuxedo suite tools

BOW TIE **Bowtie 2**
Fast and sensitive read alignment

JOHNS HOPKINS UNIVERSITY

INSTALL MANUAL GETTING STARTED TOOLS HELP HOW IT WORKS PROTOCOL BENCHMARKS CODE BLOG FEED

Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s or 1,000s of characters, and particularly good at aligning to relatively long (e.g. mammalian) genomes. To keep its memory footprint low, Bowtie 2 uses several different modes.



Bowtie
リードアラインメント

Cufflinks
Transcriptome assembly and differential expression analysis for RNA-Seq.

Cufflinks assembles transcripts and performs gene regulation analysis using parsimonious models based on high-throughput RNA-Seq data.

Cufflinks
遺伝子構造・発現量推定

TopHat
A spliced read mapper for RNA-Seq

MCKUSICK-NATHANS Institute of Genetic Medicine

CUMME RBUND

CUMME RBUND
Exploration, analysis and visualization of Cufflinks high-throughput RNA-Seq data

CSAIL MIT

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

TopHat is a collaboration between the Broad Institute, the University of California, Berkeley, and the University of California, San Francisco.

TopHat
splice-awareアラインメント



CummeRbund is an R package that is designed to aid and simplify the task of analyzing Cufflinks RNA-Seq output.

CummeRbund is developed by Manolis Kellis and the Rinn Laboratory at the Whitehead Institute for Biomedical Research and the Massachusetts Institute of Technology (MIT).

CummeRbund
発現解析結果の可視化



Tuxedo suite toolsの使い方の解説論文

Nat Protoc. 2012 Mar 1;7(3):562-78

PROTOCOL

Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks

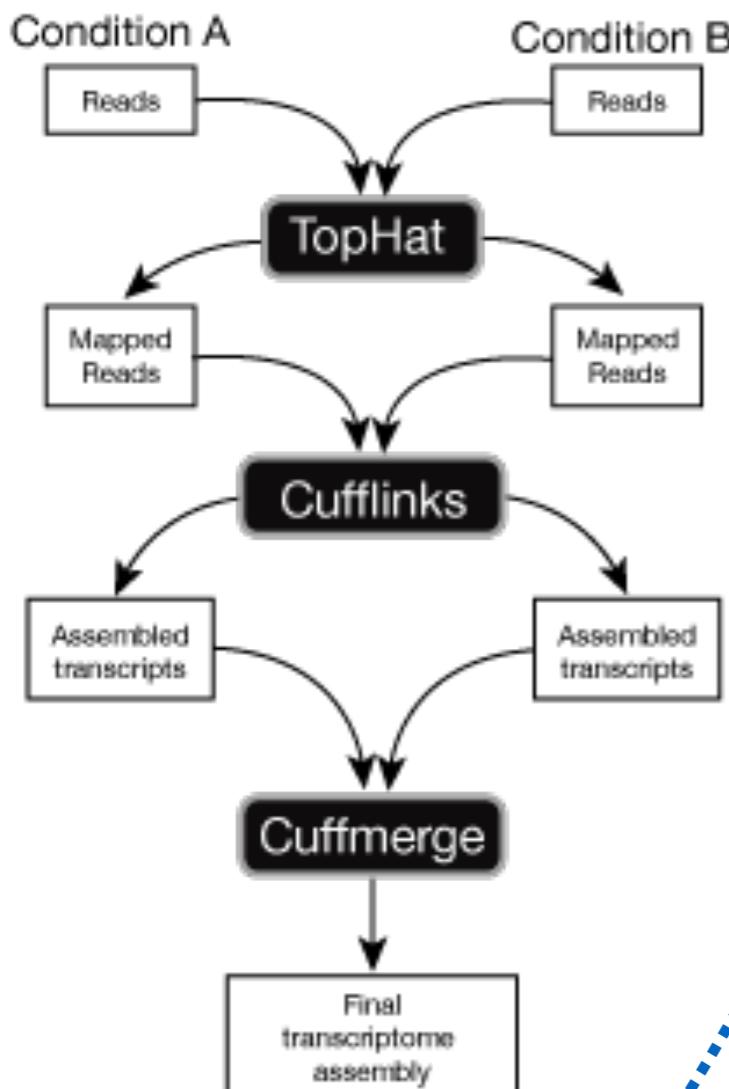
Cole Trapnell^{1,2}, Adam Roberts³, Loyal Goff^{1,2,4}, Geo Pertea^{5,6}, Daehwan Kim^{5,7}, David R Kelley^{1,2}, Harold Pimentel³, Steven L Salzberg^{5,6}, John L Rinn^{1,2} & Lior Pachter^{3,8,9}

参考にはなるが、ツールは更新され続けており、情報は古くなりつつある。

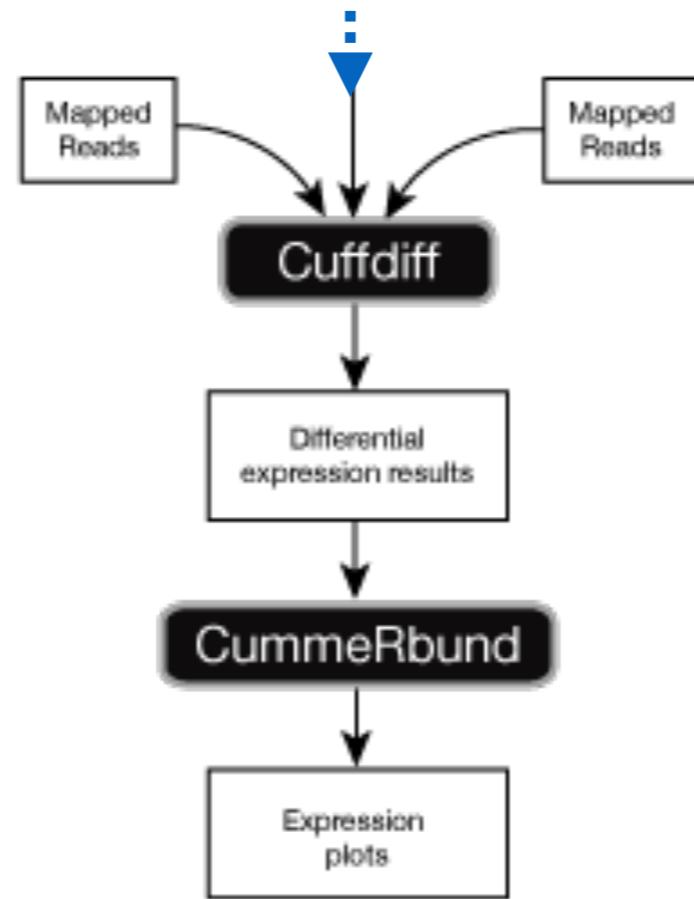
Cufflinks のウェブサイトでは最新のツールの使い方が紹介されている

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>

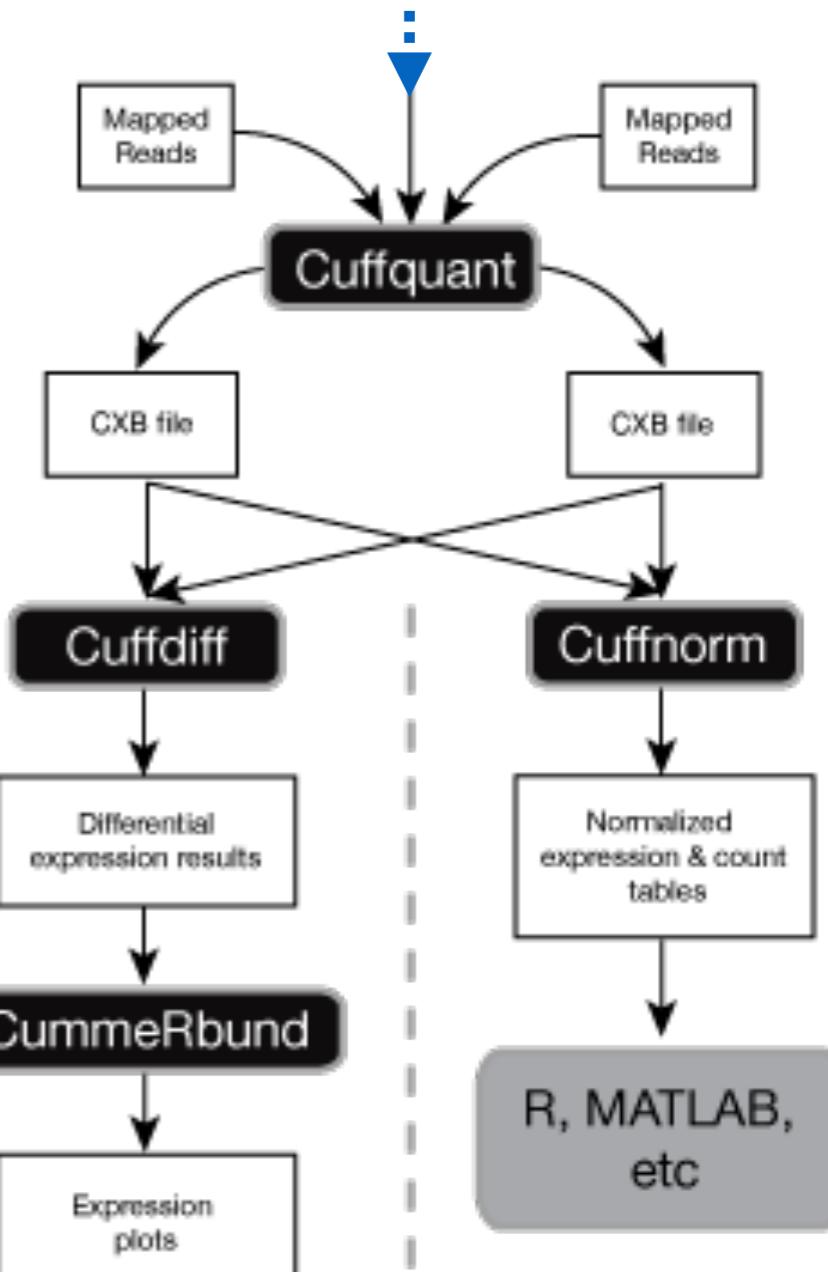
共通ステップ



Cufflinks version < 2.2.0
(still supported)



Cufflinks version >= 2.2.0
(optional)

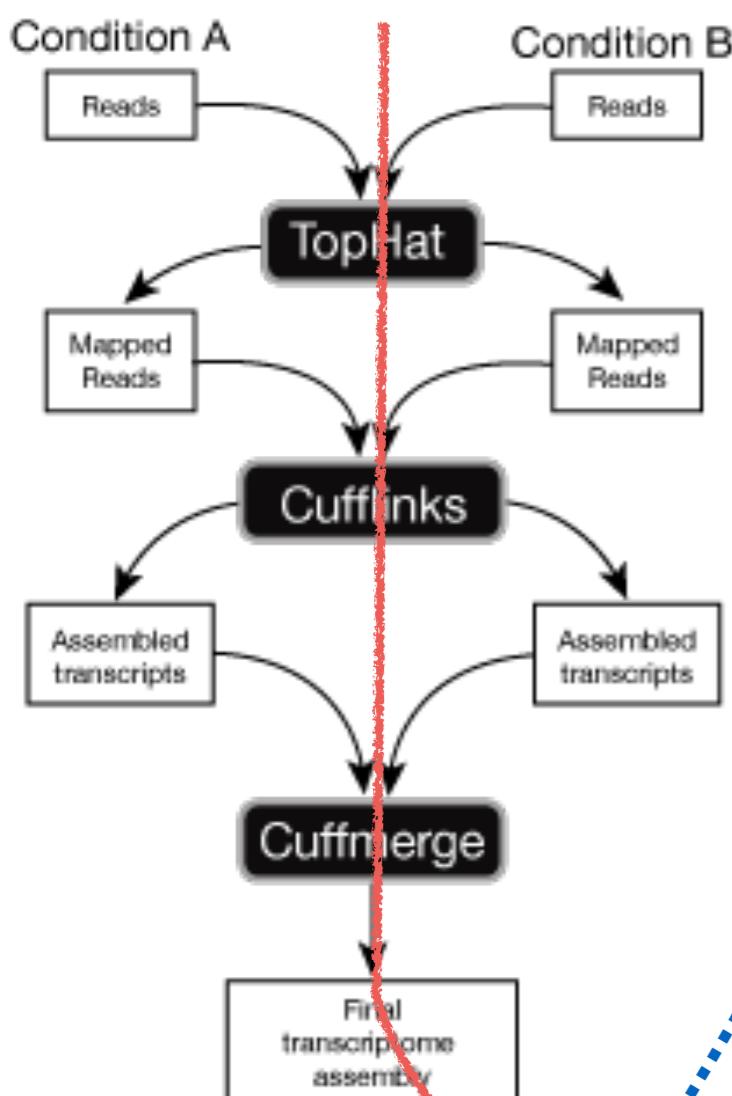


v2.2.0以前とそれ以降では解析の流れや使うツールが若干異なる。

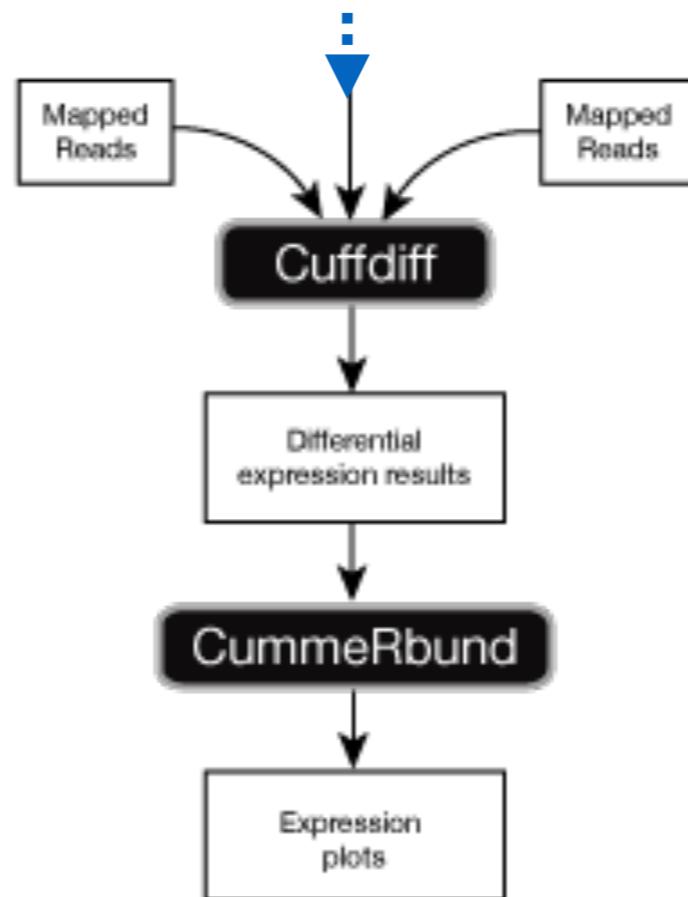
本実習でおこなう解析の流れ

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>

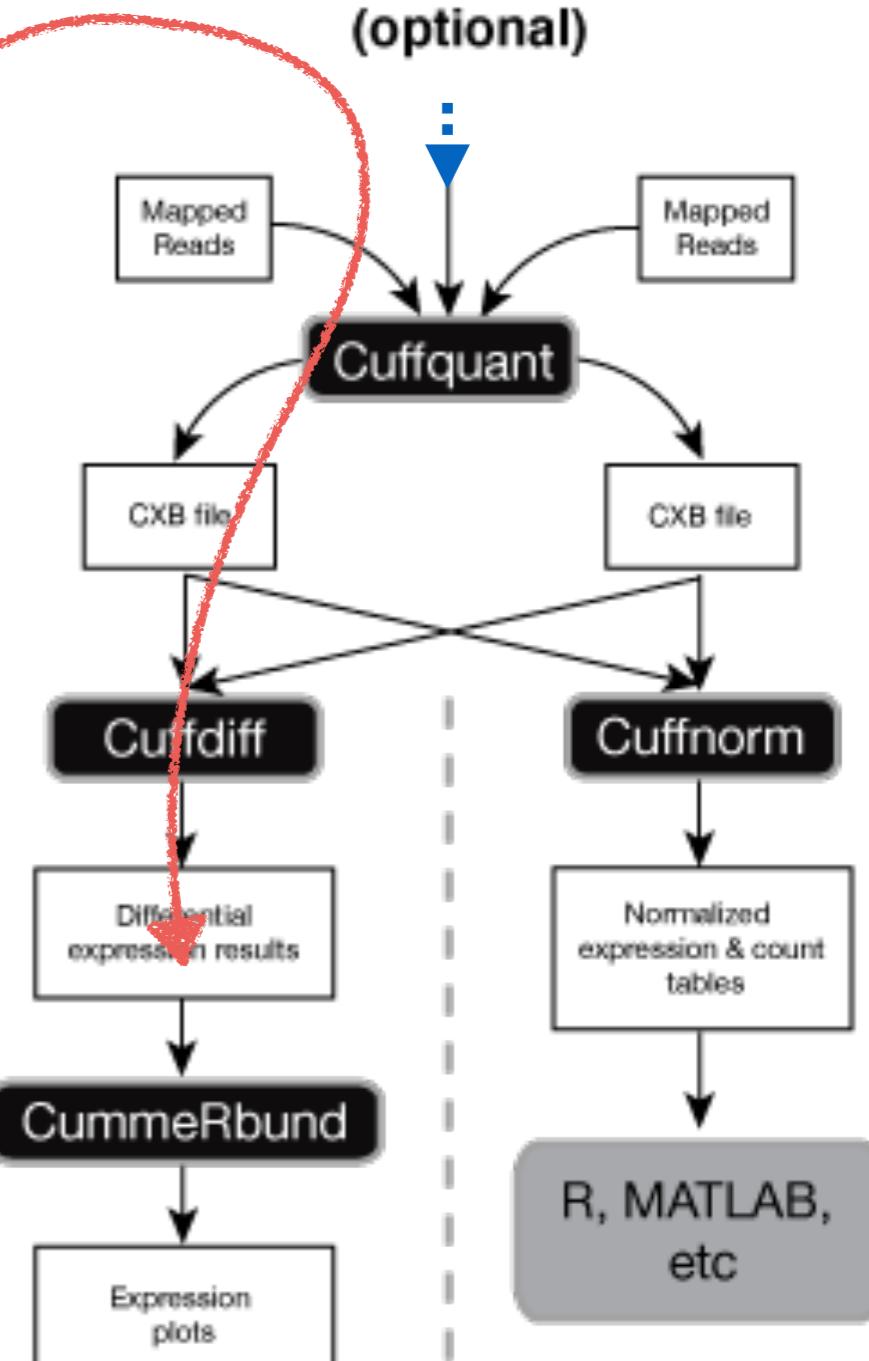
共通ステップ



Cufflinks version < 2.2.0 (still supported)



Cufflinks version >= 2.2.0 (optional)



v2.2.0以前とそれ以降では解析の流れや使うツールが若干異なる。

本実習でおこなう解析と用いるデータ

2群間の遺伝子発現比較

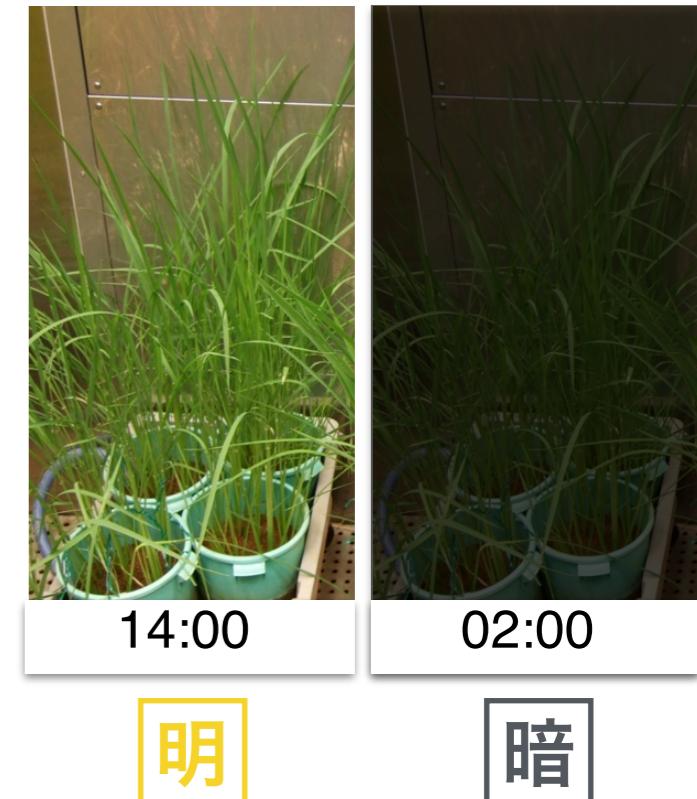
サンプル：イネ（日本晴）の葉@人工気象室

14時 vs 2時（2反復ずつ）

シーケンシング：HiSeq2000、paired-end read (100bp x 2)

解析方法：リファレンスゲノム配列を用いた発現解析

de novo assemble法による発現解析



RNA-Seqデータによる変異検出

サンプル：イネ（日本晴とコシヒカリ）の葉@人工気象室

14時（1反復ずつ）

シーケンシング：HiSeq2000、paired-end read (100bp x 2)

解析方法：リファレンスゲノム配列へのアラインメント、変異検出

*一部のゲノム領域のデータを切り出してきた実習用のデータを使います

実習用データ

0. 実習用データのコピー

```
$ cd  
$ cp -r /work/NGSworkshop2015/rnaseq.tar.gz ./  
$ tar xfz rnaseq.tar.gz  
$ cd rnaseq/ref_alignment/
```

ホームに移動
データ一式を各自のホームにコピー
圧縮ファイルの展開
コピーしたディレクトリに移動

1. 解析前のデータの確認

```
$ ls  
NPB_0200_rep1_r1.fq.gz  NPB_1400_rep2_r1.fq.gz  step_03_cufflinks.sh  
NPB_0200_rep1_r2.fq.gz  NPB_1400_rep2_r2.fq.gz  step_04_cuffmerge.sh  
NPB_0200_rep2_r1.fq.gz  annotation.gtf        step_05_cuffquant.sh  
NPB_0200_rep2_r2.fq.gz  genome.fa              step_06_cuffdiff.sh  
NPB_1400_rep1_r1.fq.gz  step_01_bowtie2-build.sh  
NPB_1400_rep1_r2.fq.gz  step_02_tophat.sh
```

- ・ 日本晴の14時と2時の葉のRNA-Seqデータを利用
- ・ 各サンプル2反復（別の日の同時刻のサンプル）
- ・ 解析対象の領域は第6番染色体の一部（約200kb）

遺伝子アノテーションファイル (GTF2/GFF3)

RAP-DBでイネの代表転写産物のアノテーション情報 (GFF3) がダウンロードできます。

<http://rapdb.dna.affrc.go.jp/download/irgsp1.html>

Annotation data on Os-Nipponbare-Reference-IRGSP-1.0

Genome sequences

- Genome sequence (Os-Nipponbare-Reference-IRGSP-1.0)
Genome assemblies (12 chromosomes)* [\[DOWNLOAD\]](#) (gz file, 116MB, [MD5 checksum](#))
Unanchored sequences* [\[DOWNLOAD\]](#) (gz file, 356KB, [MD5 checksum](#))
(*) Sequences are masked by [Censor](#) with [MIPS](#) and [MSU](#) repeat data. The masked regions are replaced by lowercase letters.
- Chromosome sequences of the aus rice cultivar 'Kasalath'.
[\[DOWNLOAD\]](#) (gz file, 199MB)

Gene set (genes supported by FL-cDNAs, ESTs or proteins)

- Gene structure and function information in GFF format.
[\[DOWNLOAD\]](#) (gz file, 13.6MB)
- Gene sequences (CDS + UTRs + introns) in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 39.5MB)
- Transcript sequences (CDS + UTRs) in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 20.8MB)
- CDS sequences in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 12.4MB)
- Protein sequences (translated CDSs) in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 7.7MB)
- 1 kb upstream sequences of genes in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 13.7MB)
- 2 kb upstream sequences of genes in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 26.9MB)
- 3 kb upstream sequences of genes in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 39.5MB)
- 1 kb downstream sequences of genes in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 13.5MB)
- 2 kb downstream sequences of genes in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 26.4MB)
- 3 kb downstream sequences of genes in FASTA format.
[\[DOWNLOAD\]](#) (gz file, 38.8MB)

様々なデータベースから遺伝子の位置や機能情報が提供されているが、書式や記載方法は必ずしも統一されていない。

TopHatやCufflinksで正しく扱えるような形式に整えたほうがのちのちの解析にも便利です。

[色々な生物のアノテーションファイルを提供しているサイト](#)

Illumina社 iGenomes

https://support.illumina.com/sequencing/sequencing_software/igenome.html

EnsemblPlants

<http://plants.ensembl.org/index.html>

Phytozome

<http://phytozome.jgi.doe.gov/pz/portal.html>

TopHatやCufflinks等で使うGTFファイル

遺伝子の位置やID、アノテーション情報などが記載されたタブ区切りのテキストファイル

```
$ less annotation.gtf
chr01  irgsp1_rep transcript      152853 156449 .       +       . gene_id "Os01g0102800";
transcript_id "Os01t0102800-01"; gene_name "CSB, OsCBS"; note "Similar to chromatin remodeling complex
subunit.";
chr01  irgsp1_rep exon       152853 153025 .       +       . gene_id "Os01g0102800";
transcript_id "Os01t0102800-01";
chr01  irgsp1_rep exon       153178 154646 .       +       . gene_id "Os01g0102800";
transcript_id "Os01t0102800-01";
chr01  irgsp1_rep exon       155010 155450 .       +       . gene_id "Os01g0102800";
transcript_id "Os01t0102800-01";
chr01  irgsp1_rep exon       155543 156449 .       +       . gene_id "Os01g0102800";
transcript_id "Os01t0102800-01";
chr01  irgsp1_rep transcript      164577 168921 .       +       . gene_id "Os01g0102850";
transcript_id "Os01t0102850-00"; gene_name "-"; note "Similar to nitrilase 2.";
chr01  irgsp1_rep exon       164577 164905 .       +       . gene_id "Os01g0102850";
transcript_id "Os01t0102850-00";
chr01  irgsp1_rep exon       168499 168921 .       +       . gene_id "Os01g0102850";
transcript_id "Os01t0102850-00";
```

- ・ 「**gene_id**」（遺伝子座ID）や「**transcript_id**」（転写産物ID）は、遺伝子座や転写産物を対象にした発現解析を行うために必須。
- ・ 「**gene_name**」（遺伝子名やシンボル）を付けておくと、結果ファイルに出力されるので便利。
- ・ 「**tss_id**」（転写開始点ID）や「**p_id**」（タンパク質ID）の情報を記載しておくと、転写開始点やタンパク質ごとの発現解析結果を出力してくれる（今回は省略している）。

実習用シェルスクリプトの使い方

本実習で実行する解析はステップごとにシェルスクリプトにまとめられており、それを順番に実行するだけで良いようになっています。

```
$ cat step_01_bowtie2-build.sh ← catコマンドでシェルスクリプトの内容を表示  
#!/bin/bash
```

```
TOOL_DIR=/usr/local  
BOWTIE_HOME=$TOOL_DIR/bowtie2-2.2.6  
TOPHAT_HOME=$TOOL_DIR/tophat-2.1.0.Linux_x86_64  
CUFFLINKS_HOME=$TOOL_DIR/cufflinks-2.2.1.Linux_x86_64  
SAMTOOLS_HOME=$TOOL_DIR/samtools-1.2  
  
export PATH=$BOWTIE_HOME:$TOPHAT_HOME:$CUFFLINKS_HOME:$SAMTOOLS_HOME:$PATH
```

```
### Step.1: Build index of the reference genome sequence by bowtie2-build  
and samtools
```

```
bowtie2-build genome.fa genome  
samtools faidx genome.fa
```

パスの設定

コマンド実行

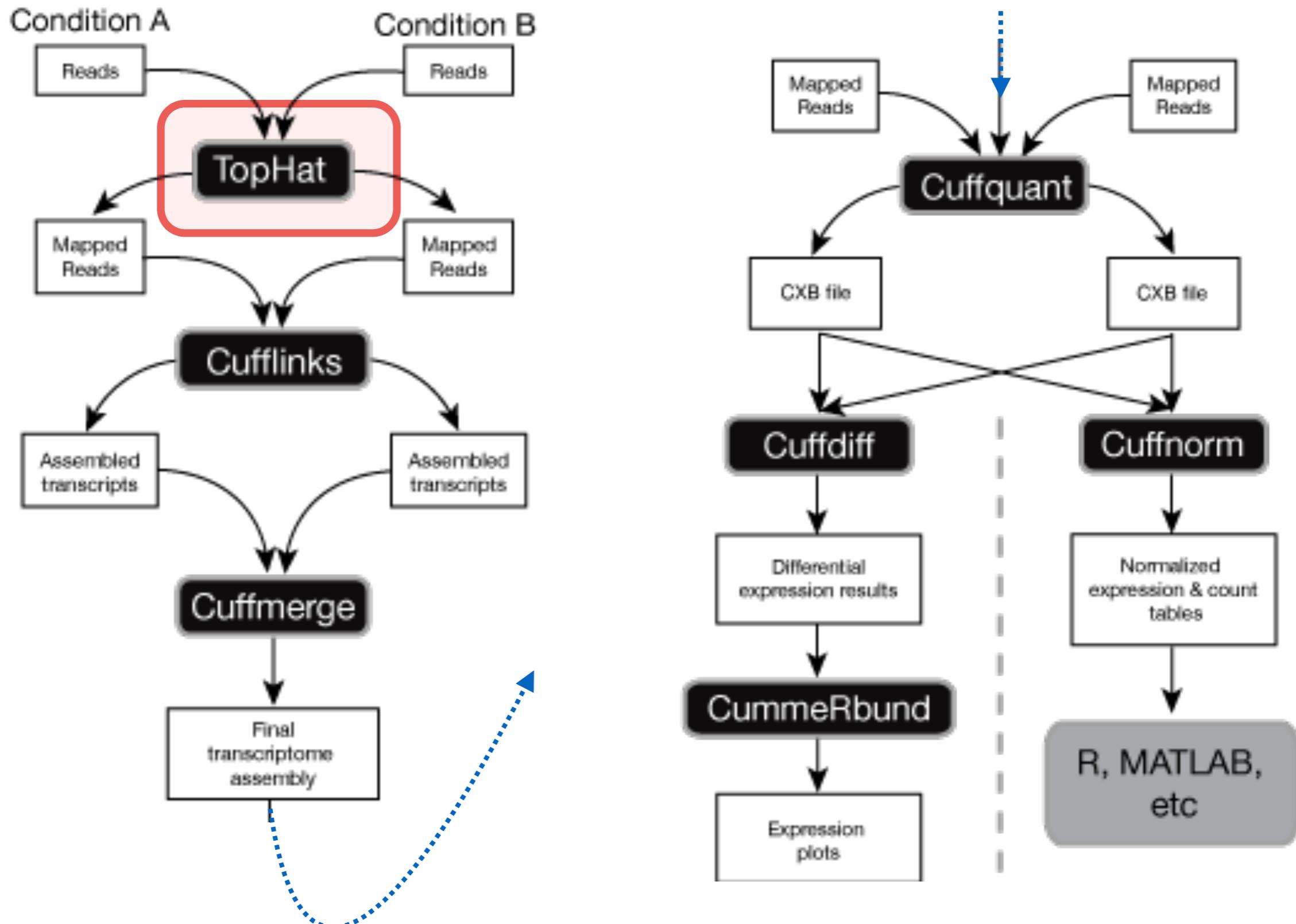
以下のように、bashコマンドに渡すことで実行できます。

```
$ bash ./step_01_bowtie2-build.sh
```

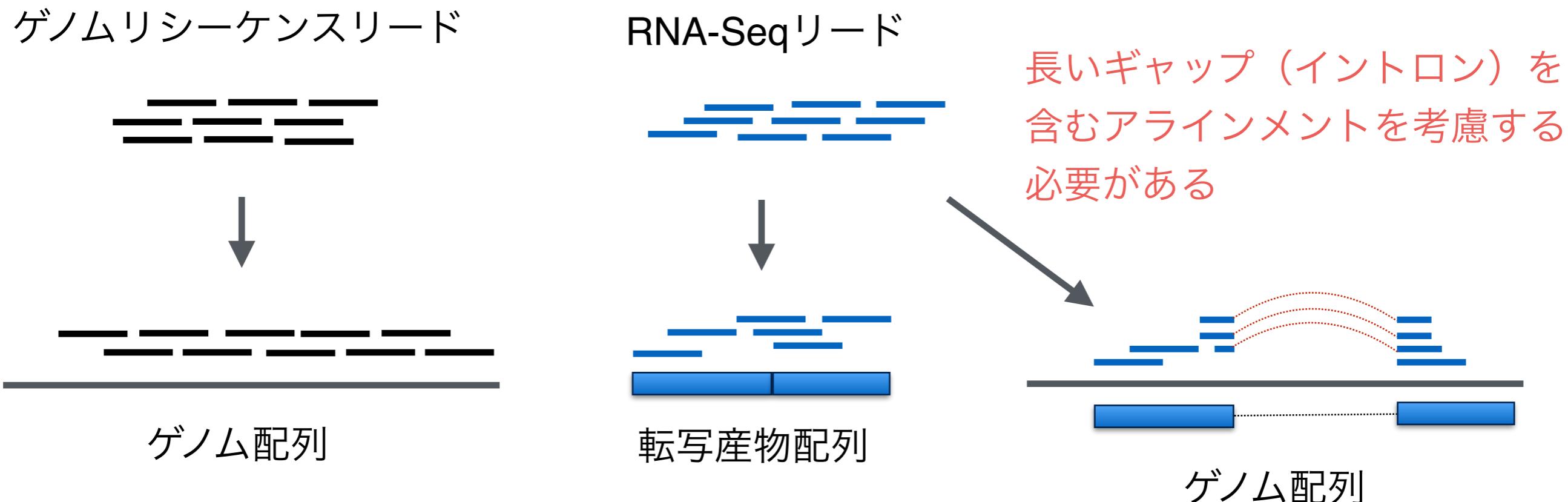
*この後、所々に出てくる「\$TOPHAT_HOME」などの表記はこのようなツールへのパスなどを格納した変数です。

TopHat : RNA-Seqリードのアライメント

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>



Genome-SeqとRNA-Seqリードのアラインメントの違い

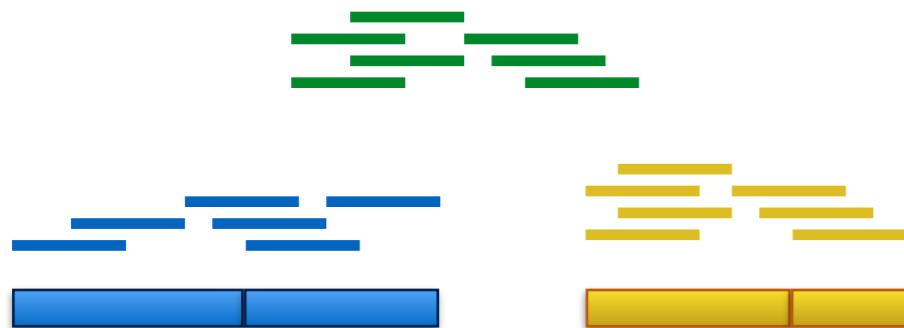


ゲノムリシーケンス解析でよく使われるBWAやBowtie、NovoalignなどはRNA-Seqデータ解析には不向き。

RNA-Seqリードアラインメントに特化した様々なツール（TopHat、STAR、GSNAPなど）が開発されている。

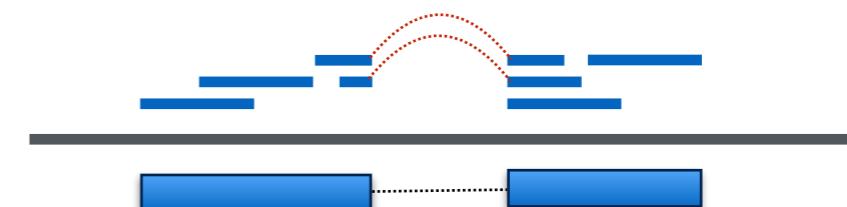
TopHat2によるsplice-awareアラインメント

1. 利用可能であれば既知の転写産物配列にアラインメント

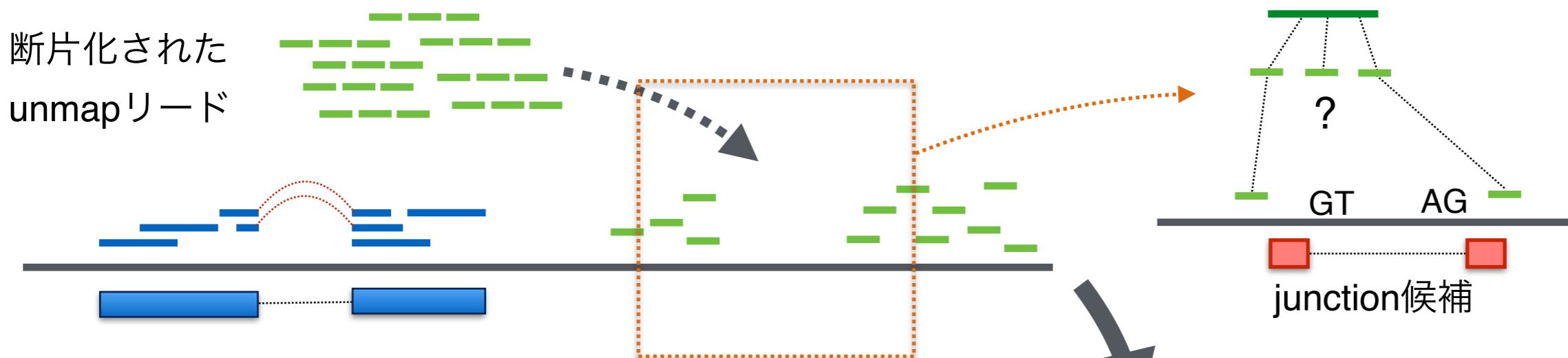


アラインメントには
Bowtie2を利用している

2. 転写産物にアラインメントされたリードをゲノム座標に変換



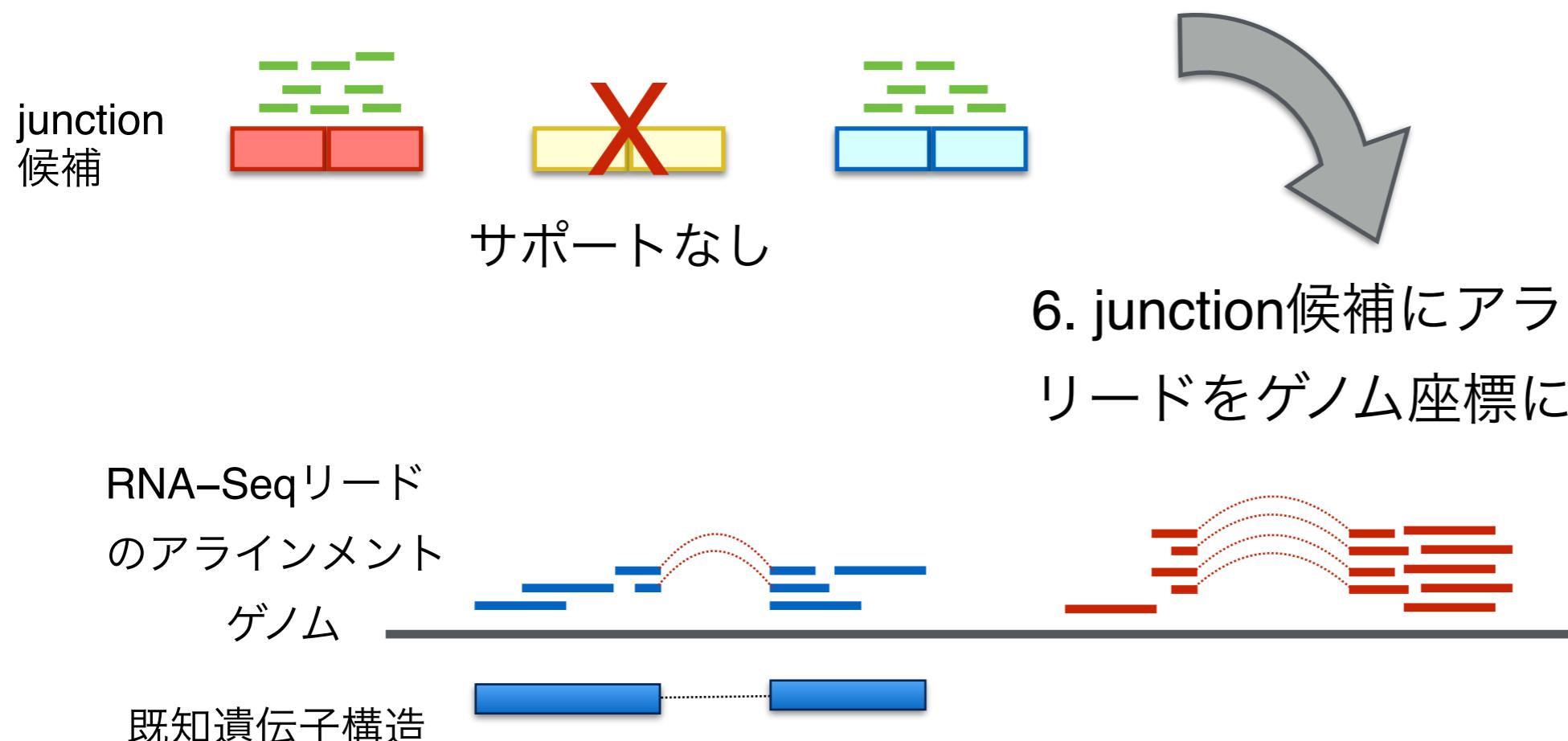
3. 1.でunmapだったリードを断片化し、ゲノム配列にアラインメント



4. ゲノムにアラインメントされたリード情報を元にjunction候補をリストアップ

TopHat2によるsplice-awareアラインメント

5. junction候補の配列ライブラリを作成し、それに対して断片化したunmapリードを再マップし、複数のリードによって支持されるjunctionであるか検証する。

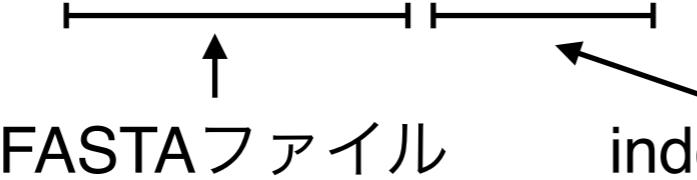


7. 各リードのアラインメント結果を出力 (accepted_hits.bam)
最終的に、既知転写配列に基づくものだけでなく、新規の転写領域や
splice-junction情報も得られる

step_01: bowtie2のインデックス作成

1. bowtie2によるアラインメントに必要なインデックス (.bt2) を作成

```
$ bowtie2-build genome.fa genome
```



FASTAファイル indexのprefix

2. IGVによる可視化などに必要なインデックス (.fai) を作成

```
$ samtools faidx genome.fa
```



FASTAファイル

以下のシェルスクリプトを実行

```
$ bash ./step_01_bowtie2-build.sh (実行時間 約10秒)
```

step_02: TopHat2によるRNA-Seqリードのアラインメント

2条件、2反復の合計4サンプルをそれぞれ別々にTopHat2でアラインメント

```
$ tophat2 --min-intron-length 10 --max-intron-length 10000 --GTF  
annotation.gtf --output-dir TopHat_out_NPB_1400_rep1 genome  
NPB_1400_rep1_r1.fq.gz NPB_1400_rep1_r2.fq.gz  
$ mv TopHat_out_NPB_1400_rep1/accepted_hits.bam TopHat_out_NPB_1400_rep1/  
NPB_1400_rep1.bam  
$ samtools index TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam
```

- ・サンプル名部分を適宜置き換え、サンプルごとに3種類のコマンドを実行する。
- ・1行目：TopHatによるアラインメント。イネ用にintronサイズの下限と上限、既知遺伝子アノテーション情報を指定している。
- ・2行目：TopHatはaccepted_hits.bamという名前でアラインメント結果を出力するため、BAMファイルの名前からでもサンプルが区別できるように「mv」コマンドでファイル名を変更している。
- ・3行目：BAMファイルのインデックス作成。IGV等でアラインメント結果を見る際に必要なインデックスファイルを作成している。

以下のシェルスクリプトを実行

```
$ bash ./step_02_tophat.sh (実行時間 約1分)
```

step_02: TopHat2によるアラインメント結果

```
$ ls TopHat_out_NPB_1400_rep1/
```

NPB_1400_rep1.bam : アラインメント情報 (元はaccepted_hits.bamという名前のファイル。次のCufflinksの入力となる)

NPB_1400_rep1.bam.bai : samtools indexコマンドで作成したbamファイルのインデックス

align_summary.txt: アラインメント結果の統計情報

deletions.bed : アラインメントから得られた欠失変異の情報

insertions.bed : アラインメントから得られた挿入変異の情報

junctions.bed : スプライシングジャンクションの情報

logs/ : 様々な処理のログの入ったディレクトリ

prep_reads.info : 入力したリードの統計情報

unmapped.bam : アンマップリードの情報

以下のコマンドで中身を確認できる。

```
$ less TopHat_out_NPB_1400_rep1/align_summary.txt
```

```
$ samtools view TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam | less
```

BAMファイルの中身は基本的にゲノム解析の時と同じだが、6カラム目のCIGARにある「49M78N51M」のような表記は、リードの前半の49bp (49M) と後半の51bp (51M) の間に78bpのイントロン (78N) が存在するという意味。

step_03: Cufflinksによる転写産物構造予測

TopHatによる4サンプルの独立なアラインメント結果を元に、サンプルごとに遺伝子構造と発現量の推定を行う。

```
$ cufflinks --output-dir Cufflinks_out_NPB_1400_rep1  
--min-intron-length 10 --max-intron-length 10000  
TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam
```

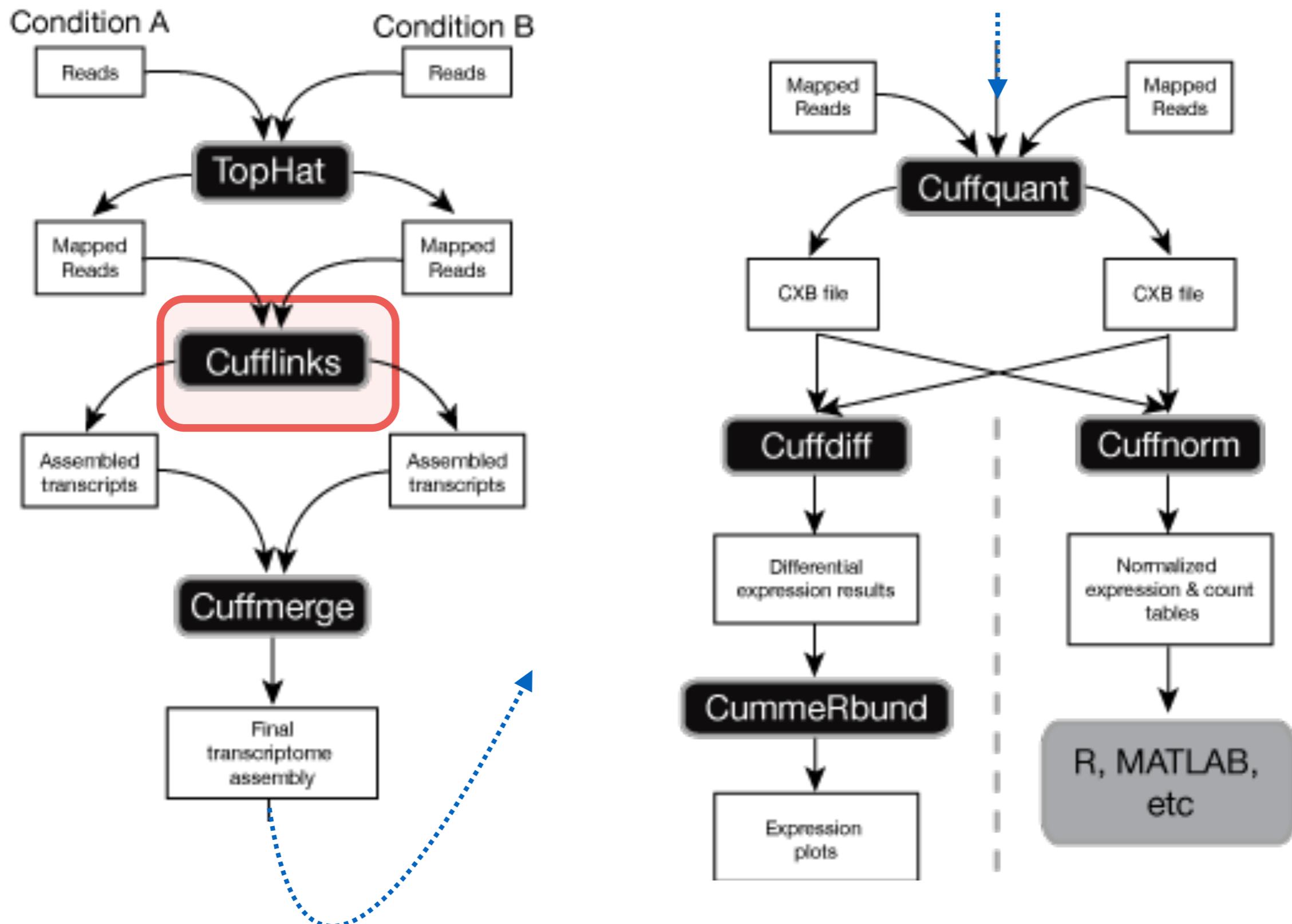
- ・サンプル名部分を適宜置き換え、サンプルごとにcufflinksを実行する。
- ・TopHatと同様にintronサイズの下限と上限を指定している。他にも--GTF-guideなど遺伝子構造構築のパラメータや、--frag-bias-correctなど遺伝子発現量推定のパラメータ等が色々ある。例えば、--GTFオプションで既知遺伝子アノテーションを与えると、新規転写産物の予測はせずに、既知遺伝子のみについて発現量の定量をおこなう。
- ・今回は新規の遺伝子構造予測をおこなうため、--GTFオプションは指定していない。

以下のシェルスクリプトを実行

```
$ bash ./step_03_cufflinks.sh      (実行時間 約30秒)
```

Cufflinksによる転写産物構造予測

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>



Cufflinksの結果の確認

- *.fpkm_tracking : 遺伝子やisoformごとの発現情報
- transcripts.gtf : 遺伝子構造、位置、および発現情報 (GTF形式)

以下のようにlessコマンドで結果ファイルを確認

```
$ less Cufflinks_out_NPB_1400_rep1/transcripts.gtf
chr06    Cufflinks      transcript      9246969 9248112 1000      +      .      gene_id "CUFF.1"; transcript_id "CUFF.1.1"; FPKM "11458.0854217572"; frac "1.000000"; conf_lo "8898.937394"; conf_hi "14017.233449"; cov "17.732623";
chr06    Cufflinks      exon       9246969 9247309 1000      +      .      gene_id "CUFF.1"; transcript_id "CUFF.1.1"; exon_number "1"; FPKM "11458.0854217572"; frac "1.000000"; conf_lo "8898.937394"; conf_hi "14017.233449"; cov "17.732623";
chr06    Cufflinks      exon       9247388 9248112 1000      +      .      gene_id "CUFF.1"; transcript_id "CUFF.1.1"; exon_number "2"; FPKM "11458.0854217572"; frac "1.000000"; conf_lo "8898.937394"; conf_hi "14017.233449"; cov "17.732623";
chr06    Cufflinks      transcript      9257043 9257160 1000      -      .      gene_id "CUFF.2"; transcript_id "CUFF.2.1"; FPKM "260393.7454311026"; frac "1.000000"; conf_lo "92260.782572"; conf_hi "428526.708291"; cov "398.152921";
chr06    Cufflinks      exon       9257043 9257160 1000      -      .      gene_id "CUFF.2"; transcript_id "CUFF.2.1"; exon_number "1"; FPKM "260393.7454311026"; frac "1.000000"; conf_lo "92260.782572"; conf_hi "428526.708291"; cov "398.152921";
```

*サンプルごとに独立に遺伝子構造と発現量が得られるため、遺伝子構造を統合し、サンプル間で比較可能にする必要がある。

遺伝子発現量の指標 (RPKM/FPKM)

シーケンス量や遺伝子の長さで正規化した発現量指標

RPKMやFPKM=Read (Fragment) 数／遺伝子の長さ(kb)／総リード数(M)

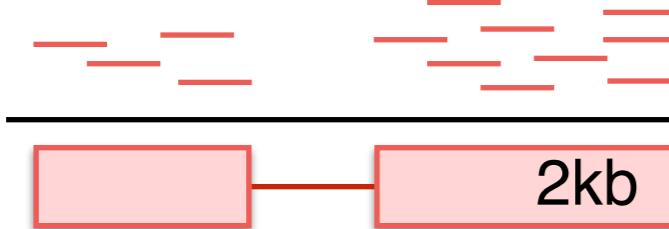
RPKM: reads per kilobase of exon model per million mapped reads

FPKM: fragments per kilobase of transcript per million fragments mapped

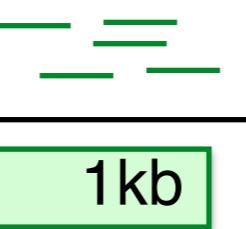
条件 1

リードカウント (≠発現量)

13リード

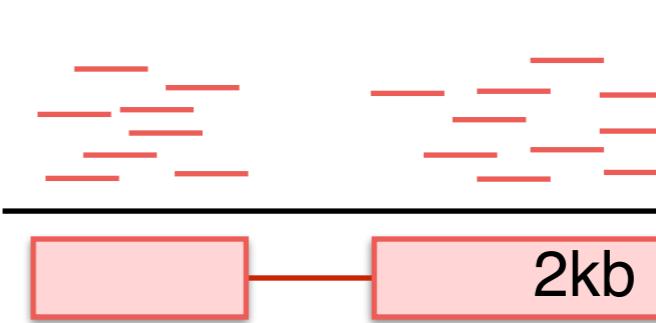


6リード

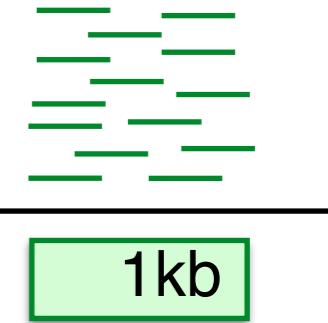


条件 2

18リード



14リード



RPKM: $13/2/19=0.34$

6/1/19=0.32

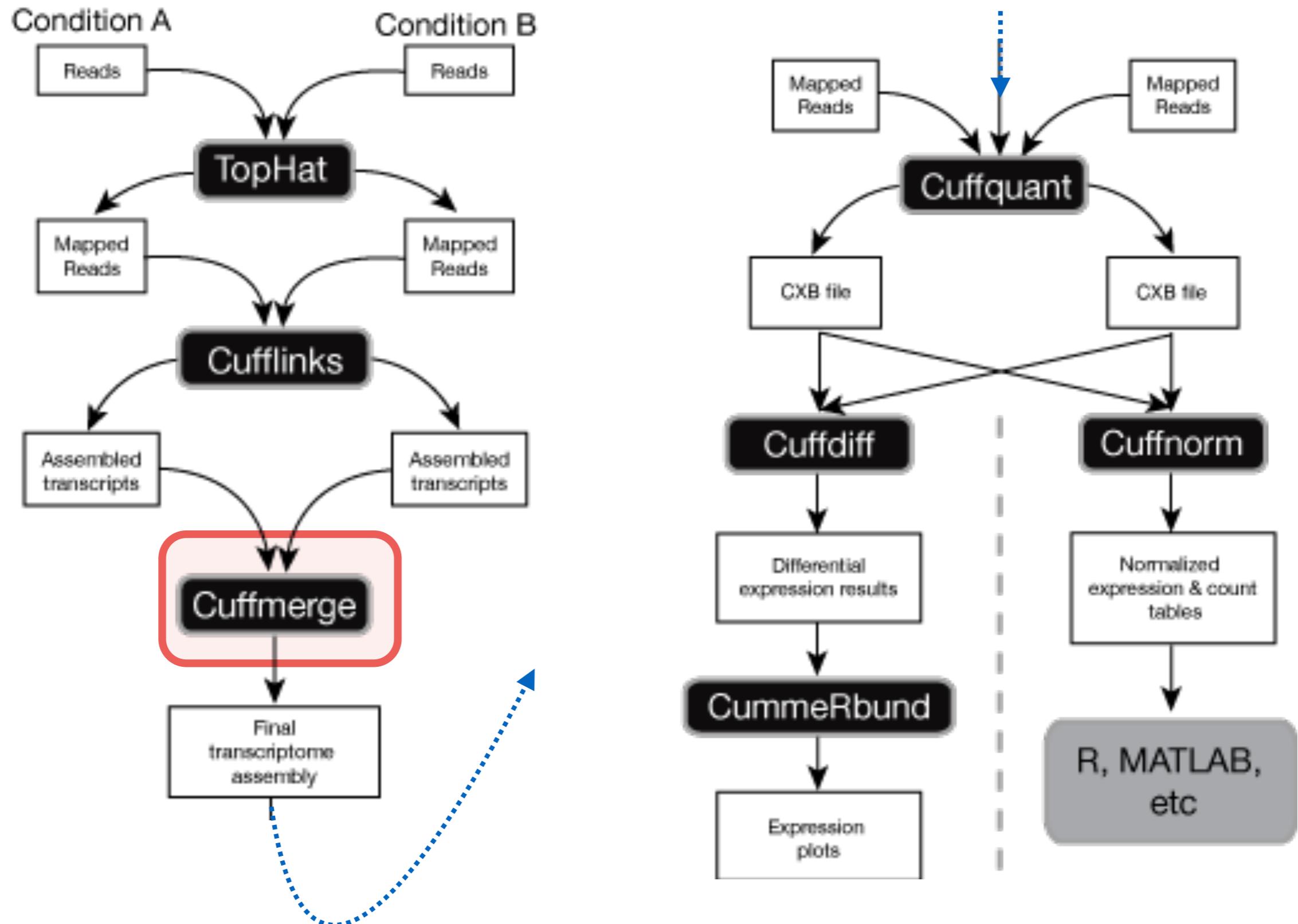
$18/2/32=0.28$

$14/1/32=0.44$

全リード中にある遺伝子由来のリードがどれくらいの割合存在するかという相対値であり、高発現遺伝子の存在やその変動の影響を強く受ける。TMM正規化など様々な手法が提唱されている。

Cuffmergeによる全サンプルの遺伝子モデルの統合

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>



step_04: Cuffmergeによる全サンプルの遺伝子モデルの統合

サンプルごとに得られた遺伝子モデルを1つの遺伝子モデルに統合する。既知の遺伝子セットも合わせて、**RNA-Seq**遺伝子と対応付ける。

1. 4サンプル分の遺伝子モデルファイル (transcripts.gtf) のリスト (assemblies.txt) を作成する

```
$ ls Cufflinks_out_*/transcripts.gtf > assemblies.txt
```

2. 複数の遺伝子モデル (transcripts.gtf) と既知の遺伝子アノテーションを合わせて、統一遺伝子モデルセットを構築する

```
$ cuffmerge --output-dir Cuffmerge_out --ref-sequence genome.fa  
--ref-gtf annotation.gtf assemblies.txt
```

- --output-dirで指定したディレクトリ中に統一遺伝子モデルセット (merged.gtf) が出力される。

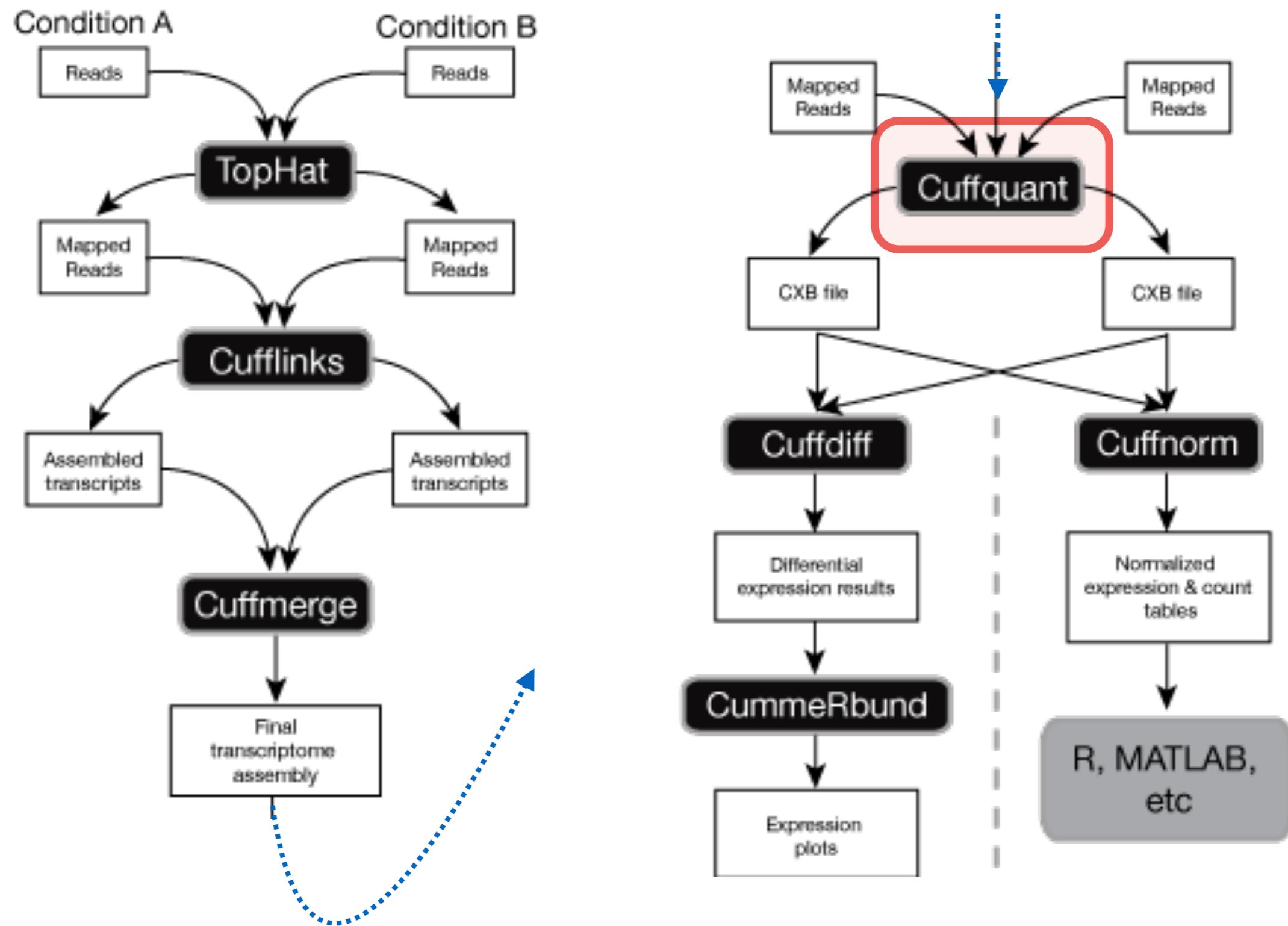
以下のシェルスクリプトを実行

```
$ bash ./step_04_cuffmerge.sh
```

(実行時間 約5秒)

Cuffquantによる遺伝子発現量の定量

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>



step_05: Cuffquantによる遺伝子発現量の定量

統合した遺伝子モデルと各サンプルのアラインメント情報を元に、各遺伝子の発現量を定量する。

```
$ cuffquant --output-dir Cuffquant_out_NPB_1400_rep1  
Cuffmerge_out/merged.gtf TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam
```

- ・ サンプル名部分を適宜置き換え、サンプルごとにcuffquantを実行する。
- ・ 発現量補正 (--frag-bias-correct) や解析対象から外す領域 (--mask-file) を指定するオプション等が用意されているが、今回は遺伝子アノテーション (merged.gtf) とアラインメントファイル (bam) のみ指定している。
- ・ 発現量データは--output-dirで指定したディレクトリ中にabundances.cxbという名前で出力される。ただし、このファイルはバイナリ形式のため中身を見ることは出来ない。

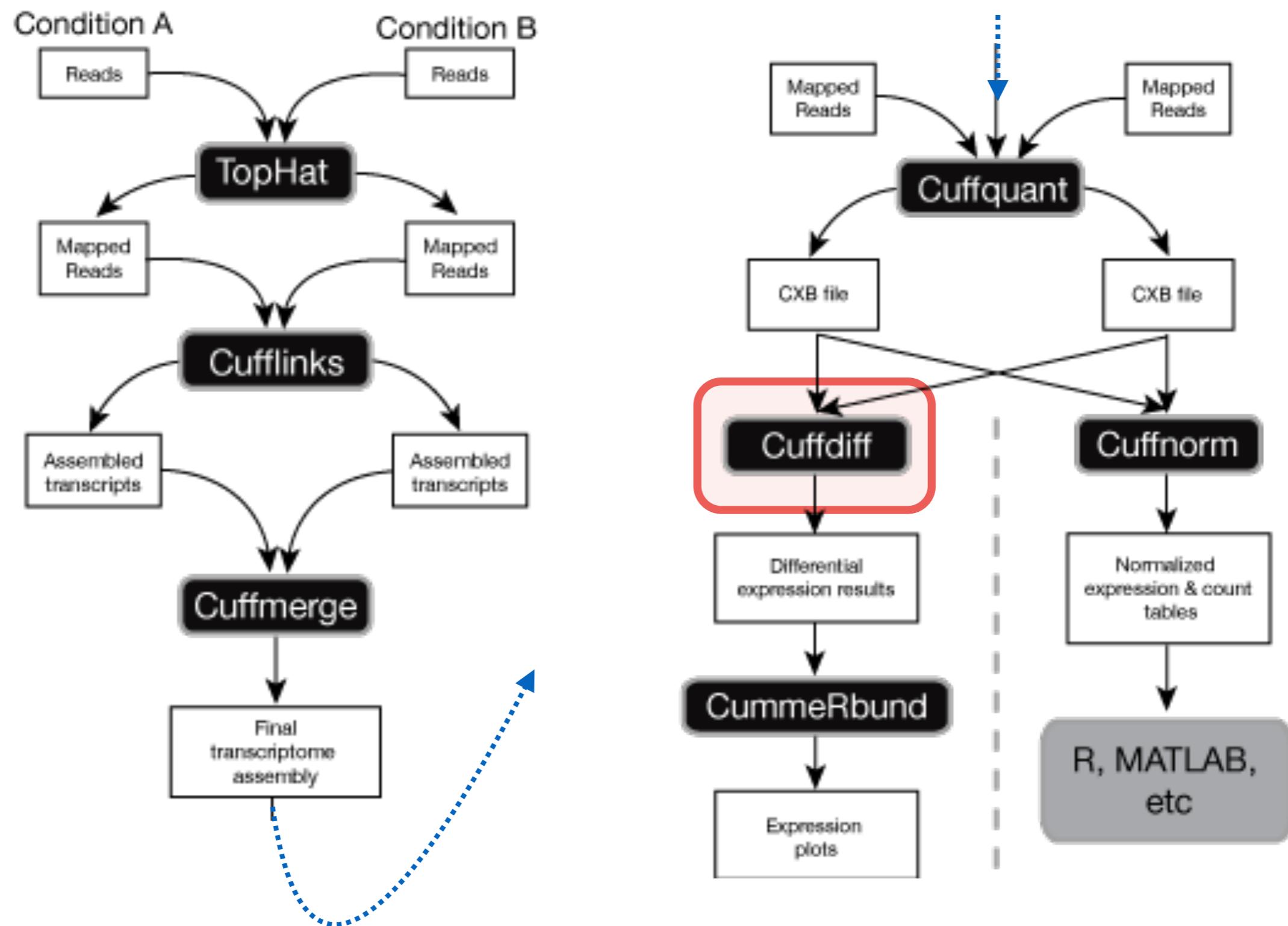
以下のシェルスクリプトを実行

```
$ bash ./step_05_cuffquant.sh
```

(実行時間 約10秒)

Cuffdiffによる遺伝子発現量比の検定

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>



step_06: Cuffdiffによる発現変動する遺伝子の検出

遺伝子構造(.gtf)と各サンプルのアラインメント情報(.bam)を元に、
サンプル間での遺伝子発現量変化の有意差を検定する。

```
$ cuffdiff --output-dir Cuffdiff_out --labels NPB_1400,NPB_0200  
Cuffmerge_out/merged.gtf  
Cuffquant_out_NPB_1400_rep1/abundances.cxb,Cuffquant_out_NPB_1400_rep2/  
abundances.cxb  
Cuffquant_out_NPB_0200_rep1/abundances.cxb,Cuffquant_out_NPB_0200_rep2/  
abundances.cxb
```

- ・ 時系列サンプル、正規化手法、バイアス補正など様々なオプションが用意されているが、
今回はサンプルのラベルのみを指定。
- ・ 反復サンプルは**カンマ区切り**で並べ、比較するサンプルの間は**スペース**を入れる。

以下のシェルスクリプトを実行

```
$ bash ./step_06_cuffdiff.sh (実行時間 約3分)
```

Cuffdiffの結果の確認

genes, isoforms, cds, tssなど様々な属性ごとに発現量 (*.fpkm_tracking) や条件間の遺伝子発現量比の有意差についての情報 (*.exp.diff) などが出力される。

```
$ ls Cuffdiff_out/  
$ less Cuffdiff_out/gene_exp.diff
```

Cuffmergeが付ける遺伝子ID。
XLOC IDがどの既知遺伝子に対応するかは、merged.gtfを参照する必要がある。

ちなみに、merged.gtfを見ると
.....gene_id "XLOC_000009"; old “Os06t0275000-01”
となっているので、この遺伝子は既知のSE1 (Hd1)であることが分かる。

XLOC_000009	XLOC_000009	SE1	chr06:9336358-9338634
NPB_1400	NPB_0200	OK	1846.27 37074.9 4.32776 14.861
5e-05	0.000105	yes	

左がFDRのq-value、右が5%水準で有意差があるかどうかの判定

左から14時のFPKM、2時のFPKM、Fold-change(log2)

発現解析の結果の可視化

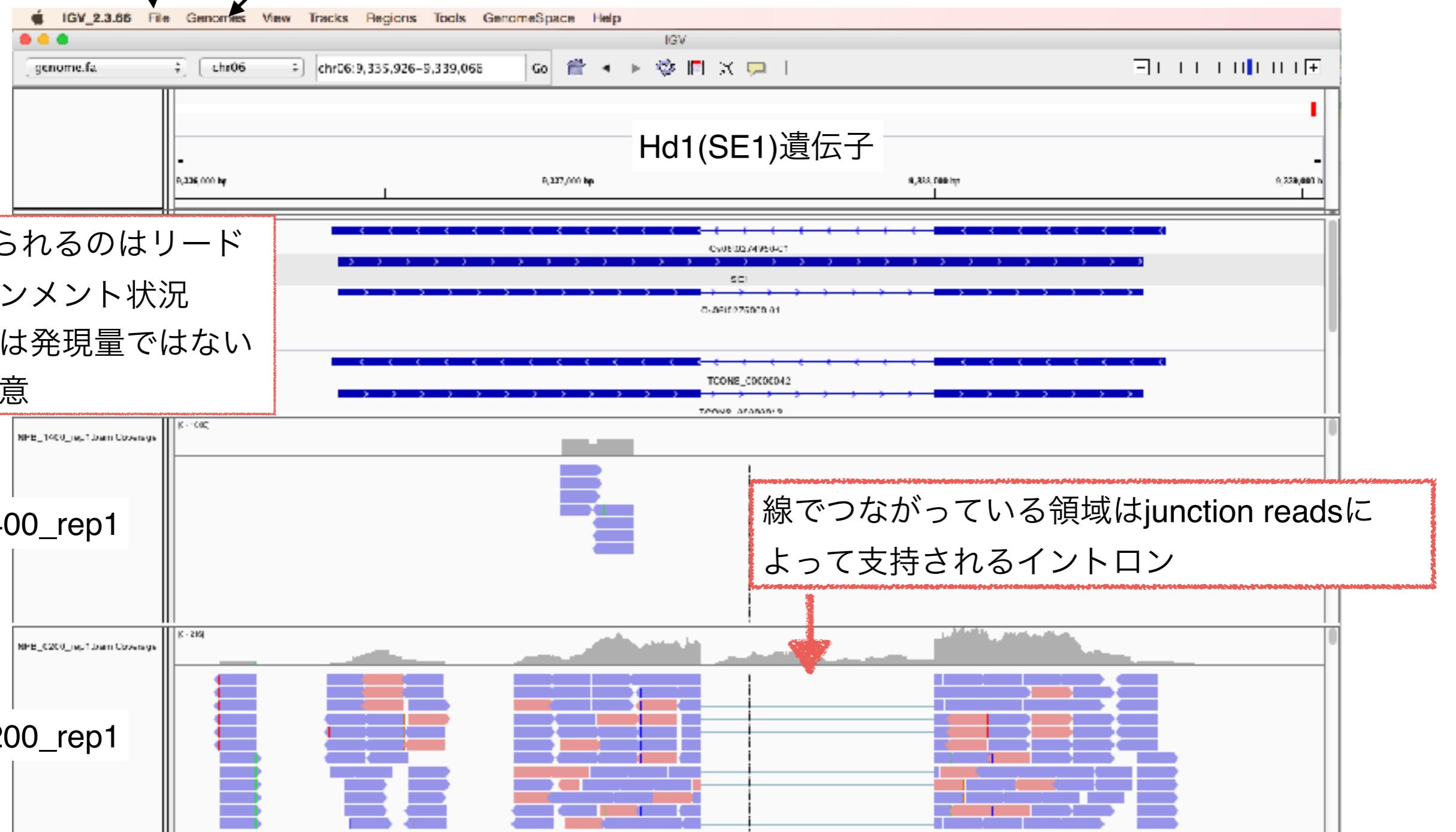
IGVによるアラインメントや転写産物構造の可視化

デスクトップ上の「workshop/ref_alignment」中のリファレンスゲノムとアノテーション、TopHatの結果のBAMファイル、Cuffmergeの結果のmerged.gtfなどが読み込み可能。

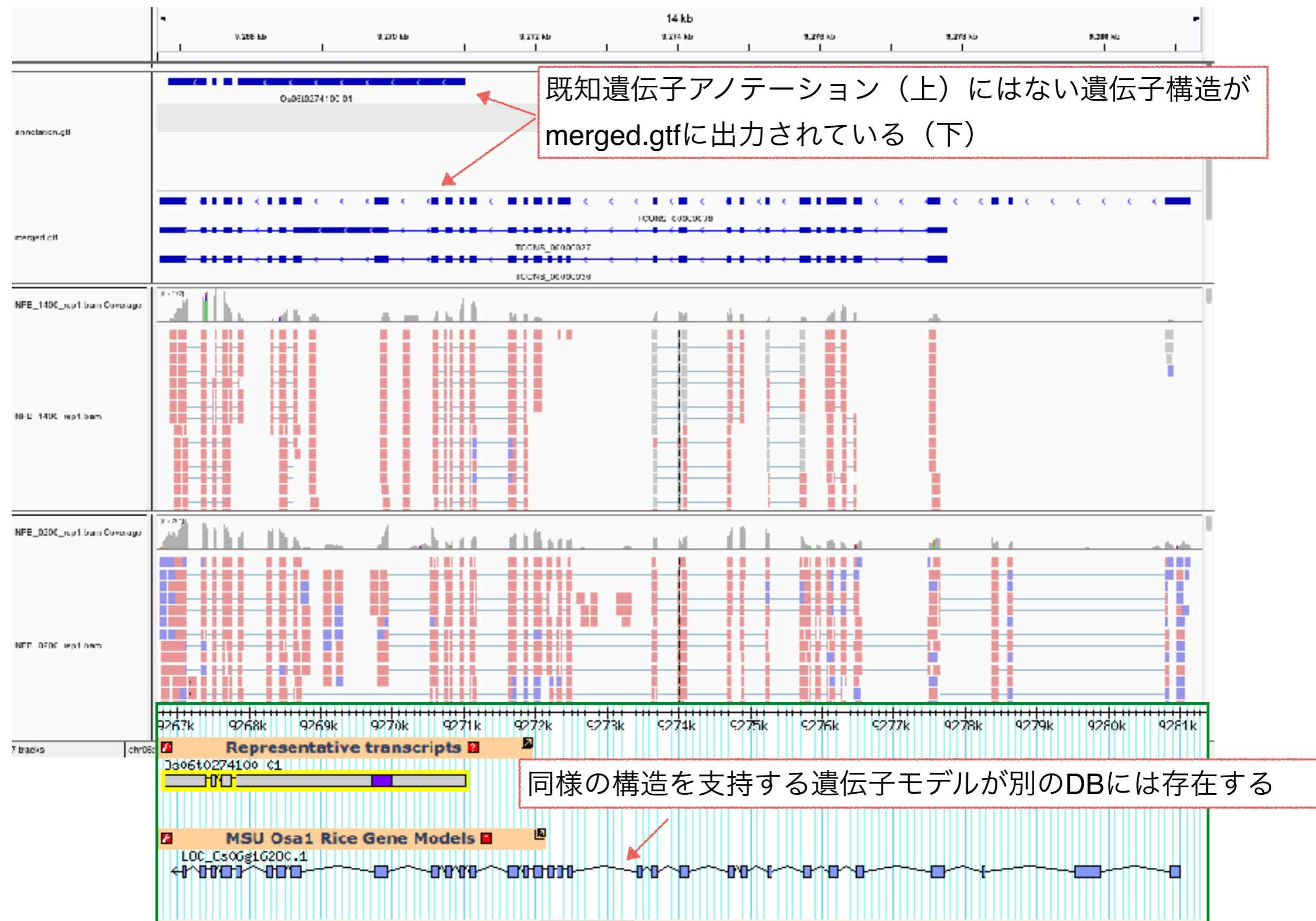
2. 「Load from File」からannotation.gtfやBAMファイルを読み込む



1. 「Load Genome from File」からgenom.faを読み込む

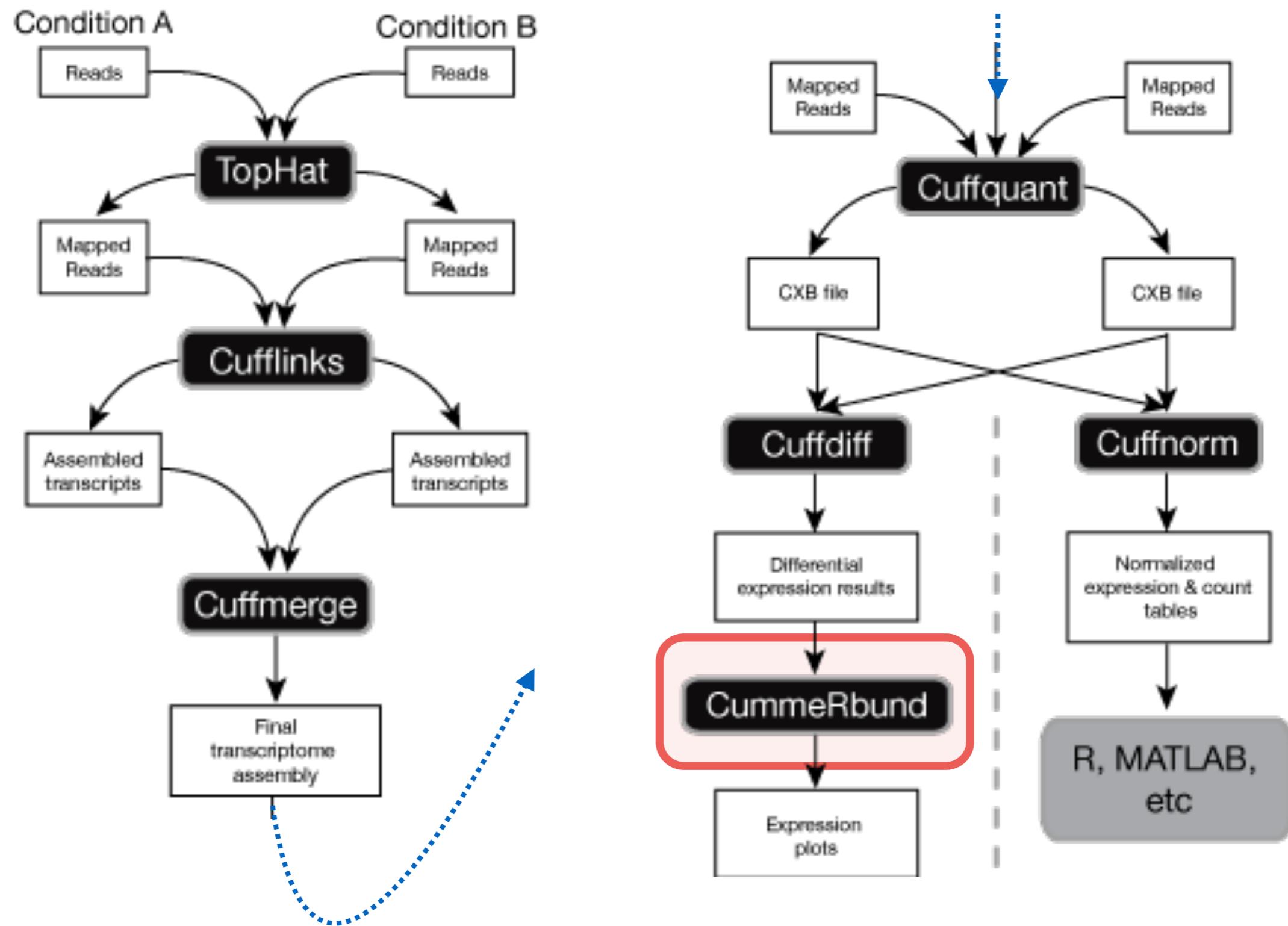


IGVによるアラインメントや転写産物構造の可視化



Tuxedo suite toolsのCummeRbundによる可視化

Modified from <http://cole-trapnell-lab.github.io/cufflinks/manual/>



CummeRbund

The screenshot shows the Bioconductor package page for 'cummeRbund'. At the top, there's a navigation bar with links for Home, Install, Help, Developers, and About. A search bar is also present. Below the navigation, a breadcrumb trail indicates the current location: Home > Bioconductor 3.2 > Software Packages > cummeRbund. The main content area features the package name 'cummeRbund' in green, followed by a brief description: 'Analysis, exploration, manipulation, and visualization of Cufflinks high-throughput sequencing data.' Below this, there are sections for Bioconductor version (3.2), package details (allows persistent storage, access, exploration, and manipulation of Cufflinks high-throughput sequencing data), author information (L. Goff, C. Trapnell, D. Kelley), maintainer (Loyal A. Goff), and citation. A large callout box highlights 'CummeRbund(Bioconductor)' and provides a link to its release page: <http://www.bioconductor.org/packages/release/bioc/html/cummeRbund.html>.

The screenshot shows the R Project for Statistical Computing homepage. It features the R logo and a navigation menu with links for Home, Download, CRAN, R Project, About R, Contributors, What's New?, Mailing Lists, Bug Tracking, Conferences, and Search. The main content area includes sections for 'Getting Started' (describing R as a free software environment for statistical computing and graphics) and 'News' (mentioning R version 3.2.3 (Wooden Christmas-Tree) prerelease versions and R version 3.2.2 (Fire Safety)). There's also a note about the R Journal (Volume 7(2) is available). At the bottom, it says 'R' and provides a link to the R Project website: <https://www.r-project.org/>.

CummeRbundは**R/Bioconductor**のパッケージの一つであり、cuffdiffの結果を読み込んで様々な情報を可視化する手法を提供している。利用するためには統計計算とグラフィックスのための言語・環境である「**R**」をインストールする必要がある。

RStudioはRの統合開発環境であり、使いやすいGUIを備えておりとても便利。

The screenshot shows the RStudio homepage. It features a large blue header with the RStudio logo and a search bar. Below the header, a welcome message reads 'Welcome to RStudio - Open source and enterprise-ready professional software for R'. There are three main calls-to-action: 'Download RStudio', 'Discover Shiny', and 'shinyapps.io Login'. To the right is a large blue circular icon with a white 'R'. Below the header, there are three columns of features: 'Powerful IDE for R' (RStudio IDE is a powerful and productive user interface for R. It has auto-completion, code highlighting, and refactoring), 'R Packages' (Our developer and open source maintainers of several popular R packages, including ggplot2, plyr, dplyr, dada, and dada2), and 'Bring R to the web' (Shiny is an elegant and powerful web framework for building interactive reports and web applications using R without front-end web development skills). At the bottom, it says 'RStudio' and provides a link to the RStudio website: <https://www.rstudio.com/>.

実習用のCuffdiff結果の準備と確認

- デスクトップ上の「workshop」ディレクトリ中にある「Cuffdiff_for_cummeRbund」が可視化の対象となるCuffdiffの出力ディレクトリ
- いもち病菌の感染前後の葉のmRNA-Seqデータ（1番染色体のみ）
- Cuffdiffによる、感染前と後の2条件（各3反復）の2群間比較の結果

 bias_params.info	53 バイト
 cds_exp.diff	670 KB
 cds.count_tracking	291 KB
 cds.diff	528 KB
 cds.fpkm_tracking	655 KB
 cds.read_group_tracking	1.5 MB
 gene_exp.diff	683 KB
 genes.count_tracking	312 KB
 genes.fpkm_tracking	671 KB
 genes.read_group_tracking	1.6 MB
 isoform_exp.diff	817 KB
 isoforms.count_tracking	394 KB
 isoforms.fpkm_tracking	907 KB
 isoforms.read_group_tracking	2.1 MB
 promoters.diff	566 KB
 read_groups.info	549 バイト
 run.info	412 バイト
 splicing.diff	584 KB
 tss_group_exp.diff	714 KB
 tss_groups.count_tracking	316 KB
 tss_groups.fpkm_tracking	697 KB
tss_groups.read_group_tracking	1.6 MB
var_model.info	614 KB

CummeRbundによる可視化の準備

1. 以下の3つのコマンドはRでcummeRbundを使うための準備

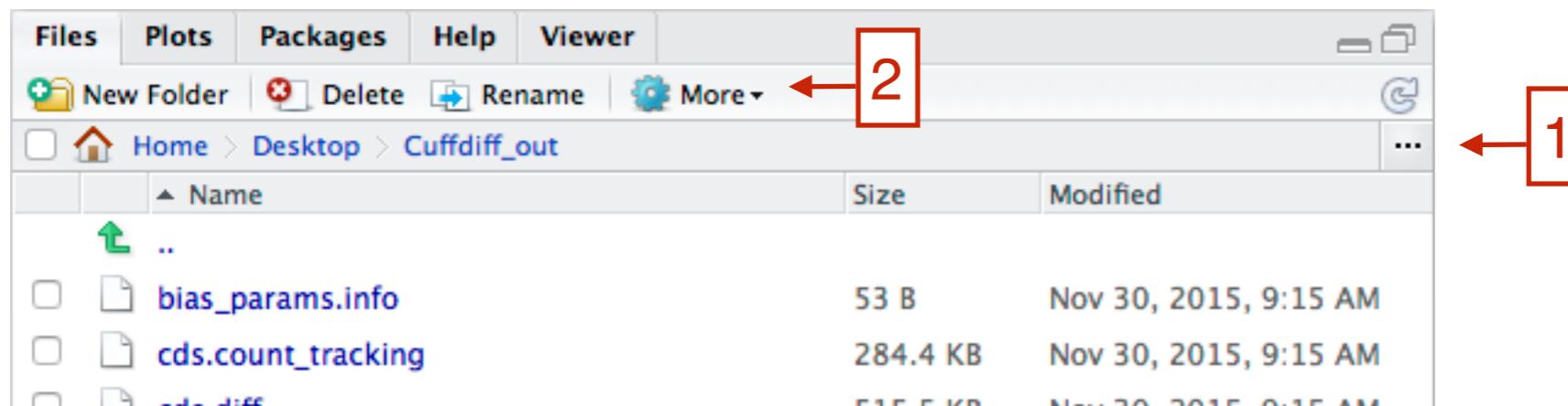
```
> source("http://bioconductor.org/biocLite.R")  
> biocLite("cummeRbund")  
> library(cummeRbund)
```

← 今回はパッケージのインストールと動作
← 確認済みのため最初の2つはスキップ

- cummeRbundはBioconductorのパッケージの1つとして提供されている。
- 途中、「Update all/some/none? [a/s/n]:」と聞かれたら、「a」と入力しリターンを押し、全てアップデートする。
- cummeRbundで必要な様々なパッケージがロードされる。

2. 作業ディレクトリの指定

```
> setwd("c:/Users/user/Desktop/workshop/Cuffdiff_for_cummeRbund/")
```



RStudioでの作業ディレクトリの指定

1. 「...」をクリックし、「Cuffdiff_for_cummeRbund」ディレクトリを選択
2. 「More」をクリックし、「Set As Working Directory」を実行

CummeRbundによる可視化の準備

3. cuffdiffデータの読み込みと整形により、様々なプロットが可能になる。

```
> cuff.data <- readCufflinks(rebuild=T)
Creating database /Users/ykawahara/Desktop/Cuffdiff_out/cuffData.db
Reading Run Info File /Users/ykawahara/Desktop/Cuffdiff_out/run.info
Writing runInfo Table
Reading Read Group Info /Users/ykawahara/Desktop/Cuffdiff_out/
read_groups.info
Writing replicates Table
Reading Var Model Info /Users/ykawahara/Desktop/Cuffdiff_out/var_model.inf
Writing varModel Table
Reading /Users/ykawahara/Desktop/Cuffdiff_out/genes.fpkm_tracking
Checking samples table...
Populating samples table...
Writing genes table
Reshaping geneData table
Recasting
Writing geneData table
...skip...
Indexing Tables...
```

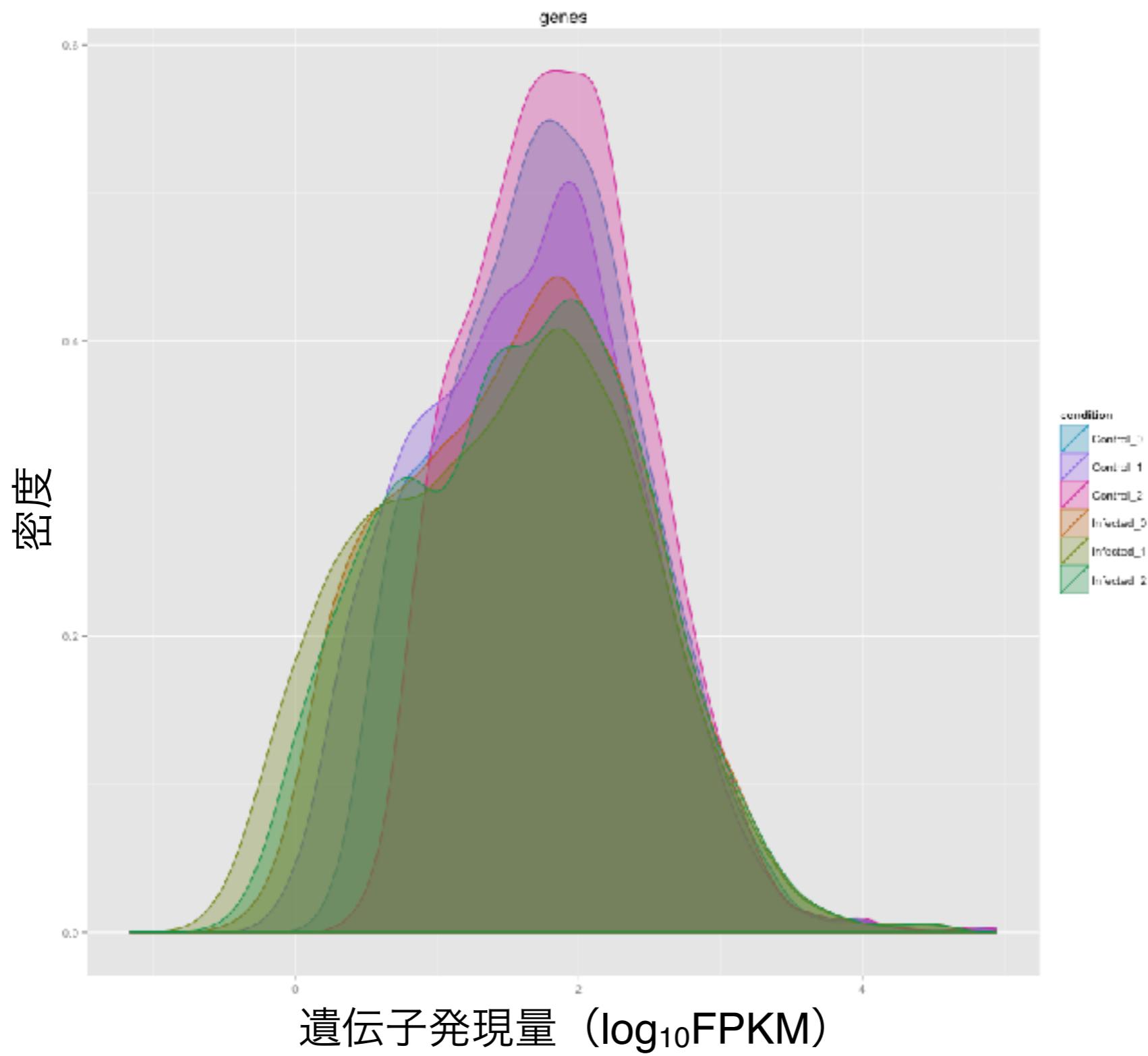
- Cuffdiffのデータ中（特に遺伝子アノテーション情報）にcummeRbundがうまく処理できない記号などがあると読み込みエラーとなる。解析ツールが読み込み可能な形式でアノテーション情報（GTF）などを準備するのが重要。

解析データ中のサンプル（実験条件）数、遺伝子数、isoform数

```
> cuff.data
CuffSet instance with:
  2 samples
  5273 genes
  6260 isoforms
  5795 TSS
  5499 CDS
  5273 promoters
  5795 splicing
  4886 relCDS
```

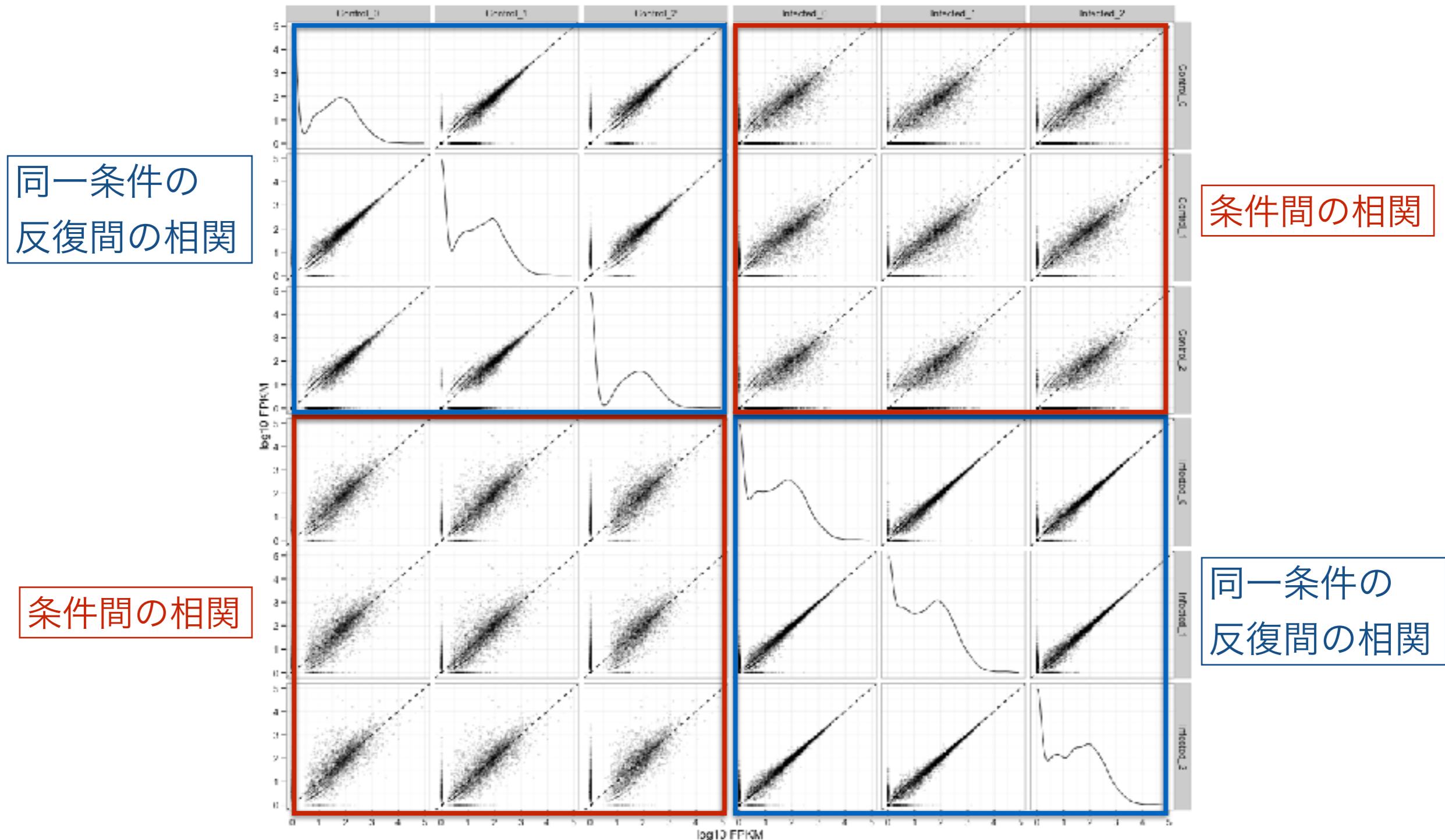
サンプルごとの遺伝子発現量の分布を調べる

```
> csDensity(genes(cuff.data),replicates=T)
```



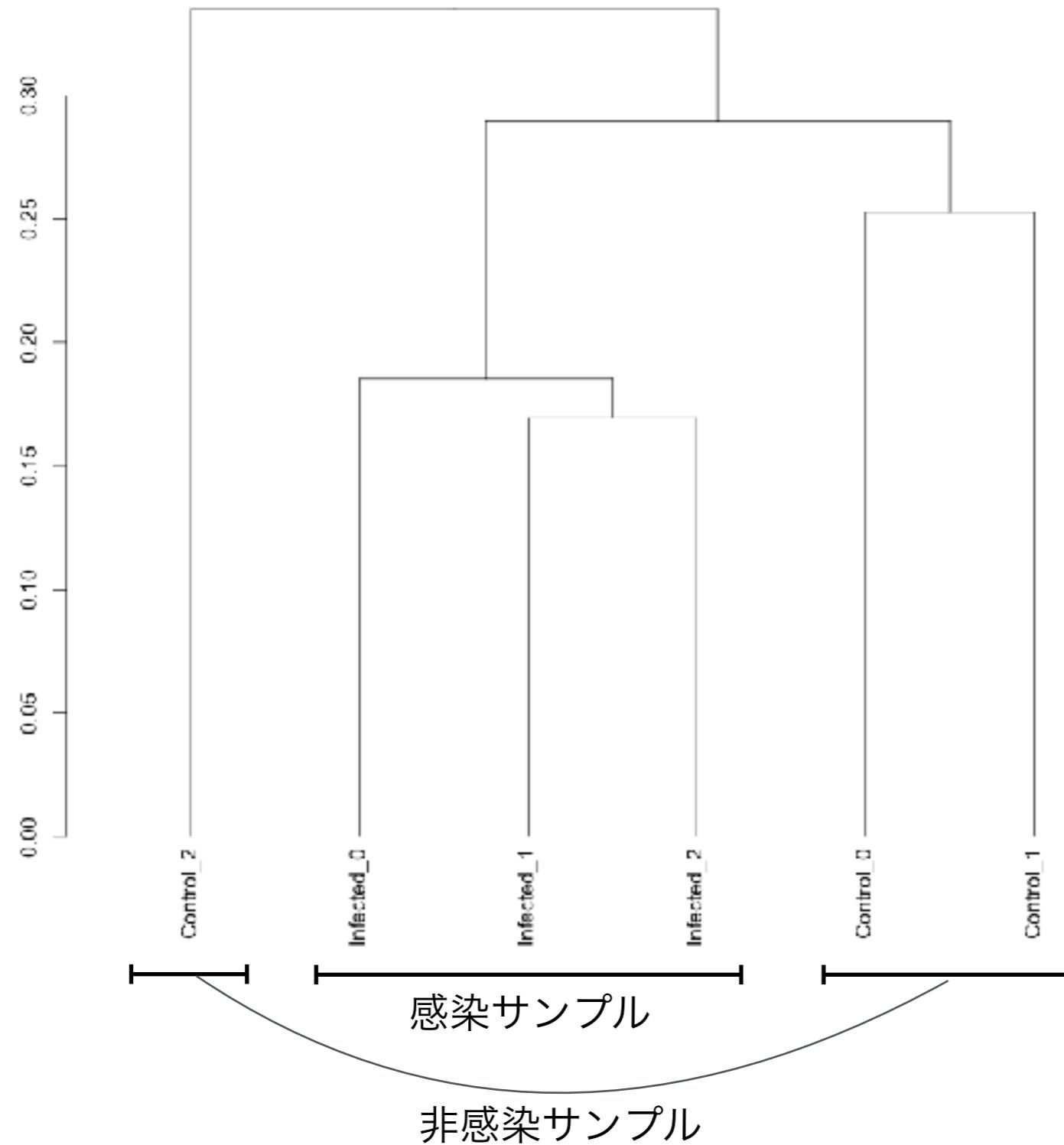
遺伝子発現量の分布とサンプル間の相関を合わせて調べる

```
> csScatterMatrix(genes(cuff.data), replicates=T)
```



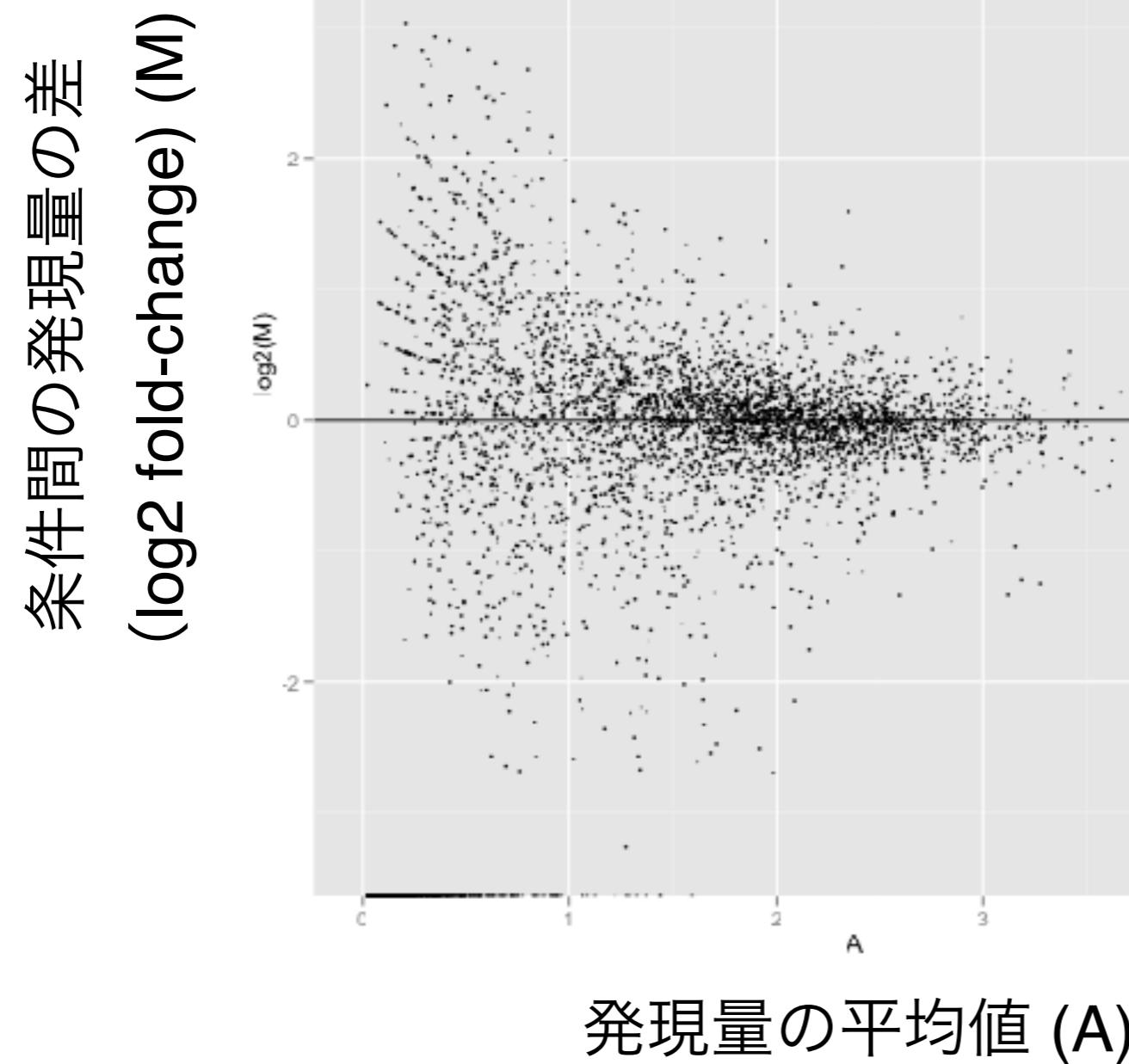
サンプル間のトランскriプトームをクラスタリングし、類似度を調べる

```
> csDendro(genes(cuff.data), replicates=T)
```



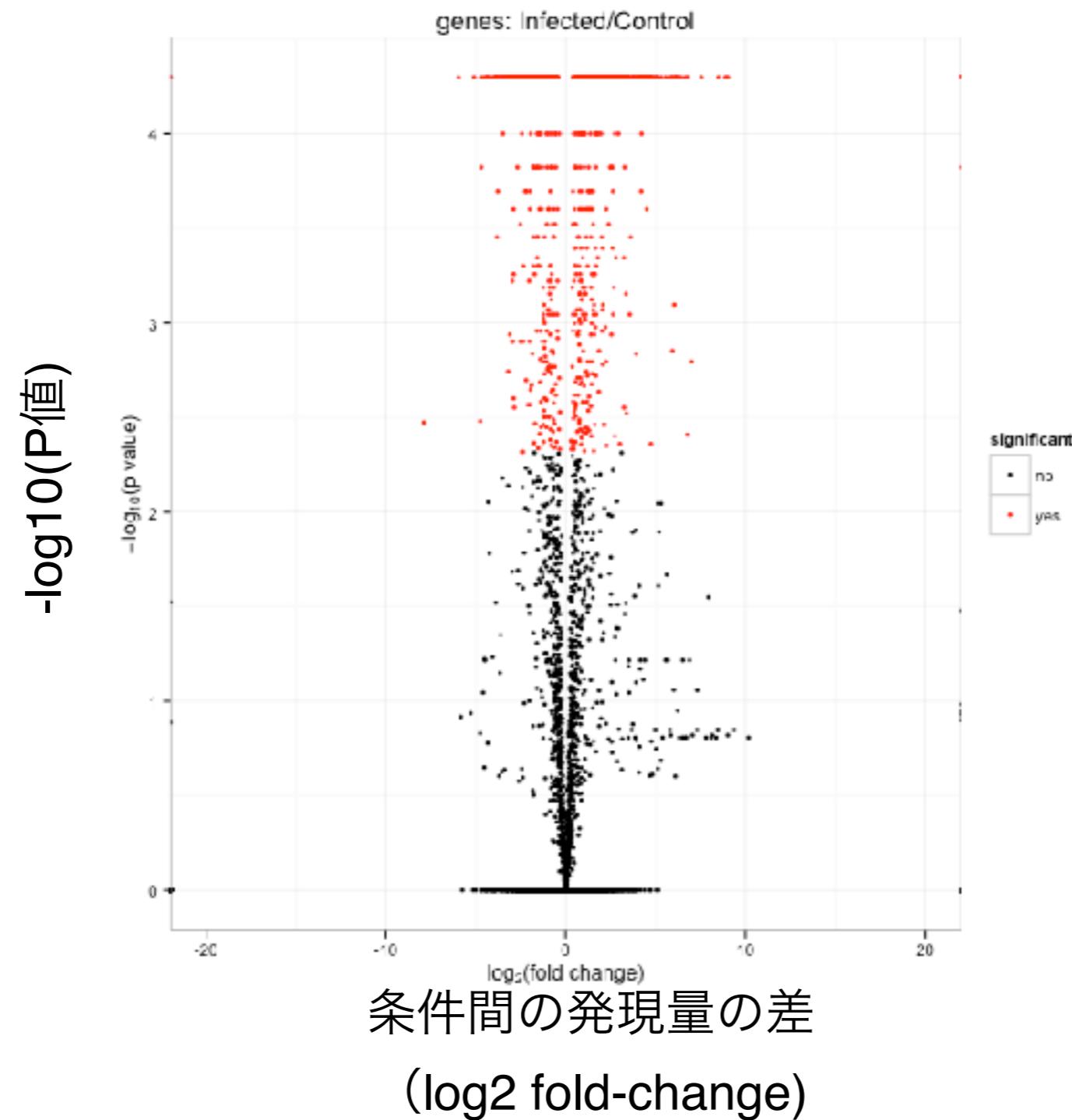
MA-plotで遺伝子発現量とサンプル間の発現変動を合わせて調べる

```
> MAplot(genes(cuff.data), "Control", "Infected")
```



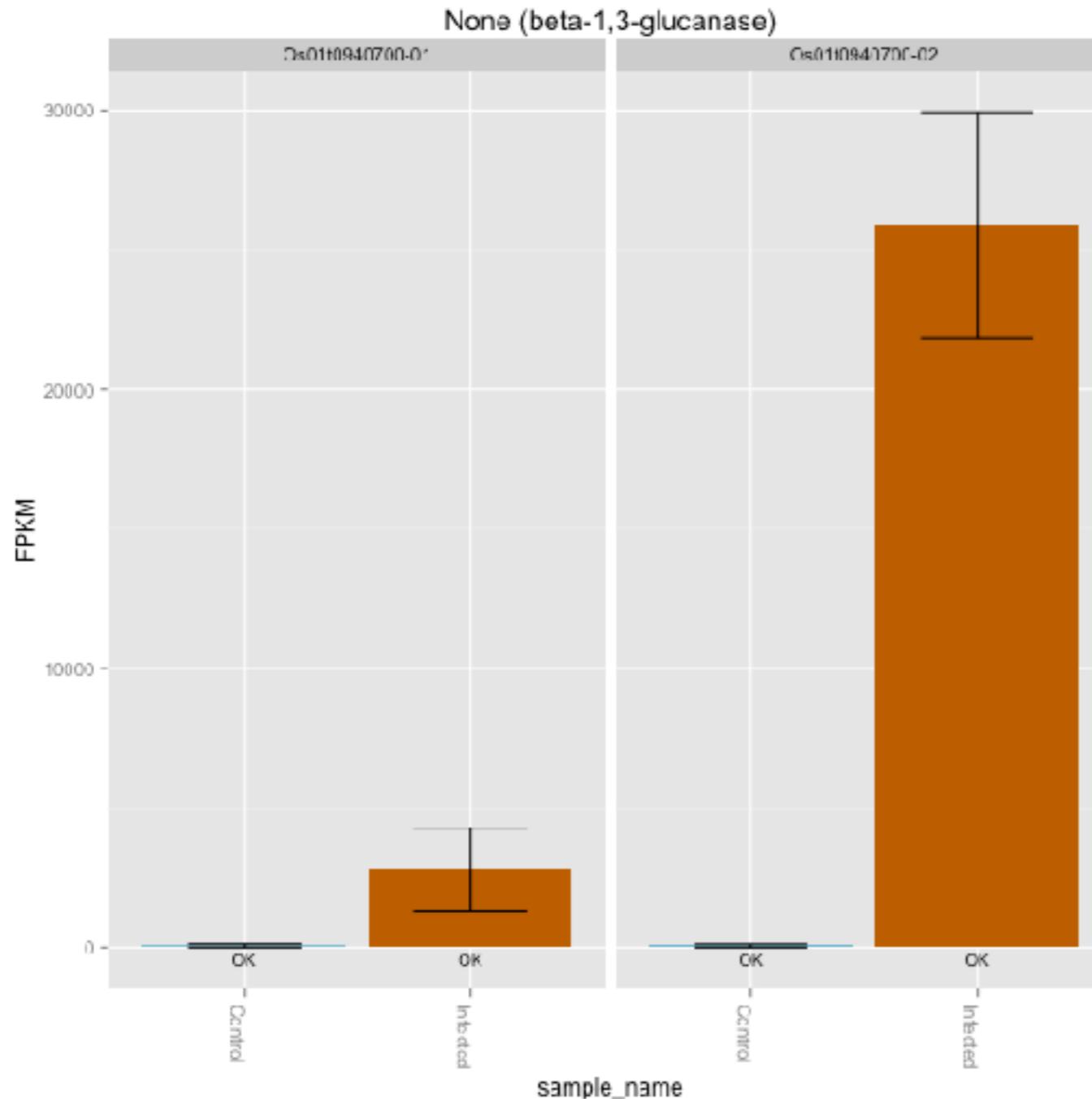
Volcano plotで遺伝子変動と有意差を合わせて調べる

```
> csVolcano(genes(cuff.data), "Control", "Infected",
alpha=0.01, showSignificant=T)
```



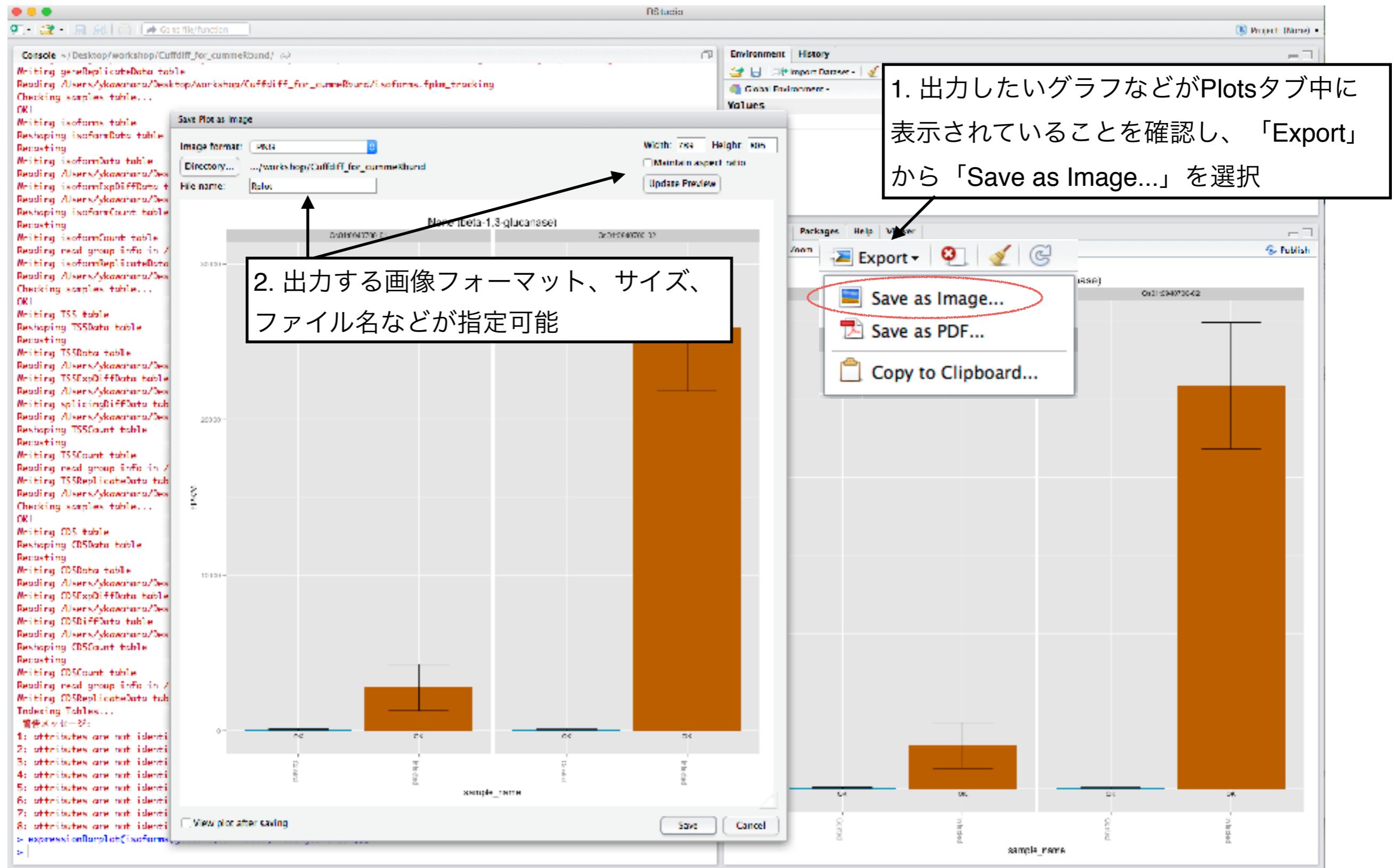
Os01g0940700のisoformごとの遺伝子発現量を調べる

```
> expressionBarplot(isoforms(getGene(cuff.data, "Os01g0940700")))
```

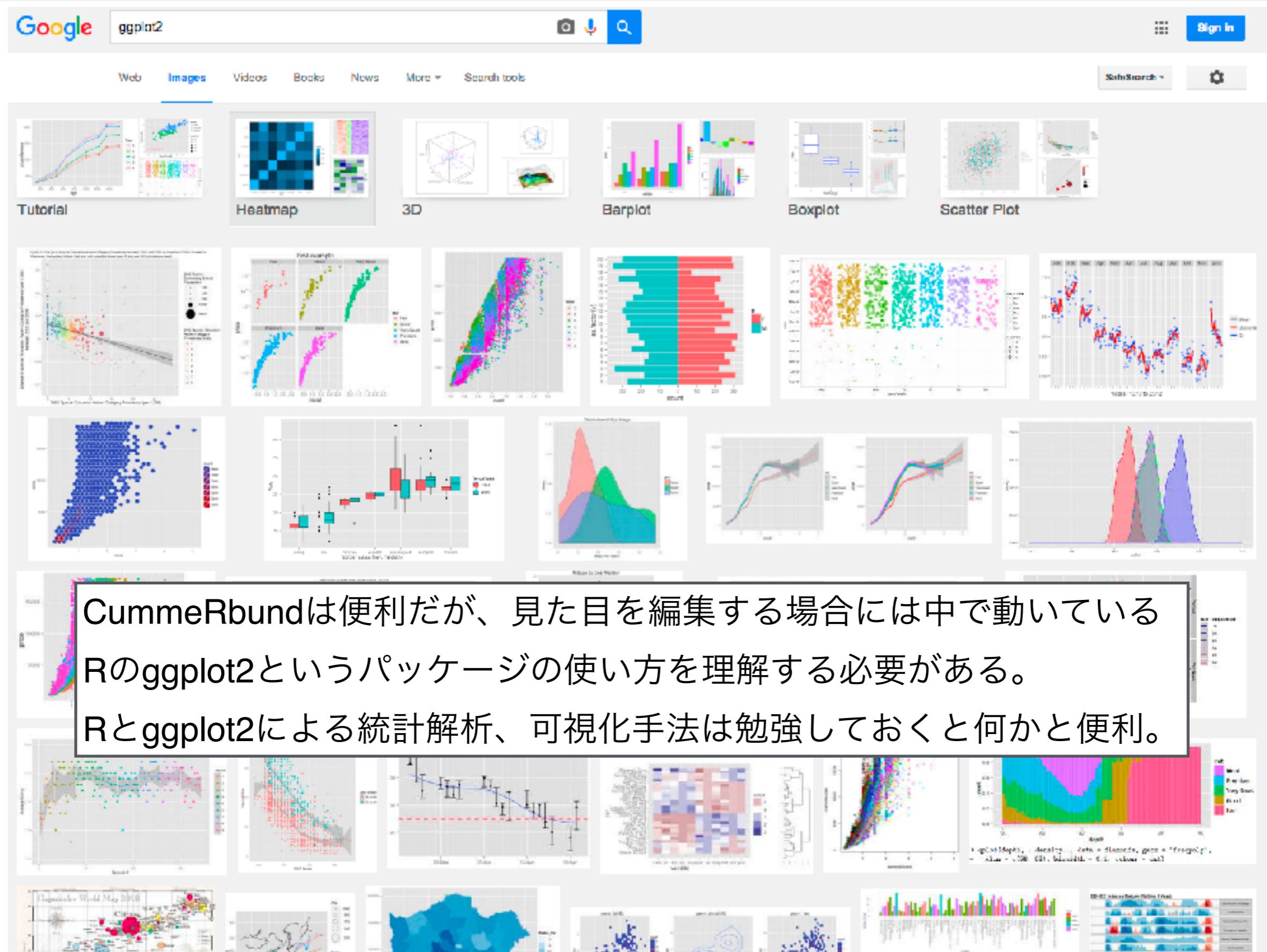


Os01g0940700はいもち病菌の細胞壁を壊すBeta-1,3-glucanase遺伝子をコードしており、イネの防御応答として感染時の発現誘導が知られている。

RStudioによる描画した図の画像出力



R/ggplot2



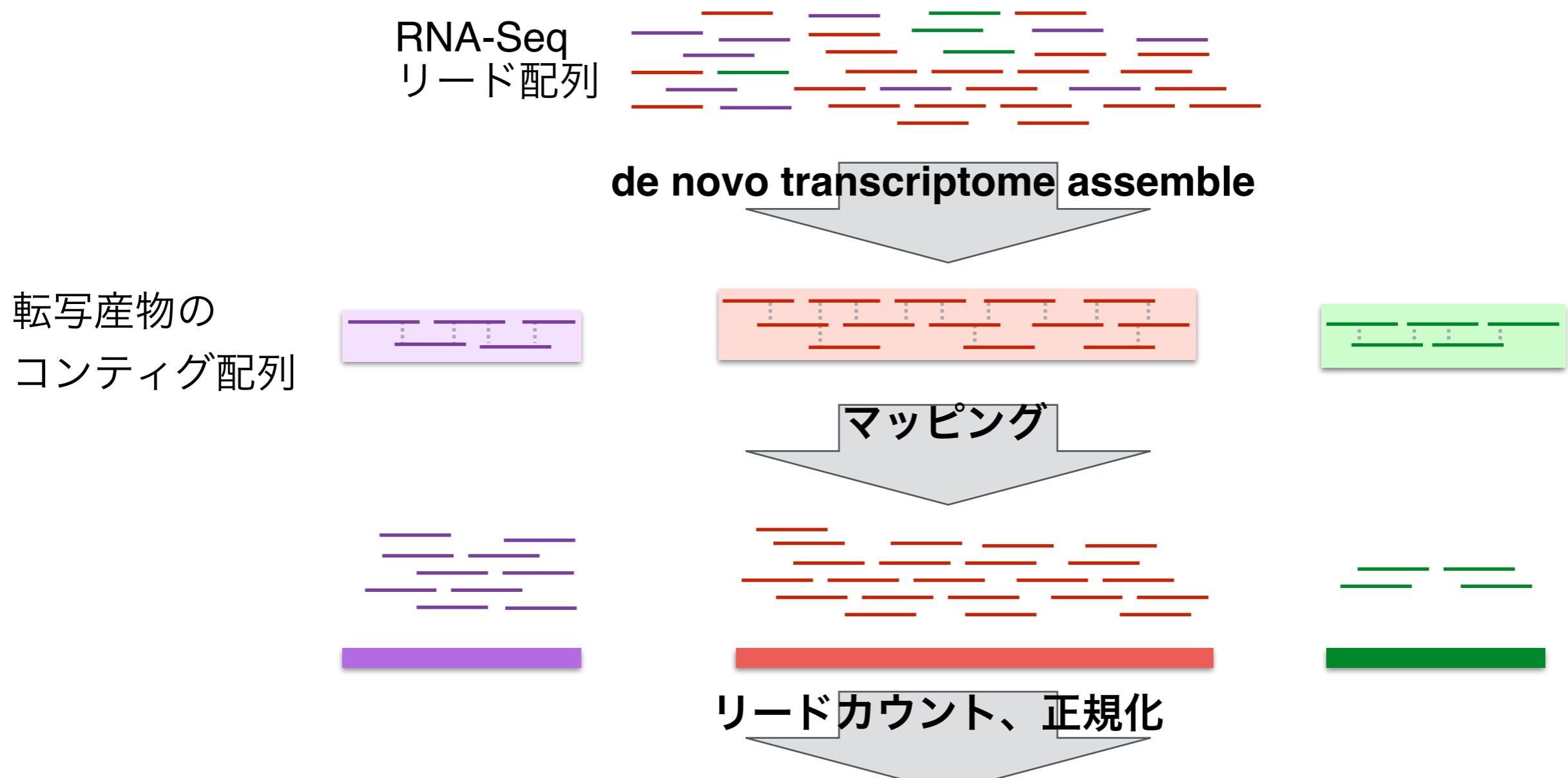
de novo assemble法

による発現解析

戦略2：de novo assemble法を用いた遺伝子発現解析

バラバラになった転写産物配列をアセンブルし、転写産物構造を再構築する。

アセンブルされた転写産物配列上にRNA-Seqリードをマッピング、カウントすることにより遺伝子発現を定量する。



遺伝子発現量：

10

11

4

de novo transcriptome assemble

良い点

- ・ゲノム配列を解読することなく、効率よく遺伝子配列を得ることができ
る上に発現情報も一緒に得られる。

悪い点

- ・リファレンスゲノムベースでの発現解析に比べ、信頼度が劣る。
- ・重複遺伝子だけではなく、選択的スプライシングによるアイソフォーム
も区別する必要があるうえに、発現量によって遺伝子間のリードの厚み
にバラつきがあり、アセンブルが困難。

それでもゲノム配列などのない非モデル生物においては、
トランскriプトーム解析の主流になっている。

本実習でおこなう解析と用いるデータ

2群間の遺伝子発現比較

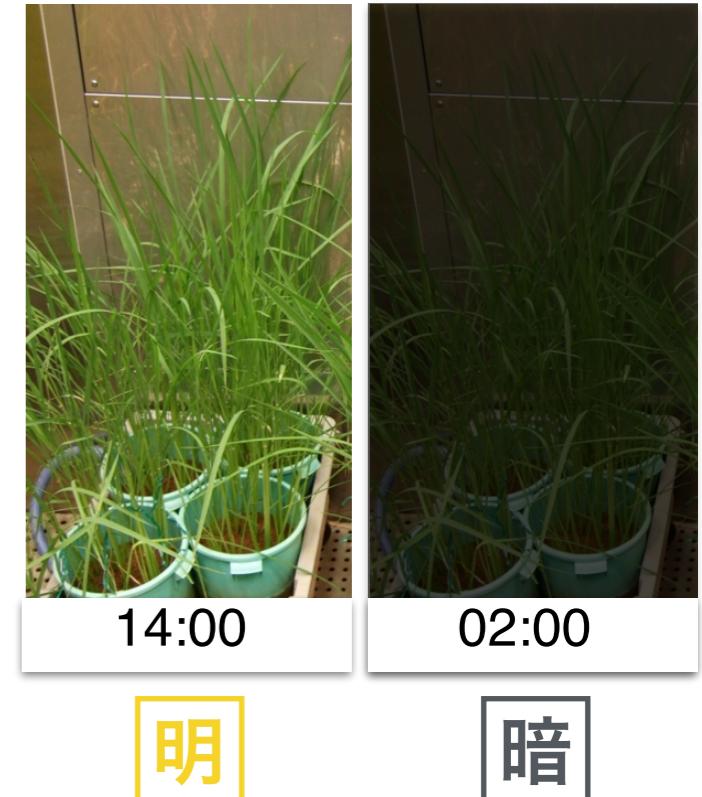
サンプル：イネ（日本晴）の葉@人工気象室

14時 vs 2時（2反復ずつ）

シーケンシング：HiSeq2000、paired-end read (100bp x 2)

解析方法：リファレンスゲノム配列を用いた発現解析

de novo assemble法による発現解析



RNA-Seqデータによる変異検出

サンプル：イネ（日本晴とコシヒカリ）の葉@人工気象室

14時（1反復ずつ）

シーケンシング：HiSeq2000、paired-end read (100bp x 2)

解析方法：リファレンスゲノム配列へのアラインメント、変異検出

実習用データ

0. denovo assemble 実習用のディレクトリに移動

```
$ cd  
$ cd rnaseq/denovo_assemble/
```

ホームに移動

denovo_assembleディレクトリに移動

1. 解析前のデータの確認

```
$ ls  
NPB_0200_rep1_r1.fq.gz  NPB_1400_rep2_r1.fq.gz  step_04_EvalByReadRep.sh  
NPB_0200_rep1_r2.fq.gz  NPB_1400_rep2_r2.fq.gz  step_05_EvalBySprotCov.sh  
NPB_0200_rep2_r1.fq.gz  samples.txt  
NPB_0200_rep2_r2.fq.gz  step_01_Trinity.sh  
NPB_1400_rep1_r1.fq.gz  step_02_RSEM.sh  
NPB_1400_rep1_r2.fq.gz  step_03_edgeR.sh
```

- ・日本晴の14時と2時の葉のRNA-Seqデータを利用
- ・各サンプル2反復（別の日の同時刻のサンプル）
- ・解析対象の領域は第6番染色体の一部（約200kb）

step_01: Trinityによるアセンブルの実行

```
$ perl $TRINITY_HOME/Trinity --seqType fq --max_memory 2G --CPU 1  
--normalize_reads --normalize_max_read_cov 50 --output Trinity_out  
--left  
NPB_1400_rep1_r1.fq.gz,NPB_1400_rep2_r1.fq.gz,NPB_0200_rep1_r1.fq.gz,  
NPB_0200_rep2_r1.fq.gz --right  
NPB_1400_rep1_r2.fq.gz,NPB_1400_rep2_r2.fq.gz,NPB_0200_rep1_r2.fq.gz,  
NPB_0200_rep2_r2.fq.gz
```

- `--seqType`, `--left`, `--right` : 入力ファイルとフォーマット
- `--output` : 出力先
- `--max_memory`, `--CPU` : 使用する計算機リソース
- `--normalize_reads`, `--normalize_max_read_cov` : *in silico normalization*に関するパラメータ
- *in silico normalization*とは、アセンブルの前におこなう冗長な配列を除く処理のこと。これをおこなうとアセンブル対象の配列が減り、必要な計算機リソースを減らすことができる。

以下のシェルスクリプトを実行

```
$ bash ./step_01_Trinity.sh (実行時間: 約3分)
```

Trinityによるアセンブルの結果

アセンブルされたコンティグ（転写産物配列）は、「Trinity_out/Trinity.fasta」

```
$ less Trinity_out/Trinity.fasta
>TRINITY_DN2_c0_g1_i1 len=215 path=[385:0-214] [-1, 385, -2]
GAECTCTAGCCAATGCTACAGTACCATGCGAAAAGAAAATTGAGGTCTGACTAACAGTAG
ACATTATGTACAAACAAAATATATTAAAACGATATTGTGCAGATTCATCTCCTCTGGA
ATTGAAAATTGCTTGAAAGCTCTTAAGAATTGAATTGGACTTGGAGGAATTATTA
TGGCAAATAATAATTGTTACCTTAGCTGGAGTG
>TRINITY_DN28_c0_g1_i1 len=643 path=[5774:0-135 5765:136-167 5766:168-191 5767:192-642]
[5774, 5765, 5766, 5767, -2]
GTGTGTTGAAGCCTGCCCTGACACTTAGCAACTGCAGAGTCCTGCTGGAGAGAGATGGG
GTTCATCGCGACACGATAGAGTCAATCCGCTCGATGCAGGTCCGCCAGGTGCTGGCGCA
AATCATCAGCTTAGGTGCAACCTTGCCTCTTGCCCTCAGTGATTATTTGCTCACAGAT
AAGTGGATTGGGTCCACTACTTCTTGATTTAGGCTTGGTATTTACCTTGATTCCAGG
...
```

「TRINITY_DN2_c0_g1_i1」や「TRINITY_DN28_c0_g1_i1」は転写産物配列ID。
それぞれ、「TRINITY_DN2_c0_g1遺伝子座のアイソフォーム1」の配列、「TRINITY_DN28_c0_g1遺伝子座のアイソフォーム1」の配列という意味。

以下のコマンドでは、転写産物数（91）と遺伝子座数（58）をカウントしている。

```
$ grep ">" Trinity_out/Trinity.fasta | wc -l
91
$ grep ">" Trinity_out/Trinity.fasta | cut -d "_" -f 1,2,3,4 | sort | uniq | wc -l
58
```

step_02: RSEMによる遺伝子発現の定量

1. RSEMを実行する

```
$ perl $TRINITY_HOME/util/align_and_estimate_abundance.pl --transcripts  
Trinity_out/Trinity.fasta --seqType fq --left NPB_1400_rep1_r1.fq.gz --  
right NPB_1400_rep1_r2.fq.gz --est_method RSEM --aln_method bowtie --  
output_dir RSEM_out_NPB_1400_rep1 --trinity_mode --prep_reference
```

- ・ 入出力部分の名前を適宜変えて、4サンプルそれぞれ独立にRSEMを実行する。
- ・ --seqType, --left, --right : 入力ファイルとそのフォーマット
- ・ --transcripts : Trinityの結果のコンティグファイル
- ・ --output_dir : 結果の出力先
- ・ --est_method RSEM : 発現量の定量方法
- ・ --aln_method bowtie : リードのアラインメントプログラム

次スライドにつづく→

step_02: RSEMによる遺伝子発現の定量（つづき）

2. RSEMによって得られた4サンプル分の発現情報をマージする

```
$ perl $TRINITY_HOME/util/abundance_estimates_to_matrix.pl --  
out_prefix trans_counts --est_method RSEM --name_sample_by_basedir  
RSEM_out_NPB_1400_rep1/RSEM.genes.results RSEM_out_NPB_1400_rep2/  
RSEM.genes.results RSEM_out_NPB_0200_rep1/RSEM.genes.results  
RSEM_out_NPB_0200_rep2/RSEM.genes.results
```

- --out_prefix trans_counts : 出力ファイルのprefix
- --est_method RSEM : 発現量の定量方法
- --name_sample_by_basedir : 出力結果のサンプル名
- 最後にRSEMの結果の4つのファイルをスペース区切りで指定

前のスライドのRSEMも合わせて、以下のシェルスクリプトを実行

```
$ bash ./step_02_RSEM.sh
```

(実行時間：約1分)

RSEMによる遺伝子発現の定量の結果

各遺伝子ごとのリードカウント数

```
$ less trans_counts.counts.matrix
    RSEM_out_NPB_1400_rep1  RSEM_out_NPB_1400_rep2  RSEM_out_NPB_0200_rep1
RSEM_out_NPB_0200_rep2
TRINITY_DN27_c0_g1      34.94    29.12    0.00    41.08
TRINITY_DN51_c0_g1      0.00     2.00     0.00     0.00
TRINITY_DN42_c0_g1      0.00     1.00     4.00     1.00
TRINITY_DN28_c0_g1      0.00    18.02    14.00    10.02
TRINITY_DN48_c0_g1      0.00     0.00     3.00     2.00
TRINITY_DN33_c0_g2      0.00     0.00     0.00     7.92
TRINITY_DN32_c2_g4     1396.00  968.00  1331.00  1481.00
```

各遺伝子ごとのTMM正規化されたリードカウント数（遺伝子発現量に相当）

```
$ less trans_counts.TMM.EXPR.matrix
    RSEM_out_NPB_1400_rep1  RSEM_out_NPB_1400_rep2  RSEM_out_NPB_0200_rep1
RSEM_out_NPB_0200_rep2
TRINITY_DN27_c0_g1      3036.474          1716.670          0.000          1533.282
TRINITY_DN51_c0_g1      0.000            4676.246          0.000          0.000
TRINITY_DN42_c0_g1      0.000            985.301          3410.889         694.251
TRINITY_DN28_c0_g1      0.000            5073.233          3297.518         1830.461
TRINITY_DN48_c0_g1      0.000            0.000            2447.591         1324.558
TRINITY_DN33_c0_g2      0.000            0.000            0.000          3007.679
TRINITY_DN32_c2_g4     300614.077        139601.706        158628.411        137821.751
```

step_03: edgeRによる発現変動する遺伝子の検出

edgeRを実行する

```
$ perl $TRINITY_HOME/Analysis/DifferentialExpression/  
run_DE_analysis.pl --matrix ./trans_counts.counts.matrix --method  
edgeR --samples_file ./samples.txt --output edgeR_out
```

- --matrix ./trans_counts.counts.matrix : マージしたRSEMの結果
- --method edgeR : 解析手法
- --sample_file samples.txt : サンプル情報。中身は下の通り
- --output edgeR_out : 結果の出力先

```
$ cat samples.txt  
NPB_1400 RSEM_out_NPB_1400_rep1  
NPB_1400 RSEM_out_NPB_1400_rep2  
NPB_0200 RSEM_out_NPB_0200_rep1  
NPB_0200 RSEM_out_NPB_0200_rep2
```

タブ区切りのテキストファイル

以下のシェルスクリプトを実行

```
$ bash ./step_03_edgeR.sh (実行時間: 数秒)
```

発現変動する遺伝子の検出結果

最初の遺伝子のみがFDR<0.01で有意に発現変動を示している。

	\$ less edgeR_out/trans_counts.counts.matrix.NPB_0200_vs_NPB_1400.edgeR.DE_results	logFC	logCPM	PValue	FDR
TRINITY_DN29_c0_g1	-4.50196521863603	14.8914251378066	0.000108396961389901	0.00596183287644458	
TRINITY_DN25_c0_g1	0.987562778190113	16.9366047408829	0.00729287440936741	0.200554046257604	
TRINITY_DN38_c0_g1	-5.11907396065862	11.5239540930861	0.0109698466087043	0.201113854492912	
TRINITY_DN30_c0_g3	7.77063724629797	10.354136550185	0.0704933878059558	0.809652883433794	
TRINITY_DN35_c0_g1	7.00303281643664	9.7100759241507	0.0736048075848904	0.809652883433794	
TRINITY_DN32_c2_g2	-0.638342681863463	17.2556382204509	0.0971102755106584	0.828909895919131	
TRINITY_DN24_c0_g1	-1.14296633437692	15.8398427784775	0.10549762311698	0.828909895919131	
TRINITY_DN25_c0_g2	1.71142905217539	13.9063924587969	0.136344744500824	0.889080890172577	
TRINITY_DN30_c0_g1	-6.12537914054916	9.09477868150286	0.145485963846422	0.889080890172577	
TRINITY_DN32_c4_g1	-3.63569390694803	9.14362497820752	0.250920168831648	1	

logFC:log2(fold change), logCPM:log2(counts per million), PValue:P-value, FDR:False Discovery Rate

step_05: アセンブルの評価（リードの再マップ率）

元のリードをコンティグに再マッピングすることで、どれぐらいのリードがアセンブルに貢献しているかを見る。

```
$ perl $TRINITY_HOME/util/bowtie_PE_separate_then_join.pl --seqType fq  
--run_rsem --trinity_mode --output Bowtie_out --left  
NPB_1400_rep1_r1.fq.gz,NPB_1400_rep2_r1.fq.gz,NPB_0200_rep1_r1.fq.gz,NPB_0200  
_rep2_r1.fq.gz --right  
NPB_1400_rep1_r2.fq.gz,NPB_1400_rep2_r2.fq.gz,NPB_0200_rep1_r2.fq.gz,NPB_0200  
_rep2_r2.fq.gz --target Trinity_out/Trinity.fasta --aligner bowtie  
-- -p 1 --all --best --strata -m 300  
  
$ perl $TRINITY_HOME/util/SAM_nameSorted_to_uniq_count_stats.pl  
Bowtie_out/Bowtie_out.nameSorted.bam > read_rep_stats.txt
```

- 「--」以下の「-p 1 --all --best --strata -m 300」はコンティグにリードをマッピングする際に用いるBowtieに渡すパラメータ。

以下のシェルスクリプトを実行

```
$ bash ./step_04_EvalByReadRep.sh (実行時間：約2分)
```

step_05: アセンブルの評価（リードの再マップ率）

今回の実習用の第6染色体の一部
のデータのみを使った結果。

```
$ cat read_rep_stats.txt
```

#read_type	count	pct
proper_pairs	93466	96.33
improper_pairs	2352	2.42
left_only	636	0.66
right_only	570	0.59

Total aligned reads: 97024

全トランск립トームデータを
使った結果。

#read_type	count	pct
proper_pairs	251938506	85.04
improper_pairs	31484108	10.63
left_only	6514770	2.20
right_only	6325174	2.13
Total aligned reads:	296262558	

「proper_pairs」が、~70-80%になるのが一般的のようです。

step_05: アセンブルの評価（既知遺伝子との比較）

既知の転写産物との構造の一致度によって、アセンブルの良し悪しを評価する。
コンティグ配列をクエリーとして、既知のタンパク質配列データベース（SwissProt）に
相同性検索をし、マップ率、カバー率を見る。

```
$ blastx -query Trinity_out/Trinity.fasta -task blastx-fast  
-evalue 1e-20 -max_target_seqs 1 -outfmt 6 -db $BLAST_DB -out blastx.out  
  
$ perl $TRINITY_HOME/util/analyze_blastPlus_topHit_coverage.pl blastx.out  
Trinity_out/Trinity.fasta $BLAST_DB > sprot_coverage.txt
```

- 「-outfmt 6」は次のカバー率等を計算するプログラムが読めるように出力フォーマットをテーブル形式に指定しています。
- 本実習では計算を早くするために「blastx-fast」を指定しています。
- \$BLAST_DBはSwissProtのBLASTデータベース「/work/NGSworkshop2015/BLAST_DB/uniprot_sprot.fasta」です。

以下のシェルスクリプトを実行

```
$ bash ./step_05_EvalBySprotCov.sh
```

(実行時間：約3分)

既知遺伝子との比較結果

今回の実習用の第6染色体の一部
のデータのみを使った結果。

```
$ cat sprot_coverage.txt
#hit_pct_cov_bin count_in_bin >bin_below
100 4 4
90 1 5
80 6 11
70 1 12
60 0 12
50 0 12
40 0 12
30 1 13
20 3 16
10 0 16
```

100 : 90% < coverage <= 100%

90 : 80% < coverage <= 90%

...

全ranscriptomeデータを使つ
た結果 (158,099転写産物)。

#hit_pct_cov_bin	count_in_bin	>bin_below
100	3902	3902
90	1559	5461
80	1144	6605
70	883	7488
60	847	8335
50	761	9096
40	677	9773
30	635	10408
20	469	10877
10	113	10990

わずかにでも既知遺伝子との相同意が確認
できたのは、 $10,990/158,099=7\%$
意外と少ない・・

SwissProtは信頼度の高い配列のみに絞っておりエント
リーが少ない (549,832本) 、閾値が厳しい、blastx-fast
を使っていることなどが理由か。

既知遺伝子との比較結果

先ほどDEGのとして検出された「TRINITY_DN29_c0_g1」を
blast検索結果から探してみると・・・

```
$ less edgeR_out/trans_counts.counts.matrix.NPB_0200_vs_NPB_1400.edgeR.DE_results
logFC  logCPM PValue FDR
TRINITY_DN29_c0_g1 -4.50196521863603 14.8914251378066 0.000108396961389901
0.00596183287644458
```

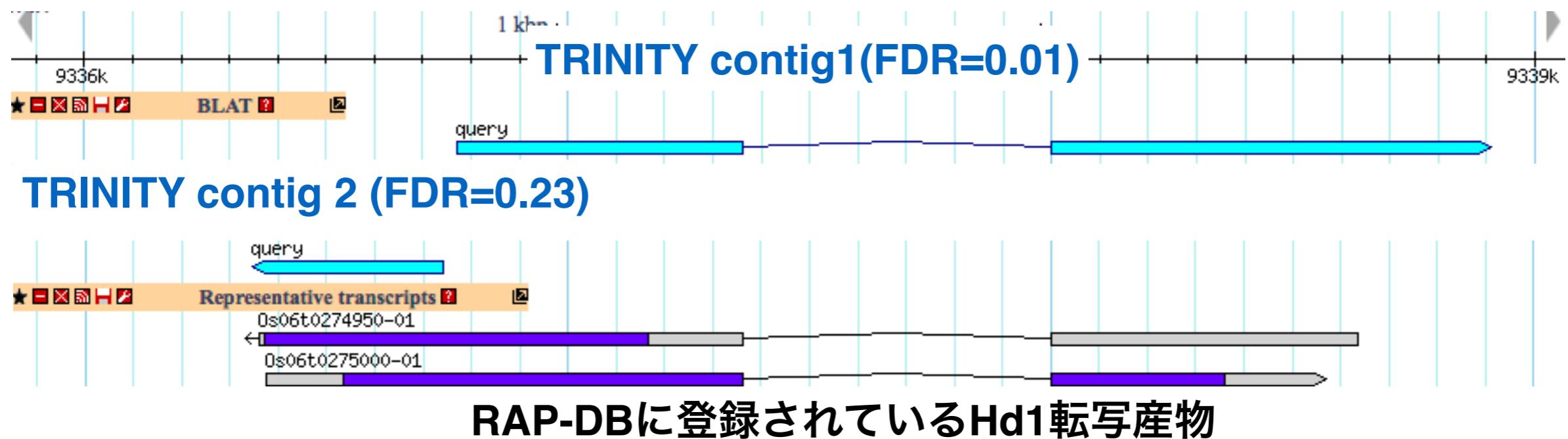
どうやら発現が日周変動することが知られている「Hd1」らしい。

```
$ grep TRINITY_DN29_c0_g1 blastx.out.w_pct_hit_length
TRINITY_DN29_c0_g1_i1  sp|Q9FDX8|HD1_ORYSJ  100.00    114    0    0    893    552    282
 395  9e-117  239  114  28.86  Zinc finger protein HD1  OS=Oryza sativa subsp.
japonica  GN=HD1  PE=2  SV=1
TRINITY_DN29_c0_g1_i2  sp|Q9FDX8|HD1_ORYSJ  100.00    114    0    0    893    552    282
 395  1e-69   239  114  28.86  Zinc finger protein HD1  OS=Oryza sativa subsp.
japonica  GN=HD1  PE=2  SV=1
TRINITY_DN29_c0_g1_i3  sp|Q9FDX8|HD1_ORYSJ  100.00    292    0    0   1427    552    104
 395  6e-148  436  292  73.92  Zinc finger protein HD1  OS=Oryza sativa subsp.
japonica  GN=HD1  PE=2  SV=1
```

アセンブル結果の評価（BLATによるマッピング）

ゲノム配列やアノテーションが整備されている場合、BLAT等でアラインメントし、マップ率や遺伝子構造を比較することで評価が可能です。

RAP-DBのBLAT検索によってイネゲノムにマッピングされたコンティグ配列



残念ながら1つの転写産物が2つのコンティグに分かれてしまっている。

遺伝子機能アノテーション

TransDecoder (Find Coding Regions Within Transcripts)

TransDecoder identifies candidate coding regions within transcript sequences, such as those generated by de novo RNA-Seq transcript assembly using Trinity, or constructed based on RNA-Seq alignments to the genome using Tophat and Cufflinks.

TransDecoder identifies likely coding sequences based on the following criteria:

- a minimum length open reading frame (ORF) is found in a transcript sequence
- a log-likelihood score similar to what is computed by the GencID software is > 0.
- the above coding score is greatest when the ORF is scored in the 1st reading frame as compared to scores in the other 5 reading frames.
- if a candidate ORF is found fully encapsulated by the coordinates of another candidate ORF, the longer one is reported. However, a single transcript can report multiple ORFs (allowing for operons, chimeras, etc).
- optional the putative peptide has a match to a Pfam domain above the noise cutoff score.

The software is primarily maintained by Brian Haas at the Broad Institute and Alexio Panagiotou at the Commonwealth Scientific and Industrial Research Organisation (CSIRO). It is integrated into other related software such as Trinity, PASA, EViidence-Mender, and Trinotate.

<http://transdecoder.github.io/>

コンティグ配列と発現量が得られただけでは何もわからない・・・。

その転写産物が何者なのかが知りたい！

- ・タンパク質配列
- ・機能が分かっているホモログ
- ・機能ドメイン
- ・機能分類 (Gene Ontology)

Trinotate: Transcriptome Functional Annotation and Analysis

Trinotate



RNA-Seq → Trinity → Transcripts/Proteins → Functional Data → Discovery

Automated Higher Order Biological Analysis

<https://trinotate.github.io/>

明日のGalaxyを使った演習で紹介予定

RNA-Seqデータによる変異解析

本実習でおこなう解析と用いるデータ

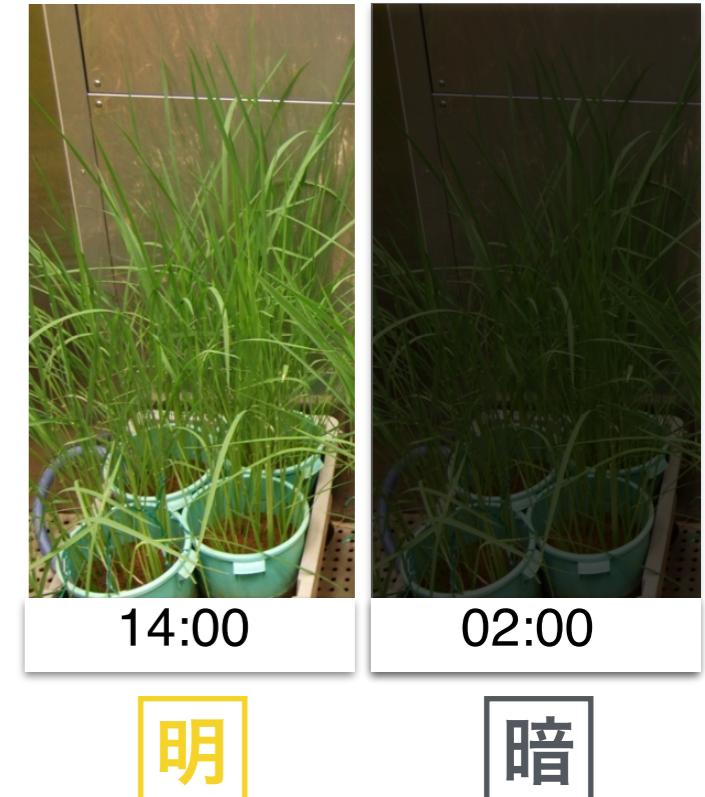
2群間の遺伝子発現比較

サンプル：イネ（日本晴）の葉@人工気象室

14時 vs 2時（2反復ずつ）

シーケンシング：HiSeq2000、paired-end read (100bp x 2)

解析方法：**リファレンスゲノム配列を用いた発現解析
de novo assemble法による発現解析**



RNA-Seqデータによる変異検出

サンプル：イネ（日本晴とコシヒカリ）の葉@人工気象室

14時（1反復ずつ）

シーケンシング：HiSeq2000、paired-end read (100bp x 2)

解析方法：**リファレンスゲノム配列へのアラインメント、変異解析**

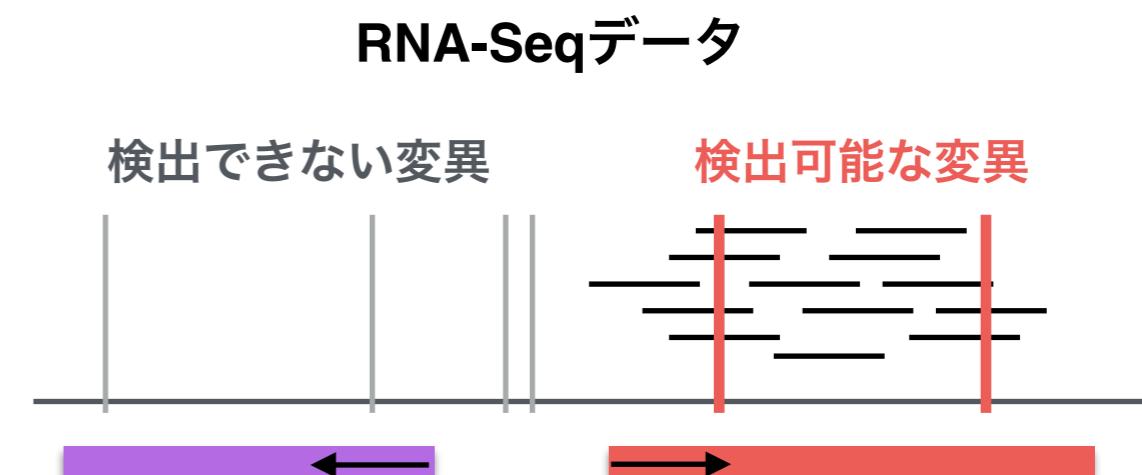
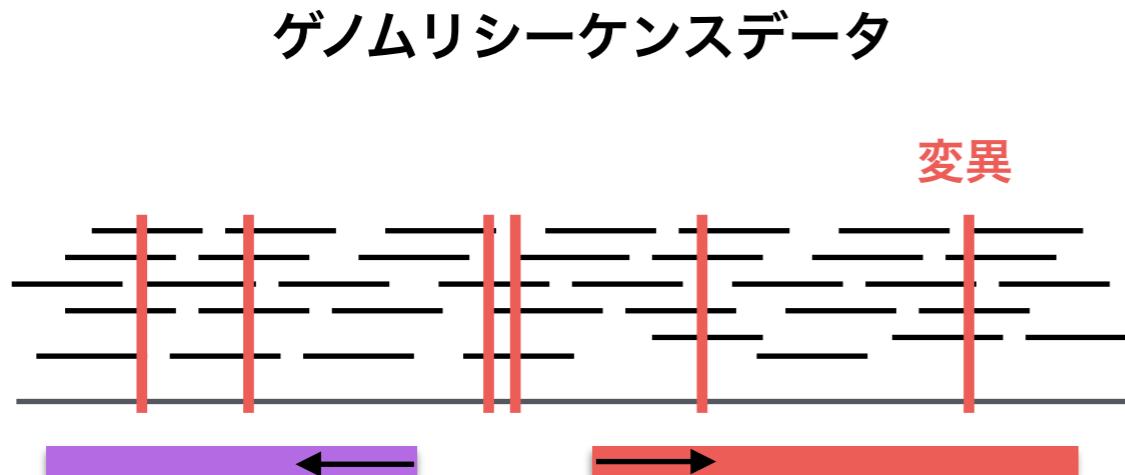
RNA-Seqデータによる変異解析の特徴

良い点

- ・ 巨大なゲノムでも効率よく多型情報が得られる。
- ・ 遺伝子発現解析と同時に多型情報が得られる。
- ・ 進化的に距離の離れた種の比較に有用な変異が得られる
(転写領域は進化的に保存されているため)。

悪い点

- ・ 発現していない遺伝子の変異は得られない。
- ・ 遺伝子発現解析と同時にできるが、転写開始点上流の発現調節領域の変異は得られない。



GATKのフォーラムに詳しい解析方法があります

Calling variants in RNAseq



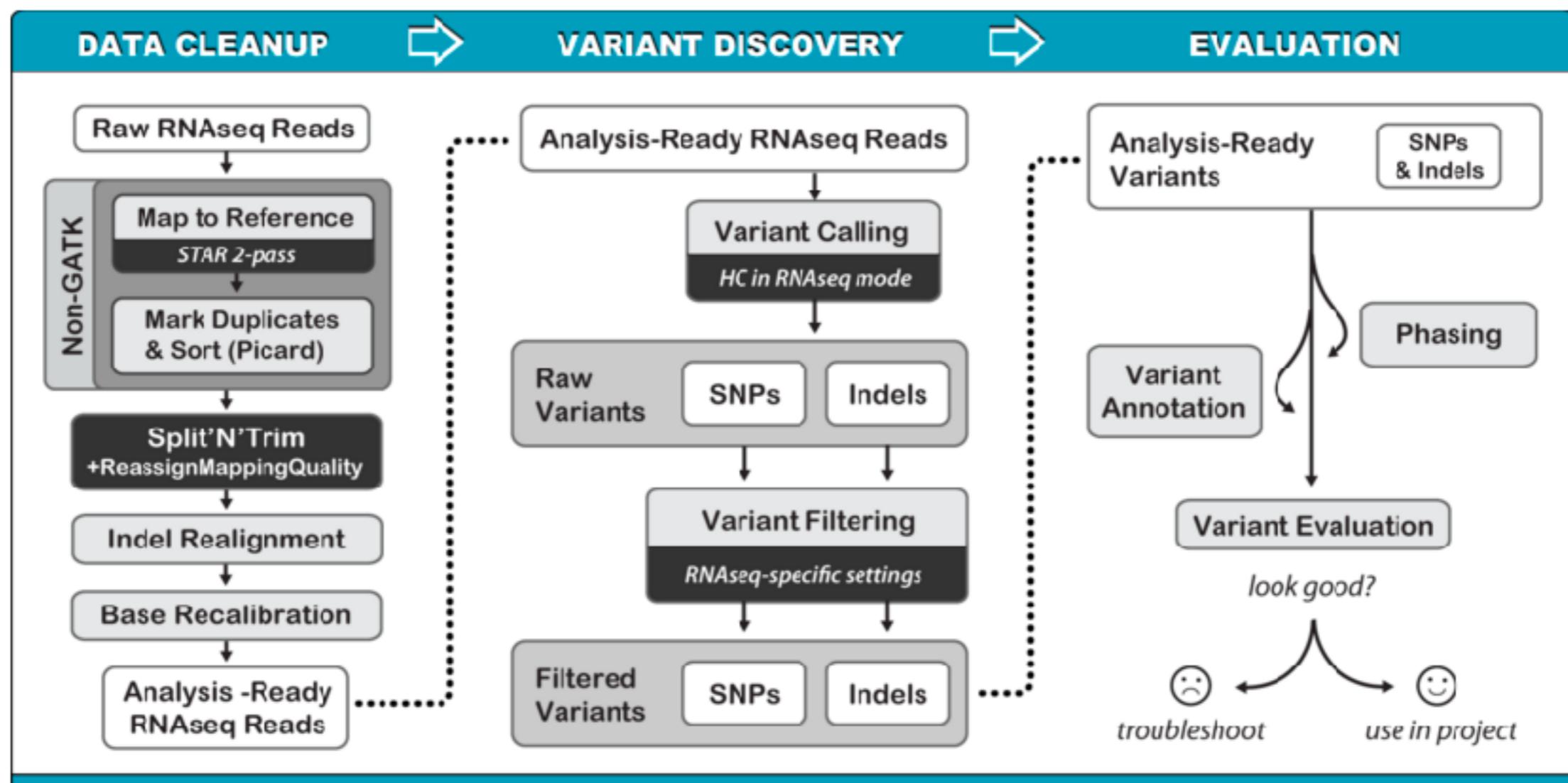
Geraldine_VdAuwera Posts: 8,748 Administrator, GATK Dev admin
March 2014 edited November 2014 in Methods and Algorithms

<http://gatkforums.broadinstitute.org/discussion/3891/calling-variants-in-rnaseq>

Overview

This document describes the details of the GATK Best Practices workflow for SNP and indel calling on RNAseq data.

Please note that any command lines are only given as example of how the tools can be run. You should always make sure you understand what is being done at each step and whether the values are appropriate for your data. To that effect, you can find more guidance [here](#).



実習用データ

0. denovo assemble 実習用のディレクトリに移動

```
$ cd  
$ cd rnaseq/var_call/
```

ホームに移動
var_callディレクトリに移動

1. 解析前のデータの確認

```
$ ls  
KOS_1400_rep1_r1.fq.gz annotation.gtf  
KOS_1400_rep1_r2.fq.gz genome.fa  
NPB_1400_rep1_r1.fq.gz run_var_call.sh  
NPB_1400_rep1_r2.fq.gz
```

- ・日本晴とコシヒカリのRNA-Seqデータを利用
- ・解析対象の領域は第3番染色体の一部（約200kb）

2. TopHatから変異検出までを実行するシェルスクリプト

```
$ bash ./run_var_call.sh
```

(実行時間：約4分)

Step1: TopHatによるRNA-Seqリードのアラインメント

ここでは、日本晴データの解析のためのシェルスクリプトのみを示すので、同様の処理をコシヒカリデータに対してもおこなう必要があります。

- 解析に必要なインデックス等の作成 -

```
$ bowtie2-build genome.fa genome  
$ samtools faidx genome.fa  
$ java -Xmx4G -Xms2G -jar $PICARD_HOME/picard.jar  
CreateSequenceDictionary R=genome.fa O=genome.dict
```

- TopHatによるRNA-Seqリードのアラインメント -

```
$ tophat2 --min-intron-length 10 --max-intron-length 10000 --output-dir  
TopHat_out_NPB_1400_rep1 genome NPB_1400_rep1_r1.fq.gz  
NPB_1400_rep1_r2.fq.gz
```

```
$ mv TopHat_out_NPB_1400_rep1/accepted_hits.bam  
TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam
```

```
$ samtools index TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam
```

Step2: アラインメント結果のフィルタリング、整形

- それぞれのアラインメントにグループ名などを付ける (GATKによる解析に必要) -

```
$ mkdir VarCall_NPB_1400_rep1  
$ java -Xmx4G -Xms2G -jar $PICARD_HOME/picard.jar AddOrReplaceReadGroups  
INPUT=TopHat_out_NPB_1400_rep1/NPB_1400_rep1.bam  
OUTPUT=VarCall_NPB_1400_rep1/NPB_1400_rep1.sorted.bam SO=coordinate  
RGID=NPB RGLB=NPB RGPL= Illumina RGPU= Illumina RGSM=NPB
```

- PCRによって冗長になったリードを取り除く (MarkDuplicates) -

```
$ java -Xmx4G -Xms2G -jar $PICARD_HOME/picard.jar MarkDuplicates  
INPUT=VarCall_NPB_1400_rep1/NPB_1400_rep1.sorted.bam  
OUTPUT=VarCall_NPB_1400_rep1/NPB_1400_rep1.rmdup.bam  
METRICS_FILE=VarCall_NPB_1400_rep1/NPB_1400_rep1.rmdup.matrix  
REMOVE_DUPLICATES=true
```

```
$ samtools index VarCall_NPB_1400_rep1/NPB_1400_rep1.rmdup.bam
```

- イントロン領域の情報を取り除く (GATKによる解析に必要) -

```
$ java -Xmx4G -Xms2G -jar $GATK_HOME/GenomeAnalysisTK.jar -T  
SplitNCigarReads -R genome.fa -I VarCall_NPB_1400_rep1/  
NPB_1400_rep1.rmdup.bam -o VarCall_NPB_1400_rep1/NPB_1400_rep1.split.bam  
-U ALLOW_N_CIGAR_READS
```

Step3: GATK HaplotypeCallerによる変異検出とフィルタリング

- HaplotypeCallerによって変異を検出する -

```
$ java -Xmx4G -Xms2G -jar $GATK_HOME/GenomeAnalysisTK.jar -T HaplotypeCaller -R genome.fa -I VarCall_NPB_1400_rep1/NPB_1400_rep1.split.bam -dontUseSoftClippedBases -stand_call_conf 20.0 -stand_emit_conf 20.0 -o VarCall_NPB_1400_rep1/NPB_1400_rep1.vcf
```

- HaplotypeCallerによって変異を検出する -

```
$ java -Xmx4G -Xms2G -jar $GATK_HOME/GenomeAnalysisTK.jar -T VariantFiltration -R genome.fa -V VarCall_NPB_1400_rep1/NPB_1400_rep1.vcf -window 20 -cluster 3 -filterName QUAL -filter "QUAL < 100" -filterName QD -filter "QD < 2.0" -filterName FS -filter "FS > 30.0" -o VarCall_NPB_1400_rep1/NPB_1400_rep1.filtered.vcf
```

- 基本的にはゲノムリシーケンスデータによる変異検出と同じ。
- stranded (strand-specific) RNA-Seqという方法によって得られたRNA-Seqデータの利用はおすすめできない。特定の配列にエラーに入る傾向のあるIlluminaのシークエンシングデータによる疑陽性を除くために、一般的に片側方向のリードのみで指示される変異はフィルタリング (FS) によって除くが、それができないため。

変異検出、フィルタリングの結果

```
$ awk -F "\t" '$7 == "PASS"' VarCall_NPB_1400_rep1/NPB_1400_rep1.filtered.vcf
1 chr03 32985032 . A G 162.90 PASS
AC=2;AF=1.00;AN=2;DP=5;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=50.00;QD=32.58;SOR=1.022
GT:AD:DP:GQ:PL 1/1:0,5:5:15:191,15,0

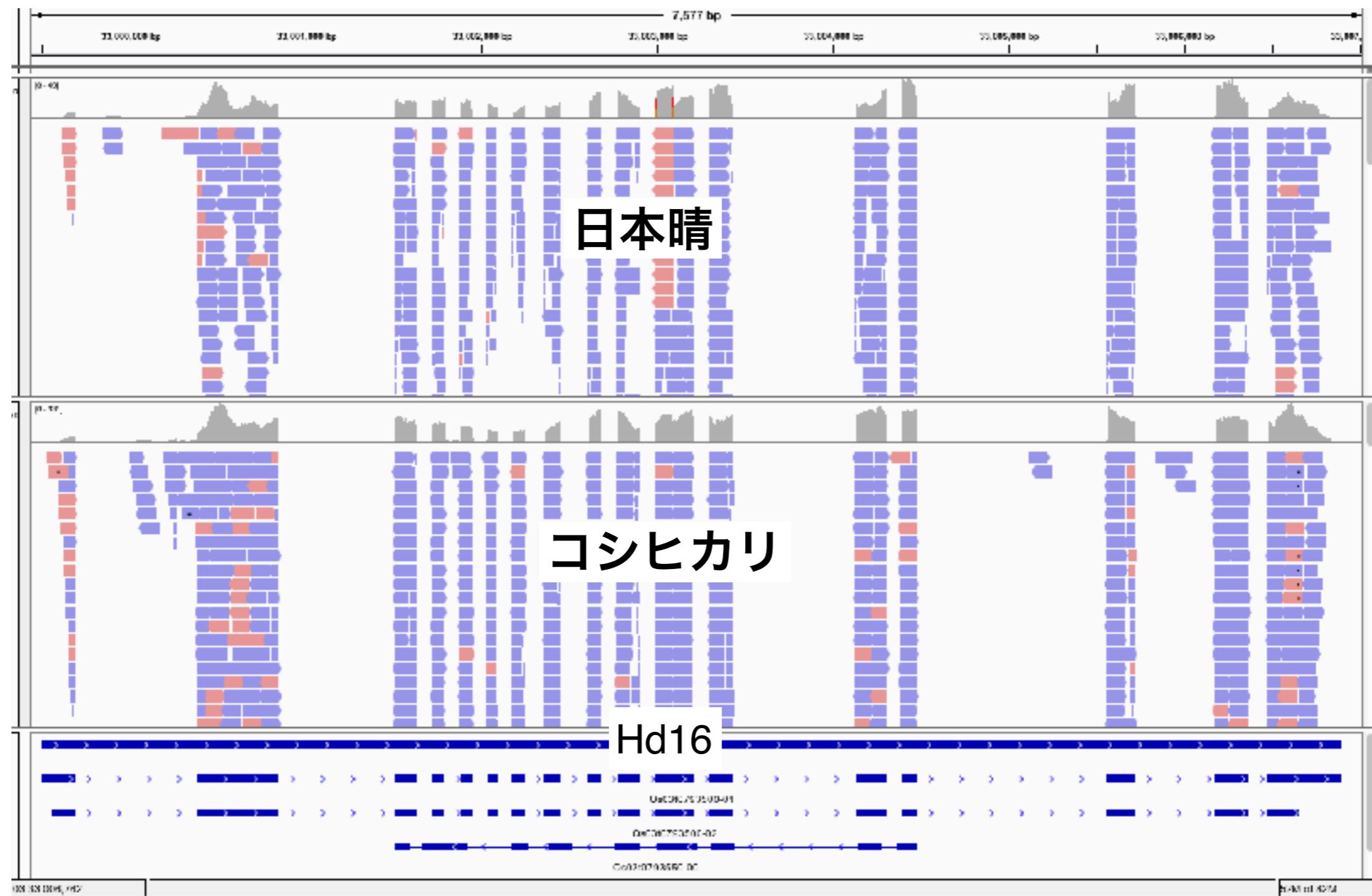
$ awk -F "\t" '$7 == "PASS"' VarCall_KOS_1400_rep1/KOS_1400_rep1.filtered.vcf
1 chr03 32961688 . G GC 122.06 PASS
AC=2;AF=1.00;AN=2;DP=13;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=50.00;QD=9.39;SOR=3.258
GT:AD:DP:GQ:PL 1/1:0,4:4:11:159,11,0
2 chr03 32985032 . A G 603.77 PASS
AC=1;AF=0.500;AN=2;BaseQRankSum=0.636;ClippingRankSum=-0.058;DP=26;FS=0.000;MLEAC=1
;MLEAF=0.500;MQ=50.00;MQRankSum=-0.636;QD=23.22;ReadPosRankSum=-0.289;SOR=0.412
GT:AD:DP:GQ:PL 0/1:7,19:26:99:632,0,180
3 chr03 33002789 . G A 2671.77 PASS
AC=2;AF=1.00;AN=2;BaseQRankSum=-0.507;ClippingRankSum=1.056;DP=82;FS=0.000;MLEAC=2;
MLEAF=1.00;MQ=50.00;MQRankSum=1.225;QD=32.58;ReadPosRankSum=0.591;SOR=0.262
GT:AD:DP:GQ:PL 1/1:1,81:82:99:2700,206,0
4 chr03 33006724 . G GA 117.73 PASS
AC=1;AF=0.500;AN=2;BaseQRankSum=-0.819;ClippingRankSum=0.637;DP=50;FS=0.000;MLEAC=1
;MLEAF=0.500;MQ=50.00;MQRankSum=1.105;QD=2.35;ReadPosRankSum=1.677;SOR=0.813
GT:AD:DP:GQ:PL 0/1:37,12:49:99:155,0,897
5 chr03 33064837 . C CA 227.73 PASS
AC=1;AF=0.500;AN=2;BaseQRankSum=-3.061;ClippingRankSum=0.122;DP=83;FS=4.469;MLEAC=1
;MLEAF=0.500;MQ=50.00;MQRankSum=-0.674;QD=2.74;ReadPosRankSum=0.901;SOR=0.142
GT:AD:DP:GQ:PL 0/1:50,21:71:99:265,0,1215
6 chr03 33082634 . A G 103.28 PASS
AC=2;AF=1.00;AN=2;DP=3;FS=0.000;MLEAC=2;MLEAF=1.00;MQ=50.00;QD=34.43;SOR=1.179
GT:AD:DP:GQ:PL 1/1:0,3:3:9:131,9,0
```

RNA-Seqデータを用いて品種間のFNPを検出

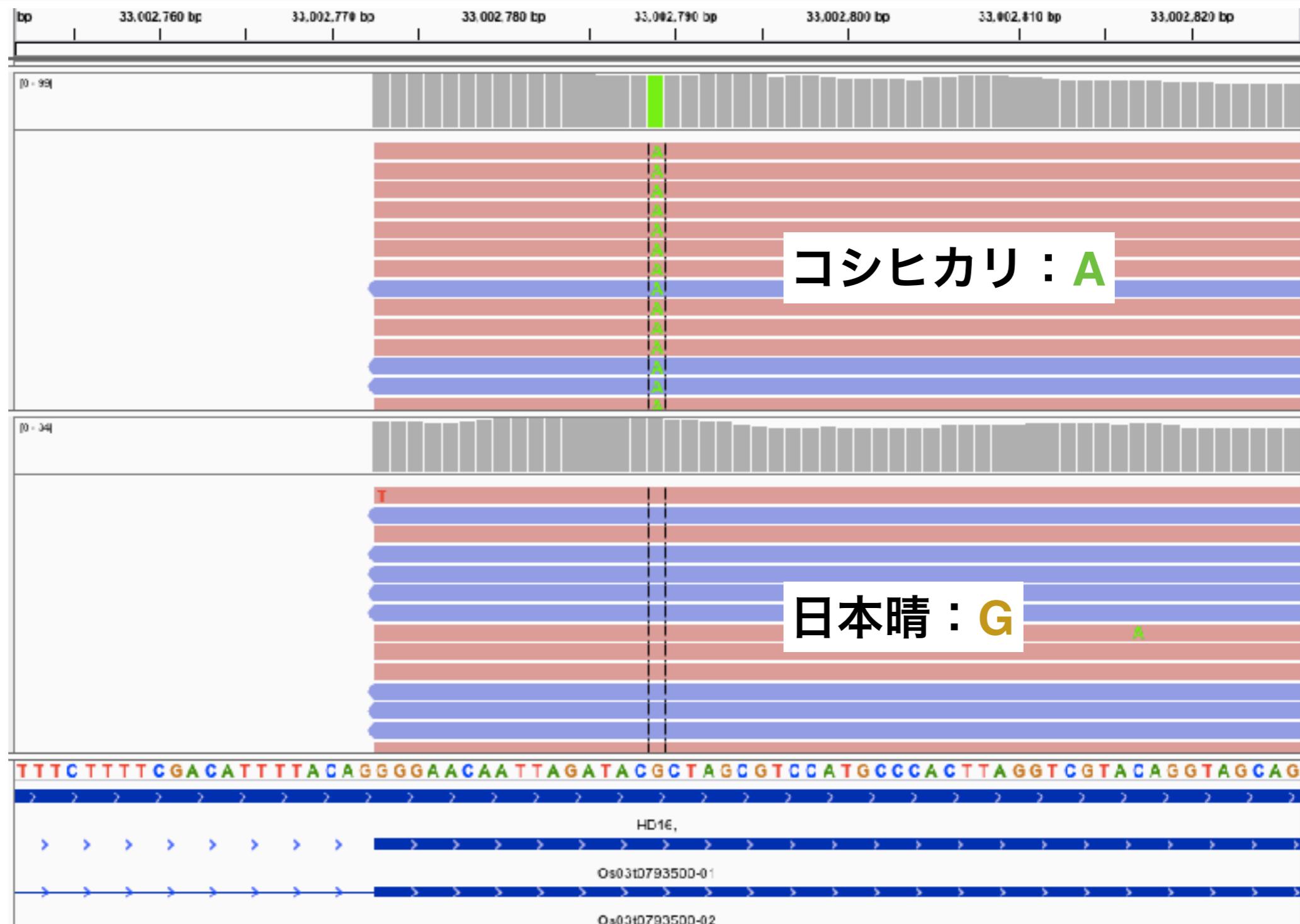
日本晴とコシヒカリの開花期の違いに関する既知のFNP

FNP=Functional Nucleotide Polymorphism

Reported in Hori K et al. Plant J. 2013



品種間のFNPを正しく検出できている



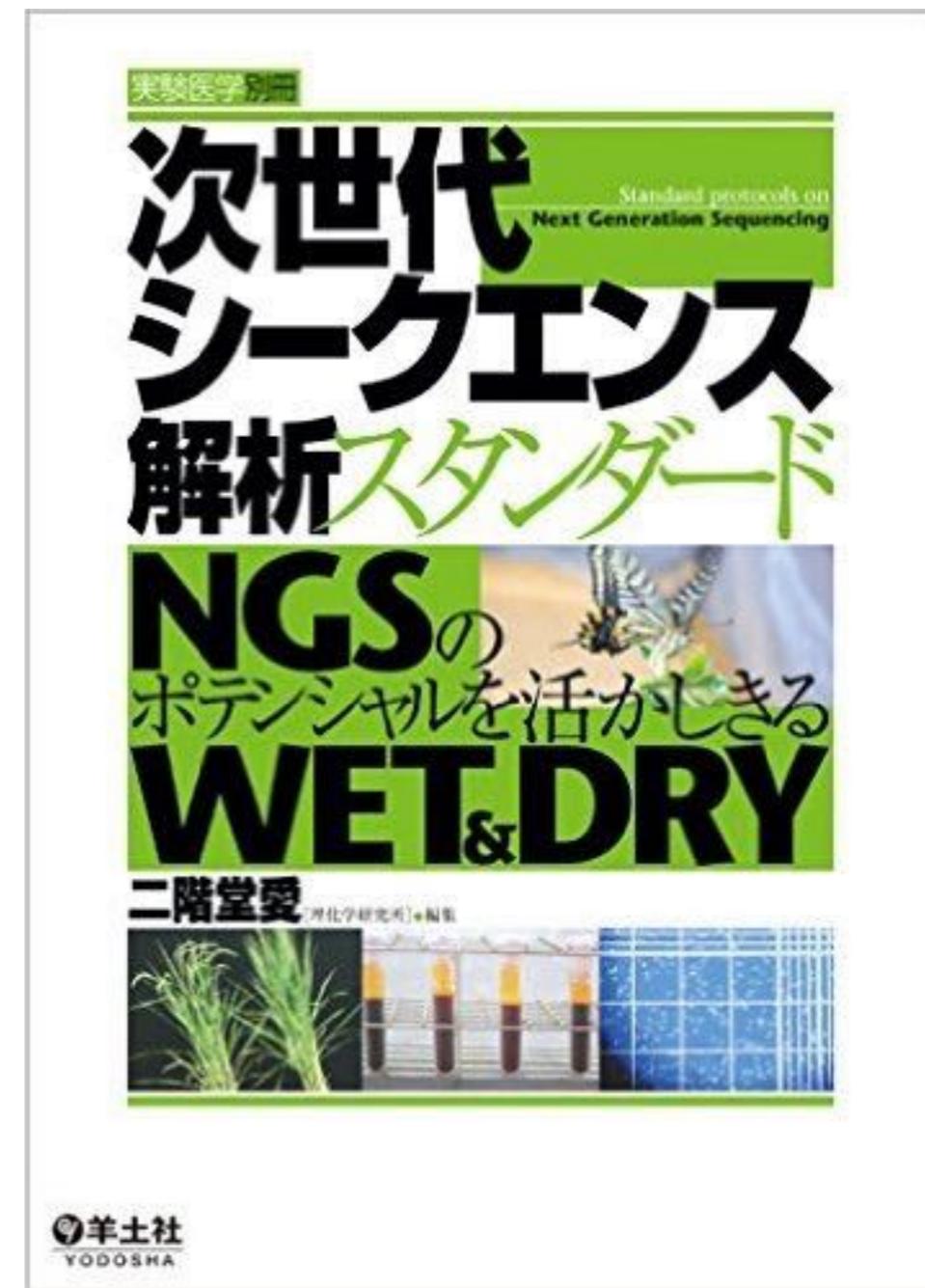
chr03 33002789 . G A 2671.77 PASS
VCF
AC=2;AF=1.00;AN=2;BaseQRankSum=-0.507;ClippingRankSum=1.056;DP=82;FS
ファイル =0.000;MLEAC=2;MLEAF=1.00;MQ=50.00;MQRankSum=1.225;QD=32.58;ReadPosR
ankSum=0.591;SOR=0.262 GT:AD:DP:GQ:PL 1/1:1,81:82:99:2700,206,0

参考書籍

次世代シークエンサーDRY解析教本 (細胞工学別冊)
(監修) 清水 厚志, 坊農 秀雅



次世代シークエンス解析スタンダード～NGSのポテンシャルを活かしきるWET&DRY (実験医学別冊)
(編集) 二階堂 愛



参考書籍

次世代シークエンサーDRY解析教本 (細胞工学別冊)

(監修) 清水 厚志, 坊農 秀雅

**次世代シークエンサー
DRY**

清水厚志／坊農秀雅

★★★★★ マストアイテム

投稿者 MOAOL 投稿日 2015/12/7

形式: 単行本 | [Amazonで購入](#)

この数年間、ネットを見ながらこつこつ勉強してきた基礎的なことが、この一冊に網羅されているのを拝見し、悔しいくらいの感動を覚えました。この値段は安いです。特筆すべきは、研究者では無い一般の方々によるNGS解析体験記があることです。ハードルの高さを感じて手を出せない方々に勇気をくれますし、著者らの「伝えたい」という熱意も感じられます。この本を参考に、いろいろ遊んでみたいです。

コメント | このレビューは参考になりましたか? はい いいえ [不正使用の報告](#)

★★★☆☆ 初心者には良い参考書ですが、動かないコードがあります。

投稿者 Amazon カスタマー 投稿日 2015/12/9

形式: 単行本

初心者には役に立つ参考書ですが、幾つかのツールは既にバージョンアップされ、本書に記載された通りのコードでは動きません。そういうコードを自分で新バージョンに対応するよう書き換えるスキルの有る人には、この本は必要ありません。

訂正。Lv1の② コマンドラインの使い方。p38 5行目のコマンドが...

誤) curl -L <http://gakken-mesh/dry/cmdline.tar.gz>

正) curl -L http://shujunsha.com/NGS_DAT/Lv1_2/cmdline.tar.gz

本実習で使用したツール

- 前処理 -

FastQC

<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>

Trimmomatic

<http://www.usadellab.org/cms/?page=trimmomatic>

- リファレンスゲノム配列を用いた発現解析 -

Bowtie v2.2.6

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

TopHat v2.1.0

<https://ccb.jhu.edu/software/tophat/index.shtml>

Cufflinks v2.2.1

<http://cole-trapnell-lab.github.io/cufflinks/>

SAMtools v1.2

<http://www.htslib.org/>

- 発現解析の結果の可視化 -

IGV v2.3.66

<https://www.broadinstitute.org/software/igv/home>

R v3.2.2

<https://www.r-project.org/>

RStudio Desktop 0.99.489

<https://www.rstudio.com/>

Rのパッケージ

CummeRbund (Bioconductor)

- de novo assemble法による発現解析 -

Bowtie v1.1.2

<http://bowtie-bio.sourceforge.net/index.shtml>

Trinity v2.1.1

<https://github.com/trinityrnaseq/trinityrnaseq/wiki>

RSEM v1.2.25

<http://deweylab.github.io/RSEM/>

NCBI BLAST+ v2.2.31

<ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>

Rのパッケージ

edgeR (Bioconductor)

- RNA-Seqデータによる変異解析 -

Bowtie v2.2.6

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

TopHat v2.1.0

<https://ccb.jhu.edu/software/tophat/index.shtml>

SAMtools v1.2

<http://www.htslib.org/>

Picard v1.140

<http://broadinstitute.github.io/picard/>

GATK v3.4-46

<https://www.broadinstitute.org/gatk/>

Tuxedo suite tools関連の論文

- ・ [アラインメントツールBowtie2について](#)

Ben Langmead & Steven L Salzberg (2012) **Nature Methods** “Fast gapped-read alignment with Bowtie 2”

- ・ [splice-awareアラインメントツールTopHat2について](#)

Daehwan Kim et al. (2013) **Genome Biology** “TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions”

- ・ [Cufflinksの論文](#)

Cole Trapnell et al. (2010) **Nature Biotechnology** “Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation”

- ・ [CufflinksのRABT assembly法について](#)

Adam Roberts et al. (2011) **Bioinformatics** “Identification of novel transcripts in annotated genomes using RNA-Seq”

- ・ [Cufflinksの遺伝子発現量推定における補正について](#)

Adam Roberts et al. (2011) **Genome Biology** “Improving RNA-Seq expression estimates by correcting for fragment bias”

- ・ [Cufflinksのcuffdiffについて](#)

Cole Trapnell et al. (2013) **Nature Biotechnology** “Differential analysis of gene regulation at transcript resolution with RNA-seq”

- ・ [Cufflinksのcuffdiff2について](#)

Cole Trapnell et al. (2014) **Nature Biotechnology** “The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells”

- ・ [TopHat-Cufflinks系の使い方のお作法](#)

Cole Trapnell et al. (2012) **Nature Protocol** “Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks”

その他のRNA-Seqデータ解析関連のツール

- QC -

RNA-SeQC

<http://www.broadinstitute.org/cancer/cga/rna-seqc>

- Splice-awareなアラインメントツール -

STAR

<https://code.google.com/p/rna-star/>

GSNAP

<http://www.molecularevolution.org/software/genomics/gmap>

様々なアラインメントツールを比較した論文

Pär G Engström et al. (2013) **Nature Methods** 10, 1185–1191
“Systematic evaluation of spliced alignment programs for RNA-seq data”

- 遺伝子構造予測プログラム -

AUGUSTUS

<http://bioinf.uni-greifswald.de/augustus/>

Scripture

<http://www.broadinstitute.org/software/scripture/>

StringTie

<https://ccb.jhu.edu/software/stringtie/>

様々な遺伝子構造予測ツールを比較した論文

Tamara Steijger et al. (2013) **Nature Methods** 10, 1177–1184
“Assessment of transcript reconstruction methods for RNA-seq”

- 発現比較解析 -

DESeq2

<http://bioconductor.org/packages/release/bioc/html/DESeq2.html>

baySeq

<http://www.bioconductor.org/packages/release/bioc/html/baySeq.html>

TCC

<http://www.bioconductor.org/packages/release/bioc/html/TCC.html>
様々なツールを比較した論文

Franck Rapaport et al. (2013) **Genome Biology** 14:R95
“Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data”

- De novo transcriptome assemble -

SOAPdenovo-Trans

<http://soap.genomics.org.cn/SOAPdenovo-Trans.html>

Oases

<https://www.ebi.ac.uk/~zerbino/oases/>

- De novo assembleによる発現解析 -

eXpress

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

kallisto

<http://pachterlab.github.io/kallisto/>