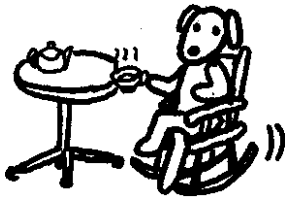
RNA-Seqデータ解析リテラシー

東京大学大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究ユニット
門田 幸二（かどた　こうじ）
http://www.iu.a.u-tokyo.ac.jp/~kadota/
kadota@iu.a.u-tokyo.ac.jp

2009年ごろの私

- 次世代シーケンサー（NGS）解析についての認識
  - 単に短い塩基配列が沢山あるだけでしょ
  - 得られる配列データって、multi-FASTA形式のもので、単にそれをリファレンス配列にマッピングしてカウントするだけでしょ
  - それ以降の解析はマイクロアレイと同じなんじゃないのー



- 私について
  - **マイクロアレイ**を中心としたデータ解析手法の開発
  - 主に遺伝子発現行列の**数値データのみ**を取り扱ってきた
  - 配列解析系のスキルはほぼゼロで、用語がまるでわかっていない
  - アグリバイオインフォマティクス教育研究プログラムの活動の一環でsmallRNAのNGS（Illumina）解析をやりはじめた
  - 自分の研究テーマとして主体的にやり始めたのは2011年～

取り扱うRNA-Seqデータの基本形式

```
@SRR037439.375
GCGGTGTGTTTGTGGTATAGTGGTGCCCCGCCCCG
+SRR037439.375
IIII&IIIII?223<(<I2B*4@#/I"#"#''""+
@SRR037439.376
CTCCCTGGAGGGACAGCAGAAATCTTTTGGGGCTG
+SRR037439.376
II*II"AI/II#I%IA-5II1?$#=G%I*#(""$3
@SRR037439.377
CAAGGGTACAAAGTTTTAGGTAAACAAGGGGAGTA
+SRR037439.377
IIIIIIIA4IIII3II"=I1I*II*I"I&I$#I$0
@SRR037439.378
GTGGCTTTTAAATTTTTCTTTTTTTTTTTTTTTAG
+SRR037439.378
IIIIIIII-*8E;III01G#"III?%A,(I:I#-%
@SRR037439.379
AGCCATTCTTATTCCAAACCCCCCGAAGCTTTCCC
+SRR037439.379
IIIIIII#2."(43$II0IIIII."C")&9(,-'2
@SRR037439.380
CGACTCCGAGCTGCCTACAAACCCGGGGGGAGGGG
+SRR037439.380
IIIIII>0II"II*I&,(+I7'"%'BB?99"I8BI
```

FASTQ形式?

Contents

- RNA-Seqデータ取得（マップする側）
  - 基本形式（FASTQ形式）
  - 公共データベースから取得する場合
  - クオリティのカットオフ
- マッピングに使うリファレンス配列（マップされる側）
  - ゲノム配列、（RefSeqのような）トランスクリプトーム配列
- リード数のカウントやデータの正規化（RPKM）
- 分布の話（ポアソン分布、負の二項分布）
- Rを使って二群間比較（発現変動遺伝子検出の流れ）
  - なぜ倍率変化（何倍発現が変化したかでの議論）がだめなのか

（自分のデータ解析の際に路頭に迷わなくてすむよう）
標準的なRNA-Seqデータ解析を一通り眺める

http://www.iu.a.u-tokyo.ac.jp/~kadota/r_seq.html

参考URL

(Rで)塩基配列解析(主に次世代シーケンサーのデータ) by 門田幸二 (last modified 2011/08/26)

What's new?
・2011年9月以降、次世代シーケンサー解析周辺の話をいくつかやります。初心者向けのが9/8と11/17, 私の最新の手法の話が9/29と11/11の予定です。定員に限りがあるようですので、詳細は私のホームページの「講演など」の項目をご覧ください。(2011/08/16)NEW
・アノテーション情報取得(BioMart and biomaRt)のところで配列長情報取得時の誤りに気づきましたm(_ _)m 2011年8月16日14:20までに一通り修正してあります。(2011/08/10-16)NEW
・FASTQ形式ファイル周辺の記述を追加しています。(2011/08/1-4)
・Bioconductorのリンク先をver. 2.7 --> 2.8に変更しました。(2011/07/20)
・R2.13.1がリリースされていたのでこれに変更しました。(2011/07/14)
・DEGseqパッケージ関連のパラメータ指定ミスを修正しました。具体的には例えば「expCol1=1」→「expCol1=2」でしたm(_ _)m(2011/06/09)

- はじめに (last modified 2011/07/19)
- Rのインストールと起動 (last modified 2011/08/18) NEW
- サンプルデータ (last modified 2011/02/03)
- イントロダクション | NGS | 各種覚書 (last modified 2010/12/10)
- イントロダクション | NGS | 様々なプラットフォーム (last modified 2011/07/15)
- イントロダクション | NGS | リファレンス配列取得(マップされる側) (last modified 2011/02/03)
- イントロダクション | NGS | リファレンス配列取得後の各種情報抽出(特にRefSeq) (last modified 2011/03/20)
- イントロダクション | NGS | リファレンス配列取得後の各種情報抽出2(readFASTA関数の利用) (last modified 2011/04/07)
- イントロダクション | NGS | アノテーション情報取得(refFlatファイル) (last modified 2010/12/07)
- イントロダクション | NGS | アノテーション情報取得(BioMart and biomaRt) (last modified 2011/08/26) NEW
- イントロダクション | 一般 | 配列取得 (last modified 2010/7/7)
- イントロダクション | 一般 | 指定した範囲の配列を取得 (last modified 2011/07/27)
- イントロダクション | 一般 | 翻訳配列(translate)を取得 (last modified 2011/07/27)
- イントロダクション | 一般 | 相補鎖(complement)を取得 (last modified 2011/07/27)
- イントロダクション | 一般 | 逆相補鎖(reverse complement)を取得 (last modified 2011/07/27)
- イントロダクション | 一般 | 逆鎖(reverse)を取得 (last modified 2011/07/27)
- イントロダクション | 一般 | 二連続塩基の出現頻度情報を取得 (last modified 2011/07/25)

FASTQ形式（とFASTA形式）

- FASTA形式
  - 「“>”ではじまる一行のdescription行」と「配列情報」からなる形式
  - NGSのread長は短いので、実質的に一つのリードを二行で表現

```
>SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
```

- FASTQ形式
  - 一行目：「“@”ではじまる一行のdescription行」
  - 二行目：「配列情報」
  - 三行目：「”+”からはじまる一行（のdescription行）」
  - 四行目：「クオリティ情報」

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*(((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

http://en.wikipedia.org/wiki/FASTQ_format

塩基配列のクオリティ情報といえば…

- Phredスコア
  - Phredというベースコールプログラムから得られるQuality Value（QV値）のこと

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90 %
20	1 in 100	99 %
30	1 in 1000	99.9 %
40	1 in 10000	99.99 %
50	1 in 100000	99.999 %

http://en.wikipedia.org/wiki/Phred_quality_score

なぜFASTQ形式では、Phredスコアそのものでクオリティ情報を表現しないの？

理由:(容量)節約のため

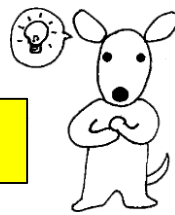
Phred スコア	ASCII印字可能文字		Phred スコア	ASCII印字可能文字	
0	!	ASCII 33	21	6	ASCII 54
1	"	ASCII 34	22	7	ASCII 55
2	#	ASCII 35	23	8	ASCII 56
3	$	ASCII 36	24	9	ASCII 57
4	%	ASCII 37	25	:	ASCII 58
5	&	ASCII 38	26	;	ASCII 59
6	'	ASCII 39	27	<	ASCII 60
7	(	ASCII 40	28	=	ASCII 61
8	)	ASCII 41	29	>	ASCII 62
9	*	ASCII 42	30	?	ASCII 63
10	+	ASCII 43	31	@	ASCII 64
11	,	ASCII 44	32	A	ASCII 65
12	-	ASCII 45	33	B	ASCII 66
13	.	ASCII 46	34	C	ASCII 67
14	/	ASCII 47	35	D	ASCII 68
15	0	ASCII 48	36	E	ASCII 69
16	1	ASCII 49	37	F	ASCII 70
17	2	ASCII 50	38	G	ASCII 71
18	3	ASCII 51	39	H	ASCII 72
19	4	ASCII 52	40	I	ASCII 73
20	5	ASCII 53	…	…	…

- FASTQ形式中のクオリティ情報部分

```
@SRR037439.375
GCGGTGTGTTTGTGGTATAGTGGTGCCCCGCCCCG
+SRR037439.375
IIII&IIIII?223<(<I2B*4@#/I"#"#''""+
```

- Phredスコア(QUAL形式)

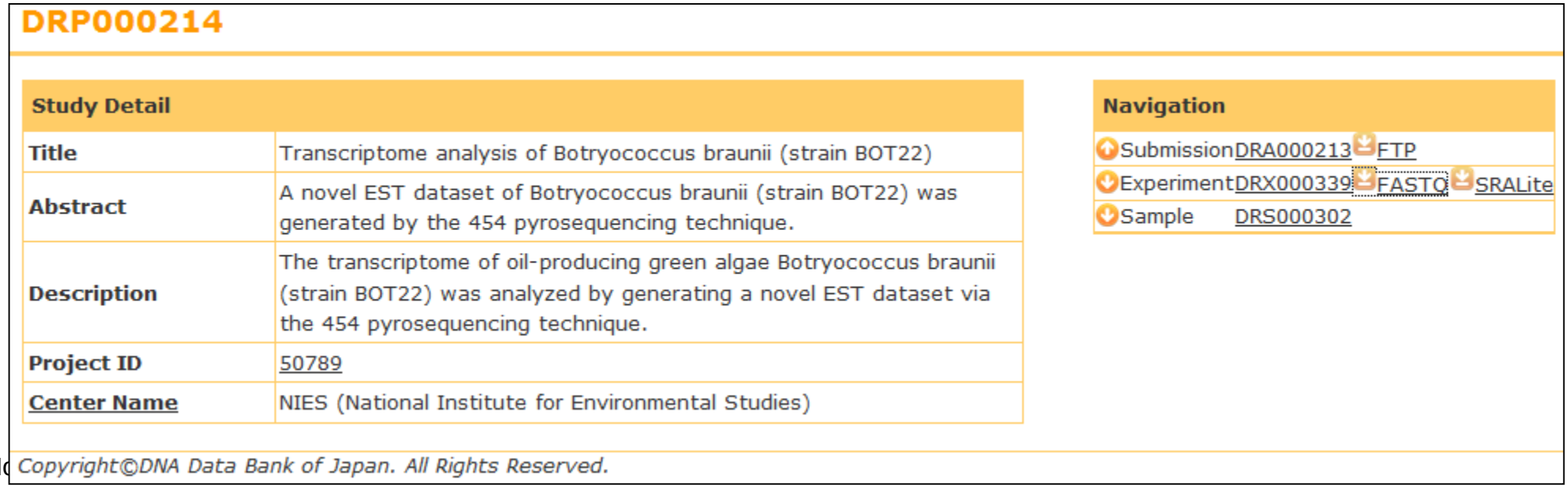
```
40 40 40 40  5 40 40 40 40 40 30 17 17 18 27  7 27 40 17 33
 9 19 31  2 14 40  1  2  1  2  6  6  1  1 10
```



Phredスコアが*X*の場合「ASCII (*X*+33)」に対応する文字コードを割り当てる

公共DBからデータを取得する場合

- ENA Sequence Read Archive (ERA; 欧)
  - FASTQ形式でダウンロード可能
- NCBI Sequence Read Archive (SRA; 米)
  - (sra形式と)sra-lite形式でダウンロード可能
- DDBJ Sequence Read Archive (DRA; 日)
  - FASTQ形式とsra-lite形式でダウンロード可能



DRP000214

Study Detail	
Title	Transcriptome analysis of Botryococcus braunii (strain BOT22)
Abstract	A novel EST dataset of Botryococcus braunii (strain BOT22) was generated by the 454 pyrosequencing technique.
Description	The transcriptome of oil-producing green algae Botryococcus braunii (strain BOT22) was analyzed by generating a novel EST dataset via the 454 pyrosequencing technique.
Project ID	50789
Center Name	NIES (National Institute for Environmental Studies)

Navigation			
Submission	DRA000213	FTP	
Experiment	DRX000339	FASTQ	SRALite
Sample	DRS000302		

sra.lite形式 → FASTQ形式

- *.lite.sraファイルがあるフォルダにおいて、Linuxシステム上で、以下のコマンドを実行（例：SRR002324.lite.sraファイル）
- 「`fastq-dump -A SRR002324 -D SRR002324.lite.sra`」

```
[kadota@hpcs02 SRR002324]$ ls
SRR002324.lite.sra
[kadota@hpcs02 SRR002324]$ fastq-dump -A SRR002324 -D SRR002324.lite.sra


[kadota@hpcs02 SRR002324]$
[kadota@hpcs02 SRR002324]$
[kadota@hpcs02 SRR002324]$ ls
SRR002324.fastq  SRR002324.lite.sra
[kadota@hpcs02 SRR002324]$ head SRR002324.fastq
@SRR002324.1 080317_CM-KID-LIV-2-REPEAT_0003:4:1:874:369 length=36
TGGAGCTTTTTCTTGTGTTTTAATTTTCTTATCAAC
+SRR002324.1 080317_CM-KID-LIV-2-REPEAT_0003:4:1:874:369 length=36
IIIIIIIIIIIIIIIIIIIII>IIIIIIIII.I)4I
@SRR002324.2 080317_CM-KID-LIV-2-REPEAT_0003:4:1:897:455 length=36
TGGGAGTATAGGGCTGTGACTAGTATGTTGAGTCCT
+SRR002324.2 080317_CM-KID-LIV-2-REPEAT_0003:4:1:897:455 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIBIIIIII
@SRR002324.3 080317_CM-KID-LIV-2-REPEAT_0003:4:1:890:253 length=36
TAATTTGGTTTCTCGGAAACCTTCCTTCAAGGCCTC
[kadota@hpcs02 SRR002324]$
```

13分程度かかる

Q & A

間違いが残ったままになっていたので変更しました（20130806）

- Q：なぜsra.lite形式で配布するんですか？
  - A：よくわかりません
- Q：Linuxが使えないとだめ…ってことですよね?!
  - A：（今のところ）そう…ですね。…しかも…それ以外の様々な局面でLinux環境での作業が必要…

NGS解析はLinux上で行うのが基本

様々なファイル形式…

前頁のお詫びに講義資料で用いていた
別のスライドを追加しました（20130806）

- 情報量：SRA-full ＞ SRA-lite ＞ FASTQ (＞ FASTA)
  - SRA-full：塩基配列、クオリティ情報、Intensity情報など画像以外の全て
  - SRA-lite：SRA-fullからIntensity情報を除いて軽量化したもの
  - FASTQ：塩基配列とクオリティ情報のみからなるもの
  - (FASTA：塩基配列のみからなるもの)
  - ファイルサイズ (SRA-full : SRA-lite : FASTQ : FASTA)
    - 6 : 3 : 2 : 1
    - 例：SRA-fullはFASTQの約3倍

FASTQ形式ファイルの利用が基本

様々な選択肢があります

- 自前で大容量メモリ計算サーバ(Linux)を購入し、必要なソフトのインストールからスタート

  特徴:難易度は高いが思い通りの解析が可能

- Linuxサーバをもつバイオインフォ系の人にお願いする

  特徴:気軽に頼める知り合いがいればいいが、その人次第

- DDBJ Read Annotation Pipelineを利用

  特徴:一番お手軽な選択肢だが、サポートしているプログラムのみ

DDBJ
DNA Data Bank of Japan

DDBJ Read Annotation Pipeline

>>English

DDBJ Read Annotation Pipelineは、次世代シーケンサ配列のクラウド型データ解析プラットフォームです。

LOGIN

新規アカウント作成 guest としてログイン

User ID:

Password:

Login

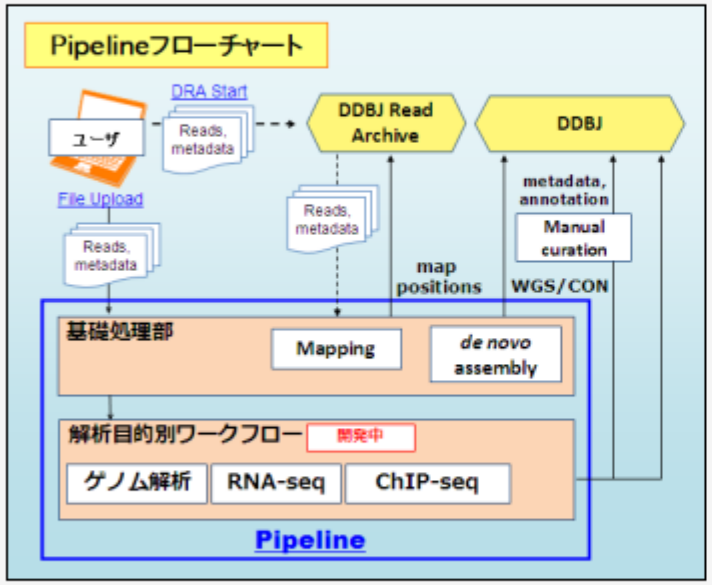
新規アカウントの作成は こちらです。

動作中JOBの確認

PipelineのIDをお持ちでない場合、
ゲストとしてログインすることができます。

User ID, Passwordの入力は、Query Fileの指定方法により異なります。

DRAへ登録したQueryファイルを使用する場合
DRA登録後にDRAのID、Passwordでログイン
Queryファイルをアップロードして使用する場合
新規アカウントを作成（DRAの登録なしでご利用になれます。）



マニュアルおよびチュートリアル

- 日本語マニュアル
- Manual(英語)
- FTPクライアント資料
- DBCLS 統合TV チュートリアル1 - 今日からはじめるDDBJ Read Annotation Pipeline

入力と出力のイメージ

- 入力データ

  1：解析したいRNA-Seqデータ（マップする側）

  2：マッピングに使うリファレンス配列（マップされる側）

- マッピングプログラム（bowtie, bwaなど）

  許容するミスマッチ数、複数個所にマップされるものの取り扱い、など多数のオプションが存在

- 出力結果（SAM/BAM形式、BED形式など）

  リファレンス配列中のどこにマップされたかという座標情報を返すのが基本形

  例1：4番染色体の78943-78977の領域（にマップされた）

  例2：Ensembl Gene ID XXX（のyyyy-zzzzの領域にマップされた）

Q & A

- Q:クオリティスコアでのフィルタリングは？
  - A(一般論):研究者の哲学次第。
  - A(私の思想):スコアが極端に低いものはFASTQファイルの段階ですでに"N"になっている → マップされる確率が低い → RNA-Seqの場合は特に気にする必要はないのでは…

- フィルタリング | NGS | quality scoreの評価1 (FASTQファイルを読み込んでPHREDスコア化した結果を眺める)(last modified 2011/08/16) NEW
- フィルタリング | NGS | quality scoreの評価2 (FASTQファイルを読み込んでPHREDスコアが低いリードを除去する)(last modified 2011/08/03)
- フィルタリング | NGS | quality scoreの評価3 (FASTQファイルを読み込んでPHREDスコアが低い塩基をNに置換する)(last modified 2011/08/04)

「(Rで)塩基配列解析」のウェブページ上でもPhredスコアの任意の閾値でフィルタリングするやり方を紹介しています

Q & A

http://hgdownload.cse.ucsc.edu/downloads.html

- Q:マッピングに使うリファレンス配列は？
  - A:好きなものを使ってください。ゲノム配列でもトランスクリプトーム配列でも結構です。
- Q:どこから取得できるんですか？
  - A:「UCSC Sequence and Annotation Downloads」などから取得できます（アノテーション情報も）。以下はほんの一例
    - ヒト全ゲノム配列の場合
      ftp://genome-ftp.cse.ucsc.edu/goldenPath/hg19/bigZips/hg19.2bit
    - ヒトトランスクリプトーム配列（RefSeq mRNA）の場合
      ftp://genome-ftp.cse.ucsc.edu/goldenPath/hg19/bigZips/refMrna.fa.gz
    - ヒトアノテーション情報の場合
      http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/refFlat.txt.gz

こんな感じ

ヒトゲノム配列

```
[kadota@hpcs02 hg19]$ head hg19.fasta
>chr1
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
[kadota@hpcs02 hg19]$
```

・1-22番染色体＋X＋Y
・約6200万行のファイル
・約3GBのサイズ

ヒトトランスクリプトーム配列

```
[kadota@hpcs02 h_rna]$ head h_rna.fasta
>NM_203348.1
CAGCCCAGGGCCGGGCTGCCCACCGTGCCAGCCCAGTTTCAATGACCCACCTGAGGGTTT
CCATCGTGGGCCAGGGGACCGGCGCAGGCGGGCATCCGGAGCCAGGCAGTGGCCAGCCCA
TCCCGGGCAGGGTCATAGGTGGGGCTGCAGGCCTCGAAGCCGCCATGTGAAGGCAGCAGC
GACCCCAGGCAGGGCGGGCCAGGTTGACCTTGCACCTGCTCTCCCCTCAGCCCCGCGGCC
ATGCCGACCCTGGTCGTGGGCACGCCGCCCACCTGCCTGGGGGACACACCTCAGCCCTGC
CACAAGAACAGCCAGAGGCAGGGCCCCTTCTCCCATGGGGCCCCAGGGAGAGCAGCCGAC
TGGAAGGCTGTTGCCAAGCCCAGGCTCTGCGCACCTGCAGCTGAGGATGACGTGGCAGCC
CTGAGGTGGCCCGGGCCCTCCCAGCAGCCAGACCCACCCTGGGCAGCTCCCCACGTGGTC
GGGTCTGACGACCTCAAGGAACCAGGCCCCTGGGGGAAGGCGTGCAGCCTGCCCATGTGG
[kadota@hpcs02 h_rna]$
```

・46,XXX mRNA sequences
ftp://ftp.ncbi.nih.gov/refseq/H_sapiens/mRNA_Prot/human.rna.fna.gz

アノテーション情報ファイル(refFlat形式)

symbol (A), name (B), 染色体名 (C), strand (D), 転写開始位置 (E), 転写終結位置 (F), CDS start (G), CDS end (H), Exon数 (I)

	A	B	C	D	E	F	G	H	I	J	K
1	SNAR-G2	NR_024244	chr19	−	49534925	49535044	49535044	49535044	1	49534925,	49535044,
2	SNAR-D	NR_024243	chr19	−	50643458	50643577	50643577	50643577	1	50643458,	50643577,
3	SNORD113-5	NR_003233	chr14	+	101404523	101404600	101404600	101404600	1	101404523,	101404600,
4	UBL5	NM_024292	chr19	+	9938567	9940797	9939013	9940684	5	9938567,9939002,9939267,9939512,9940640,	9938740,9939069,9

Q & A

- Q:ドラフトゲノム配列しかないんですけど…
  - A:マッピングの際のオプションで許容するミスマッチ数を増やすなどの措置をとることである程度の解析は可能だと思います。
- Q:手持ちのRNA-Seqデータしかないんですけど…
  - A:2010年頃から提供されはじめた*de novo* transcriptome assembly用のプログラム(TrinityやTrans-ABySSなど;もちろんLinux用です)を利用すればトランスクリプトームの配列セット(RefSeqのようなイメージ)を得ることができます。

入力:RNA-Seqデータ

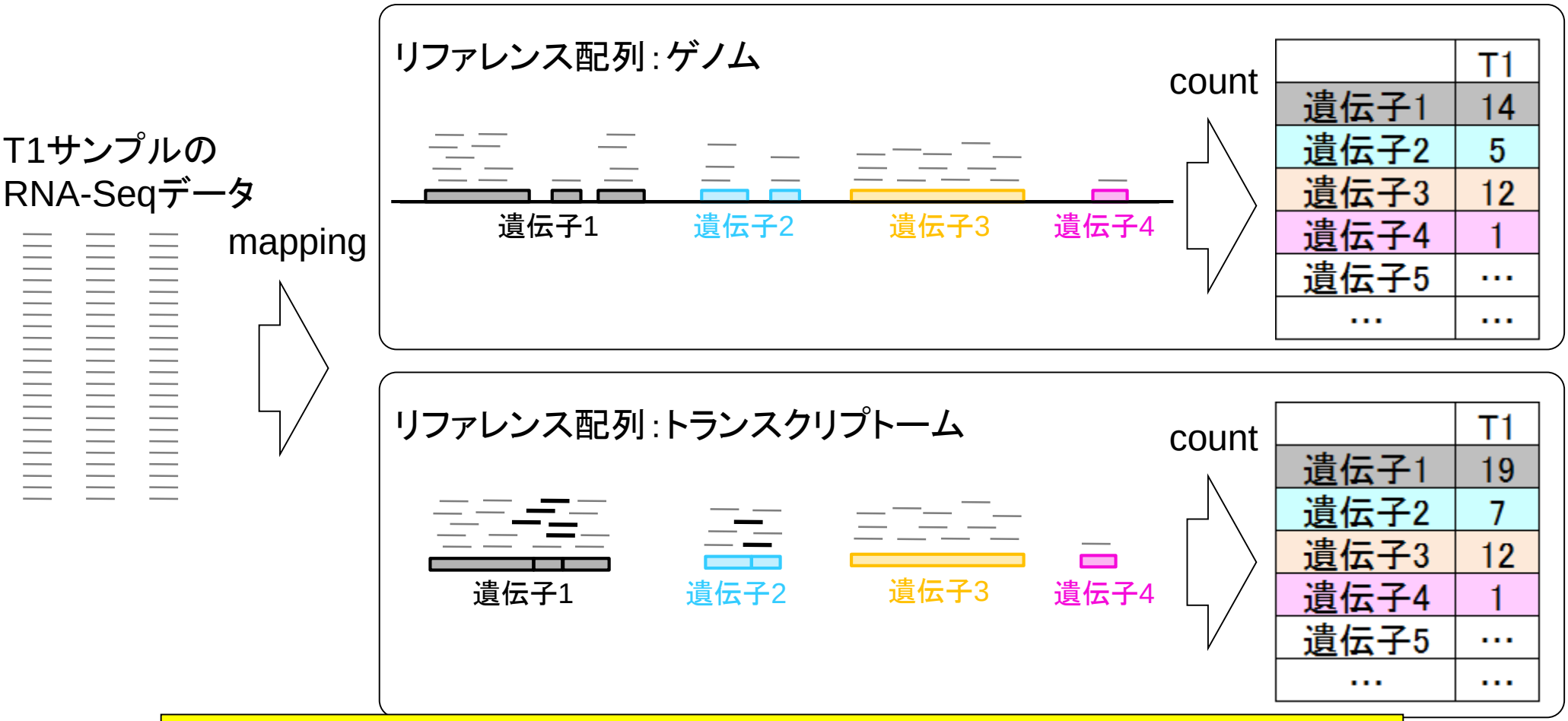
```
>read1
GGGGTTCAAAGCAGTATCGATCAAATAGTA
>read2
GTTCAAAGCAGTATCGATCAAATAGTAAAT
>read3
ACGATGCAGCCTTAACGATGGTCCACAATT
>read4
```

出力:コンティグ(≒転写物配列)

```
>contig1(transcript1)
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAA
CTCACAGTTTGGAGCTTATCAGTCAA…
>contig2(transcript2)
ACGATGCAGCCTTAACGATGGTCCACAATTATCGGGAATCA…
>contig3(transcript3)
…
```

マッピングの基本的なイメージ

- 基本的なマッピングプログラム（bowtieなど）を用いた場合

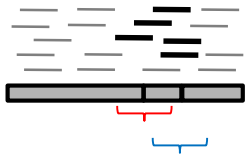


ゲノム配列へのマッピングの場合、複数のエクソンにまたがるリード（**spliced reads**）はマップされないのでは?!

対策（リードが短かったころ；<50bp）

- 既知のsplice junction周辺配列を予め組み込んだリファレンスゲノム配列側にマッピング

遺伝子1



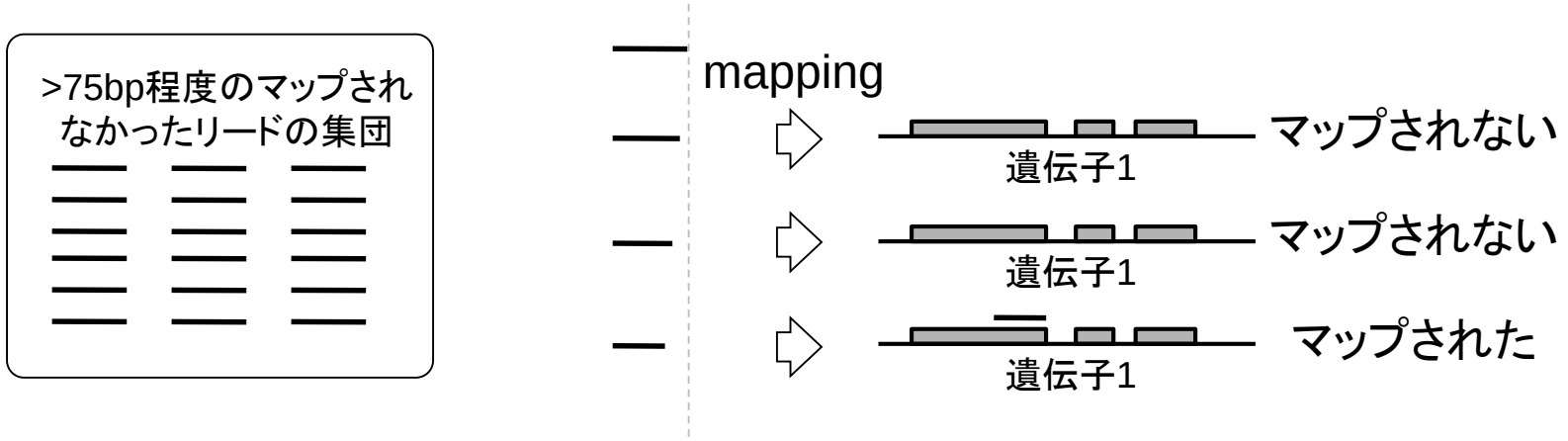
リファレンスゲノム配列への組み込み後のイメージ

```
>chr1
GGGGTTCAAAGCAGTATCGATCAAATAGTA
>chr2
GTTCAAAGCAGTATCGATCAAATAGTAAAT
…
>遺伝子1の「Exon1のend-20bp」から「exon2のstart+20bp」
ACGATGCAGCCTTAACGATGGTCCACAATT…
>遺伝子1の「Exon2のend-20bp」から「exon3のstart+20bp
```

（少なくとも）既知のexon間をまたぐリードのマッピングは可能

対策(ーリード>75bp程度の現在)

- 再帰的にマッピングする戦略(recursive mapping strategy)
  - (通常のマッピングプログラムでマップされなかったものに対して)リードを短くしてマップされるかどうか、を繰り返す



・(原理的に)未知のアイソフォームの発見も可能
～リード長などによっても戦略が異なる～

マッピング結果の出力ファイル形式

- (ゲノム配列の場合)どの染色体上のどの位置に(どのリードが)マッピングされたか、あるいは(トランスクリプトーム配列の場合)どの転写物配列上のどの位置に(どのリードが)マッピングされたかを表すファイル形式(フォーマット)は複数あります:
  - **BED** (Browser Extensible Data) format
    - BEDtools (Quinlan et al., *Bioinformatics*, **26**: 841-842, 2010)
  - GFF (General Feature Format) format
  - **SAM** (Sequence Alignment/Map) format
    - SAMtools (Li et al., *Bioinformatics*, **25**: 2078-2079, 2009)
  - …

マッピング結果ファイルから、どうやって転写物ごとのマップされたリード数をカウントするのか？

http://genome.ucsc.edu/FAQ/FAQformat.html#format1

BED形式

• イントロダクション | NGS | マッピング | (short) readの出力形式について

マッピング | (short) readを眺めると、いろいろな出力形式があることがわかります。
注目すべきは、Sequence Alignment/Map (SAM) formatです。この形式は国際共同研究の1000人のゲノムを解析するという1000 Genomes Projectで採用された(開発された)フォーマットで、("@"から始まる)header sectionと(そうでない)alignment sectionから構成されています。このヒトの目で解読可能な形式がSAMフォーマットで、このバイナリ版がBinary Alignment/Map (BAM)フォーマットというものです。今後SAM/BAM formatという記述をよく見かけるようになることでしょう

代表的な出力ファイル形式

- BED format
- ELAND format
- GFF (General Feature
- GFF3 (General Feat
- SAM (Sequence Alig
- SOAP format
- ZOOM format

UCSC Genome Bioinformatics

Home - Genomes - Blat - Tables - Gene Sorter - PCR - Proteome - Help

Frequently Asked Questions: Data File Formats

- BED format
- bigBed format
- BED
- PSL f
- GFF f
- GTF f

BED format

BED format provides a flexible way to define the data lines that are displayed in an annotation track. BED lines have three r
and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an ann
The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are use

The first three required BED fields are:

1. **chrom** - The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. **chromStart** - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is nu
3. **chromEnd** - The ending position of the feature in the chromosome or scaffold. The *chromEnd* base is not included in
   of the feature. For example, the first 100 bases of a chromosome are defined as *chromStart=0, chromEnd=100*, an
   bases numbered 0-99.

The 9 additional optional BED fields are:

BED形式

- あるトランスクリプトーム配列（RefSeq）にマップした結果

```
NM_001190702.1   235     271     U0      0       +
NM_024408.3      7718    7754    U0      0       +
NM_000110.3      2390    2426    U0      0       -
NR_002819.2      2753    2789    U0      0       +
NR_002819.2      2322    2358    U0      0       -
NR_003286.2      1359    1395    U0      0       -
NM_001190470.1   91      127     U0      0       +
NM_014918.4      1389    1425    U0      0       -
NM_002046.3      275     311     U0      0       -
NM_001419.2      1424    1460    U0      0       +
```

転写物ID　　Start　　End

転写物IDごとの出現数＝マップされたリード数

IDごとにマップされたリード数をカウント

- R上でBED形式ファイルを入力として下記スクリプトを実行

• 前処理 | トランスクリプトーム配列へのマップ後のファイルからマップされたリード数をカウント(BED形式ファイル)

手元に「RefSeqのhuman mRNAのmulti-fasta形式のファイル (h_rna.fasta; マップされる側の配列) に対して、参考文献1のKid
たNGSデータ(SRA ID: SRR002324; マップする側の配列)をBowtieプログラムを用いてマッ
す。
ここでは、各RefSeq IDに対してマップされたリード数をカウントした結果をファイルに保存す

「ファイル」－「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以

```
------    ここから    ------
in_f <- "SRR002324_t.bed"                          #読み込みたいBED形式
out_f <- "output.txt"                              #出力ファイル名を指定

data <- read.table(in_f)                           #in_fで指定したファイ
ID_list <- as.vector(data[,1])                     #行列dataから1列目の
out <- rle(sort(ID_list))                          #どのIDがいくつあった

tmp <- cbind(out$values, out$lengths)              #結果ファイルの1列目
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F, col.names=F)#tmpの中

------    ここまで    ------
```

	A	B
1	NM_000014.4	2198
2	NM_000015.2	1
3	NM_000016.4	8
4	NM_000017.2	157
5	NM_000018.2	46
6	NM_000019.3	1421
7	NM_000022.2	4
8	NM_000023.2	1
9	NM_000024.5	4
10	NM_000025.2	1
11	NM_000026.2	18
12	NM_000027.3	1
13	NM_000029.3	292

結果ファイル(output.txt)

Rを用いてコピペでマップされた
リード数情報を得ることができます

データ解析の前に…

- 研究目的(と手持ちのデータ)をおさらい
  - 一つのサンプル内でどの転写物(or 遺伝子)の発現レベルが高いか低いかを調べたい場合
    - RPKMやFPKMなどの「転写物の長さを考慮して正規化されたデータ」で解析
    - トータルのリード数を補正する必要はないがやってもよい
      - 遺伝子間の発現レベルの大小関係を調べたいだけなので、解析データを定数倍したところで何ら影響を与えないから…
  - サンプル間比較(sample A vs. Bなど)で、発現変動遺伝子(Differentially Expressed Genes; DEGs)を調べたい場合
    - 「トータルのリード数を補正したデータ」で解析
      - 正確には、「サンプル間で発現変動していない遺伝子(non-DEGs)ができるだけ発現変動していないと判定されるように正規化したデータ」
    - 既存のRパッケージを用いて解析を行う場合には、「(整数値のみからなる)生のリードカウントデータ」を入力とし、内部的に上記正規化を行う。

研究目的によってやっていい正規化とやってはいけない(と言われている)正規化がある

正規化（マップされたリード数 → 発現レベル）

- 基本的な考え
  - サンプル間の総リード数の違いをいかに補正するか
  - 配列長由来の偏り（長いほど沢山sequenceされる）をいかに補正するか
    （長さの異なる複数のisoformsが存在する場合にその遺伝子の配列長をいかに定義するか）

- RPKM (Mortazavi et al., *Nat Methods*, 2008; **ERANGEの論文**)
  - Reads per kilobase of exon per million mapped reads
- NAC (Griffith et al., *Nat Methods*, 2010; **ALEXA-seqの論文**)
  - Normalized average coverage

本質的に同じ

- FPKM (Trapnell et al., *Nat Biotechnol*., 2010; **Cufflinksの論文**)
  - Fragments per kilobase of transcript per million mapped fragments
- FVKM (Lee et al., *Nucleic Acids Res*., 2010; **NEUMAの論文**)
  - Fragments per virtual kilobase per million mapped reads

Multiple isoforms

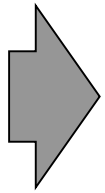
…

マイクロアレイデータの正規化

- 「各サンプルから測定されたシグナル強度の和は一定」と仮定
  - チップ上の遺伝子数が少ない場合は非現実的だが、数千～数万種類の遺伝子が搭載されているので妥当（だろう）

	sample1	sample2
gene1	10.5	12.4
gene2	6.4	7.1
gene3	8.0	8.5
gene4	10.8	11.4
gene5	5.6	6.7
gene6	8.4	8.9
gene7	6.2	7.0
gene8	6.1	6.8
gene9	6.6	6.5
gene10	5.1	5.8
総和	73.7	81.1

グローバル正規化

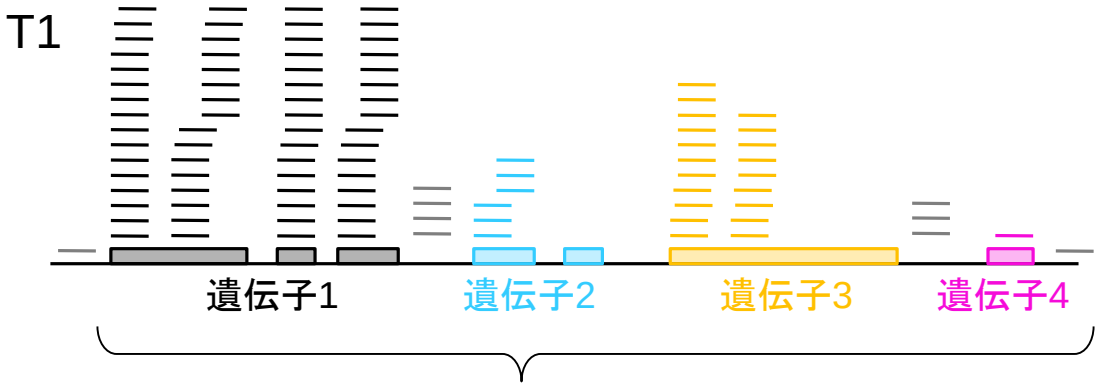


	sample1	sample2
gene1	14.2	15.3
gene2	8.7	8.8
gene3	10.9	10.5
gene4	14.7	14.1
gene5	7.6	8.3
gene6	11.4	11.0
gene7	8.4	8.6
gene8	8.3	8.4
gene9	9.0	8.0
gene10	6.9	7.2
総和	100.0	100.0

背景：サンプル（or chip）ごとにシグナル強度の総和は異なる
対策：総和が任意の値（例では100）になるような正規化係数を掛ける
例：sample1の正規化係数= 100 / **73.7**

RNA-Seqデータの正規化（の一部）

- 「各サンプルからsequenceされた総リード数は一定」と仮定



	T1	T2
遺伝子1	40	7
遺伝子2	6	15
遺伝子3	20	5
遺伝子4	1	1
総リード数	67	28

RPM正規化

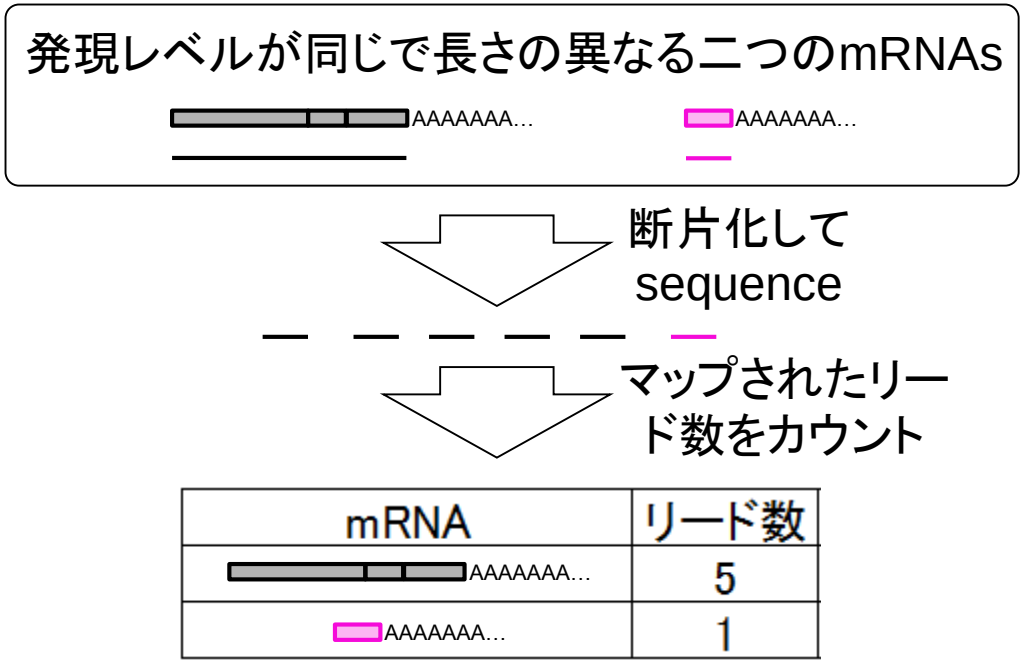
	T1	T2
遺伝子1	597014.9	250000.0
遺伝子2	89552.2	535714.3
遺伝子3	298507.5	178571.4
遺伝子4	14925.4	35714.3
総リード数	1000000	1000000

Reads Per Million mapped reads（RPM）
正規化後の総リード数が100万（one million）になるように補正
例：T1の正規化係数 = 1000000 / **67**

配列長の補正

Mortazavi et al., *Nature Methods*, **5**: 621-628, 2008

- 配列長が長い遺伝子ほど沢山sequenceされる
  - それらの遺伝子上にマップされる生のリード数が増加傾向
  - 配列長が長い遺伝子ほど発現レベルが高い傾向になる

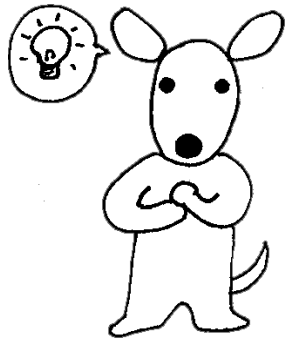


配列長の補正

Mortazavi et al., *Nature Methods*, **5**: 621-628, 2008

mRNA	リード数	配列長（in bp）
 AAAAAAA…	5	1500
 AAAAAAA…	1	300

- 前提条件：配列長が既知
- 補正の基本戦略：配列長で割る
  - 「1 / 配列長」を掛ける場合
    → 「塩基あたりの平均のリード数」を計算しているのと等価
  - 「1000 / 配列長」を掛ける場合
    → 「その遺伝子の配列長が1000bpだったときのリード数」と等価
    Reads Per Kilobase of exon



RPKM

Mortazavi et al., *Nature Methods*, **5**: 621-628, 2008

- RPM正規化（マイクロアレイなどと同じところ）
  - Reads per million mapped reads
  - サンプルごとにマップされた総リード（塩基配列）数が異なる。
    →各遺伝子のマップされたリード数を「総read数が100万（one million）だった場合」に補正

「raw counts : all reads= RPM : 1,000,000 」
A1BGの場合は「744 : 5,087,097 = RPM : 1,000,000」

$$\text{RPM} = \text{raw counts} \times \frac{1{,}000{,}000}{\text{all reads}} = 744 \times \frac{1{,}000{,}000}{5{,}087{,}097} = 146.3$$

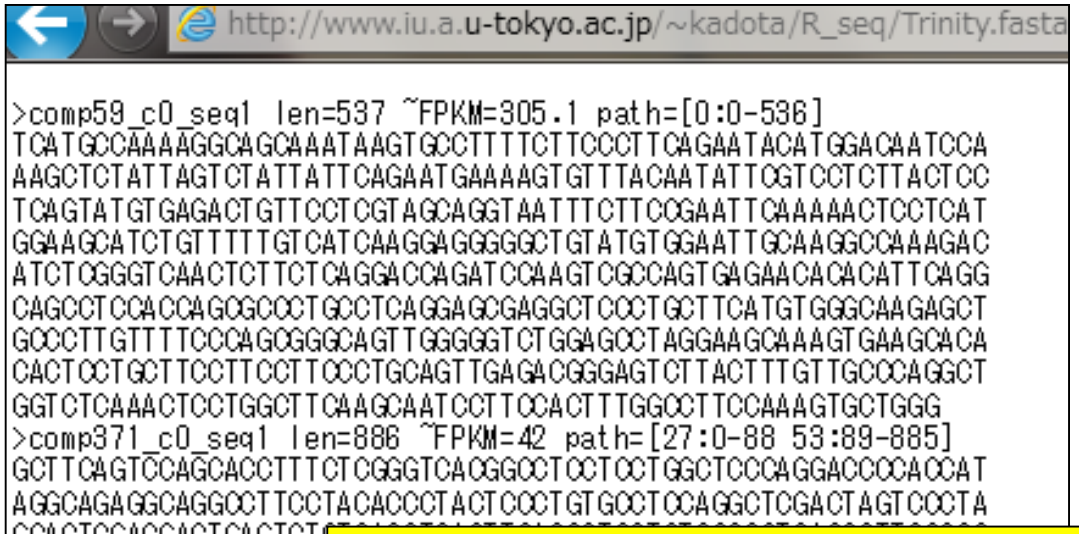
geneName	raw counts	RPKM	all reads	gene length	RPM
A1BG	744	82.9	5087097	1764	146.3
A1CF	159	13.7	5087097	2278	31.3
A2BP1	1	0.0	5087097	5415	0.2
A2LD1	4	0.6	5087097	1226	0.8

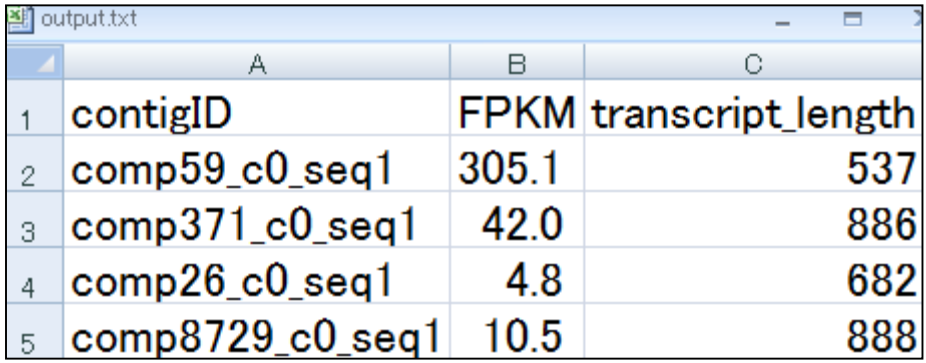
- RPKM正規化（RNA-Seq特有）
  - Reads per kilobase of exon per million mapped reads
  - 遺伝子の配列長が長いほど配列決定(sequence)される確率が上昇
    →各遺伝子の配列長を「1000塩基（one kilobase）の長さだった場合」に補正

$$\text{RPKM} = \text{raw counts} \times \underbrace{\frac{1{,}000{,}000}{\text{all reads}}}_{\text{RPM}} \times \frac{1{,}000}{\text{gene length}} = \text{raw counts} \times \frac{1{,}000{,}000{,}000}{\text{gene length} \times \text{all reads}}$$

Trinity出力結果からFPKM値を取得

- Trinity (Grabherr et al., *Nat Biotechnol*., **29**: 644-652, 2011)
  - RNA-Seqデータを入力としてトランスクリプトームのアセンブリを行うプログラム
  - 出力はmulti-fasta形式のファイル（ファイル名 : Trinity.fasta）
    - 転写物（コンティグ）の塩基配列情報
    - description行に配列長や発現レベルに相当するFPKM値が含まれる





Rを使って簡単にFPKM値の情報を取得することができます

利用可能なRパッケージたち

- *DEGseq* (Wang et al., *Bioinformatics*, **26**: 136-138, 2010)
  - ポワソン分布（variance = mean）を仮定しているためばらつきを過少評価
- *edgeR* (Robinson et al., *Bioinformatics*, **26**: 139-140, 2010)
  - 正規化法：TMM法
  - 負の二項分布（variance > mean）を仮定。meanのみのパラメータを用いて現実のばらつきを表現
- *DESeq* (Anders and Huber, *Genome Biol.*, **11**: R106, 2010)
  - 正規化法：RLE法(relative log expression)
  - edgeRのモデルをさらに拡張（しているらしい）
- *baySeq* (Hardcastle and Kelly, *BMC Bioinformatics*, **11**:422, 2010)
  - 正規化法：RPM (たぶん)
  - 配列の長さ情報を与えるオプションがあ
  - **データセット中に占めるDEGの割合（P_{DEG}）を一意に返す**
- *NBPSeq* (Di et al., *SAGMB*, **10**:24, 2011)

入力：生のリードカウントからなる遺伝子発現行列
出力：遺伝子ごとの発現変動の度合い（p値など）

理想的な実験デザイン(二群間比較)

- サンプルA vs. Bの比較(Kidney vs. Liver; 腎臓 vs. 肝臓)
  - 生のリードカウントのデータ(整数値)

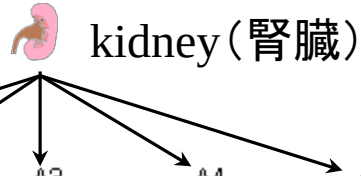
Gene ID	A1	A2	A3	A4	…	B1	B2	B3	B4	…
Gene1										
Gene2										
Gene3										
Gene4										
Gene5										
Gene6										
Gene7										
…										

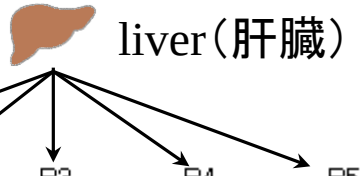
A1: ある生物の**腎臓**
A2: 同じ生物種の別個体の**腎臓**
A3: 同じ生物種のさらに別個体の**腎臓**
…
B1: ある生物の**肝臓**
B2: 同じ生物種の別個体の**肝臓**
…

Biological replicatesのデータ
生物学的なばらつき(個体間の違い)を考慮すべし

分布の話

- 例題：Marioni et al., *Genome Res.*, **18**: 1509-1517, 2008のデータ（の一部）





	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5
EnsemblGeneID	R1 L1 Kidney	R1 L3 Kidney	R1 L7 Kidney	R2 L2 Kidney	R2 L6 Kidney	R1 L2 Liver	R1 L4 Liver	R1 L6 Liver	R1 L8 Liver	R2 L3 Liver
ENSG00000146556	0	0	0	0	0	0	0	0	0	0
ENSG00000197194										
ENSG00000197490										
ENSG00000205292										
ENSG00000177693										
ENSG00000209338										
ENSG00000196573										
ENSG00000177799										
ENSG00000209341										
ENSG00000209342										
ENSG00000209343										
ENSG00000209344										
ENSG00000209346										
ENSG00000209349										
ENSG00000209350	4	7	3	6	7	35	32	31	29	34
ENSG00000209351	0	0	0	0	0	0	0	0	0	0
ENSG00000209352	0	0	1	1	0	2	0	0	0	0
ENSG00000212679	110	131	149	112	118	177	135	141	148	145
ENSG00000212678	12685	13204	12403	13031	13268	9246	9312	8746	8496	9070
ENSG00000185097	0	0	0	0	0	0	0	0	0	0

Technical replicatesのデータ

サンプル内の技術的なばらつき（例：レーン間の違い）の度合いを調べるためのデータであり、このようなデータで二群間比較し、**発現変動遺伝子がどの程度あるかといった数に関する議論は無意味**

解析例：アリエナイ?!数（50%とか）が発現変動遺伝子として検出される

理由：Biological variation > Technical variation

分布の話

- 例題：Marioni et al., *Genome Res.*, **18**: 1509-1517, 2008のデータ（の一部）

kidney（腎臓）

EnsemblGeneID	A1	A2	A3	A4	A5
ENSG00000146556	0	0	0	0	0
ENSG00000197194	0	0	0	0	0
ENSG00000197490	0	0	0	0	0
ENSG00000205292	0	0	0	0	0
ENSG00000177693	0	0	0	0	0
ENSG00000209338	0	0	0	0	0
ENSG00000196573	0	0	0	0	0
ENSG00000177799	0	0	0	0	0
ENSG00000209341	0	0	0	0	0
ENSG00000209342	0	0	2	4	3
ENSG00000209343	0	0	0	0	0
ENSG00000209344	0	0	0	0	0
ENSG00000209346	0	0	0	0	0
ENSG00000209349	0	0	0	0	0
ENSG00000209350	4	7	3	6	7
ENSG00000209351	0	0	0	0	0
ENSG00000209352	0	0	1	1	0
ENSG00000212679	110	131	149	112	118
ENSG00000212678	12685	13204	12403	13031	13268
ENSG00000185097	0	0	0	0	0
…	…	…	…	…	…
総リード数	1804977	1855190	1742426	1927517	1963420

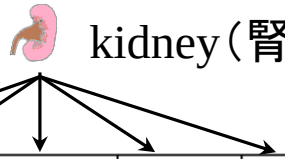
RPM
正規化

EnsemblGeneID	A1	A2	A3	A4	A5
ENSG00000209342	0.0	0.0	1.1	2.1	1.5
ENSG00000209350	2.2	3.8	1.7	3.1	3.6
ENSG00000209352	0.0	0.0	0.6	0.5	0.0
ENSG00000212679	60.9	70.6	85.5	58.1	60.1
ENSG00000212678	7027.8	7117.3	7118.2	6760.5	6757.6
ENSG00000197049	0.0	0.0	0.0	0.5	0.0
ENSG00000177757	1.1	0.0	1.1	0.5	1.5
ENSG00000177750	0.6	2.2	1.7	1.6	3.6
ENSG00000177741	0.6	0.5	0.0	3.1	0.0
ENSG00000198907	3.3	0.0	3.4	1.0	0.0
ENSG00000187634	27.1	23.2	23.5	21.8	23.9
ENSG00000188976	40.4	41.5	39.0	36.3	41.8
ENSG00000187961	8.3	8.1	7.5	6.2	7.6
ENSG00000187583	0.6	0.5	1.7	0.0	1.5
ENSG00000187642	2.2	2.7	6.9	4.7	4.6
ENSG00000188290	5.0	5.4	6.9	5.2	6.6
ENSG00000187608	6.6	5.9	4.0	8.3	6.6
ENSG00000188157	227.1	223.2	200.9	239.7	240.4
ENSG00000131591	5.5	4.9	4.0	6.2	8.1
ENSG00000215916	5.5	4.9	4.6	6.7	8.7
…	…	…	…	…	…
総リード数	1000000	1000000	1000000	1000000	1000000

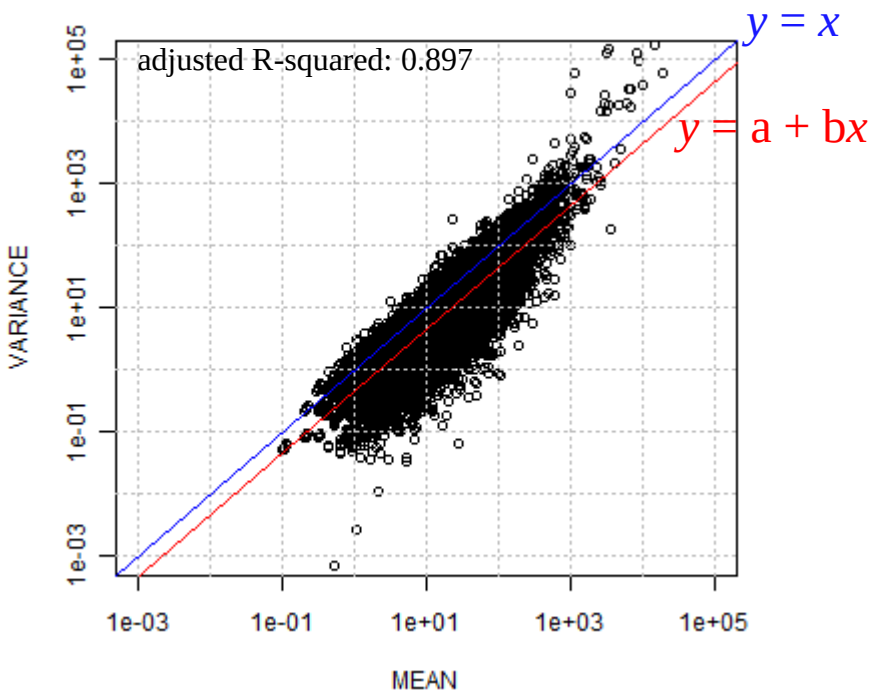
$$12{,}685 \times \frac{1{,}000{,}000}{1{,}804{,}977} = 7027.8$$

分布の話

- 例題：Marioni et al., *Genome Res.*, **18**: 1509-1517, 2008のデータ（の一部）



EnsemblGeneID	A1	A2	A3	A4	A5	MEAN	VARIANCE
ENSG00000209342	0.0	0.0	1.1	2.1	1.5	1.0	0.9
ENSG00000209350	2.2	3.8	1.7	3.1	3.6	2.9	0.8
ENSG00000209352	0.0	0.0	0.6	0.5	0.0	0.2	0.1
ENSG00000212679	60.9	70.6	85.5	58.1	60.1	67.1	129.8
ENSG00000212678	7027.8	7117.3	7118.2	6760.5	6757.6	6956.3	33770.4
ENSG00000197049	0.0	0.0	0.0	0.5	0.0	0.1	0.1
ENSG00000177757	1.1	0.0	1.1	0.5	1.5	0.9	0.4
ENSG00000177750	0.6	2.2	1.7	1.6	3.6	1.9	1.2
ENSG00000177741	0.6	0.5	0.0	3.1	0.0	0.8	1.7
ENSG00000198907	3.3	0.0	3.4	1.0	0.0	1.6	2.9
ENSG00000187634	27.1	23.2	23.5	21.8	23.9	23.9	3.9
ENSG00000188976	40.4	41.5	39.0	36.3	41.8	39.8	5.0
ENSG00000187961	8.3	8.1	7.5	6.2	7.6	7.5	0.7
ENSG00000187583	0.6	0.5	1.7	0.0	1.5	0.9	0.5
ENSG00000187642	2.2	2.7	6.9	4.7	4.6	4.2	3.4
ENSG00000188290	5.0	5.4	6.9	5.2	6.6	5.8	0.8
ENSG00000187608	6.6	5.9	4.0	8.3	6.6	6.3	2.4
ENSG00000188157	227.1	223.2	200.9	239.7	240.4	226.3	258.8
ENSG00000131591	5.5	4.9	4.0	6.2	8.1	5.8	2.5
ENSG00000215916	5.5	4.9	4.6	6.7	8.7	6.1	2.8
…	…	…	…	…	…	…	…
総リード数	1000000	1000000	1000000	1000000	1000000		

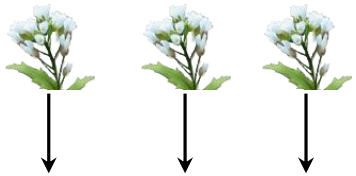


Technical replicatesのデータは：
・（遺伝子の）VARIANCEはそのMEANで説明可能である
・VARIANCE ≒ MEAN
・ポアソン分布に従う
・ポアソンモデルが適用可能

分布の話

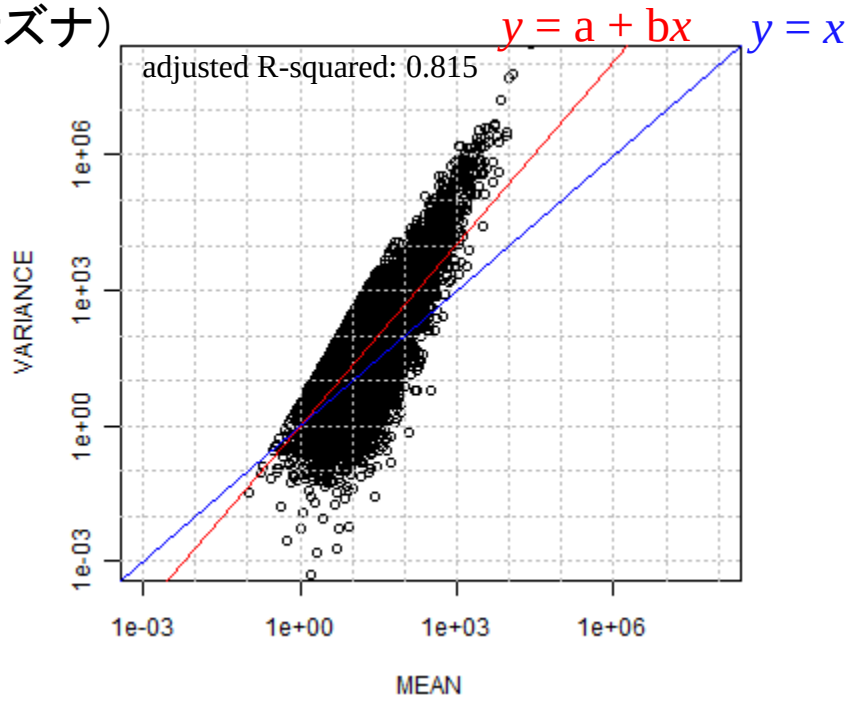
生物アイコン(http://biosciencedbc.jp/taxonomy_icon/taxonomy_icon.cgi)

- 例題:Cumbie et al., *PLoS ONE*, **6**: e25279, 2011のデータ(の一部)



Arabidopsis(シロイヌナズナ)

	mock1	mock2	mock3	MEAN	VARIANCE
AT1G01010	18.4	39.8	12.3	23.5	209.1
AT1G01020	22.6	23.3	9.8	18.6	57.5
AT1G01030	8.4	12.4	8.0	9.6	6.0
AT1G01040	37.9	22.2	19.6	26.6	97.1
AT1G01050	25.8	40.3	27.6	31.2	62.9
AT1G01060	0.0	7.8	0.6	2.8	18.6
AT1G01070	8.4	17.6	1.8	9.3	62.5
AT1G01080	89.4	98.8	117.2	101.8	200.2
AT1G01090	153.0	178.9	172.7	168.2	183.1
AT1G01100	59.4	64.6	75.5	66.5	67.1
AT1G01110	0.0	0.5	0.3	0.3	0.1
AT1G01120	119.9	97.7	82.8	100.1	347.3
AT1G01130	4.7	5.7	0.3	3.6	8.2
AT1G01140	95.2	62.0	43.6	66.9	683.3
…	…	…	…	…	…
総リード数	1000000	1000000	1000000		



Biological replicatesのデータは:
- **VARIANCE > MEAN**
- 負の二項(NB)分布に従う
- NBモデルが適用可能

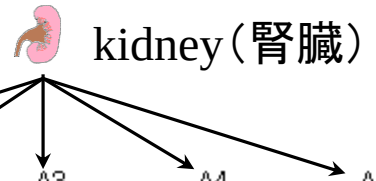
なぜ沢山の方法が存在しうるのか？

- *DEGseq* (Wang et al., *Bioinformatics*, **26**: 136-138, 2010) $\mathrm{VAR} = \mu$
  - ポワソン分布（variance = mean）を仮定しているためばらつきを過少評価
- ***edgeR*** (Robinson et al., *Bioinformatics*, **26**: 139-140, 2010) $\mathrm{VAR} = \mu(1+\phi\mu)$
  - 正規化法：TMM法
  - 負の二項分布（variance > mean）を仮定。
- *DESeq* (Anders and Huber, *Genome Biol.*, **11**: R106, 2010) $\mathrm{VAR} = \mu(1+\phi_\mu\mu)$
  - 正規化法：RLE法(relative log expression)
  - edgeRのモデルをさらに拡張（しているらしい）
- ***baySeq*** (Hardcastle and Kelly, *BMC Bioinformatics*, **11**:422, 2010)
  - 正規化法：RPM (たぶん)
  - 配列の長さ情報を与えるオプションがある
  - データセット中に占めるDEGの割合（P_{DEG}）を一意に返す
- *NBPSeq* (Di et al., *SAGMB*, **10**:24, 2011) $\mathrm{VAR} = \mu(1+\phi\mu^{\alpha-1})$

Ans. VarianceとMeanの関係を表現する手段が沢山あるから

edgeRを使ってみる

- 例題：Marioni et al., *Genome Res.*, **18**: 1509-1517, 2008のデータ



	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5
EnsemblGeneID	R1 L1 Kidney	R1 L3 Kidney	R1 L7 Kidney	R2 L2 Kidney	R2 L6 Kidney	R1 L2 Liver	R1 L4 Liver	R1 L6 Liver	R1 L8 Liver	R2 L3 Liver
ENSG00000146556	0	0	0	0	0	0	0	0	0	0
ENSG00000197194	0	0	0	0	0	0	0	0	0	0
ENSG00000197490	0	0	0	0	0	0	0	0	0	0
ENSG00000205292	0	0	0	0	0	0	0	0	0	0
ENSG00000177693	0	0	0	0	0	0	0	0	0	0
ENSG00000209338	0	0	0	0	0	0	0	0	0	0
ENSG00000196573	0	0	0	0	0	0	0	0	0	0
ENSG00000177799	0	0	0	0	0	0	0	0	0	0
ENSG00000209341	0	0	0	0	0	0	0	0	0	0
ENSG00000209342	0	0	2	4	3	0	0	0	1	0
ENSG00000209343	0	0	0	0	0	0	0	0	0	0
ENSG00000209344	0	0	0	0	0	0	0	0	0	0
ENSG00000209346	0	0	0	0	0	0	0	0	0	0
ENSG00000209349	0	0	0	0	0	0	0	0	0	0
ENSG00000209350	4	7	3	6	7	35	32	31	29	34
ENSG00000209351	0	0	0	0	0	0	0	0	0	0
ENSG00000209352	0	0	1	1	0	2	0	0	0	0
ENSG00000212679	110	131	149	112	118	177	135	141	148	145
ENSG00000212678	12685	13204	12403	13031	13268	9246	9312	8746	8496	9070
ENSG00000185097									0	0

ファイル名：**SupplementaryTable2_changed.txt**
内容：A群が最初の**5**列、B群が残りの**5**列のデータ
解析結果を**hoge2.txt**という名前でファイルに出力したい

edgeRを使ってみる

• 解析 | NGS(RNA-seq) | 発現変動遺伝子 | 二群間 | edgeR (Robinson_2010)

参考文献1のedgeRパッケージを用いて解析を行
ものです。(おそらくedgeRの論文が受理された後
(normalization factor)をどのように組み込むかに
ベーションとなった偏りのあるデータセットでの計
ません。なぜなら極端でない場合はTMM法で得ら
入力ファイルは、"遺伝子発現行列"形式のもの
ここでは、サンプルデータ2 (つまりSupplementar

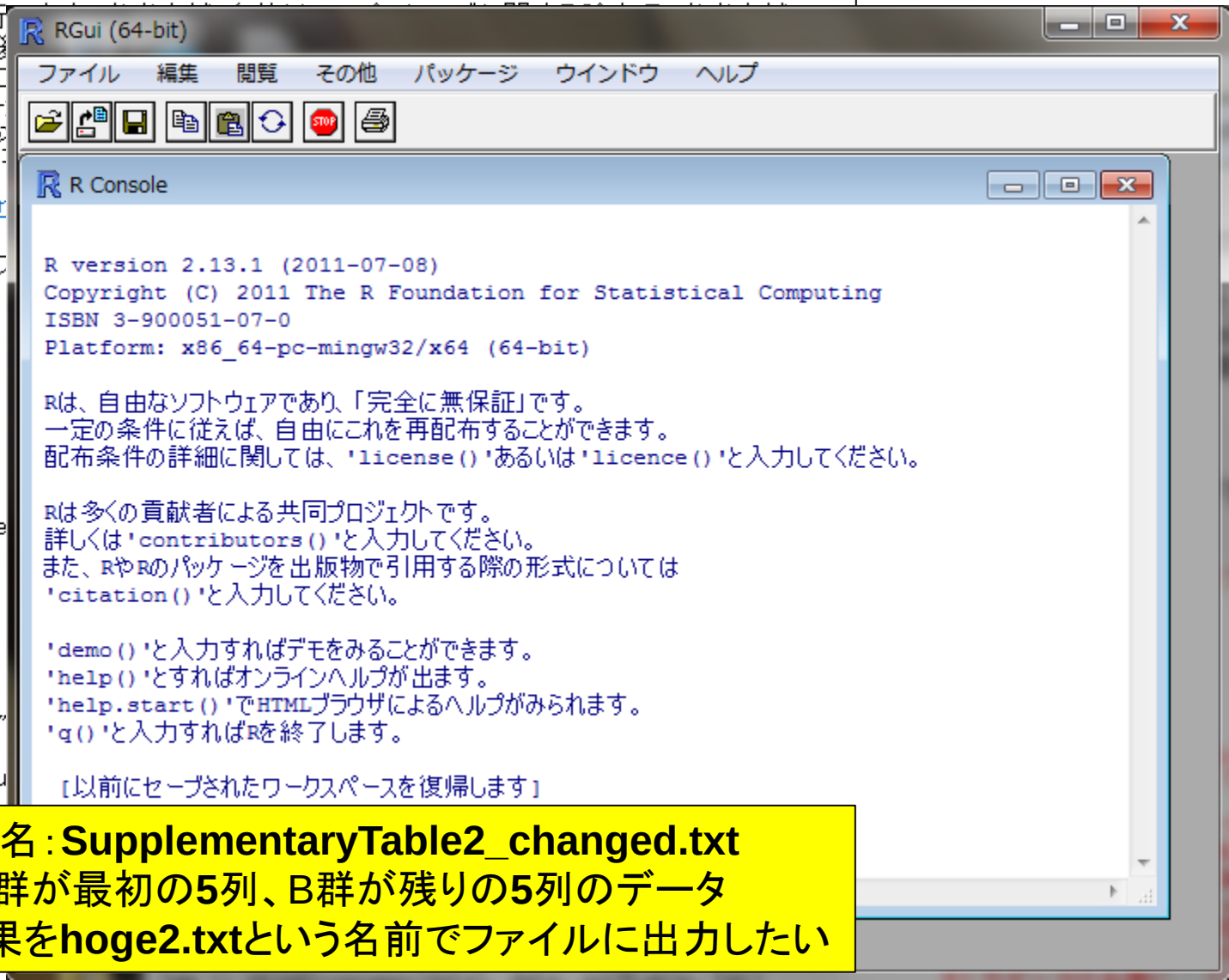
「ファイル」－「ディレクトリの変更」で解析したいフ

```
------    ここから    ------
in_f <- "SupplementaryTable2_changed.txt"
out_f <- "hoge2.txt"
param1 <- 5
param2 <- 5

library(DEGseq)
library(edgeR)
data <- read.table(in_f, header=TRUE, row.name
data <- as.matrix(data)

data.cl <- c(rep(1, param1), rep(2, param2))
d <- DGEList(counts=data, group=data.cl)
d <- calcNormFactors(d)
d <- estimateCommonDisp(d)
out <- exactTest(d)
fdr <- p.adjust(out$table$p.value, method="BH"
tmp <- cbind(rownames(data), data, out$table,
write.table(tmp, out_f, sep="¥t", append=F, qu

------    ここまで    ------
```



ファイル名：**SupplementaryTable2_changed.txt**
内容：A群が最初の**5**列、B群が残りの**5**列のデータ
解析結果を**hoge2.txt**という名前でファイルに出力したい

edgeRを使ってみる

• 解析 | NGS(RNA-seq) | 発現変動遺伝子 | 二群間 | edgeR (Robinson_2010)

参考文献1のedgeRパッケージを用いて解析を行います。参考文献1自体はRのパッケージに関する論文で、参考文献2-4の
ものです。(おそらくedgeRの論文が受理された後
(normalization factor)をどのように組み込むかに
ベーションとなった偏りのあるデータセットでの計
ません。なぜなら極端でない場合はTMM法で得ら
入力ファイルは、"遺伝子発現行列"形式のもの
ここでは、サンプルデータ2 (つまりSupplementar

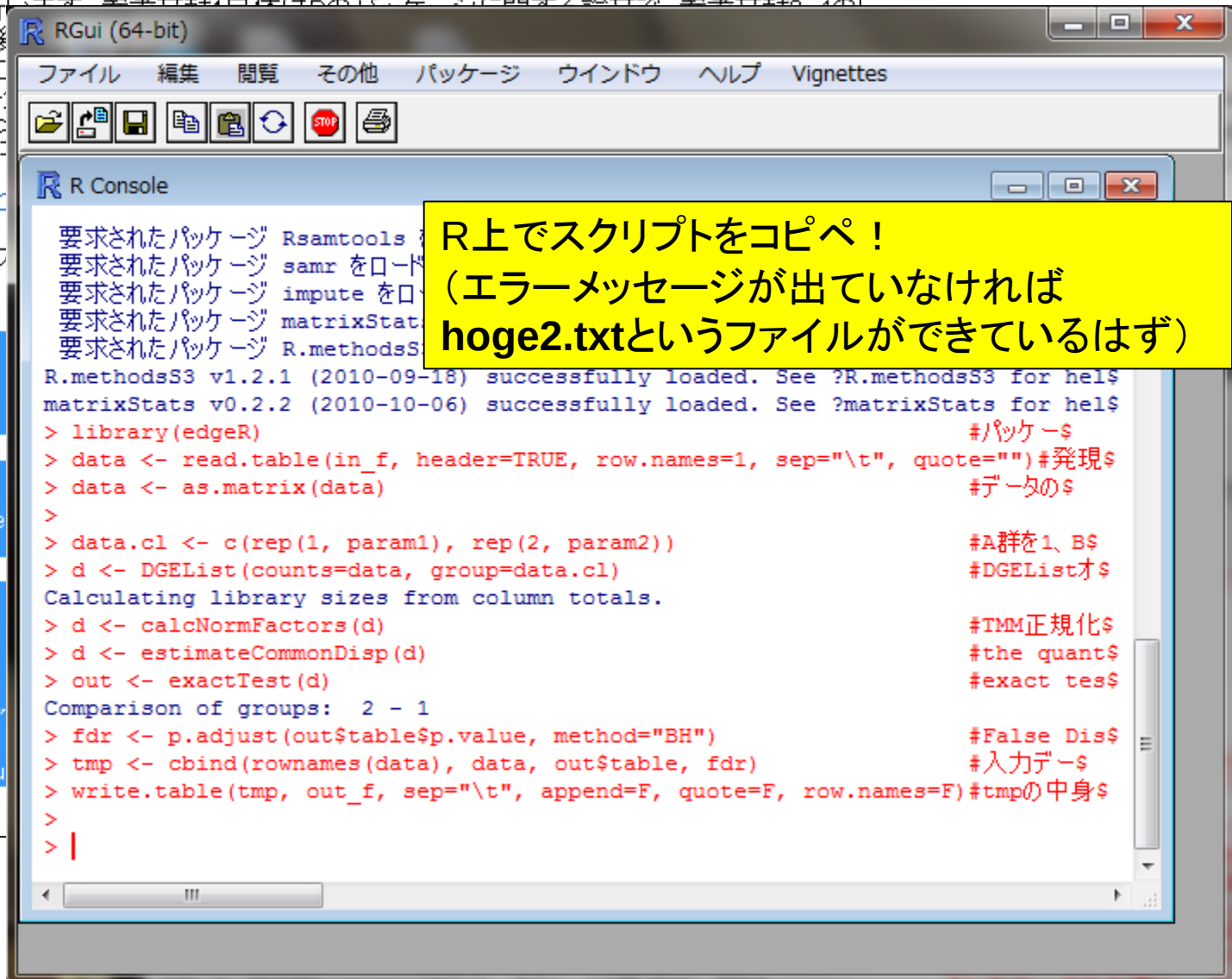
「ファイル」-「ディレクトリの変更」で解析したいフ

```
------    ここから    ------
in_f <- "SupplementaryTable2_changed.txt"
out_f <- "hoge2.txt"
param1 <- 5
param2 <- 5

library(DEGseq)
library(edgeR)
data <- read.table(in_f, header=TRUE, row.name
data <- as.matrix(data)

data.cl <- c(rep(1, param1), rep(2, param2))
d <- DGEList(counts=data, group=data.cl)
d <- calcNormFactors(d)
d <- estimateCommonDisp(d)
out <- exactTest(d)
fdr <- p.adjust(out$table$p.value, method="BH"
tmp <- cbind(rownames(data), data, out$table,
write.table(tmp, out_f, sep="¥t", append=F, qu

------    ここまで    ------
```



R上でスクリプトをコピペ！
（エラーメッセージが出ていなければ
hoge2.txtというファイルができているはず）

edgeRを使ってみる

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
rownames(data)	R1 L1 Kidn	R1 L3 Kidn	R1 L7 Kidn	R2 L2 Kidn	R2 L6 Kidn	R1 L2 Live	R1 L4 Live	R1 L6 Live	R1 L8 Live	R2 L3 Live	logConc	logFC	p.value	fdr
ENSG00000146556	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000197194	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000197490	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000205292	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000177693	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209338	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000196573	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000177799	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209341	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209342	0	0	2	4	3	0	0	0	1	0	-21.487	-2.4472	0.17983	0.40224
ENSG00000209343	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209344	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209346	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209349	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209350	4	7	3	6	7	35	32	31	29	34	-17.029	3.29875	1.13E-40	1.61E-39
ENSG00000209351	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209352	0	0	1	1	0	2	0	0	0	0	-22.072	0.72267	1	1
ENSG00000212679	110	131	149	112	118	177	135	141	148	145	-13.662	0.98919	7.82E-33	9.51E-32
ENSG00000212678	12685	13204	12403	13031	13268	9246	9312	8746	8496	9070	-7.3545	0.19636	2.63E-14	1.88E-13
ENSG00000185097	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1
ENSG00000209353	0	0	0	0	0	0	0	0	0	0	-50.016	0	1	1

一番右側の数値がFalse Discovery Rate (FDR)
この列（O列）で昇順にソートすれば任意の閾値を満たす遺伝子数がわかる
・9,862個がFDR < 0.01を満たす
・11,172個がFDR < 0.05を満たす

edgeRを使ってみる

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
rownames(data)	R1 L1 Kidn	R1 L3 Kidn	R1 L7 Kidn	R2 L2 Kidn	R2 L6 Kidn	R1 L2 Live	R1 L4 Live	R1 L6 Live	R1 L8 Live	R2 L3 Live	logConc	logFC	p.value	fdr
ENSG00000116285	115	144	115	143	153	1669	1753	1710	1675	1794	−11.842	4.40591	0	0
ENSG00000049239	183	232	179	207	199	838	822	814	773	895	−12.081	2.77294	0	0
ENSG00000186510	515	564	516	568	590	6	1	1	3	2	−15.508	−7.0035	0	0
ENSG00000184908	484	486	463	573	512	4	2	5	4	3	−15.338	−6.4052	0	0
ENSG00000142949	332	320	312	350	354	732	772	716	711	808	−11.786	1.88711	0	0
ENSG00000117472	572	614	603	624	688	14	17	15	13	16	−14.158	−4.647	0	0
ENSG00000162366	730	782	720	832	866	4	8	7	7	6	−14.602	−6.217	0	0
ENSG00000121310	229	223	247	228	239	1832	1805	1812	1693	1954	−11.402	3.68579	0	0
ENSG00000116171	542	568	545	548	601	1777	1800	1817	1663	1845	−10.785	2.38936	0	0
ENSG00000162391	435	444	414	455	450	5	2	5	6	7	−15.199	−5.7356	0	0
ENSG00000116133	632	681	622	733	702	3534	3396	3178	3196	3657	−10.188	3.05395	0	0
ENSG00000169174	10	8	8	7	13	223	230	221	173	219	−15.281	5.25688	0	0
ENSG00000157131	14	11	13	7	14	1352	1405	1400	1345	1402	−13.753	7.594	0	0
ENSG00000021852	10	12	11	4	20	968	1002	969	982	982	−14.025	7.15013	0	0
ENSG00000132855	82	96	86	76	90	822	874	823	821	885	−12.675	4.01933	0	0
ENSG00000079739	194	198	189	213	194	564	540	506	500	575	−12.402	2.16468	0	0
ENSG00000134215	521	526	514	476	559	14	10	15	7	7	−14.536	−4.8921	0	0
ENSG00000134243	919	875	849	883	937	86	93	77	75	94	−12.644	−2.6705	0	0
ENSG00000163399	7334	7494	6959	7702	7744	284	272	272	250	243	−10.295	−4.0942	0	0
ENSG00000134240	170	189	180	191	199	2161	2229	2166	2019	2393	−11.432	4.28375	0	0
ENSG00000168509	9	10	7	8	8	696	710	736	666	711	−14.485	7.11179	0	0

Top-ranked geneの生リードカウントを眺めても確かに発現変動 (Kidney << Liver)していることが分かる

edgeRを使ってみる

- M-A plotを描画（FDR < 0.01を満たすものを赤色で表示）

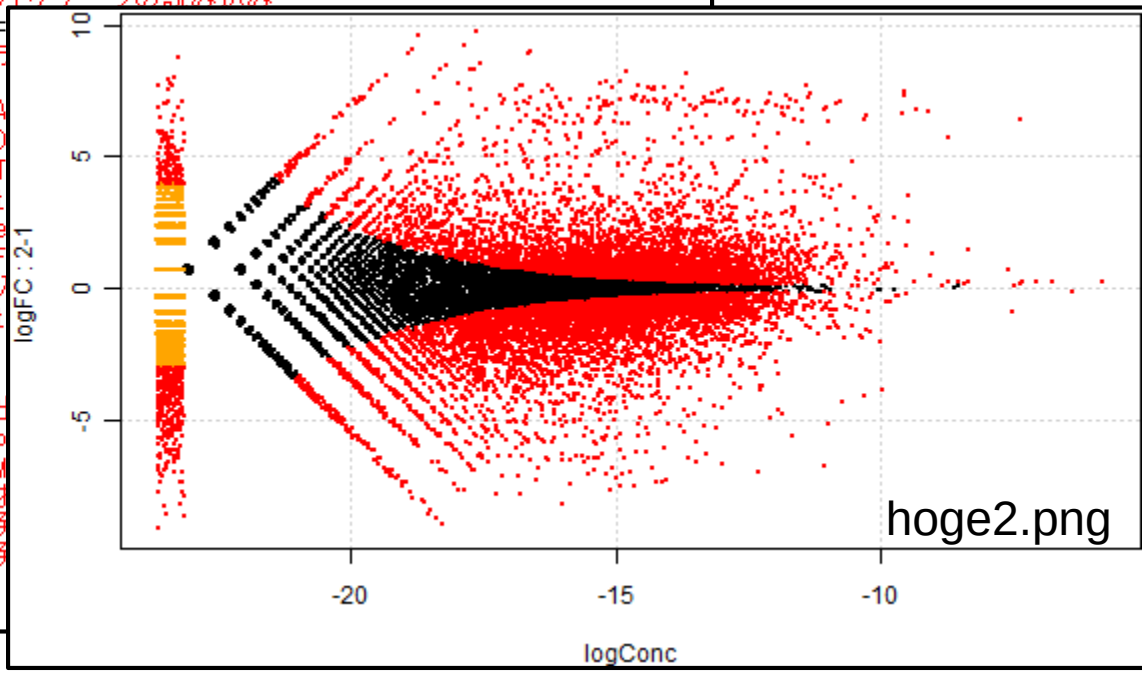
```
7. MA-plotも描く場合(FDR < 0.01を満たすものを赤色で示したMA-plotをファイルに保存)
------    ここから    ------
in_f <- "SupplementaryTable2_changed.txt"                    #読み込みたい発現データファイルを指定してin_fに格納
out_f1 <- "hoge2.txt"                                        #出力ファイル名を指定
out_f2 <- "hoge2.png"                                        #出力ファイル名を指定
param1 <- 5                                                  #A群のサンプル数を指定
param2 <- 5                                                  #B群のサンプル数を指定
param3 <- 0.01                                               #MA-plot描画時のFDRの閾値を指定

library(DEGseq)                                              #パッケージの読み込み
library(edgeR)                                               #パッケージの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="¥t", quote=
data <- as.matrix(data)                                      #

data.cl <- c(rep(1, param1), rep(2, param2))                 #A
d <- DGEList(counts=data, group=data.cl)                     #D
d <- calcNormFactors(d)                                      #T
d <- estimateCommonDisp(d)                                   #t
out <- exactTest(d)                                          #e
fdr <- p.adjust(out$table$p.value, method="BH")              #F
tmp <- cbind(rownames(data), data, out$table, fdr)           #入
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#t

#MA-plotを描画
png(out_f2, width=param4[1], height=param4[2])               #ヒ
obj <- rownames(data)[fdr < param3]                          #p
plotSmear(d, de.tags=obj)                                    #M
dev.off()                                                    #お
length(obj)                                                  #発
length(obj)/nrow(data)                                       #発

------    ここまで    ------
```



9877個（全遺伝子数のうち約31%がFDR < 0.01を満たす）

edgeRを使ってみる

- M-A plotを描画（2倍以上発現変動しているものを赤色で表示）

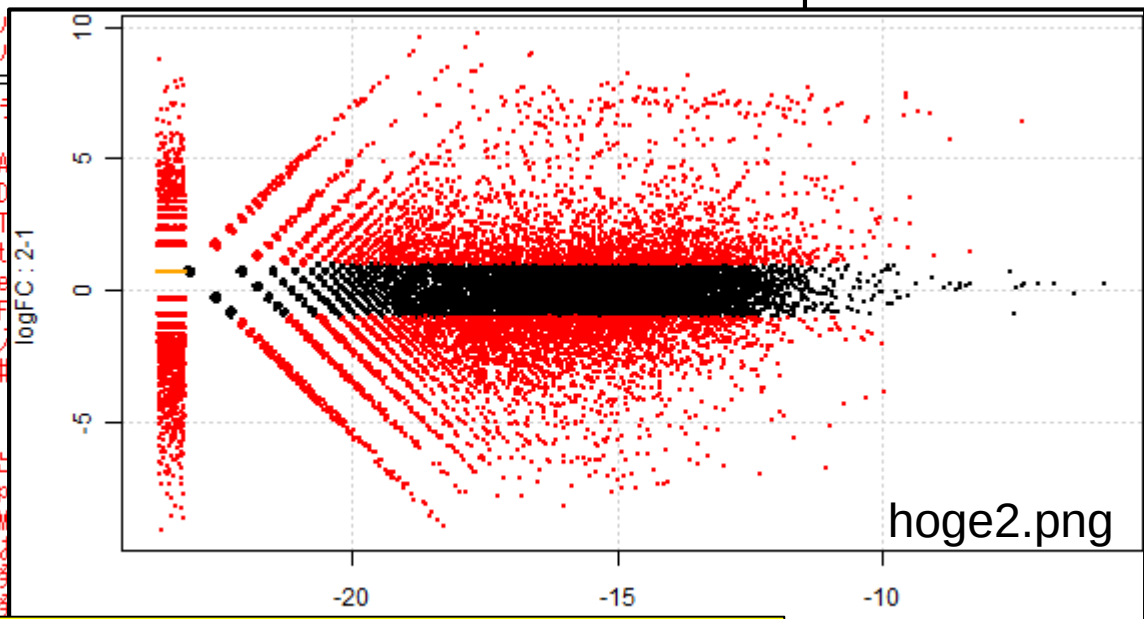
```
6. MA-plotも描く場合(5.のMA-plotで大きさを指定してpng形式ファイルに保存したいとき)
------    ここから    ------
in_f <- "SupplementaryTable2_changed.txt"          #読み込みたい発現データファイルを指定してin_fに格納
out_f1 <- "hoge2.txt"                              #出力ファイル名を指定
out_f2 <- "hoge2.png"                              #出力ファイル名を指定
param1 <- 5                                        #A群のサンプル数を指定
param2 <- 5                                        #B群のサンプル数を指定
param3 <- 2                                        #MA-plot描画時の倍率変化の閾値を指定
param4 <- c(600, 400)                              #MA-plotのファイル出力時の横幅(単位はピクセル)と縦幅を指定


library(DEGseq)
library(edgeR)
data <- read.table(in_f, header=TRUE, row.names=1, sep="¥t", quote=
data <- as.matrix(data)

data.cl <- c(rep(1, param1), rep(2, param2))
d <- DGEList(counts=data, group=data.cl)
d <- calcNormFactors(d)
d <- estimateCommonDisp(d)
out <- exactTest(d)
fdr <- p.adjust(out$table$p.value, method="BH")
tmp <- cbind(rownames(data), data, out$table, fdr)
write.table(tmp, out_f1, sep="¥t", append=F, quote=F, row.names=F)

#MA-plotを描画
png(out_f2, width=param4[1], height=param4[2])
obj <- rownames(data)[abs(out$table$logFC) >= log2(param3)]
plotSmear(d, de.tags=obj)
dev.off()
length(obj)
length(obj)/nrow(data)

------    ここまで    --
```



11786個（全遺伝子数のうち約37%が2倍以上発現変動している）
このやり方はダメなんです

倍率変化がだめな理由をデモ

- 例題：Marioni et al., *Genome Res.*, **18**: 1509-1517, 2008のデータ


kidney（腎臓）


liver（肝臓）

	A1	A2	A3	A4	A5	B1	B2	B3	B4	B5
EnsemblGeneID	R1 L1 Kidney	R1 L3 Kidney	R1 L7 Kidney	R2 L2 Kidney	R2 L6 Kidney	R1 L2 Liver	R1 L4 Liver	R1 L6 Liver	R1 L8 Liver	R2 L3 Liver
ENSG00000146556	0	0	0	0	0	0	0	0	0	0
ENSG00000197194	0	0	0	0	0	0	0	0	0	0
ENSG00000197490	0	0	0	0	0	0	0	0	0	0
ENSG00000205292	0	0	0	0	0	0	0	0	0	0
ENSG00000177693	0	0	0	0	0	0	0	0	0	0
ENSG00000209338	0	0	0	0	0	0	0	0	0	0
ENSG00000196573	0	0	0	0	0	0	0	0	0	0
ENSG00000177799	0	0	0	0	0	0	0	0	0	0
ENSG00000209341	0	0	0	0	0	0	0	0	0	0
ENSG00000209342	0	0	2	4	3	0	0	0	1	0
ENSG00000209343	0	0	0	0	0	0	0	0	0	0
ENSG00000209344	0	0								
ENSG00000209346	0	0								
ENSG00000209349	0	0	0	0	0	0	0	0	0	0
ENSG00000209350	4	7	3	6	7	35	32	31	29	34
ENSG00000209351	0	0	0	0	0	0	0	0	0	0
ENSG00000209352	0	0	1	1	0	2	0	0	0	0
ENSG00000212679	110	131	149	112	118	177	135	141	148	145
ENSG00000212678	12685	13204	12403	13031	13268	9246	9312	8746	8496	9070
ENSG00000185097	0	0	0	0	0	0	0	0	0	0

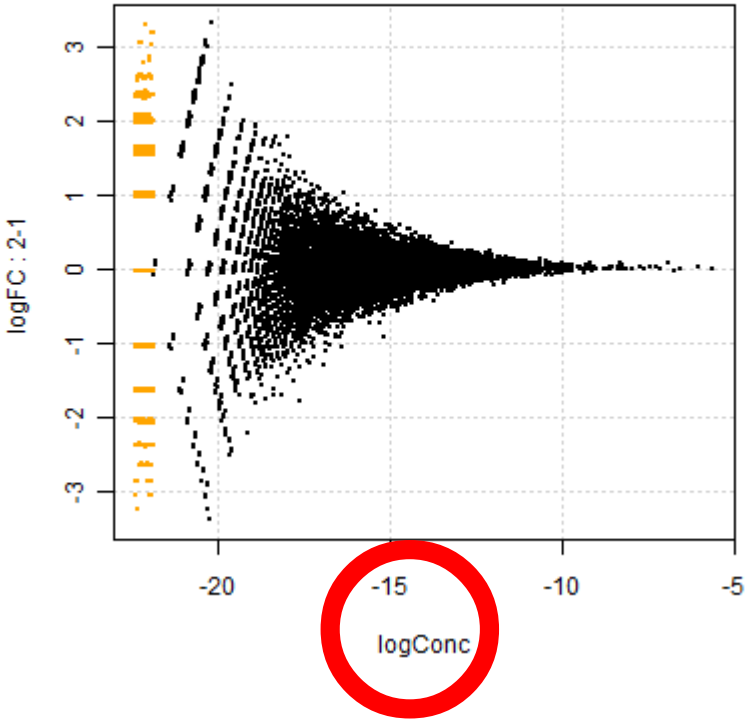
発現変動遺伝子がないデータで二群間比較をしてみる

A群　　B群

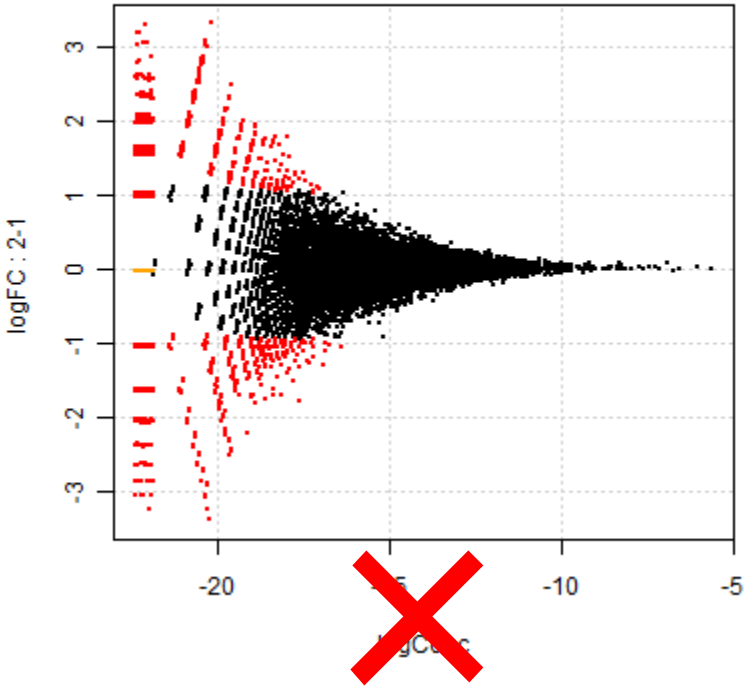
倍率変化がだめな理由をデモ

- 例題：Marioni et al., *Genome Res.*, **18**: 1509-1517, 2008のデータ（の一部）
  - (A1, A2) vs. (A3, A4)の二群間比較結果

*edgeR*でFDR < 0.01を満たすものは0個



（*edgeR*で）2倍以上発現変動しているものは3813個

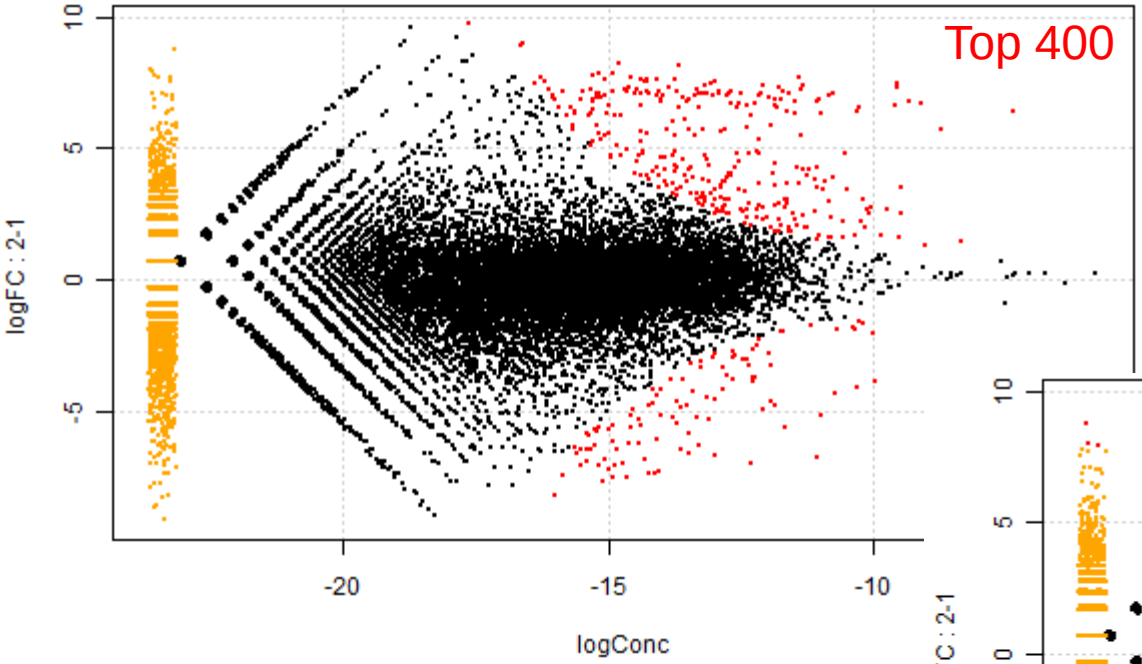


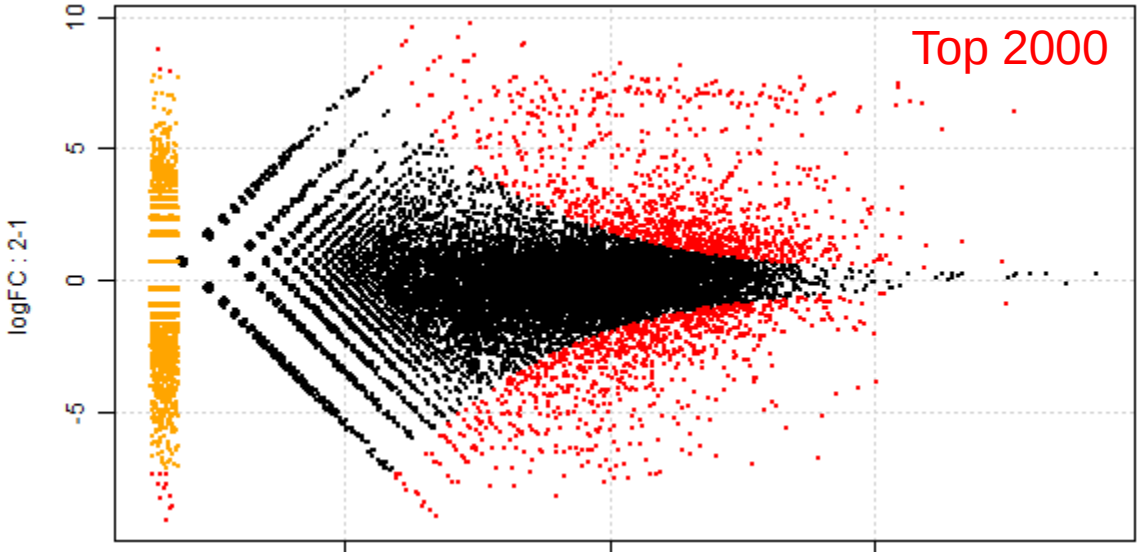
低発現領域でlog比が大きくなる現象をうまくモデル化することが重要

まとめ

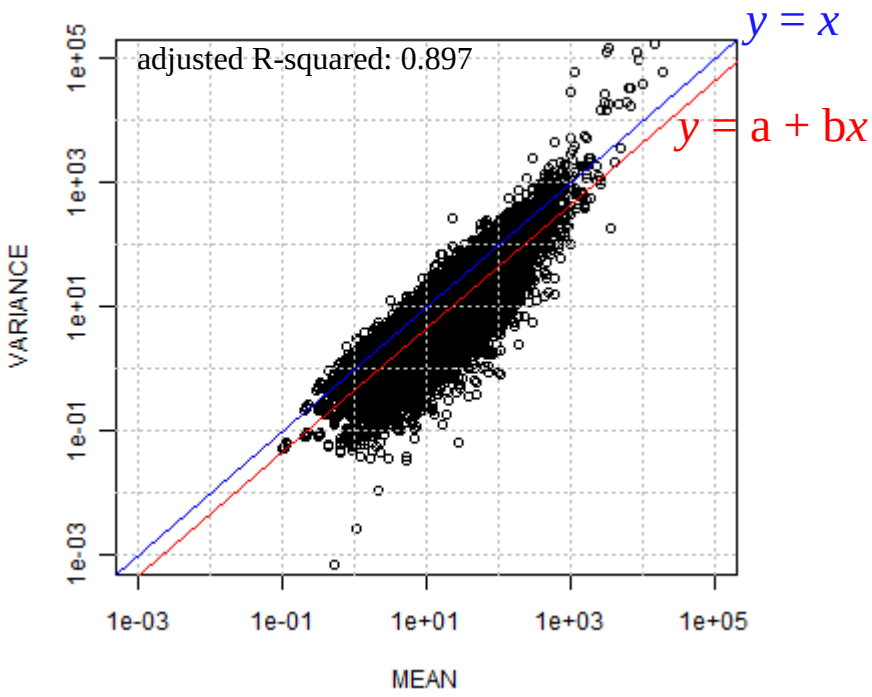
- RNA-Seqデータ取得から標準的なデータ解析の流れを説明
  - 一般的なファイル形式（FASTQ）について説明
  - 一通りの解析を自力で行うためにはLinux系スキルが必要
  - マッピング（やアセンブル）以降は基本的にRで解析可能
  - 研究目的によってデータ解析時の入力データが異なる
    - サンプル間比較：生のリードカウントデータ
    - サンプル内比較：長さ補正を行ったデータ（RPKMやFPKMなど）
  - 分布を考えることは重要（DEG数を議論したい場合）
    - technical replicatesや**biological replicates**
    - Rパッケージを用いれば発現変動遺伝子の検出から描画まで簡単
    - 「二倍（倍率変化）じゃだめなんです。〇〇さん」

「（Rで）塩基配列解析」のウェブページを用いて…なるべく自力で解析

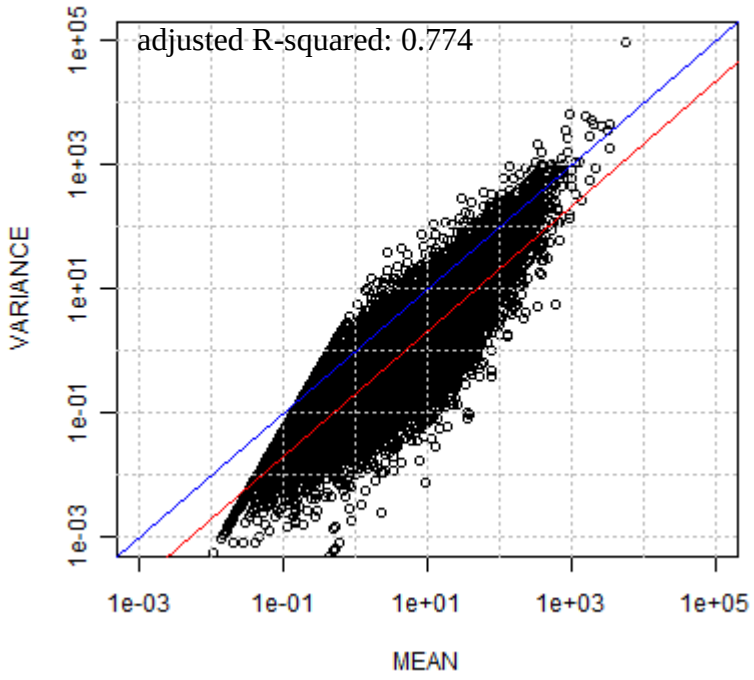




低い ← 全体的な発現レベル → 高い



RPM正規化データ



RPKM正規化データ

東大生以外の方も受講可能です（来年度もやります）