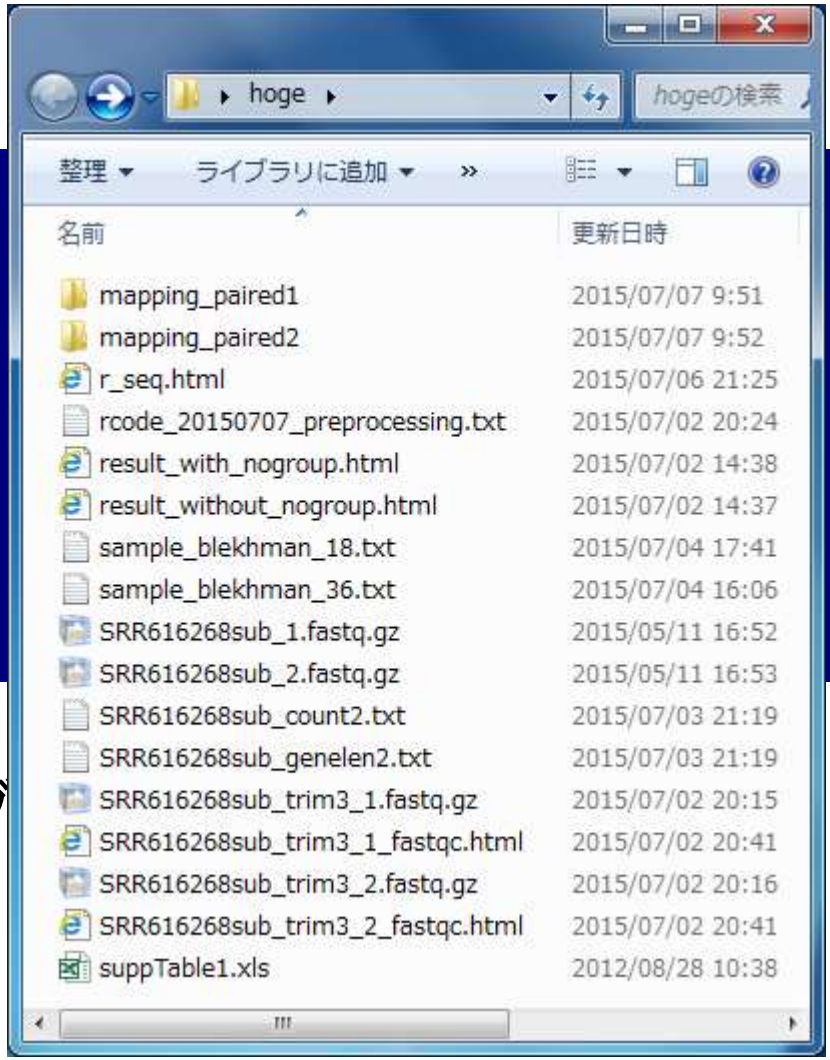
USBメモリ中のhogeフォルダをデスクトップにコピーしておいてください。

前回(6/30)のhogeフォルダがデスクトップに残っているかもしれないのでご注意ください。



農学生命情報科学特論I 第4回

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プログ
門田幸二(かどた　こうじ)
kadota@iu.a.u-tokyo.ac.jp
http://www.iu.a.u-tokyo.ac.jp/~kadota/

講義予定

NGSの普及により、以前は主にゲノム解析系で必要とされていた配列解析のためのスキルがトランスクリプトーム解析においても要求される時代になっています。本科目では、様々な局面で応用可能な配列解析系のスキルアップを目指し、RNA-Seqに基づくトランスクリプトーム解析を題材とした講義を行います。

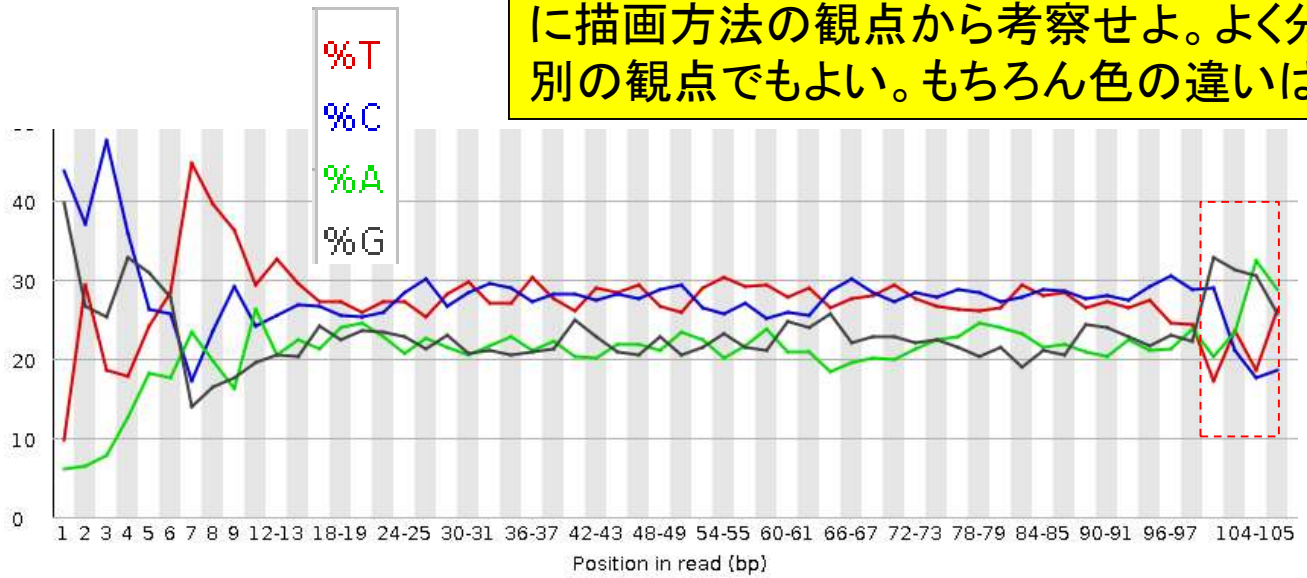
- 第1回（2015年6月16日）
  - データベース、データ取得、ファイル形式、Quality Control
  - 教科書の1.3節周辺
- 第2回（2015年6月23日）
  - Quality Control、k-mer解析、トリミング（アダプター配列除去）
- 第3回（2015年6月30日）
  - フィルタリング、アセンブル、マッピング、カウント情報取得
  - 教科書の2.3節周辺
- 第4回（2015年7月7日）
  - クラスタリング、実験デザイン、分布（モデル）、発現変動解析
  - 教科書の3.3節周辺

Contents

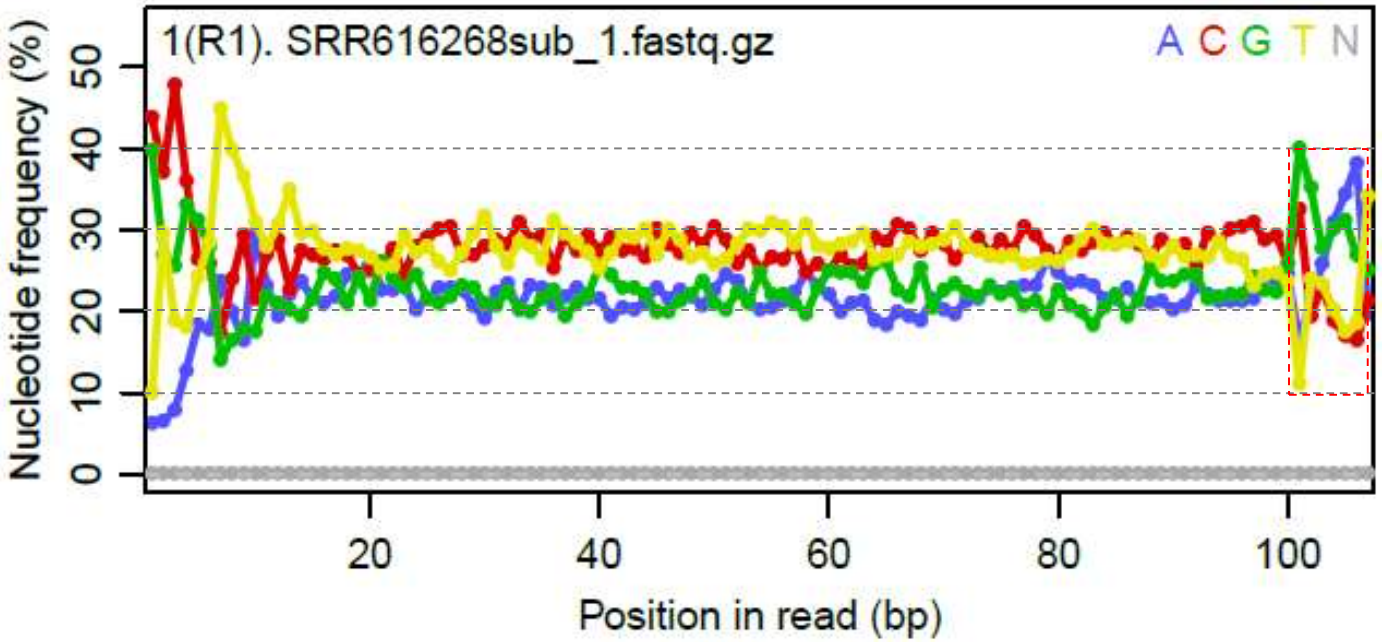
- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) ：発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

先週の課題4

課題4：①はFastQCの「Per base sequence content」、②はQuasRのQCレポートファイル中の同様な結果である。赤点線枠部分に違いが見られるが、この理由について主に描画方法の観点から考察せよ。よく分からないヒトは別の観点でもよい。もちろん色の違いは本質的ではない



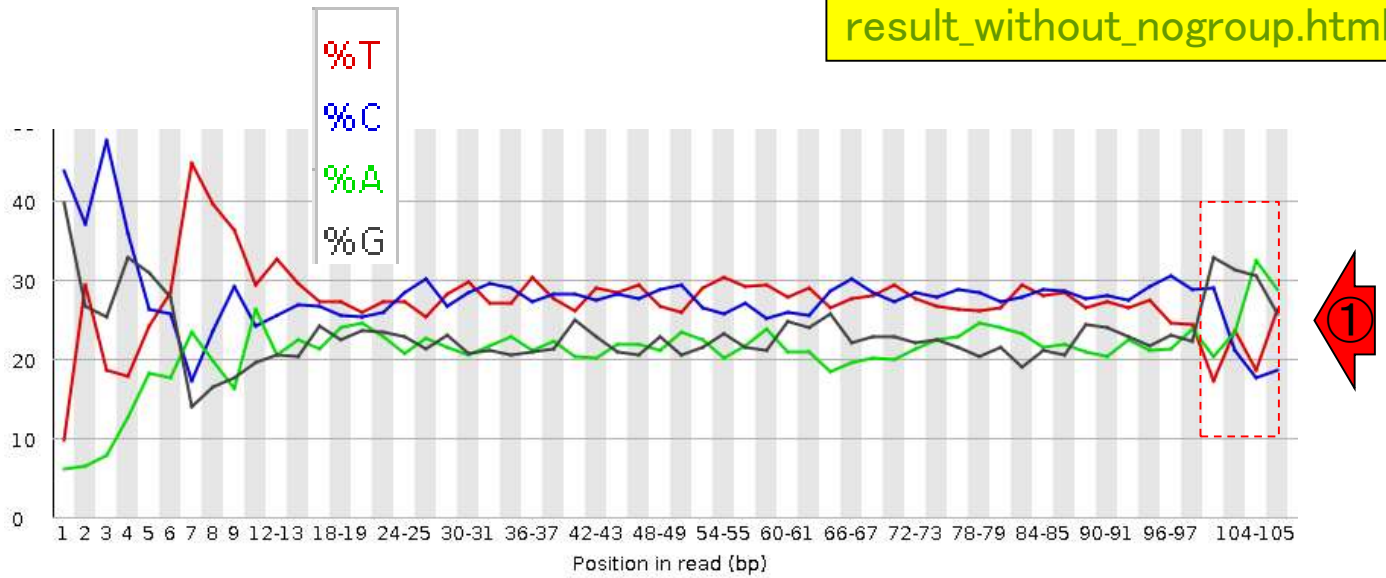
①



②

FastQC

①このFastQCレポートhtmlは、②FastQCをデフォルトオプションで実行した結果。
③得られたhtmlファイル名を
result_without_nogroup.htmlに変更。




```
iu@bielinux[~/Desktop/mac_share]                                  Ja  14:40
iu@bielinux[mac_share] pwd                                        [ 2:35午後]
/home/iu/Desktop/mac_share
iu@bielinux[mac_share] ls -la SRR616268sub_1.fastq.gz             [ 2:36午後]
-rwxrwxrwx 1 root root 74906576  5月  11 16:52 SRR616268sub_1.fastq.gz
②
iu@bielinux[mac_share] fastqc2 -q SRR616268sub_1.fastq.gz         [ 2:36午後]
iu@bielinux[mac_share] ls -la SRR616268sub_1_fastqc.html          [ 2:36午後]
-rwxrwxrwx 1 root root 365018  7月   2 14:37 SRR616268sub_1_fastqc.html
iu@bielinux[mac_share] date                                       [ 2:37午後]
2015年  7月   2日 木曜日 14:37:06 JST
iu@bielinux[mac_share] mv SRR616268sub_1_fastqc.html result_without_nogroup.html
                                                                  ③
```

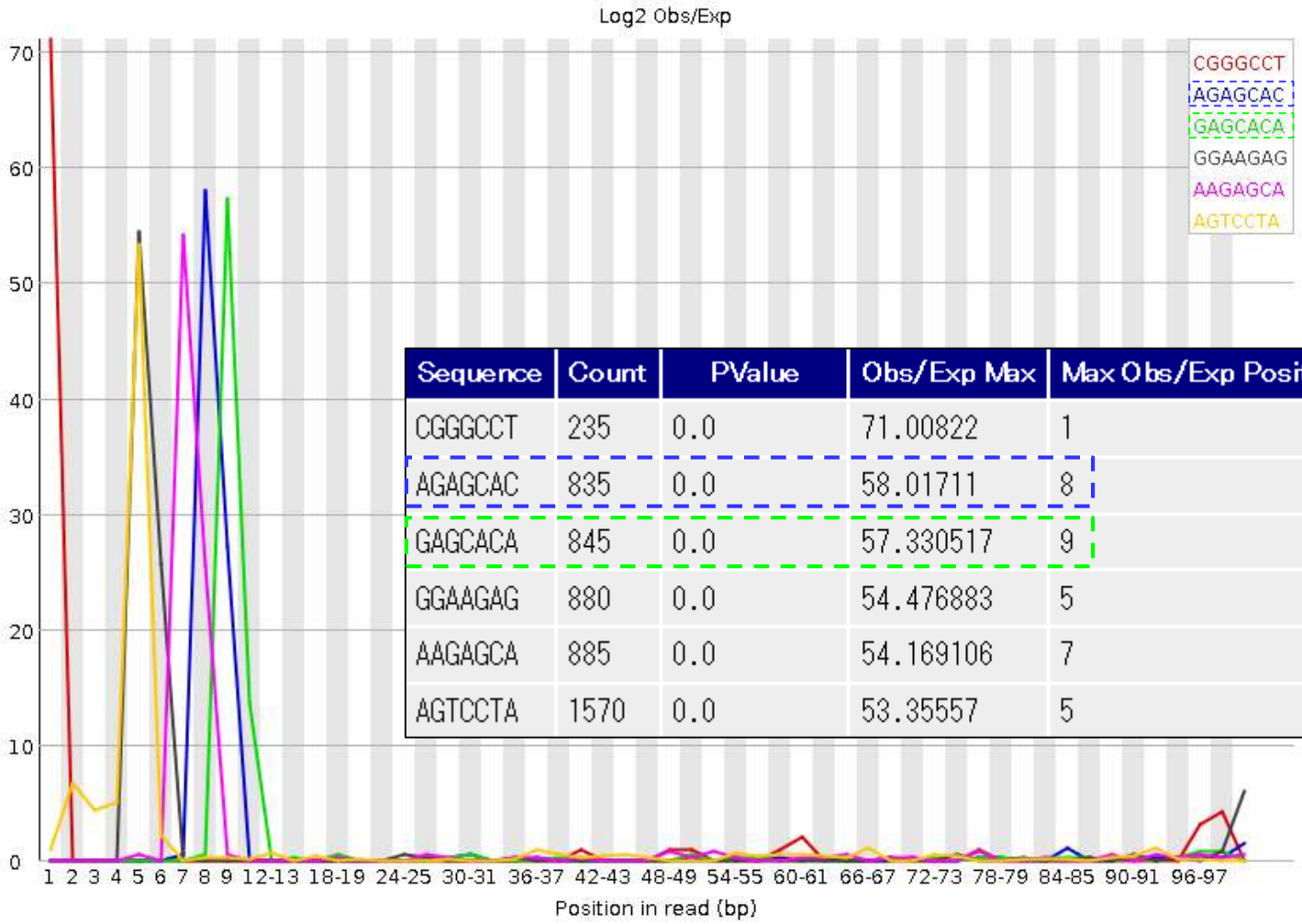
result_without_nogroup.html 参考

①Kmer Contentの項目はこんな感じでした。
(2015.06.23の第2回講義資料も参照のこと)

FastQC

Summary

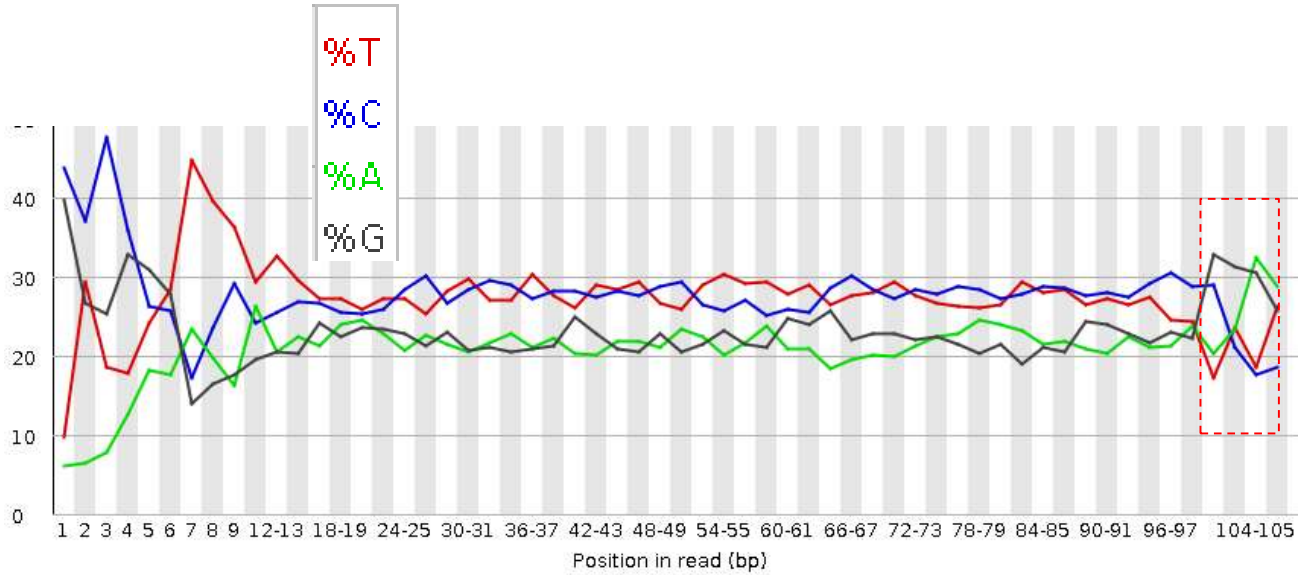
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content ①



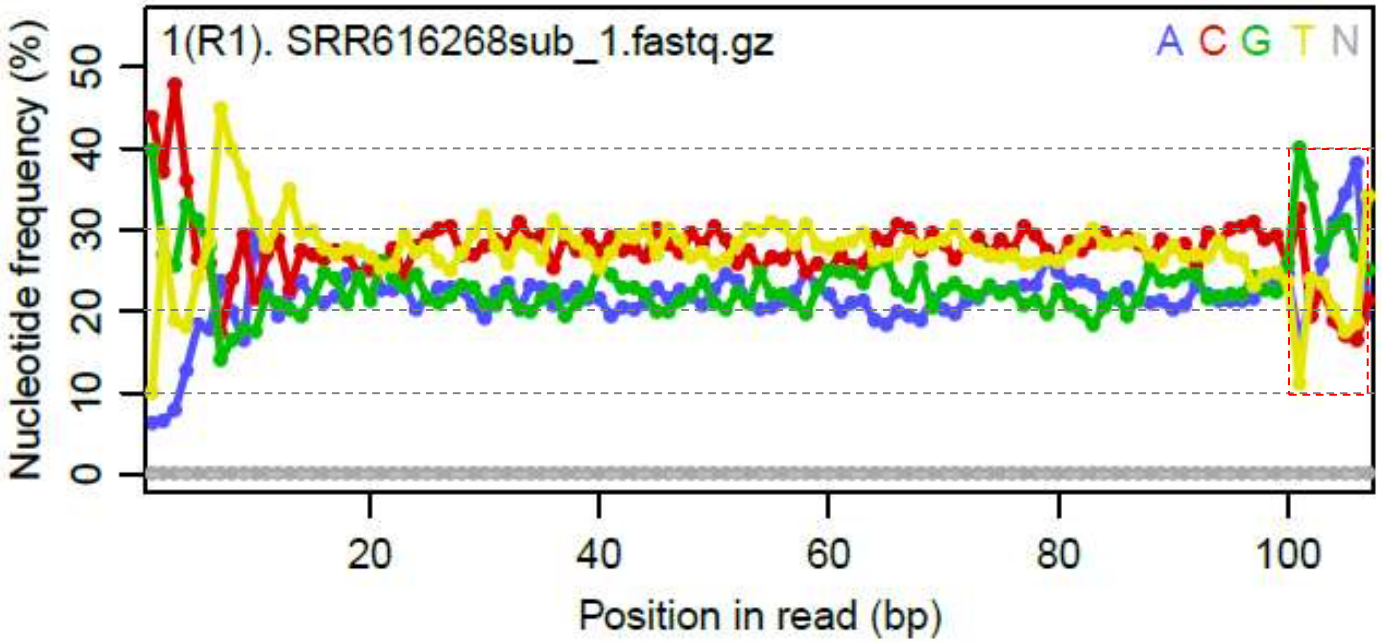
Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9
GGAAGAG	880	0.0	54.476883	5
AAGAGCA	885	0.0	54.169106	7
AGTCCTA	1570	0.0	53.35557	5

FastQC

もちろん①FastQCも②QuasRのように**塩基ごと**に表示させることはできる



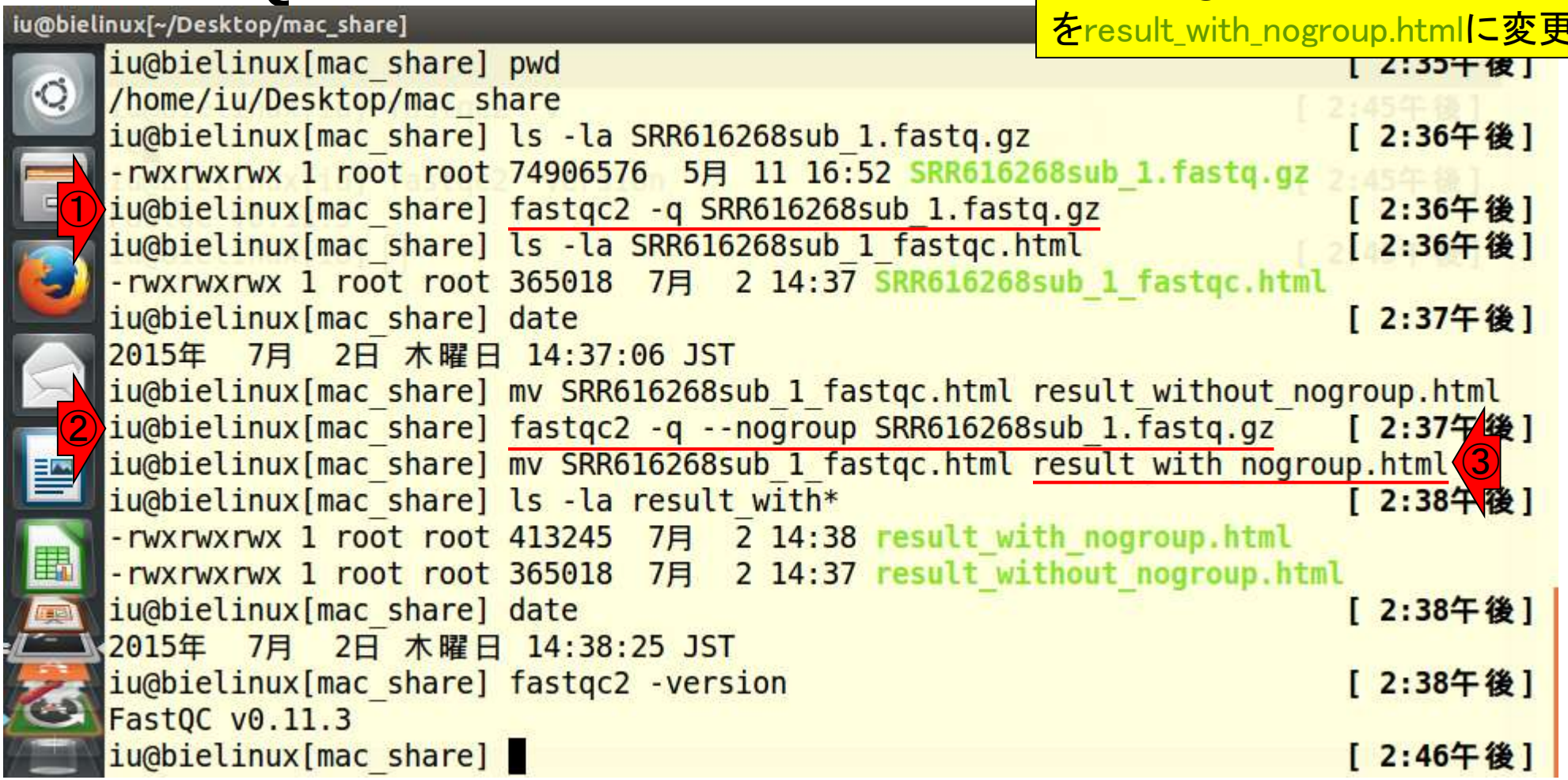
①



②

FastQC

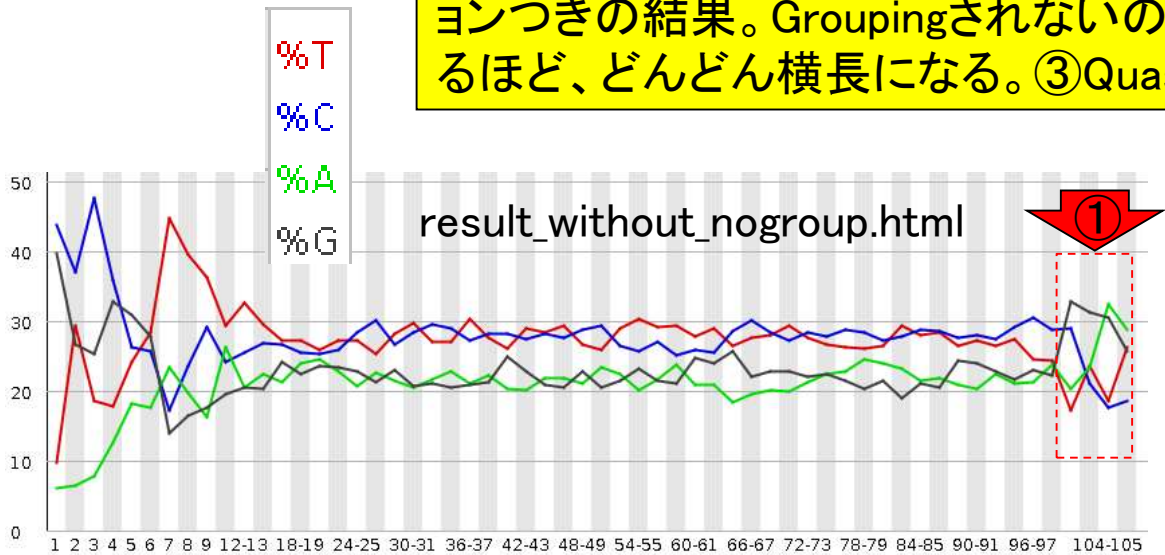
FastQC (ver. 0.11.3)を①デフォルトで実行、②--nogroupオプションつきで実行。③得られたhtmlファイル名をresult_with_nogroup.htmlに変更。

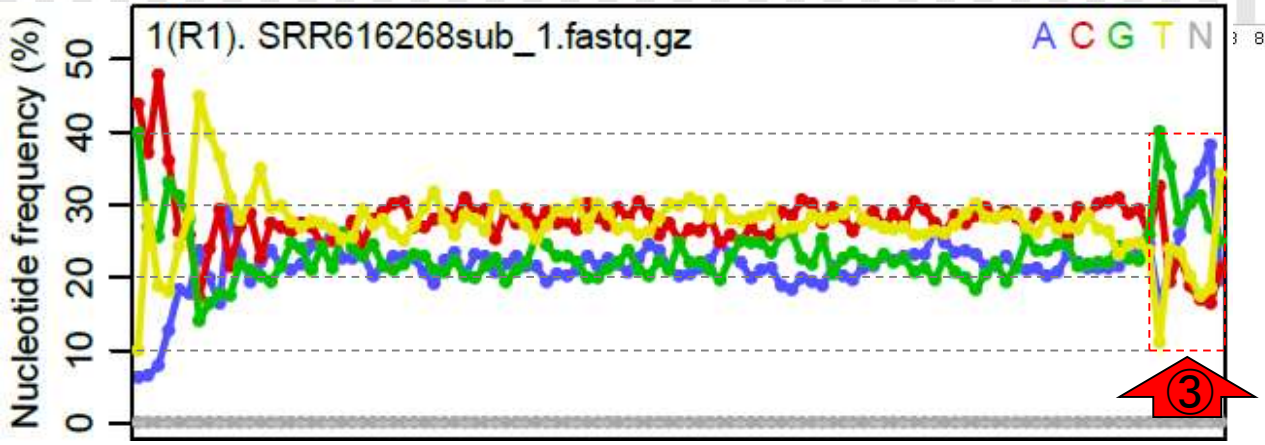


水越美咲 氏、孫建強 氏提供情報

FastQC

①FastQCデフォルトの結果。②FastQC --nogroupオプションつきの結果。Groupingされないので、配列長が長くなるほど、どんどん横長になる。③QuasRデフォルトの結果



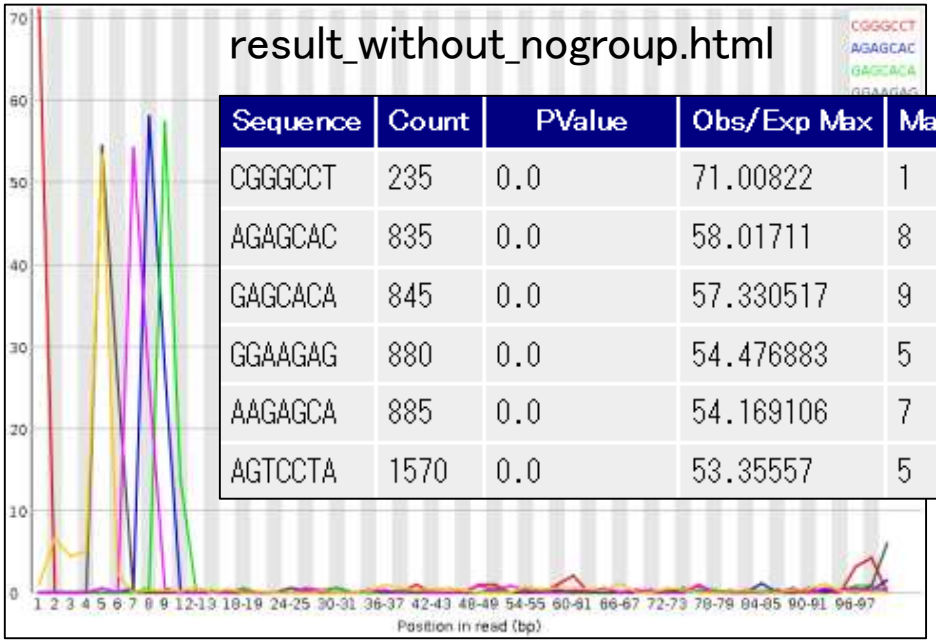


FastQC

Kmer Contentの項目を比較。違いは--nogroupの有無のみ…何か変。結論としては、--nogroupをつけないときの結果はおそらくバグ

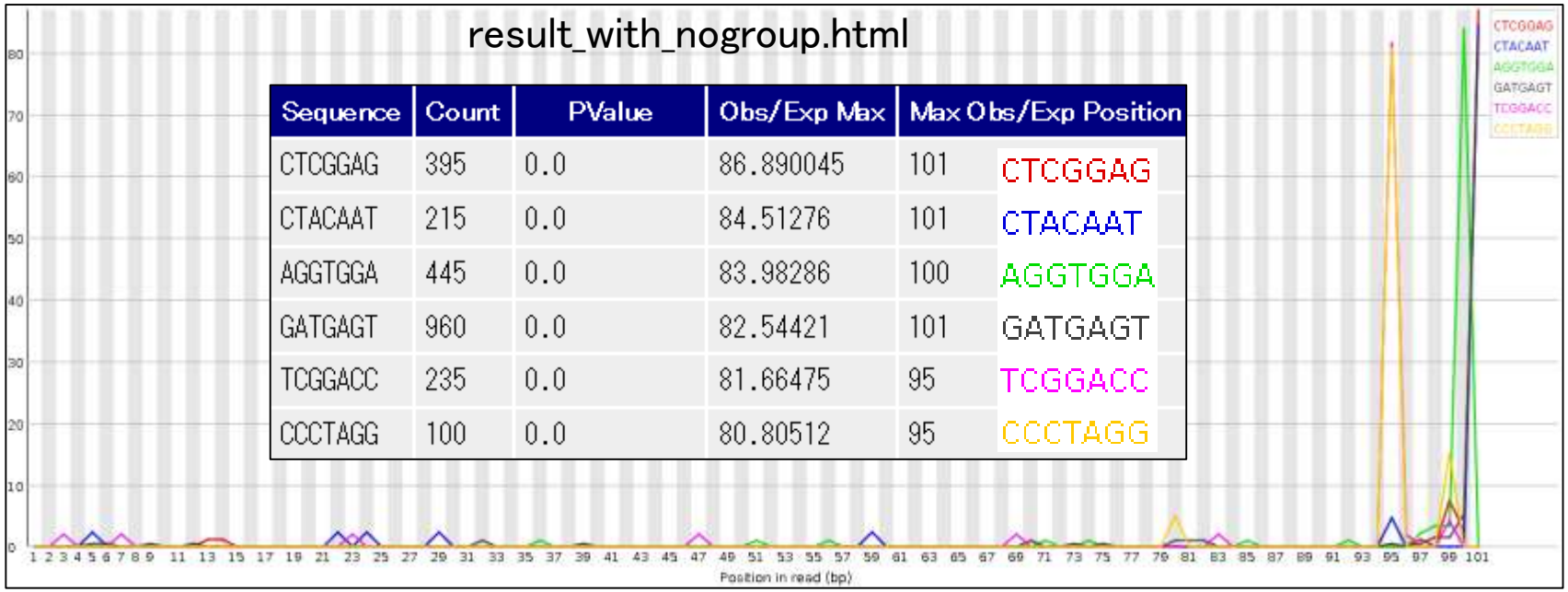
result_without_nogroup.html

Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position	
CGGGCCT	235	0.0	71.00822	1	CGGGCCT
AGAGCAC	835	0.0	58.01711	8	AGAGCAC
GAGCACA	845	0.0	57.330517	9	GAGCACA
GGAAGAG	880	0.0	54.476883	5	GGAAGAG
AAGAGCA	885	0.0	54.169106	7	AAGAGCA
AGTCCTA	1570	0.0	53.35557	5	AGTCCTA



result_with_nogroup.html

Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position	
CTCGGAG	395	0.0	86.890045	101	CTCGGAG
CTACAAT	215	0.0	84.51276	101	CTACAAT
AGGTGGA	445	0.0	83.98286	100	AGGTGGA
GATGAGT	960	0.0	82.54421	101	GATGAGT
TCGGACC	235	0.0	81.66475	95	TCGGACC
CCCTAGG	100	0.0	80.80512	95	CCCTAGG



CGGGCCTは、position 1で高い値だった

様々な角度で検証

- イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得 (last modified 2014/08/18)
- イントロ | NGS | 読み込み | FASTA形式 | description行の記述を整形 (last modified 2014/04/05)
- イントロ | NGS | 読み込み | FASTQ形式 | 基礎 (last modified 2015/06/24) NEW
- イントロ | NGS | 読み込み | FASTQ形式 | 応用 (last modified 2015/06/18) NEW

①

イントロ | NGS | 読み込み | FASTQ形式 | 基礎 NEW

Sanger FASTQ形式ファイルを読み込むやり方を示します。「基礎」では、FASTQファイルの中身を全て読み込む手順を示します。入力、出力形式は、ともに非圧縮(.fastq)、gzip圧縮(.fastq.gz)ファイルが可能です。

②

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107 bpです。NAを含む場合への各種対応策を2015年6月24日に追加しました(茂木朋貴氏、野間口達洋氏、他多くの受講生提供情報)。

```
#以下はおまけ(forループを用いて美しく...上級1)
param_len_ngs <- 107                          #リード長を指定
param_obj <- "CGGGCCT"                        #調べたいk-merを指定
Obs <- NULL                                   #おまじない
hoge <- param_len_ngs - nchar(param_obj) + 1#positionの右端の値を計算してhogeに格
for(i in 1:hoge){                             #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nchar(param_obj)))[param_
}
Obs[is.na(Obs)] <- 0                          #NAの位置に0を代入
head(Obs)                                     #最初の6個の要素を表示
mean(Obs, na.rm=TRUE)                         #平均値
Exp <- mean(Obs, na.rm=TRUE)                  #平均値をExpとして取り扱う
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")

#以下はおまけ(forループを用いて美しく...上級2)
param_obj <- "CGGGCCT"                        #調べたいk-merを指定
Obs <- NULL                                   #おまじない
hoge <- width(fasta)[1] - nchar(param_obj) + 1#positionの右端の値を計算してhogeに
for(i in 1:hoge){                             #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nchar(param_obj)))[param_
}
```

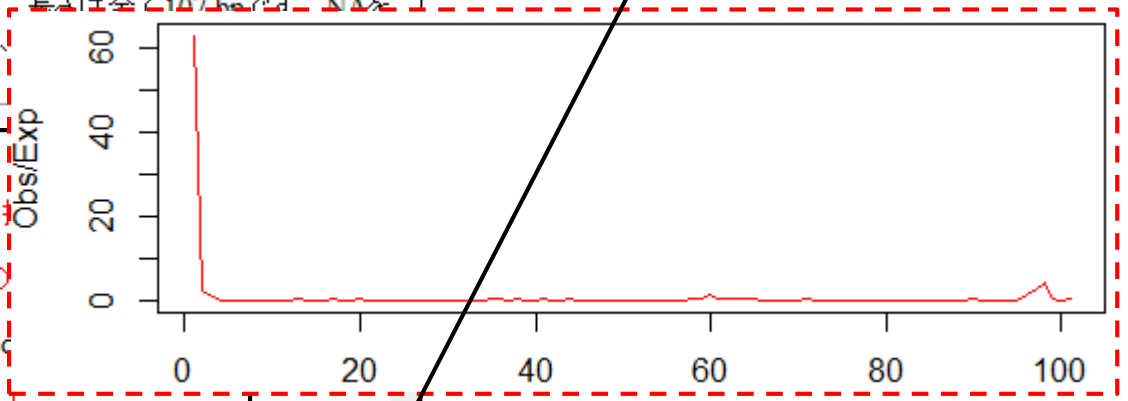
CGGGCCTは、position 1で高い値だった

様々な角度で検証

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107bpです。NAを含む場合への各種対応策を2015年6月24日に追加しました(茂木朋貴氏、生提供情報)。

```
#以下はおまけ(forループを用いて美しく...上級1)
param_len_ngs <- 107                    #リード長を指定
param_obj <- "CGGGCCT"                  #調べたいk-merを
Obs <- NULL                             #おまじない
hoge <- param_len_ngs - nchar(param_obj) + 1#positionの
for(i in 1:hoge){                       #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nc
}
Obs[is.na(Obs)] <- 0                    #NAの位置に0を代入
head(Obs)                               #最初の6個の
mean(Obs, na.rm=TRUE)                   #平均値
Exp <- mean(Obs, na.rm=TRUE)            #平均値をExp
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")

#以下はおまけ(forループを用いて美しく...上級2)
param_obj <- "CGGGCCT"                  #調べたいk-m
Obs <- NULL                             #おまじない
hoge <- width(fasta)[1] - nchar(param_obj) + 1#posi
for(i in 1:hoge){                       #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, widt
```



R Console

```
> head(Obs)                                    #最初の6$
CGGGCCT CGGGCCT CGGGCCT CGGGCCT CGGGCCT CGGGCCT
   1804      69      41       9       3       1
> mean(Obs, na.rm=TRUE)                        #平均値
[1] 28.58416
> Exp <- mean(Obs, na.rm=TRUE)                 #平均値を$
> head(Obs/Exp)
    CGGGCCT     CGGGCCT     CGGGCCT     CGGGCCT    $
63.11188085  2.41392449  1.43436093  0.31485972   0$
    CGGGCCT
 0.03498441
> plot(Obs/Exp, type="l", col="red")
>
```

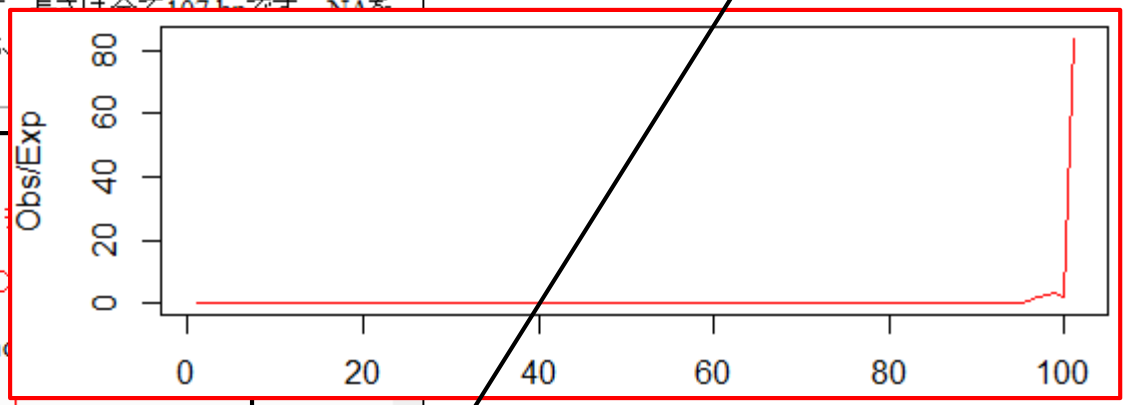
様々な角度で検証

CTCGGAGは、確かに最後のポジション(position 101)で高い値。

8. FASTQ形式ファイル(SRR616268sub_1.fastq.gz)の場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分(約73MB)です。長さは全て107bpです。NAを含む場合への各種対応策を2015年6月24日に追加しました(茂木朋貴氏、生提供情報)。

```
#以下はおまけ(forループを用いて美しく...上級1)
param_len_ngs <- 107                    #リード長を指定
param_obj <- "CGGGCCT"                  #調べたいk-merを
Obs <- NULL                             #おまじない
hoge <- param_len_ngs - nchar(param_obj) + 1#positionの
for(i in 1:hoge){                       #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, width=nc
}
Obs[is.na(Obs)] <- 0                    #NAの位置に0を代入
head(Obs)                               #最初の6個の
mean(Obs, na.rm=TRUE)                   #平均値
Exp <- mean(Obs, na.rm=TRUE)            #平均値をExp
head(Obs/Exp)
plot(Obs/Exp, type="l", col="red")

#以下はおまけ(forループを用いて美しく...上級2)
param_obj <- "CGGGCCT"                  #調べたいk-m
Obs <- NULL                             #おまじない
hoge <- width(fasta)[1] - nchar(param_obj) + 1#posi
for(i in 1:hoge){                       #ループを回す
    Obs <- c(Obs, table(subseq(fasta, start=i, widt
```



R Console

```
> tail(Obs)                                          #最後の6$
CTCGGAG CTCGGAG CTCGGAG CTCGGAG CTCGGAG CTCGGAG
     20      75     120     144      79    3356
> mean(Obs, na.rm=TRUE)                              #平均値
[1] 40.0297
> Exp <- mean(Obs, na.rm=TRUE)                       #平均値を$
> tail(Obs/Exp)
  CTCGGAG   CTCGGAG   CTCGGAG   CTCGGAG   CTCGGAG
0.499629  1.873609  2.997774  3.597329  1.973535
  CTCGGAG
83.837744
> plot(Obs/Exp, type="l", col="red")
>
```

様々な角度で検証

「--nogroupオプションつきの①」と「オプションなしの②」は確かに高い値になっていることをRでも確認した。②のCGGGCCTは、③--nogroupの結果の13番目にいた。

result_without_nogroup.html

Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9
GGAAGAG	880	0.0	54.476883	5
AAGAGCA	885	0.0	54.169106	7
AGTCCTA	1570	0.0	53.35557	5
GTCCAGT	1555	0.0	53.00536	1
CCAGTCC	1655	0.0	51.225075	3
CAGTCCT	1650	0.0	51.07447	4
CCTCTAT	30	0.008129199	50.4628	2
GAAGAGC	975	0.0	49.68645	6
GTCCTAC	1720	0.0	49.289246	6
TCCTACA	1760	0.0	48.169033	7
TCGGAAG	1005	0.0	47.701153	3
GGGGCCT	160	0.0	47.40606	1
GATCGGA	1035	0.0	46.90223	1
CCTCTCT	65	1.707027E-6	46.581047	3
CGGAAGA	1040	0.0	46.09583	4
CCTACAA	1840	0.0	46.07473	8
GGCCTAT	2465	0.0	45.33545	1

result_with_nogroup.html

Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CTCGGAG	395	0.0	86.890045	101
CTACAAT	215	0.0	84.51276	101
AGGTGGA	445	0.0	83.98286	100
GATGAGT	960	0.0	82.54421	101
TCGGACC	235	0.0	81.66475	95
CCCTAGG	100	0.0	80.80512	95
CCTAGGT	45	1.4897523E-9	78.56054	96
GTTGTGG	965	0.0	75.840096	101
GGCCCTG	180	0.0	75.75481	100
TACTACA	20	0.0016262057	75.75481	96
TAGGTGG	225	0.0	74.07136	99
GTTCTCT	75	0.0	74.02692	101
CGGGCCT	235	0.0	71.00822	1
TCCCTCG	50	3.430614E-9	70.70448	98
CCCACAC	440	0.0	70.0158	95
TACTTAG	405	0.0	68.5846	95
TCGACGC	450	0.0	68.45989	95
ACTGGAC	510	0.0	68.28686	101
GTCGAAG	540	0.0	68.23187	101
TACCTCT	45	1.3395584E-7	67.3376	98

様々な角度で検証

FastQC (ver. 0.11.3)は、おそらくK-mer計算のところにバグがあるが、--nogroupオプションをつければ大丈夫。デフォルトは、groupingしてからK-mer解析しちゃってるのかもしれない。その根拠は、①Max Obs/Exp Positionの上位全てがposition1-9に偏っているから。この程度のことで目くじらを立てる暇があったら、開発者に詳細かつ丁寧なバグレポートを送付。我々はあくまでも有難く使わせていただく立場です。いずれにしても次期バージョンに期待！

result_without_nogroup.html

Sequence	Count	PValue	Obs/Exp Max	Max Obs/Exp Position
CGGGCCT	235	0.0	71.00822	1
AGAGCAC	835	0.0	58.01711	8
GAGCACA	845	0.0	57.330517	9
GGAAGAG	880	0.0	54.476883	5
AAGAGCA	885	0.0	54.169106	7
AGTCCTA	1570	0.0	53.35557	5
GTCCAGT	1555	0.0	53.00536	1
CCAGTCC	1655	0.0	51.225075	3
CAGTCCT	1650	0.0	51.07447	4
CCTCTAT	30	0.008129199	50.4628	2
GAAGAGC	975	0.0	49.68645	6
GTCCTAC	1720	0.0	49.289246	6
TCCTACA	1760	0.0	48.169033	7
TCGGAAG	1005	0.0	47.701153	3
GGGGCCT	160	0.0	47.40606	1
GATCGGA	1035	0.0	46.90223	1
CCTCTCT	65	1.707027E-6	46.581047	3
CGGAAGA	1040	0.0	46.09583	4
CCTACAA	1840	0.0	46.07473	8
GGCCTAT	2465	0.0	45.33545	1

①

TCGGACC	235	0.0	81.66475	95
CCCTAGG	100	0.0	80.80512	95
CCTAGGT	45	1.4897523E-9	78.56054	96
GTTGTGG	965	0.0	75.840096	101
GGCCCTG	180	0.0	75.75481	100
TACTACA	20	0.0016262057	75.75481	96
TAGGTGG	225	0.0	74.07136	99
GTTCTCT	75	0.0	74.02692	101
CGGGCCT	235	0.0	71.00822	1
TCCCTCG	50	3.430614E-9	70.70448	98
CCCACAC	440	0.0	70.0158	95
TACCTCT	45	1.3395584E-7	67.3376	98

```
If you have any comments about FastQC we would like to hear them.  You
can either enter them in our bug tracking system at:

http://www.bioinformatics.babraham.ac.uk/bugzilla/

..or send them directly to simon.andrews@babraham.ac.uk.
```

Contents

- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) ：発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

前処理

とりあえずこのような手順で行ったものでマッピングを行います

- 100万リードのオリジナル?!ファイル
  - SRR616268sub_1.fastq.gz：107 bp, 74,906,576 bytes
  - SRR616268sub_2.fastq.gz：93 bp, 67,158,462 bytes
- Step1：SRR616268sub_1.fastq.gzの前処理
  - 1-1. 3’側の7 bpをトリム
  - 1-2. 5’側の「TruSeq Adapter, Index 3」を除去
  - 1-3. 5’側の「TruSeq Adapter, Index 2」を除去
- Step2：SRR616268sub_2.fastq.gzの前処理
  - 2-1. 3’側の2 bpをトリム
  - 2-2. 5’側の「Illumina Single End PCR Primer 1」を除去
- Step3：共通リード(998,521 reads)を抽出
  - SRR616268sub_trim3_1.fastq.gz, 59,092,219 bytes
  - SRR616268sub_trim3_2.fastq.gz, 54,667,920 bytes

カウント情報取得1

アノテーション情報(GFFファイル)を用いずに、マップされた和集合領域を同定してカウントデータを得る一連のやり方を示します。

- マップ後 | 出力ファイルの読み込み | htSeqTools(Planet_2012) (last modified 2013/06/19)
- マップ後 | カウント情報取得 | について (last modified 2014/12/17)
- マップ後 | カウント情報取得 | single-end | ゲノム | アノテーション有 | QuasR(Gaidatzis_2015)
- マップ後 | カウント情報取得 | single-end | ゲノム | アノテーション無 | QuasR(Gaidatzis_2015)
- マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis_2015)
- マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション無 | QuasR(Gaidatzis_2015) ①
- マップ後 | カウント情報取得 | トランスクリプトーム | BEDファイルから (last modified 2014/06/
- マップ後 | 配列長とカウント数の関係 (last modified 2014/07/03)
- 正規化 | について (last
- 正規化 | 基礎 | RPK or
- 正規化 | 基礎 | RPM or
- 正規化 | 基礎 | RPKM
- 正規化 | 基礎 | RPKM
- 正規化 | サンプル内 |
- 正規化 | サンプル内 |

マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション無 | QuasR(Gaidatzis_2015)

NEW

QuasRパッケージを用いたpaired-end RNA-seqデータのリファレンスゲノム配列へのBowtieによるマッピングから、カウントデータ取得までの一連の流れを示します。アノテーション情報がない場合を想定しているので、GenomicAlignments パッケージを利用して、マップされたリードの和集合領域(union range)を得たのち、領域ごとにマップされたリード数をカウントしています。
「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下をコピペ。

1. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分のpaired-endファイル(SRR616268sub_1.fastq.gzとSRR616268sub_2.fastq.gz)から5'および3'側をrcode_20150707_preprocessing.txtに書いてある手順でトリムして得られた998,521リードからなるpaired-endのファイルです。SRR616268sub_trim3_1.fastq.gz (59,092,219 bytes)とSRR616268sub_trim3_2.fastq.gz (54,667,920 bytes)です。Ensembl (Flicek et al., 2014)から提供されている Lactobacillus casei 12Aの multi-FASTA形式ゲノム配列ファイル (Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa) がリファレンス配列です。マッピングオプションはデフォルトです。

```
in_f1 <- "mapping_paired_genome2.txt"   #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
out_f1 <- "hoge1_count.txt"             #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1_genelength.txt"        #出力ファイル名を指定してout_f2に格納
```

FileName1	FileName2	SampleName
SRR616268sub_trim3_1.fastq.gz	SRR616268sub_trim3_2.fastq.gz	namae_paired

```
#本番(マッピング)
time_s <- proc.time()                   #計算時間を計測するため
out <- qAlign(in_f1, in_f2)             #マッピングを行うqAlign関数を実行した結果をoutに格納
time_e <- proc.time()                   #計算時間を計測するため
```

カウント情報取得1

作業ディレクトリと入力ファイルの確認を行ってコピペ。約7分

1. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分のpaired-endファイル(SRR616268sub_1.fastq.gzとSRR616268sub_2.fastq.gz)から5'および3'側をrcode_20150707_preprocessing.txtに書いてある手順でトリムして得られた998,521リードからなるpaired-endのファイルです。SRR616268sub_trim3_1.fastq.gz (59,092,219 bytes)とSRR616268sub_trim3_2.fastq.gz (54,667,920 bytes)です。Ensembl (Flicek et al., 2014)から提供されているLactobacillus casei 12Aの multi-FASTA形式ゲノム配列ファイル (Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa) がリファレンス配列です。マッピングオプションはデフォルトです。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
out_f1 <- "hoge1_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1_genelength.txt"       #出力ファイル名を指定してout_f2に格納

#必要なパッケージをロード
library(QuasR)                         #パッケージの読み込み
library(GenomicAlignments)             #パッケージの読み込み

#本番(マッピング)
time_s <- proc.time()                  #計算時間を計測するため
out <- qAlign(in_f1, in_f2)            #マッピングを行うqAlign関数を実行した結果をoutに格納
time_e <- proc.time()                  #計算時間を計測するため
time_e - time_s                        #計算時間を表示(一番右側の数字、単位はsecond)
out
alignmentStats(out)

#本番(マップされ
tmpfname <- ou
tmpsname <- ou
for(i in 1:len
  if(i == 1){
```

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/mapping_paired1"
> list.files()
[1] "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"
[2] "mapping_paired_genome2.txt"
[3] "SRR616268sub_trim3_1.fastq.gz"
[4] "SRR616268sub_trim3_2.fastq.gz"
>
```

カウント情報取得1

途中経過。①マッピングのところだけだと約4分。②リードの3’側のトリミングをしたおかげで、693,500 / (693,500 + 1,303,542) = 34.73%のリードがマップされていることがわかる

R Console

```
> time_e - time_s                          #計算時間を表示(一番右
   ユーザ    システム        経過
      0.85        0.28      242.89 ①
> out                                      #マッピングに用いたパラメータや入力$
Project: qProject
 Options    : maxHits          : 1
              paired           : fr
              splicedAlignment: FALSE
              bisulfite        : no
              snpFile          : none
 Aligner    : Rbowtie v1.8.0 (parameters: -m 1 --best --strata --maxins 500)
 Genome     : .../Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Ch$

 Reads      : 1 pair of files, 1 sample (fastq format):
   1. SRR616268..._1.fastq.gz  SRR616268..._2.fastq.gz  namae_paired (phred33)

 Genome alignments: directory: same as reads
   1. SRR616268sub_trim3_1_1f8423da460e.bam

 Aux. alignments: none

> alignmentStats(out)                      #マッピング結果(alignment statistics$
                    seqlength mapped unmapped
namae_paired:genome   2907892 693500  1303542 ②
>
```

最後までエラーなく終えるとこんな感じ

カウント情報取得1

1. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分のpaired-endファイル(SRR616268sub_1.fastq.gzとSRR616268sub_2.fastq.gz)から5'および3'側をrcode_20150707_preprocessing.txtに書いてある手順でトリムして得られた998,521リードからなるpaired-endのファイルです。SRR616268sub_trim3_1.fastq.gz (59,092,219 bytes)とSRR616268sub_trim3_2.fastq.gz (54,667,920 bytes)です。Ensembl (Flicek et al., 2014)から提供されているLactobacillus casei 12Aの multi-FASTA形式ゲノム配列ファイル (Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa) がリファレンス配列です。マッピングオプションはデフォルトです。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.
out_f1 <- "hoge1_count.txt"            #出力ファイル名
out_f2 <- "hoge1_genelength.txt"       #出力ファイル名

#必要なパッケージをロード
library(QuasR)                         #パッケージの読
library(GenomicAlignments)             #パッケージの読

#本番(マッピング)
time_s <- proc.time()                  #計算時間を計測
out <- qAlign(in_f1, in_f2)            #マッピングを行
time_e <- proc.time()                  #計算時間を計測
time_e - time_s                        #計算時間を表示
out                                    #マッピングに用
alignmentStats(out)                    #マッピング結果

#本番(マップされたリードの和集合領域同定)
tmpfname <- out@alignments[,1]         #ファイル名(in_
tmpsname <- out@alignments[,2]         #サンプル名(in_
for(i in 1:length(tmpfname)){          #サンプル数(フ
  if(i == 1){
```

R Console

```
> #ファイルに保存(QCレポート用のpdfファイル作成)
> out_f <- sub(".bam", "_QC.pdf", out@alignments$
> qQCReport(out, pdfFilename=out_f)       #QCレポ$
collecting quality control data
creating QC plots
> out_f                                   #ファイ$
[1] "C:/Users/kadota/Desktop/hoge/mapping_paired$
>
> #ファイルに保存(BED形式ファイル)
> tmpfname <- out@alignments[,1]          #ファイ$
> for(i in 1:length(tmpfname)){           #サンプ$
+   hoge <- readGAlignments(tmpfname[i]) #BAM形$
+   hoge <- as.data.frame(hoge)           #データ$
+   tmp <- hoge[, c("seqnames","start","end")]#$
+   out_f <- sub(".bam", ".bed", tmpfname[i])#BE$
+   out_f                                 #ファイ$
+   write.table(tmp, out_f, sep="\t", append=F, $
+ }
>
```

カウント情報取得1

2つの出力ファイルはこんな感じです。10,329行。

1. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

乳酸菌RNA-seqデータSRR616268の最初の100万リード分のpaired-endファイル(SRR616268sub_1.fastq.gzとSRR616268sub_2.fastq.gz)から5'および3'側をrcode_20150707_preprocessing.txtに書いてある手順でトリムして得られた998,521リードからなるpaired-endのファイルです。SRR616268sub_trim3_1.fastq.gz (59,092,219 bytes)とSRR616268sub_trim3_2.fastq.gz (54,667,920 bytes)です。Ensembl (Flicek et al., 2014)から提供されているLactobacillus casei 12Aの multi-FASTA形式ゲノム配列ファイル (Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa) がリファレンス配列です。マッピングオプションはデフォルトです。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
out_f1 <- "hoge1_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1_genelength.txt"       #出力ファイル名を指定してout_f2に格納

#必要なパッケージをロード
library(QuasR)
library(GenomicAlignm

#本番(マッピング)
time_s <- proc.time()
out <- qAlign(in_f1,
time_e <- proc.time()
time_e - time_s
out
alignmentStats(out)

#本番(マップされたリー
tmpfname <- out@align
tmpsname <- out@align
for(i in 1:length(tmp
  if(i == 1){
```

①hoge1_count.txt (カウント情報)

tmp	namae_paired
Chromosome_5_2615_2611_+	538
Chromosome_2914_3033_120_+	2
Chromosome_3190_9498_6309_+	1082
Chromosome_9539_10270_732_+	62
Chromosome_10380_10495_116_+	2
Chromosome_10855_10945_91_+	1
Chromosome_11066_12348_1283_+	2657
Chromosome_12433_12645_213_+	3
Chromosome_12758_12856_99_+	1
Chromosome_12884_12974_91_+	1
Chromosome_13220_13310_91_+	1

②hoge1_genelength.txt (配列長情報)

tmp	
Chromosome_5_2615_2611_+	2611
Chromosome_2914_3033_120_+	120
Chromosome_3190_9498_6309_+	6309
Chromosome_9539_10270_732_+	732
Chromosome_10380_10495_116_+	116
Chromosome_10855_10945_91_+	91
Chromosome_11066_12348_1283_+	1283
Chromosome_12433_12645_213_+	213
Chromosome_12758_12856_99_+	99
Chromosome_12884_12974_91_+	91
Chromosome_13220_13310_91_+	91

Contents

- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) ：発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

カウント情報取得2

アノテーション情報(GFFファイル)を用いてカウントデータを得る一連のやり方を示します。

- マップ後 | 出力ファイルの読み込み | htSeqTools(Planet_2012) (last modified 2013/06/19)
- マップ後 | カウント情報取得 | について (last modified 2014/12/17)
- マップ後 | カウント情報取得 | single-end | ゲノム | アノテーション有 | QuasR(Gaidatzis_2015)
- マップ後 | カウント情報取得 | single-end | ゲノム | アノテーション無 | QuasR(Gaidatzis_2015)
- マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis_2015)
- マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション無 | QuasR(Gaidatzis_2015)
- マップ後 | カ
- マップ後 | 配
- 正規化 | に
- 正規化 | 基
- 正規化 | 基
- 正規化 | 基
- 正規化 | 基
- 正規化 | サ
- 正規化 | サ

①

マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis_2015)

NEW

②

QuasRパッ
での一連の
「ファイル」-「デ

1. mapping_pa

乳酸菌RNA-
から5'および
ルです。SRR
(Flicek et al.,
(Lactobacillus
(Lactobacillus
ルトです。

```
in_f1 <-
in_f2 <-
in_f3 <-
out_f <-
param_rep

#必要なパッ
library(Q
library(G
```

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させるやり方を示します。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定し
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge2_genelength.txt"       #出力ファイル名を指定してout_f2に格納
param_reportlevel <- "gene"            #カウントデータ取得時のレベルを指定："gene", "exon", "promoter",

#必要なパッケージをロード
library(QuasR)                         #パッケージの読み込み
library(GenomicFeatures)               #パッケージの読み込み

#前処理(アノテーション情報を取得)
#txdb <- makeTranscriptDbFromGFF(in_f3,#TranscriptDbオブジェクトを取得してtxdbに格納
#          format="gff3", useGenesAsTranscripts=T)#TranscriptDbオブジェクトを取得してtxdbに格納
txdb <- makeTxDbFromGFF(in_f3, format="gff3")#TranscriptDbオブジェクトを取得してtxdbに格納
txdb                                   #確認してるだけです

#本番(マッピング)
time_s <- proc.time()                  #計算時間を計測するため
out <- qAlign(in_f1, in_f2)            #マッピングを行うqAlign関数を実行した結果をoutに格納
time_e <- proc.time()                  #計算時間を計測するため
time_e - time_s                        #計算時間を表示(一番右側の数字。単位はsecond)
```

アノテーション情報(GFFファイル)の中身

カウント情報取得2

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させるやり方を示します。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定し
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
```

```
##gff-version 3
##sequence-region   Chromosome 1 2907892
```

Chromosome	ena	gene	1	1350	.	+	.	ID=gene:LCA12A_0617;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	1523	2662	.	+	.	ID=gene:LCA12A_0618;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	3240	3452	.	+	.	ID=gene:LCA12A_0619;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	3449	4564	.	+	.	ID=gene:LCA12A_0620;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	4817	6778	.	+	.	ID=gene:LCA12A_0621;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	6840	9461	.	+	.	ID=gene:LCA12A_0622;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	9566	10270	.	−	.	ID=gene:LCA12A_0623;assembly_name=GCA_000309565.2;b
Chromosome	ensembl	CDS	1	1350	.	+	0	ID=CDS:EKP96483;Parent=transcript:EKP96483;assembly_r
Chromosome	ensembl	exon	1	1350	.	+	.	Name=EKP96483-1;Parent=transcript:EKP96483;assembly
Chromosome	ensembl	CDS	1523	2662	.	+	0	ID=CDS:EKP96484;Parent=transcript:EKP96484;assembly_r
Chromosome	ensembl	exon	1523	2662	.	+	.	Name=EKP96484-1;Parent=transcript:EKP96484;assembly
Chromosome	ensembl	CDS	3240	3452	.	+	0	ID=CDS:EKP96485;Parent=transcript:EKP96485;assembly_r
Chromosome	ensembl	exon	3240	3452	.	+	.	Name=EKP96485-1;Parent=transcript:EKP96485;assembly
Chromosome	ensembl	CDS	3449	4564	.	+	0	ID=CDS:EKP96486;Parent=transcript:EKP96486;assembly_r
Chromosome	ensembl	exon	3449	4564	.	+	.	Name=EKP96486-1;Parent=transcript:EKP96486;assembly
Chromosome	ensembl	CDS	4817	6778	.	+	0	ID=CDS:EKP96487;Parent=transcript:EKP96487;assembly_r
Chromosome	ensembl	exon	4817	6778	.	+	.	Name=EKP96487-1;Parent=transcript:EKP96487;assembly
Chromosome	ensembl	CDS	6840	9461	.	+	0	ID=CDS:EKP96488;Parent=transcript:EKP96488;assembly_r
Chromosome	ensembl	exon	6840	9461	.	+	.	Name=EKP96488-1;Parent=transcript:EKP96488;assembly

カウント情報取得2

作業ディレクトリと入力ファイルの確認を行ってコピペ。約5分。Rを再起動するなどしておいたほうが、エラーが出たときにわかりやすいかも

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させるやり方を示します。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定し
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge2_genelength.txt"       #出力ファイル名を指定してout_f2に格納
param_reportlevel <- "gene"            #カウントデータ取得時のレベルを指定："gene", "exon", "promoter",

#必要なパッケージをロード
library(QuasR)                         #パッケージの読み込み
library(GenomicFeatures)               #パッケージの読み込み

#前処理(アノテーション情報を取得)
#txdb <- makeTranscriptDbFromGFF(in_f3,#TranscriptDbオブジェクトを取得してtxdbに格納
#          format="gff3", useGenesAsTranscripts=T)#TranscriptDbオブジェクトを取得してtxdbに格納
txdb <- makeTxDbFromGFF(in_f3, format="gff3")#TranscriptDbオブジェクトを取得してtxdbに格納
txdb                                   #確認してるだけです

#本番(マッピング
time_s <- proc
out <- qAlign(
time_e <- proc
time_e - time_
```

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/mapping_paired2"
> list.files()
[1] "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"
[2] "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"
[3] "mapping_paired_genome2.txt"
[4] "SRR616268sub_trim3_1.fastq.gz"
[5] "SRR616268sub_trim3_2.fastq.gz"
>
```

カウント情報取得2

途中経過。countオブジェクトの①1列目が配列長、②2列目がカウント情報なので、それをうまく振り分けて2つのファイルとして出力している。③遺伝子数は2,727個。

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させる例を示します。

```
alignmentStats(out)                                    #

#本番(カウントデータ取得)
count <- qCount(out, txdb, reportLevel=p
dim(count)                                             #
head(count)                                            #

#ファイルに保存(カウント情報)
data <- as.data.frame(count[,-1])                      #
tmp <- cbind(rownames(data), data)                     #
write.table(tmp, out_f1, sep="\t", apper

#ファイルに保存(遺伝子配列長情報)
genelength <- count[,1]                                #
tmp <- cbind(names(genelength), geneleng
write.table(tmp, out_f2, sep="\t", apper

#ファイルに保存(QCレポート用のpdfファイル作
out_f <- sub(".bam", "_QC.pdf", out@ali
qQCReport(out, pdfFilename=out_f)
out_f
```

③

R Console

```
    1. SRR616268sub_trim3_1_1f608b54dc4.bam

 Aux. alignments: none

> alignmentStats(out)                                  #マッピング結$
                      seqlength mapped unmapped
namae_paired:genome     2907892 693500   1303542
>
> #本番(カウントデータ取得)
> count <- qCount(out, txdb, reportLevel=param_reportle$
extracting gene regions from TxDb...done
counting alignments...done
collapsing counts by query name...done
> dim(count)                                           #行数と列数を$
[1] 2727    2
> head(count)                                          #確認してるだ$
           width namae_paired
LCA12A_0001   549           40
LCA12A_0002  1719           99
LCA12A_0003  1209          479
LCA12A_0004  1029          303
LCA12A_0005  1014          108
LCA12A_0006   354           22
>
```

① ②

カウント情報取得2

2つの出力ファイルはこんな感じです。2,728行。

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させるやり方を示します。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定し
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge2_genelength.txt"       #出力ファイル名を指定してout_f2に格納
param_reportlevel <- "gene"            #カウントデータ取得時のレベルを指定："gene", "exon", "promoter",

#必要なパッケージをロード
library(QuasR)                         #パッケージの読み込み
library(GenomicFeatures)               #パッケージの読み込み

#前処理(アノテーション
#txdb <- makeTranscri
#          format="gf
txdb <- makeTxDbFromG
txdb

#本番(マッピング)
time_s <- proc.time()
out <- qAlign(in_f1,
time_e <- proc.time()
time_e - time_s
```

①hoge2_count.txt (カウント情報)

rownames(data)	count[, -1]
LCA12A_0001	40
LCA12A_0002	99
LCA12A_0003	479
LCA12A_0004	303
LCA12A_0005	108
LCA12A_0006	22
LCA12A_0007	151
LCA12A_0008	222
LCA12A_0009	26
LCA12A_0011	71
LCA12A_0012	2

②hoge2_genelength.txt (配列長情報)

	genelength
LCA12A_0001	549
LCA12A_0002	1719
LCA12A_0003	1209
LCA12A_0004	1029
LCA12A_0005	1014
LCA12A_0006	354
LCA12A_0007	1176
LCA12A_0008	279
LCA12A_0009	141
LCA12A_0011	441
LCA12A_0012	1245

カウント情報取得2

アノテーションファイル中の順番通りにgene IDが並んでいるわけではないようだ。ゲノム配列のアップデートにもよるのだろう

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させるやり方を示します。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定し
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
```

```
##gff-version 3
##sequence-region   Chromosome 1 2907892
```

Chromosome	ena	gene	1	1350	.	+	.	ID=gene:LCA12A_0617;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	1523	2662	.	+	.	ID=gene:LCA12A_0618;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	3240	3452	.	+	.	ID=gene:LCA12A_0619;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	3449	4564	.	+	.	ID=gene:LCA12A_0620;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	4817	6778	.	+	.	ID=gene:LCA12A_0621;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	6840	9461	.	+	.	ID=gene:LCA12A_0622;assembly_name=GCA_000309565.2;b
Chromosome	ena	gene	9566	10270	.	−	.	ID=gene:LCA12A_0623;assembly_name=GCA_000309565.2;b
Chromosome	ensembl	CDS	1	1350	.	+	0	ID=CDS:EKP96483;Parent=transcript:EKP96483;assembly_n
Chromosome	ensembl	exon	1	1350	.	+	.	Name=EKP96483-1;Parent=transcript:EKP96483;assembly
Chromosome	ensembl	CDS	1523	2662	.	+	0	ID=CDS:EKP96484;Parent=transcript:EKP96484;assembly_n
Chromosome	ensembl	exon	1523	2662	.	+	.	Name=EKP96484-1;Parent=transcript:EKP96484;assembly
Chromosome	ensembl	CDS	3240	3452	.	+	0	ID=CDS:EKP96485;Parent=transcript:EKP96485;assembly_n
Chromosome	ensembl	exon	3240	3452	.	+	.	Name=EKP96485-1;Parent=transcript:EKP96485;assembly
Chromosome	ensembl	CDS	3449	4564	.	+	0	ID=CDS:EKP96486;Parent=transcript:EKP96486;assembly_n
Chromosome	ensembl	exon	3449	4564	.	+	.	Name=EKP96486-1;Parent=transcript:EKP96486;assembly
Chromosome	ensembl	CDS	4817	6778	.	+	0	ID=CDS:EKP96487;Parent=transcript:EKP96487;assembly_n
Chromosome	ensembl	exon	4817	6778	.	+	.	Name=EKP96487-1;Parent=transcript:EKP96487;assembly
Chromosome	ensembl	CDS	6840	9461	.	+	0	ID=CDS:EKP96488;Parent=transcript:EKP96488;assembly_n
Chromosome	ensembl	exon	6840	9461	.	+	.	Name=EKP96488-1;Parent=transcript:EKP96488;assembly

	genelength
LCA12A_0001	549
LCA12A_0002	1719
LCA12A_0003	1209
LCA12A_0004	1029
LCA12A_0005	1014
LCA12A_0006	354
LCA12A_0007	1176
LCA12A_0008	279
LCA12A_0009	141
LCA12A_0011	441
LCA12A_0012	1245

カウント情報取得2

アノテーションファイル中でのLCA12A_0001検索結果。バクテリアの場合はgene = exon = transcriptなので分かりやすい。

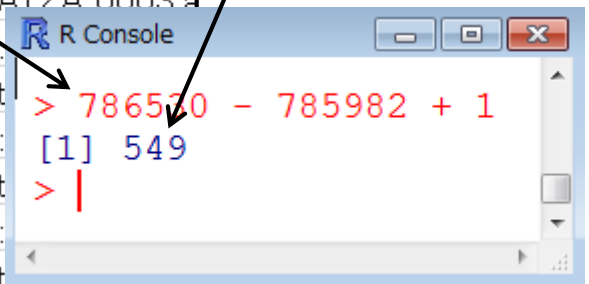
2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の出力ファイルは2列目が配列長、3列目がカウント情報でしたが、これを2つのファイルに分割して出力させるやり方を示します。

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge2_genelength.txt"       #出力ファイル名を指定してout_f2に格納
param_reportlevel <- "gene"            #カウントデータ取得時のレベルを指定："gene", "exon", "promot
```

#ena	tRNA_gene	785625	785711	.	+	.	ID=gene:LCA12A_2737;assembly_name=GCA_00
ena	gene	785982	786530	.	−	.	ID=gene:LCA12A_0001;assembly_name=GCA_00
ena	gene	786908	788626	.	+	.	ID=gene:LCA12A_0002;assembly_name=GCA_00
#ena	gene	788913	790121	.	+	.	ID=gene:LCA12A_0003;assembly_name=GCA_00
#ensembl	CDS	785982	786530	.	−	0	ID=CDS:EKQ00713;Parent=transcript:EKQ0071
ensembl	exon	785982	786530	.	−	.	Name=EKQ00713-1;Parent=transcript:EKQ0071
ensembl	CDS	786908	788626	.	+	0	ID=CDS:EKQ00714;Parent=transcript:EKQ00714;asse
#ensembl	exon	786908	788626	.	+	.	Name=EKQ00714-1;Parent=transcript:EKQ00714;ass
ensembl	CDS	788913	790121	.	+	0	ID=CDS:EKQ00715;Parent=transcript:EKQ00715;asse
ensembl	exon	788913	790121	.	+	.	Name=EKQ00715-1;Parent=transcript:EKQ00715;ass
ena	transcript	785982	786530	.	−	.	ID=transcript:EKQ00713;Parent=gene:LCA12A_0001;a
ena	transcript	786908	788626	.	+	.	ID=transcript:EKQ00714;Parent=gene:LCA12A_0002;a
ena	transcript	788913	790121	.	+	.	ID=transcript:EKQ00715;Parent=gene:LCA12A_0003;a
ena	rRNA	780180	783097	.	+	.	ID=transcript:LCA12A_2711;Parent=gene:
ensembl	exon	780180	783097	.	+	.	Name=LCA12A_2711-1;Parent=transcript
ena	rRNA	783220	783336	.	+	.	ID=transcript:LCA12A_2712;Parent=gene:
ensembl	exon	783220	783336	.	+	.	Name=LCA12A_2712-1;Parent=transcript
ena	transcript	783345	783417	.	+	.	ID=transcript:LCA12A_2713;Parent=gene:
ensembl	exon	783345	783417	.	+	.	Name=LCA12A_2713-1;Parent=transcript
ena	transcript	783420	783492	.	+	.	ID=transcript:LCA12A_2714;Parent=gene:LCA12A_27

	genelength
LCA12A_0001	549
LCA12A_0002	1719
LCA12A_0003	1209
LCA12A_0004	1029
LCA12A_0005	1014
LCA12A_0006	354
LCA12A_0007	1176
LCA12A_0008	279
LCA12A_0009	141
LCA12A_0011	441
LCA12A_0013	1245



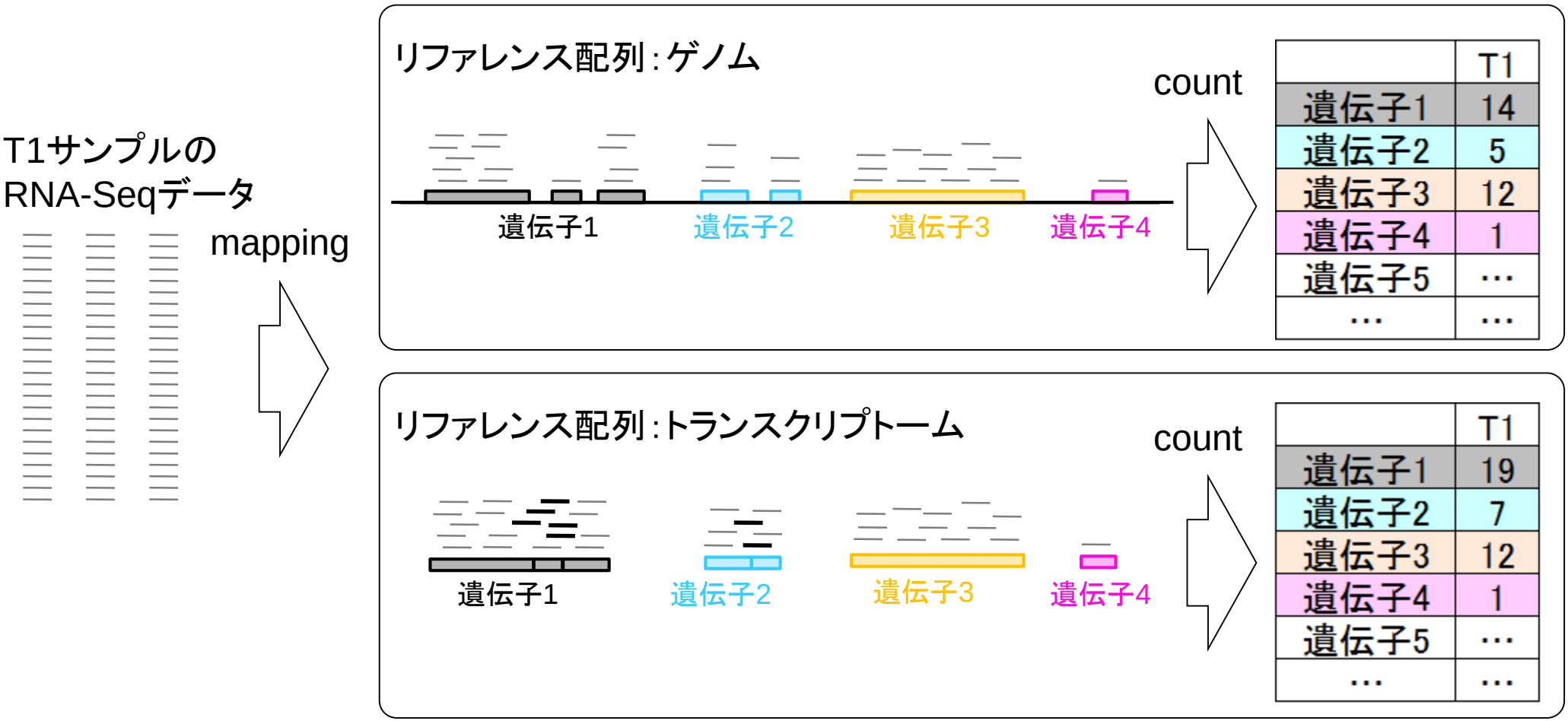
Contents

- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) : 発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

目的に応じた正規化

マップされたリード数のカウント情報は、発現量推定の基本情報だが、「マップされたリード数 = 発現量」ではない

- 基本的なマッピングプログラム（bowtieなど）を用いた場合



目的に応じた正規化

目的によってやっておかねばならない正規化（と実質的に影響がないのでやってもよい正規化）がある

- トランスクリプトーム配列取得
  - ゲノム配列既知の場合：Cufflinksなどを用いて遺伝子構造推定（アノテーション）
  - ゲノム配列未知の場合：Rockhopperなどのトランスクリプトーム用アセンブラを実行
- 遺伝子または転写物(isoform)ごとの発現量の正確な推定
  - SailfishやRNA-Skimなどを利用して発現量情報を得る
  - ある特定のサンプル内での遺伝子間の発現量の大小関係を知りたい
  - 配列長やGC biasなどの各種補正がポイント
- 比較するサンプル間で発現変動している遺伝子または転写物の同定
  - TCCパッケージなどを利用して発現変動遺伝子（DEG）を得る
  - ライブラリサイズ（総リード数）や発現している遺伝子の組成の補正がポイント
  - （GO解析など）DEG結果を用いる多くの下流解析結果に影響を及ぼす

遺伝子間比較

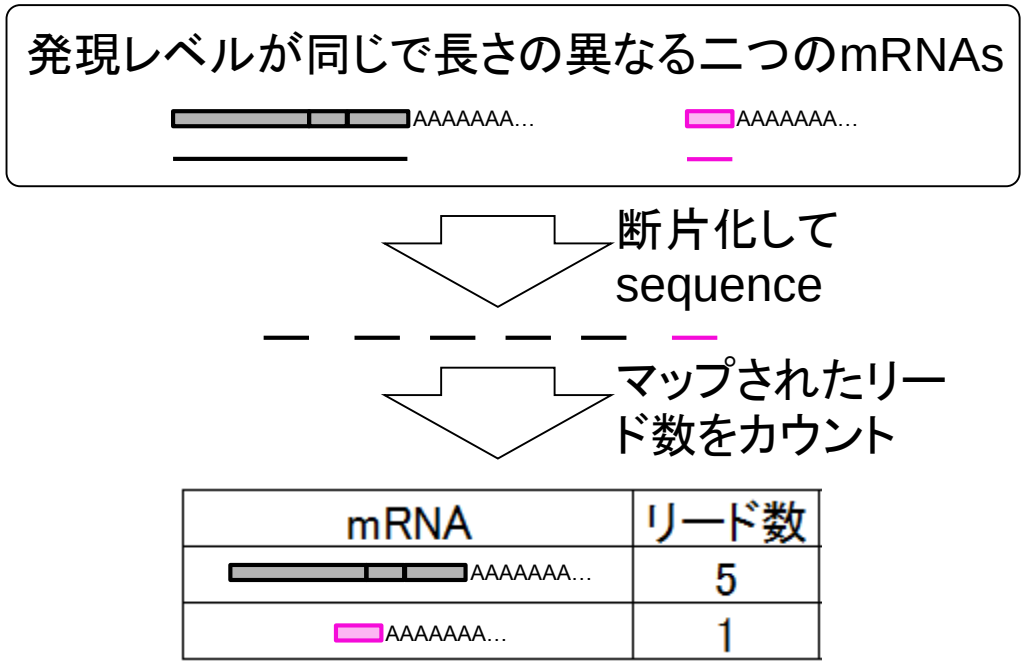
総リード数(ライブラリサイズ or sequence depth)補正は不必要
理由:遺伝子間の発現レベルの大小関係は定数倍しても不変

- 発現量補正の基本形: $\text{カウント数} \times \frac{\text{定数}}{\text{配列長} \times \text{総リード数}}$
  - RPK (Reads per kilobase)
  - RPM (Reads per million)
  - RPKM (Reads per kilobase per million)
- 同一サンプル内での異なる遺伝子間の発現レベル比較の場合
  - 配列長由来bias:長いほど沢山sequenceされる
    - RPKMやFPKMなどの配列長を考慮して正規化されたデータで解析
  - GC含量由来bias:カウント数の分布がGC含量依存的である
    - Risso et al., *BMC Bioinformatics*, **12**: 480, 2011
    - Benjamini and Speed, *Nucleic Acids Res.*, **40**: e72, 2012
    - Filloux et al., *BMC Bioinformatics*, **15**: 188, 2014

配列長の補正

1つのサンプル内で異なる遺伝子間の発現レベルの大小関係を配列長を考慮せずに比較することはできない

- 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らないように補正すべき



mRNA	リード数
AAAAAAA...	5
AAAAAAA...	1

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

カウント情報取得2

①LCA12A_0004と②LCA12A_0008の遺伝子間比較で考える。カウント数の比較だけだと ①のほうが多い(303 > 222)。しかし、配列長も考慮すると(1029 > 279)、**塩基あたりのリード数**が多いのは②。

2. mapping_paired_genome2.txt中のFASTQ形式ファイルを乳酸菌ゲノムにマッピングする場合:

1.の 出力ファイルは2列目が配列長、3列目がカウント 情報でしたが、これを2つのファイルに分割し

```
in_f1 <- "mapping_paired_genome2.txt"  #入力ファイル名を指定してin_f1に格納(RNA-seqリストファイル)
in_f2 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.dna.chromosome.Chromosome.fa"#入力ファイル名を指定
in_f3 <- "Lactobacillus_casei_12a.GCA_000309565.2.25.chromosome.Chromosome.gff3"#入力ファイル名を指定し
out_f1 <- "hoge2_count.txt"            #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge2_genelength.txt"       #出力ファイル名を指定してout_f2に格納
param_reportlevel <- "gene"            #カウントデータ取得時のレベルを指定："gene", "exon", "promoter",

#必要なパッケージをロード
library(QuasR)                         #パッケージの読み込み
library(GenomicFeatures)               #パッケージの読み込み

#前処理(アノテーション
#txdb <- makeTranscri
#          format="gf
txdb <- makeTxDbFromG
txdb

#本番(マッピング)
time_s <- proc.time()
out <- qAlign(in_f1,
time_e <- proc.time()
time_e - time_s
```

hoge2_count.txt (カウント情報)

rownames(data)	count[, -1]
LCA12A_0001	40
LCA12A_0002	99
LCA12A_0003	479
LCA12A_0004	303
LCA12A_0005	108
LCA12A_0006	22
LCA12A_0007	151
LCA12A_0008	222
LCA12A_0009	26
LCA12A_0011	71
LCA12A_0012	2

hoge2_genelength.txt (配列長情報)

	genelength
LCA12A_0001	549
LCA12A_0002	1719
LCA12A_0003	1209
LCA12A_0004	1029
LCA12A_0005	1014
LCA12A_0006	354
LCA12A_0007	1176
LCA12A_0008	279
LCA12A_0009	141
LCA12A_0011	441
LCA12A_0012	1245

配列長とカウント数の関係

配列長(横軸)が長くなるほどカウント数(縦軸)が増える傾向にあるかを実データで眺める

- マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション無 | QuasR(Gaidatzis_2015) (last mo
- マップ後 | カウント情報取得 | トランスクリプトーム | BEDファイルから (last modified 2014/06/21)
- マップ後 | 配列長とカウント数の関係 (① modified 2015/07/03) NEW
- 正規化 | について (last modified 2014/0 22)
- 正規化 | 基礎 | RPK or CPK (配列長補正) (last modified 2015/07/04) NEW
- 正規化 | 基礎
- 正規化 | 基礎
- 正規化 | 基礎
- 正規化 | サン
- 正規化 | サン
- 正規化 | サン
- 正規化 | サン
- 正規化 | サン
- 正規化 | サン
- 正規化 | サン

マップ後 | 配列長とカウント数の関係 NEW

RNA-seqデータは、原理的に配列長が長い転写物ほどその断片配列のリード数が多い傾向にあります。ここではそれを眺めます。201 年7月3日に、boxplotで示すために param個で分割(20分割など)するテクニックとして「floor (nrow(data)/param)+1 (nrow(data)/param)」に
「ファイル」-「ディレクトリの

1. 配列長とカウント情報

「マップ後 | カウント情報
長、縦軸:カウント数の
log10にしています。した
警告が出ます)

```
in_f <- "sample_l

#入力ファイルの読み
data <- read.tabl
head(data)

#本番(散布図)
plot(data, log="x
```

② **9. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_genelen2.txt)が別々の場合:**

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis_2015)」の例題2実行結果ファイル(と同じ)です。横軸:配列長、縦軸:カウント数のboxplot(箱ひげ図)をpng形式ファイルで保存したい場合です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウント情報ファイル
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(配列長情報ファイル)
out_f <- "hoge9.png"                    #出力ファイル名を指定してout_fに格納
param_bin <- 20                         #boxplotを描くときの水準数(分割数に相当)を指定
param_fig <- c(600, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#入力ファイルの読み込み
hoge <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファ
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")#in_f2で指定したファイ
data <- cbind(len, hoge)                #配列長ベクトルとカウントベクトルを列方向で結合した結果を

#前処理(ゼロカウントデータのフィルタリング)
data <- data[data[,2]>0,]               #カウントが0より大きい行のみ抽出した結果をdataに格納

#前処理(配列長の短い順にソート)
data <- data[order(data[,1]),]          #ソート

#前処理(dataを分割するためのグループラベル情報作成)
f <- gl(param_bin, ceiling(nrow(data)/param_bin), nrow(data))#nrow(data)の要素数からなるベク

#本番(ファイルに保存)
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメー
```

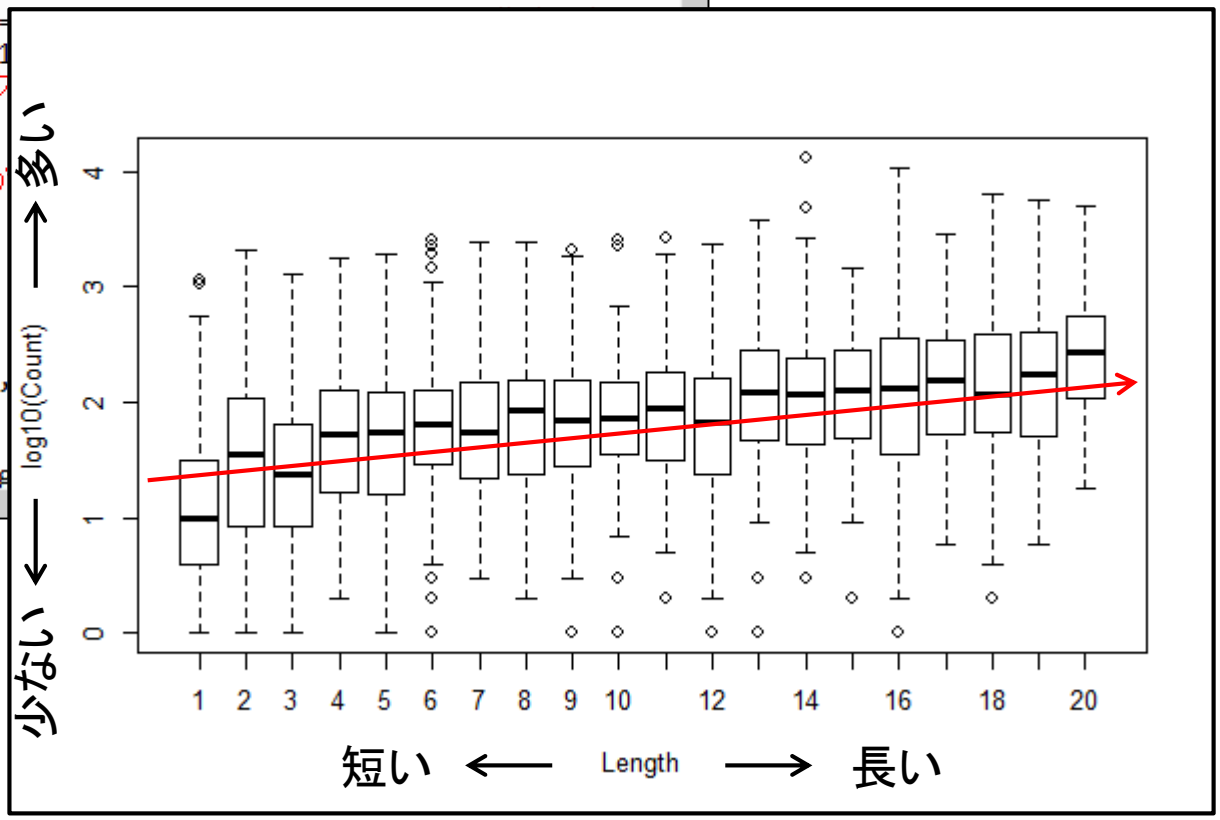
配列長とカウント数の関係

配列長(横軸)が長くなるほどカウント数(縦軸)が増える傾向にあることがわかる。

9. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_genelen2.txt)が別々の場合:

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015)」の例題2実行結果ファイル(と同じ)です。横軸:配列長、縦軸:カウント数のboxplot(箱ひげ図)をpng形式ファイルで保存したい場合です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウント情報ファイル
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(配列長情報ファイル)
out_f <- "hoge9.png"                    #出力ファイル名を指定してout_fに格納
param_bin <- 20                         #boxplotを描くときの水準数(分割数に相当)を指定
param_fig <- c(600, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#入力ファイルの読み込み
hoge <- read.table(in_f1, header=TRUE, row.names=
len <- read.table(in_f2, header=TRUE, row.names=1
data <- cbind(len, hoge)                #配列長ベク

#前処理(ゼロカウントデータのフィルタリング)
data <- data[data[,2]>0,]               #カウントが

#前処理(配列長の短い順にソート)
data <- data[order(data[,1]),]          #ソート

#前処理(dataを分割するためのグループラベル情報作成)
f <- gl(param_bin, ceiling(nrow(data)/param_bin),

#本番(ファイルに保存)
png(out_f, pointsize=13, width=param_fig[1], heig
```



配列長の補正

RPK補正(配列長が100 bp, one kilobase だったとき)後のリード数を計算してみよう

- マップ後 | カウント情報取得 | トランスクリプトーム | BEDファイルから (last modified 2
- マップ後 | 配列長とカウント数の関係 (last modified 2015/07/03) NEW
- 正規化 | について (last modified 2014/06/22)
- 正規化 | 基礎 | RPK or CPK (配列長補正) (last modified 2015/07/04) NEW
- 正規化 | 基礎 | RPM or CPM (総リード数補正) (last modified 2015/03/30)
- 正規化 | 基礎 | RPKM | トランスクリプトーム (last modified 2013/06/23)
- 正規化 | 基礎 | R
- 正規化 | サンプル
- 正規化 | サンプル
- 正規化 | サンプル
- 正規化 | サンプル
- 正規化 | サンプル
- 正規化 | サンプル
- 正規化 | サンプル

①

正規化 | 基礎 | RPK or CPK (配列長補正) NEW

ここでは、遺伝子(転写物)ごとのリード数を「配列長が1000 bp (one kilobase)だったときのリード数; Reads per kilobase (RPK)」に変換するやり方を示します。「リード数＝カウント数」なのでReadsのところをCountsに置き換えた表現(Counts per kilo
「ファイル」−「ディレク

②

1. 配列長とカウント情報

基本形です。

```
in_f <- "sample_
out_f <- "hoge1.t
param <- 1000

#入力ファイルの読み
data <- read.tabl
head(data)

#本番(正規化)
nf <- param/data[
out <- data[,2] *

#ファイルに保存
tmp <- cbind(rown
```

4. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_genelen2.txt)が別々の場合:

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015)」の例題2実行結果ファイル(と同じ)です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウントーデータ
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(長さ情報ファイル)
out_f <- "hoge4.txt"                    #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したフ
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")#in_f2で指定したファ
dim(data)                               #行数と列数を表示
dim(len)                                #行数と列数を表示
head(data, n=8)                         #確認してるだけです
head(len, n=8)                          #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPK正規化)
nf <- 1000/len[,1]                      #正規化係数(RPK補正用)を計算した結果をnfに格納
data <- sweep(data, 1, nf, "*")         #正規化係数を各行に掛けた結果をdataに格納
head(data, n=8)                         #確認してるだけです

#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(data), data)      #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイ
```

配列長の補正

①LCA12A_0004と②LCA12A_0008の遺伝子間比較で考える。カウント数の比較だけだと ①のほうが多い(303 > 222)。しかし、配列長も考慮すると(1029 > 279)、**塩基あたりのリード数**が多いのは②。

4. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_gene

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015)」の
ル(と同じ)です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウントーデ
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(長さ情報ファ
out_f <- "hoge4.txt"                    #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定し
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")#in_f2で指定した
dim(data)                               #行数と列数を表示
dim(len)                                #行数と列数を表示
head(data, n=8)                         #確認してるだけです
head(len, n=8)                          #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPK正規化)
nf <- 1000/len[,1]                      #正規化係数(RPK補正用)を計算した結果をnfに格納
data <- sweep(data, 1, nf, "*")         #正規化係数を各行に掛けた結果をdataに格納
head(data, n=8)                         #確認してるだけです

#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(data), data)      #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定した
```

```
> dim(len)
[1] 2727    1
> head(data, n=8)
             count....1.
LCA12A_0001          40
LCA12A_0002          99
LCA12A_0003         479
LCA12A_0004         303   ①
LCA12A_0005         108
LCA12A_0006          22
LCA12A_0007         151
LCA12A_0008         222   ②
> head(len, n=8)
             genelength
LCA12A_0001        549
LCA12A_0002       1719
LCA12A_0003       1209
LCA12A_0004       1029   ①
LCA12A_0005       1014
LCA12A_0006        354
LCA12A_0007       1176
LCA12A_0008        279   ②
> colSums(data)
count....1.
     595386
```

配列長の補正

①LCA12A_0004と②LCA12A_0008の遺伝子間比較で考える。カウント数の比較だけだと ①のほうが多い(303 > 222)。しかし、配列長も考慮すると(1029 > 279)、1000塩基あたりのリード数が多いのは②

4. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_gene

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015)」の
ル(と同じ)です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウントデータ
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(長さ情報ファイル)
out_f <- "hoge4.txt"                    #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定し
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")#in_f2で指定した
dim(data)                               #行数と列数を表示
dim(len)                                #行数と列数を表示
head(data, n=8)                         #確認してるだけです
head(len, n=8)                          #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPK正規化)
nf <- 1000/len[,1]                      #正規化係数(RPK補正用)を計算した結果をnfに格納
data <- sweep(data, 1, nf, "*")         #正規化係数を各行に掛けた結果をdataに格納
head(data, n=8)                         #確認してるだけです

#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(data), data)      #保存したい情報をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定した
```

R Console

```
> #本番(RPK正規化)
> nf <- 1000/len[,1]      $
> data <- sweep(data, 1, $
> head(data, n=8)         $
             count....1.
LCA12A_0001    72.85974
LCA12A_0002    57.59162
LCA12A_0003   396.19520
LCA12A_0004   294.46064   ①
LCA12A_0005   106.50888
LCA12A_0006    62.14689
LCA12A_0007   128.40136
LCA12A_0008   795.69892   ②
>
> #ファイルに保存(テキス$
> tmp <- cbind(rownames(d$
> write.table(tmp, out_f,$
>
```

配列長の補正

配列長(RPK)補正を行うことで、遺伝子間の発現レベルの大小関係を調べることができる

4. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_genelen2.txt)が別々の場合:

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015)」の例題2実行結果ファイル(と同じ)です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウントデータ
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(長さ情報ファイル)
out_f <- "hoge4.txt"                    #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したフ
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")#in_f2で指定したファ
dim(data)                               #行数と列数を表示
dim(len)                                #行数と列数を表示
head(data, n=8)                         #確認してるだけです
head(len, n=
colSums(data

#本番(RPK正規
nf <- 1000/1
data <- swee
head(data, n

#ファイルに保
tmp <- cbind
write.table(
```

R Console

```
> 303 * 1000/1029
[1] 294.4606
> 222 * 1000/279
[1] 795.6989
>
```

入力1:カウントデータ

rownames(data)	count[, -1]
LCA12A_0001	40
LCA12A_0002	99
LCA12A_0003	479
LCA12A_0004	303
LCA12A_0005	108
LCA12A_0006	22
LCA12A_0007	151
LCA12A_0008	222
LCA12A_0009	26
LCA12A_0011	71
LCA12A_0012	2

入力2:配列長情報

	genelength
LCA12A_0001	549
LCA12A_0002	1719
LCA12A_0003	1209
LCA12A_0004	1029
LCA12A_0005	1014
LCA12A_0006	354
LCA12A_0007	1176
LCA12A_0008	279
LCA12A_0009	141
LCA12A_0011	441
LCA12A_0012	1245

出力:hoge4.txt

rownames(data)	count....1.
LCA12A_0001	72.860
LCA12A_0002	57.592
LCA12A_0003	396.195
LCA12A_0004	294.461
LCA12A_0005	106.509
LCA12A_0006	62.147
LCA12A_0007	128.401
LCA12A_0008	795.699
LCA12A_0009	184.397
LCA12A_0011	160.998
LCA12A_0012	1.606

①

②

教科書p136-137

RPKMは、配列長(RPK)補正と総リード数(RPM)補正を同時に行ったもの。

RPKM

- マップ後 | 配列長とカウント数の関係 (last modified 2015/07/03) NEW
- 正規化 | について (last modified 2014/06/22)
- 正規化 | 基礎 | RPK or CPK (配列長補正) (last modified 2015/07/04) NEW
- 正規化 | 基礎 | RPM or CPM (総リード数補正) (last modified 2015/03/30)
- 正規化 | 基礎 | RPKM (last modified 2015/07/04) NEW
- 正規化 | サンプル内 | ELSeq(Risso 2011) (last modified 2013/06/24)
- 正規化 | サンプ
- 正規化 | サンプ
- 正規化 | サンプ
- 正規化 | サンプ
- 正規化 | サンプ
- 正規化 | サンプ

①

正規化 | 基礎 | RPKM NEW

遺伝子(転写物)ごとのリード数を「配列長が1000 bp (kilobase)で総リード数が100万だったときのリード数: Reads per kilobase per mill

②

「ファイル」-「デ

1. 配列長とカウント

1-3列目がそれぞれ

```
in_f <- "sampl
out_f <- "hoge

#入力ファイルの
data <- read.t
head(data)
sum(data[,2])

#本番(正規化)
nf_RPM <- 1000
nf_RPK <- 1000
data[,2] <- da
head(data)

#ファイルに保存
```

3. カウントデータファイル(SRR616268sub_count2.txt)と長さ情報ファイル(SRR616268sub_genelen2.txt)が別々の場合:

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノテーション有 | QuasR(Gaidatzis 2015)」の例題2実行結果ファイル(と同じ)です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1に格納(カウントデータファ
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2に格納(長さ情報ファイル)
out_f <- "hoge3.txt"                    #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファ
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")#in_f2で指定したファ
dim(data)                               #行数と列数を表示
dim(len)                                #行数と列数を表示
head(data)                              #確認してるだけです
head(len)                               #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPM正規化)
nf_RPM <- 1000000/colSums(data)         #正規化係数(RPM補正用)を計算した結果をnf_RPMに格納
data <- sweep(data, 2, nf_RPM, "*")     #正規化係数を各列に掛けた結果をdataに格納
head(data)                              #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPK正規化)
nf_RPK <- 1000/len[,1]                  #正規化係数(RPK補正用)を計算した結果をnf_RPKに格納
data <- sweep(data, 1, nf_RPK, "*")     #正規化係数を各行に掛けた結果をdataに格納
```

RPKM

①入力データの(マップされた)総リード数は595,386。RPM補正はサンプル間比較を可能にすることを目的としており、具体的な作業は②総リード数が100万になるような補正を行うこと。そのため、③正規化係数を1,000,000 / 595,386 = 1.679583として得ておき、それを各遺伝子に対して掛けることでRPM正規化を実現している。

3. カウントデータファイル(SRR616268sub_count2.txt)と

「マップ後 | カウント情報取得 | paired-end | ゲノム | アノ
(と同じ)です。

```
in_f1 <- "SRR616268sub_count2.txt"      #入力ファイル名を指定してin_f1
in_f2 <- "SRR616268sub_genelen2.txt"    #入力ファイル名を指定してin_f2
out_f <- "hoge3.txt"                    #出力ファイル名を指定してout_f

#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote
len <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote=
dim(data)                               #行数と列数を表示
dim(len)                                #行数と列数を表示
head(data)                              #確認してるだけです
head(len)                               #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPM正規化)
nf_RPM <- 1000000/colSums(data)         #正規化係数(RPM補正用)を計算し
data <- sweep(data, 2, nf_RPM, "*")     #正規化係数を各列に掛けた結果を
head(data)                              #確認してるだけです
colSums(data)                           #列ごとの総リード数を表示

#本番(RPK正規化)
nf_RPK <- 1000/len[,1]                  #正規化係数(RPK補正用)を計算し
data <- sweep(data, 1, nf_RPK, "*")     #正規化係数を各行に掛けた結果を
```

R Console

```
> colSums(data)
count....1.
     595386 ①
>
> #本番(RPM正規化)
> nf_RPM <- 1000000/colSums(data)
> data <- sweep(data, 2, nf_RPM, "*")
> head(data)
              count....1.
LCA12A_0001     67.18331
LCA12A_0002    166.27868
LCA12A_0003    804.52009
LCA12A_0004    508.91355
LCA12A_0005    181.39493
LCA12A_0006     36.95082
> colSums(data)
count....1.
      1e+06 ②
> nf_RPM
count....1.
   1.679583 ③
>
```

サンプル間比較

配列長やGC bias補正は理論上はおそらく不必要。理由は、同一遺伝子に対して掛かる係数はサンプル間で同じだから。

- 発現量補正の基本形：カウント数×$\frac{\text{定数}}{\text{配列長}\times\text{総リード数}}$
  - RPK (Reads per kilobase)
  - RPM (Reads per million)
  - RPKM (Reads per kilobase per million)
- 異なるサンプル間での同一遺伝子間の発現レベル比較の場合
  - 総リード数の違い：総リード数が*x*倍違うと全体的に*x*倍変動…
    - RPM正規化で全体を揃えることは基本
  - 組成の違い：サンプル特異的高発現遺伝子の存在で比較困難に…
    - TMM正規化法(Robinson and Oshlack, *Genome Biol.*, **11**: R25, 2010)
    - TbT正規化法(Kadota et al., *Algorithms Mol. Biol.*, **7**: 5, 2012)
    - DEGESに基づく正規化法(Sun et al., *BMC Bioinformatics*, **14**: 219, 2013)

グローバル正規化

- 「各サンプルから測定されたmRNAの全体量は一定」と仮定
  - アレイ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

	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
平均値	7.4	8.1

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
平均値	10.0	10.0

チップごとに独立して正規化（per-array basis）。他のアレイの影響を受けない。補正後の平均値を10にしたい場合は、sample1の正規化係数 = 10/7.4、sample2の正規化係数 = 10/8.1とする。RNA-seqの補正法であるRPM (RPKMの一部)も基本的に同じ考え方。サンプルごとの総カウント数を100万に揃えたいので、正規化係数 = 1,000,000/補正前の総リード数としているだけ。尚、総和(sum)と平均(mean)は数学的には等価。「機能ゲノム学」第2回(2015.05.19)より。

Contents

- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) : 発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

データ取得

GEOのサイトからもカウントデータが提供されている。GSE17274で検索。

- (削除予定)個別パッケージのインストール (last modified 2015/02/20)
- 基本的な利用法 (last modified 2015/04/03)
- サンプルデータ (last modified 2015/06/15) NEW
- バイオインフォマティクス人材育成カリキュラム(次世代シークエンサ) | NGSハンズオン
- 書籍
- 書籍
- 書籍

①

サンプルデータ NEW

②

41. Blekhman et al., Genome Res., 2010のリアルカウントデータです。Supplementary Table1で提供されているエクセルファイル (http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls; 約4.3MB) からカウントデータのみ抽出し、きれいに整形しなおしたものがここでの出力ファイルになります。20,689 genes×36 samplesのカウントデータ(sample_blekhman_36.txt)です。実験デザインの詳細はFigure S1中に描かれていますが、ヒト(Homo Sapiens; HS), チンパンジー(Pan troglodytes; PT), アカゲザル(Rhesus macaque; RM)の3種類の生物種の肝臓サンプル(liver sample)の比較を行っています。生物種ごとにオス3個体メス3個体の計6個体使われており(six individuals; six biological replicates)、技術的なばらつき(technical variation)を見積もるべく各個体は2つに分割されてデータが取得されています(duplicates; two technical replicates)。それゆえ、ヒト12サンプル、チンパンジー12サンプル、アカゲザル12サンプルの計36サンプル分のデータということになります。以下で行っていることはカウントデータの列のみ「ヒトのメス(HSF1, HSF2, HSF3)」,「ヒトのオス(HSM1, HSM2, HSM3)」,「チンパンジーのメス(PTF1, PTF2, PTF3)」,「チンパンジーのオス(PTM1, PTM2, PTM3)」,「アカゲザルのメス(RMF1, RMF2, RMF3)」,「アカゲザルのオス(RMM1, RMM2, RMM3)」の順番で並び替えたものをファイルに保存しています。もう少し美しくやることも原理的には可能ですが、そこは本質的な部分ではありませんので、ここではアドホック(その場しのぎ、の意味)な手順で行っています。当然ながら、エクセルなどでファイルの中身を眺めて完全に列名を把握しているという前提です。
尚、"R1L4.HSF1"と"R4L2.HSF1"が「HSF1というヒトのメス一個体のtechnical replicates」であることは列名や文脈から読み解けます。

```
#in_f <- "http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls"#入力ファイ
in_f <- "suppTable1.xls"                    #入力ファイル名を指定してin_fに格納
out_f <- "sample_blekhman_36.txt"           #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
hoge <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイルの読み込み
dim(hoge)                                   #行数と列数を表示

#サブセットの取得
data <- cbind(                              #必要な列名の情報を取得したい列の順番で結合した結果をdataに格納
  hoge$R1L4.HSF1, hoge$R4L2.HSF1, hoge$R2L7.HSF2, hoge$R3L2.HSF2, hoge$R8L1.HSF3, hoge$R8L2.HSF3,
  hoge$R1L1.HSM1, hoge$R5L2.HSM1, hoge$R2L3.HSM2, hoge$R4L8.HSM2, hoge$R3L6.HSM3, hoge$R4L1.HSM3,
  hoge$R1L2.PTF1, hoge$R4L4.PTF1, hoge$R2L4.PTF2, hoge$R6L6.PTF2, hoge$R3L7.PTF3, hoge$R5L3.PTF3,
  hoge$R1L6.PTM1, hoge$R3L3.PTM1, hoge$R2L8.PTM2, hoge$R4L6.PTM2, hoge$R6L2.PTM3, hoge$R6L4.PTM3,
  hoge$R1L7.RMF1, hoge$R5L1.RMF1, hoge$R2L2.RMF2, hoge$R5L8.RMF2, hoge$R3L4.RMF3, hoge$R4L7.RMF3,
  hoge$R1L3.RMM1, hoge$R3L8.RMM1, hoge$R2L6.RMM2, hoge$R5L4.RMM2, hoge$R3L1.RMM3, hoge$R4L3.RMM3)
```

サブセット抽出と整形

①xls形式のエクセルファイルを通常の手順で読み込むことができる。②それほど大きなサイズでなければ、ネットワーク経由で直接読み込むこともできる。③約4.5MB。

41. Blekhman et al., Genome Res., 2010のリアルカウントデータです。Supplementary Table1で提
(http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls; 約4.3
に整形しなおしたものがここでの出力ファイルになります。20,689 genes×36 samplesのカウントデータ(sample_blekhman_36.txt)です。実験デザインの詳細はFigure S1中に描かれていますが、ヒト(Homo Sapiens; HS), チンパンジー(Pan troglodytes; PT), アカゲザル(Rhesus macaque; RM)の3種類の生物種の肝臓サンプル(liver sample)の比較を行っています。生物種ごとにオス3個体メス3個体の計6個体使われており(six individuals; six biological replicates)、技術的なばらつき(technical variation)を見積もるべく各個体は2つに分割されてデータが取得されています(duplicates; two technical replicates)。それゆえ、ヒト12サンプル、チンパンジー12サンプル、アカゲザル12サンプルの計36サンプル分のデータということになります。以下で行っていることはカウントデータの列のみ「ヒトのメス(HSF1, HSF2, HSF3)」,「ヒトのオス(HSM1, HSM2, HSM3)」,「チンパンジーのメス(PTF1, PTF2, PTF3)」,「チンパンジーのオス(PTM1, PTM2, PTM3)」,「アカゲザルのメス(RMF1, RMF2, RMF3)」,「アカゲザルのオス(RMM1, RMM2, RMM3)」の順番で並び替えたものをファイルに保存しています。もう少し美しくやることも原理的には可能ですが、そこは本質的な部分ではありませんので、ここではアドホック(その場しのぎ、の意味)な手順で行っています。当然ながら、エクセルなどでファイルの中身を眺めて完全に列名を把握しているという前提です。
尚、"R1L4.HSF1"と"R4L2.HSF1"が「HSF1というヒトのメス一個体のtechnical replicates」であることは列名や文脈から読み解けます。

①

②

```
#in_f <- "http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls"  #入力ファイ
in_f <- "suppTable1.xls"                    #入力ファイル名を指定してin_fに格納
out_f <- "sample_blekhman_36.txt"           #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
hoge <- read.table(in_f, header=TRUE, row.n
dim(hoge)                                   #行数

#サブセットの取得
data <- cbind(                              #必要
  hoge$R1L4.HSF1, hoge$R4L2.HSF1, hoge$R2L7
  hoge$R1L1.HSM1, hoge$R5L2.HSM1, hoge$R2L3
  hoge$R1L2.PTF1, hoge$R4L4.PTF1, hoge$R2L4
  hoge$R1L6.PTM1, hoge$R3L3.PTM1, hoge$R2L8
  hoge$R1L7.RMF1, hoge$R5L1.RMF1, hoge$R2L2
  hoge$R1L3.RMM1, hoge$R3L8.RMM1, hoge$R2L6
colnames(data) <- c(                        #列名
  "R1L4.HSF1", "R4L2.HSF1", "R2L7.HSF2", "R
  "R1L1.HSM1", "R5L2.HSM1", "R2L3.HSM2", "R
  "R1L2.PTF1", "R4L4.PTF1", "R2L4.PTF2", "R
  "R1L6.PTM1", "R3L3.PTM1", "R2L8.PTM2", "R
  "R1L7.RMF1", "R5L1.RMF1", "R2L2.RMF2", "R
```

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="supp")
[1] "suppTable1.xls"
> file.info("suppTable1.xls")
                 size isdir mode               mtime
suppTable1.xls 4531819 FALSE  666 2012-08-28 10:38:12
                                ctime               atime
suppTable1.xls 2015-07-04 15:54:57 2015-07-04 15:54:57
               exe
suppTable1.xls  no
>
```

③

湯 敏氏提供情報

サブセット抽出と整形

①出力ファイルは20,689遺伝子×36サンプルのカウントデータ。ヒト(HS)、チンパンジー(PT)、アカゲザル(RM)の3生物種のデータ。各12サンプル。ここのコードは、予め列名を把握しておき、②欲しいサブセットのみ任意の列の順番で並べ替えて、③任意の列名を与えて出力させています。

41. Blekhman et al., Genome Res., 2010のリアルカウントデータです。Supplementary (http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1 に整形しなおしたものがここでの出力ファイルになります。20,689 genes×36 samp 験デザインの詳細はFigure S1中に描かれていますが、ヒト(Homo Sapiens; HS), macaque; RM)の3種類の生物種の肝臓サンプル(liver sample)の比較を行っ れており(six individuals; six biological replicates)、技術的なばらつき(technical variation)を見積もるべく各個体は2つに分割されてデータ が取得されています(duplicates; two technical replicates)。それゆえ、ヒト12サンプル、チンパンジー12サンプル、アカゲザル12サンプル の計36サンプル分のデータということになります。以下で行っていることはカウントデータの列のみ「ヒトのメス(HSF1, HSF2, HSF3)」,「ヒトのオス(HSM1, HSM2, HSM3)」,「チンパンジーのメス(PTF1, PTF2, PTF3)」,「チンパンジーのオス(PTM1, PTM2, PTM3)」,「アカゲザルのメス(RMF1, RMF2, RMF3)」,「アカゲザルのオス(RMM1, RMM2, RMM3)」の順番で並び替えたものをファイルに保存しています。もう少し美しくやることも原理的には可能ですが、そこは本質的な部分ではありませんので、ここではアドホック(その場しのぎ、の意味)な手順で行っています。当然ながら、エクセルなどでファイルの中身を眺めて完全に列名を把握しているという前提です。
尚、"R1L4.HSF1"と"R4L2.HSF1"が「HSF1というヒトのメス一個体のtechnical replicates」であることは列名や文脈から読み解けます。

```
#in_f <- "http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls"#入力ファイ
in_f <- "suppTable1.xls"                    #入力ファイル名を指定してin_fに格納
out_f <- "sample_blekhman_36.txt"           #出力ファイル名を指定してout_fに

#入力ファイルの読み込み
hoge <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="
dim(hoge)                                   #行数と列数を表示

#サブセットの取得
data <- cbind(                              #必要な列名の情報を取得したい列の
  hoge$R1L4.HSF1, hoge$R4L2.HSF1, hoge$R2L7.HSF2, hoge$R3L2.HSF2, ho
  hoge$R1L1.HSM1, hoge$R5L2.HSM1, hoge$R2L3.HSM2, hoge$R4L8.HSM2, ho
  hoge$R1L2.PTF1, hoge$R4L4.PTF1, hoge$R2L4.PTF2, hoge$R6L6.PTF2, ho
  hoge$R1L6.PTM1, hoge$R3L3.PTM1, hoge$R2L8.PTM2, hoge$R4L6.PTM2, ho
  hoge$R1L7.RMF1, hoge$R5L1.RMF1, hoge$R2L2.RMF2, hoge$R5L8.RMF2, ho
  hoge$R1L3.RMM1, hoge$R3L8.RMM1, hoge$R2L6.RMM2, hoge$R5L4.RMM2, ho
colnames(data) <- c(                        #列名を付加
  "R1L4.HSF1", "R4L2.HSF1", "R2L7.HSF2", "R3L2.HSF2", "R8L1.HSF3", "
  "R1L1.HSM1", "R5L2.HSM1", "R2L3.HSM2", "R4L8.HSM2", "R3L6.HSM3", "
  "R1L2.PTF1", "R4L4.PTF1", "R2L4.PTF2", "R6L6.PTF2", "R3L7.PTF3", "
  "R1L6.PTM1", "R3L3.PTM1", "R2L8.PTM2", "R4L6.PTM2", "R6L2.PTM3", "
  "R1L7.RMF1", "R5L1.RMF1", "R2L2.RMF2", "R5L8.RMF2", "R3L4.RMF3", "
```

②

③

①

R Console

```
+    "R1L3.RMM1", "R3L8.RMM1", "R2L6$
> rownames(data)<- rownames(hoge)   $
> dim(data)                         $
[1] 20689    36
>
> #ファイルに保存(テキストファイル)
> tmp <- cbind(rownames(data), data$
> write.table(tmp, out_f, sep="\t",$
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="blek")
[1] "sample_blekhman_36.txt"
>
```

出力ファイルの中身の一部

サブセット抽出と整形

41. Blekhman et al., Genome Res., 2010のリアルカウントデータです。Supplementary Table1で提供されているエクセルファイル(http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls; 約4.3MB)からカウントデータのみ抽出し、きれいに整形しなおしたものがここでの出力ファイルになります。20,689 genes×36 samplesのカウントデータ(sample_blekhman_36.txt)です。実験デザインの詳細はFigure S1中に描かれていますが、ヒト(Homo Sapiens; HS), チンパンジー(Pan troglodytes; PT), アカゲザル(Rhesus macaque; RM)の3種類の生物種の肝臓サンプル(liver sample)の比較を行っています。生物種ごとにオス3個体メス3個体の計6個体使われており(six individuals; six biological replicates)、技術的なばらつき(technical variation)を見積もるべく各個体は2つに分割されてデータが取得されています(duplicates; two technical replicates)。それゆえ、ヒト12サンプル、チンパンジー12サンプル、アカゲザル12サンプルの計36サンプル分のデータということになります。以下で行っていることはカウントデータの列のみ「ヒトのメス(HSF1, HSF2, HSF3)」,「ヒトのオス(HSM1, HSM2, HSM3)」,「チンパンジーのメス(PTF1, PTF2, PTF3)」,「チンパンジーのオス(PTM1, PTM2, PTM3)」,「アカゲザルのメス(RMF1, RMF2, RMF3)」,「アカゲザルのオス(RMM1, RMM2, RMM3)」の順番で並び替えたものをファイルに保存しています。もう少し美しくやることも原理的には可能ですが、そこは本質的な部分ではありませんので、ここではアドホック(その場しのぎ、の意味)な手順で行っています。当然ながら、エクセルなどでファイルの中身を眺めて完全に列名を把握しているという前提です。
尚、"R1L4.HSF1"と"R4L2.HSF1"が「HSF1というヒトのメス一個体のtechnical replicates」であることは列名や文脈から読み解けます。

```
#in_f <- "http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls"#入力ファイ
in_f <- "suppTable1.xls"                    #入力ファイル名を指定してin_fに格納
out_f <- "sample_blekhman_36.txt"           #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
hoge <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイルの読み込み
dim(hoge)                                   #行数と列数を表示

#サブセットの取得
data <- cbind(                              #必要な列名の情報を取得したい列の順番で結合した結果をdataに格納
  hoge$R1L4.HSF1, hoge$R4L2.HSF1, hoge$R2L7.HSF2, hoge$R3L2.HSF2, hoge$R8L1.HSF3, hoge$R8L2.HSF3,
  hoge$R1L1.HSM1, hoge$R5L2.HSM1, hoge$R2L3.HSM2, hoge$R4L8.HSM2, hoge$R3L6.HSM3, hoge$R4L1.HSM3,
  hoge$R1L2.PTF1, hoge$R4L4.PTF1, hoge$R2L4.PTF2, hoge$R6L6.PTF2, hoge$R3L7.PTF3, hoge$R5L3.PTF3,
  hog
  hog
  hog
colna
  "R1
  "R1
  "R1
  "R1
  "R1
```

	R1L4.HSF1	R4L2.HSF1	R2L7.HSF2	R3L2.HSF2	R8L1.HSF3	R8L2.HSF3	R1L1.HSM1	R5L2.HSM1
ENSG00000000003	172	157	147	153	78	90	60	61
ENSG00000000005	0	0	0	0	0	0	0	0
ENSG00000000419	36	45	26	35	16	40	17	22
ENSG00000000457	41	50	28	34	34	42	50	64
ENSG00000000460	3	3	8	9	7	5	9	6
ENSG00000000938	22	21	20	25	112	98	22	41

クラスタリング

入力ファイルは20,689遺伝子×36サンプルのカウントデータ。ヒト(HS)、チンパンジー(PT)、アカゲザル(RM)の3生物種のデータ。各12サンプル。サンプル間クラスタリング結果から、実験デザインに関する説明を行います。

- 解析 | 発現量推定(トランスクリプトーム配列を利用) (last modified 201
- 解析 | クラスタリング | について (last modified 2014/02/05)
- 解析 | クラスタリング | サンプル間 | hclust (last modifie 2015/02/26) NEW
- 解析 | クラスタリング | サンプル間 | TCC(Sun_2013) ① modified 2015/03/02) NEW
- 解析 | クラスタリング | 遺伝子間 | MBCluster.Seq (Si 14) (last modified 2014/02/05)
- 解析
- 解析

解析 | クラスタリング | サンプル間 | TCC(Sun_2013) NEW

TCCパッケージを用いてサンプル間クラスタリングを行うやり方を示します。clusterSample関数を利用した頑健なクラスタリング結果を返します。

「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下をコピペ。

② 7. サンプルデータ41のリアルデータ(sample_blekhman_36.txt)の場合:

Blekhman et al., Genome Res., 2010の 20,689 genes×36 samplesのカウントデータです。

```
in_f <- "sample_blekhman_36.txt"          #入力ファイル名を指定してin_fに格納
out_f <- "hoge7.png"                      #出力ファイル名を指定してout_fに格納
param_fig <- c(700, 400)                  #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(TCC)                              #パッケージの読み込み

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイルの読み込み
dim(data)                                 #オブジェクトdataの行数と列数を表示

#本番
out <- clusterSample(data, dist.method="spearman",#クラスタリング実行結果をoutに格納
            hclust.method="average", unique.pattern=TRUE)#クラスタリング実行結果をoutに格納

#ファイルに保存
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定
par(mar=c(0, 4, 1, 0))                    #下、左、上、右の順で余白（行）を指定
plot(out, sub="", xlab="", cex.lab=1.2,#樹形図(デンドログラム)の表示
  cex=1.3, main="", ylab="Height")        #樹形図(デンドログラム)の表示
dev.off()                                 #おまじない
```

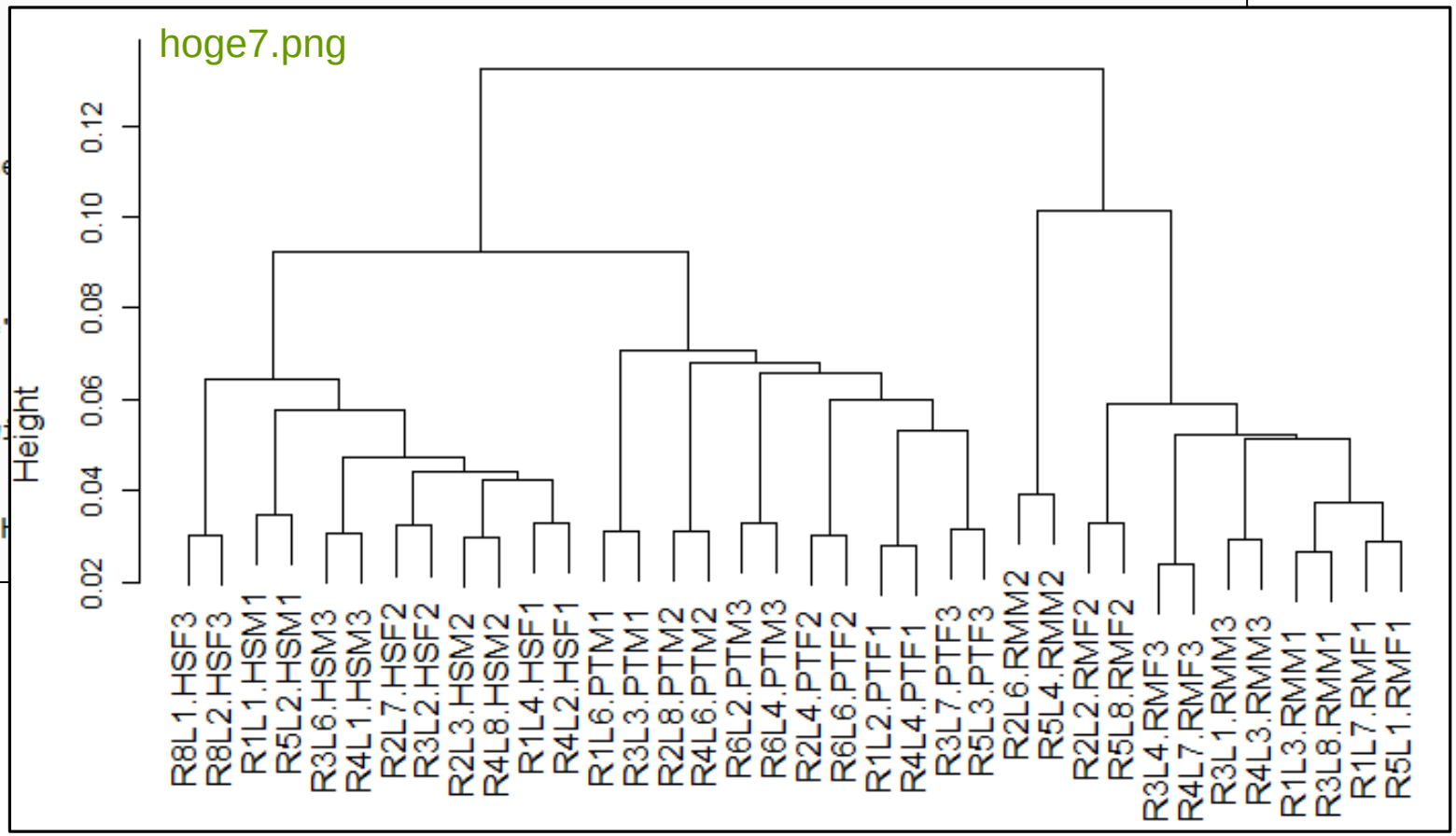
クラスタリング

入力ファイルは20,689遺伝子×36サンプルのカウントデータ。ヒト(HS)、チンパンジー(PT)、アカゲザル(RM)の3生物種のデータ。各12サンプル。

7. サンプルデータ41のリアルデータ(sample_blekhman_36.txt)の場合:

Blekhman et al., Genome Res., 2010の 20,689 genes×36 samplesのカウントデータです。

```
in_f <- "sample_blekhman_36.txt"        #入力ファイル名を指定してin_fに格納
out_f <- "hoge7.png"                    #出力ファイル名を指定してout_fに格納
param_fig <- c(700, 400)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(TCC)

#入力ファイルの読み込み
data <- read.table(in_f, he
dim(data)

#本番
out <- clusterSample(data,
            hclust.method="

#ファイルに保存
png(out_f, pointsize=13, wi
par(mar=c(0, 4, 1, 0))
plot(out, sub="", xlab="",
  cex=1.3, main="", ylab="H
dev.off()
```

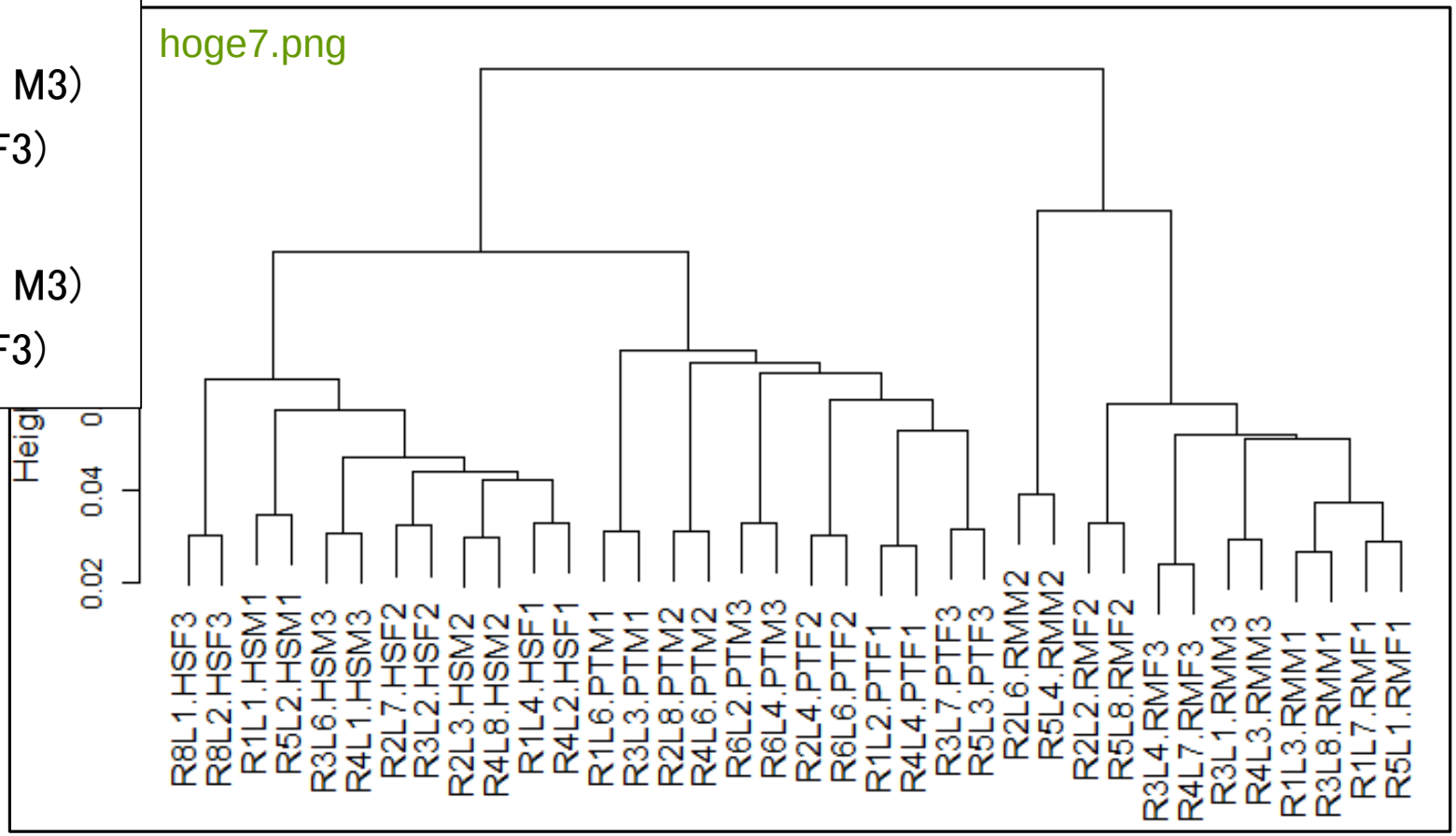


クラスタリング

生物種ごとに用いた個体数は6。雄雌を考慮しなければbiological replicates（生物学的な反復)は6。個体ごとにサンプルを分割して得たデータがあり、technical replicates（技術的な反復)は2。これらを合わせることで生物種ごとに計12サンプルになる。

- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)

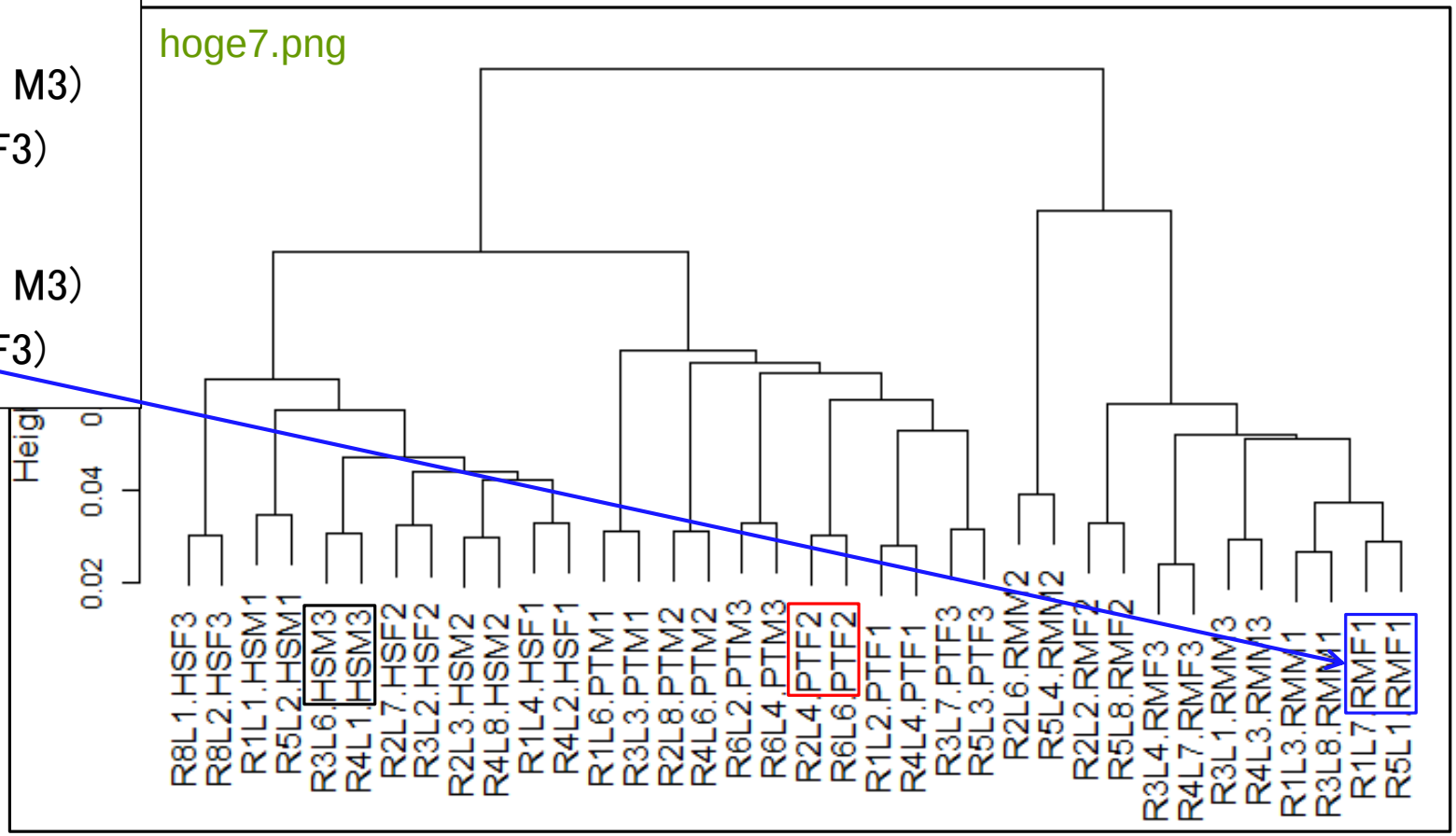
hoge7.png



クラスタリング

全個体について、同一個体を分割したtechnical replicatesのデータで末端のクラスターを形成していることが分かる。これはtechnical replicatesのデータ同士の類似度が非常に高いことを示している。

- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)

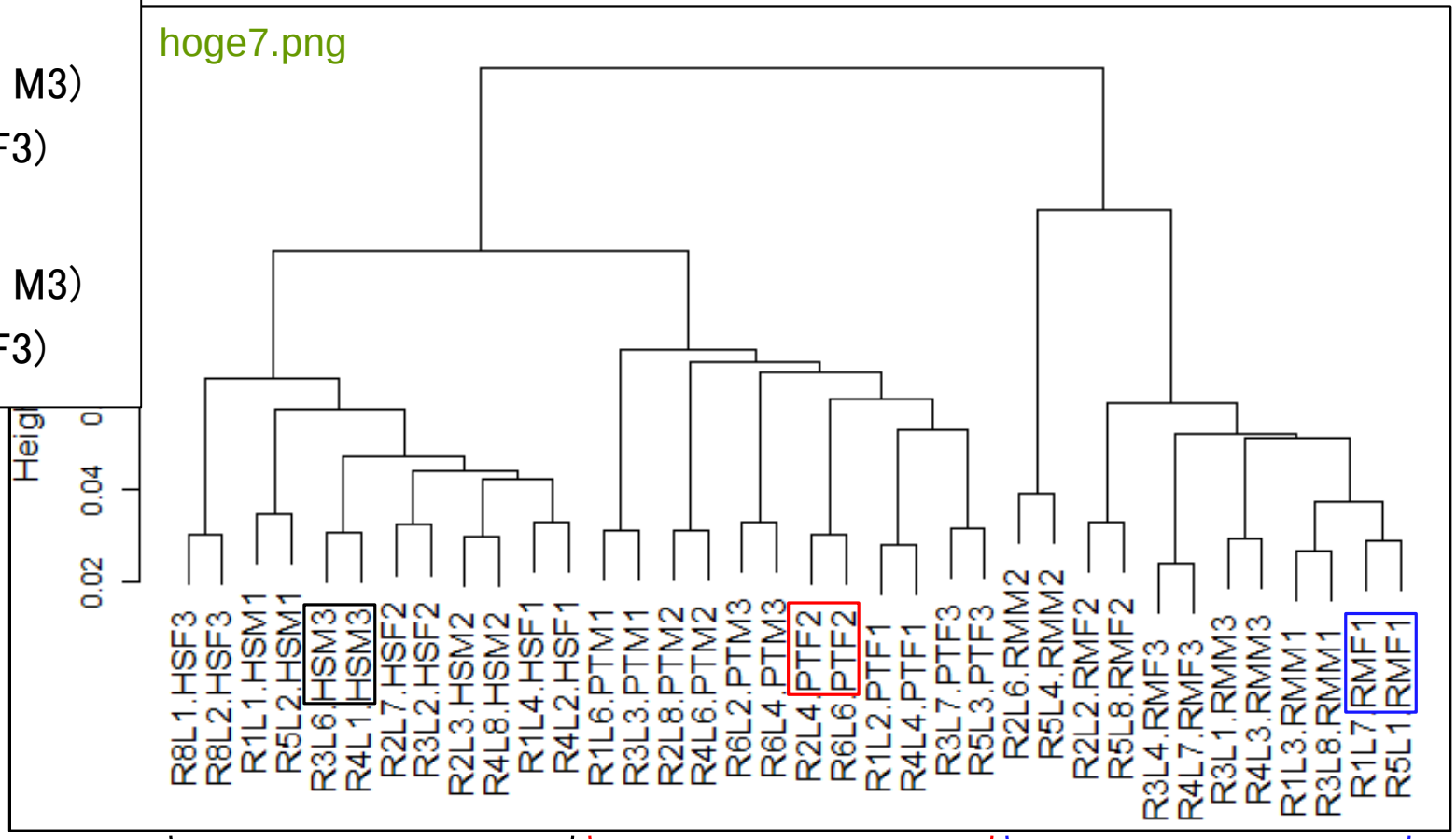


クラスタリング

- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)

統計的手法で2群間比較(例えばMales vs. Females)をする目的は、同一群内の別個体(biological replicates)のばらつきの程度を見積もっておき(モデル構築)、比較する2群間で発現に変動がないという前提(帰無仮説)からどれだけ離れているのかをp値で評価することである。p値が低ければ低いほど「発現変動していない(帰無仮説に従う)」とは考えにくい、つまり帰無仮説を棄却してDEGと判定することになる。

hoge7.png



サブセット抽出と整形

統計的手法の多くは、biological replicatesのデータを前提としている。Technical replicatesのデータをマージ(merge; collapseともいうらしい)したものを作成。サンプル名部分は余計なものを削除している。

- (削除予定)個別パッケージのインストール (last modified 2015/02/20)
- 基本的な利用法 (last modified 2015/04/03)
- サンプルデータ (last modified 2015/06/15) NEW
- バイオインフォマティクス人材育成カリキュラム(次世代シークエンサ) | NGSハンズオン
- 書籍
- 書籍
- 書籍

① ②

サンプルデータ NEW

42. Blekhman et al., Genome Res., 2010のリアルカウントデータです。
1つ前のsample41.txtとは違って、technical replicatesの2列分のデータは足して1列分のデータとしています。20,689 genes×18 samplesのカウントデータ(sample_blekhman_18.txt)です。

```
#in_f <- "http://genome.cshlp.org/content/suppl/2009/12/16/gr.099226.109.DC1/suppTable1.xls"#入力ファイ
in_f <- "suppTable1.xls"                    #入力ファイル名を指定してin_fに格納
out_f <- "sample_blekhman_18.txt"           #出力ファイル名を指定してout_fに格納

#入力ファイルの読み込み
hoge <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイルの読み込み
dim(hoge)                                   #行数と列数を表示

#サブセットの取得
data <- cbind(                              #必要な列名の情報を取得したい列の順番で結合した結果をdataに格納
  hoge$R1L4.HSF1 + hoge$R4L2.HSF1, hoge$R2L7.HSF2 + hoge$R3L2.HSF2, hoge$R8L1.HSF3 + hoge$R8L2.HSF3,
  hoge$R1L1.HSM1 + hoge$R5L2.HSM1, hoge$R2L3.HSM2 + hoge$R4L8.HSM2, hoge$R3L6.HSM3 + hoge$R4L1.HSM3,
  hoge$R1L2.PTF1 + hoge$R4L4.PTF1, hoge$R2L4.PTF2 + hoge$R6L6.PTF2, hoge$R3L7.PTF3 + hoge$R5L3.PTF3,
  hoge$R1L6.PTM1 + hoge$R3L3.PTM1, hoge$R2L8.PTM2 + hoge$R4L6.PTM2, hoge$R6L2.PTM3 + hoge$R6L4.PTM3,
  hoge$R1L7.RMF1 + hoge$R5L1.RMF1, hoge$R2L2.RMF2 + hoge$R5L8.RMF2, hoge$R3L4.RMF3 + hoge$R4L7.RMF3,
  hoge$R1L3.RMM1 + hoge$R3L8.RMM1, hoge$R2L6.RMM2 + hoge$R5L4.RMM2, hoge$R3L1.RMM3 + hoge$R4L3.RMM3)
colnames(data) <- c(                        #列名を付加
  "HSF1", "HSF2", "HSF3", "HSM1", "HSM2", "HSM3",
  "PTF1", "PTF2", "PTF3", "PTM1", "PTM2", "PTM3",
  "RMF1", "RMF2", "RMF3", "RMM1", "RMM2", "RMM3")
rownames(data)<- rownames(hoge)             #行名を付加
dim(data)                                   #行数と列数を表示
```

クラスタリング

20,689遺伝子×18サンプルの
biological replicatesのみからなる
カウントデータでクラスタリング。

- 解析 | 発現量推定(トランスクリプトーム配列を利用) (last modified 2014/07/09)
- 解析 | クラスタリング | について (last modified 2014/02/05)
- 解析 | クラスタリング | サンプル間 | hclust (last modified 2015/02/26) NEW
- 解析 | クラスタリング | サンプル間 | TCC(Sun_2013) (last modified 2015/03/02) NEW
- 解析 | クラスタリング | 遺伝子間 | MBCluster.Seq (Si 2014) (last modified 2014/02/05)

① ②

解析 | クラスタリング | サンプル間 | TCC(Sun_2013) NEW

TCCパッケージを用いてサンプル間クラスタリングを行うやり方を示します。clusterSample関数を利用した頑健なクラスタリング結果を返します。

8. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:

Blekhman et al., Genome Res., 2010の 20,689 genes×18 samplesのカウントデータです。

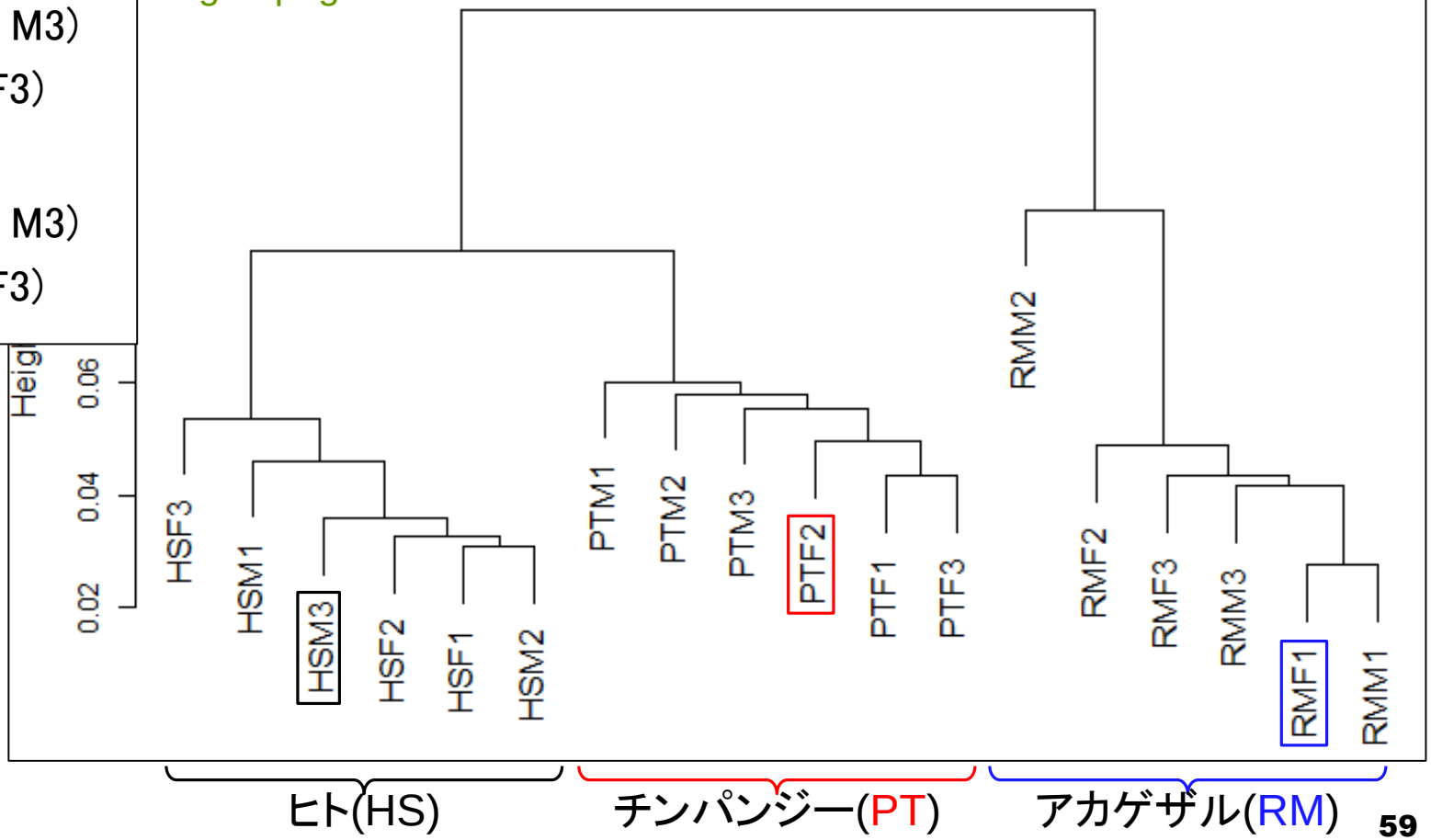
```
in_f <- "sample_blekhman_18.txt"          #入力ファイル名を指定してin_fに格納
out_f <- "hoge8.png"                      #出力ファイル名を指定してout_fに格納
param_fig <- c(700, 400)                  #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#必要なパッケージをロード
library(TCC)                              #パッケージの読み込み

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定したファイ
dim(data)                                 #オブジェクトdataの行数と列数を表示

#本番
out <- clusterSample(data, dist.method="spearman",#クラスタリング実行結果をoutに格納
           hclust.method="average", unique.pattern=TRUE)#クラスタリング実行結果をoutに格納

#ファイルに保存
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメー
par(mar=c(0, 4, 1, 0))                    #下、左、上、右の順で余白（行）を指定
plot(out, sub="", xlab="", cex.lab=1.2,#樹形図(デンドログラム)の表示
```

クラスタリング

- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)

36サンプルのときの結果と同様、全体的なトポロジーは同じ。このクラスタリング結果を眺めるだけで、DEG検出結果のイメージは大体つかめる。例1:「HS vs. RMで得られるDEG数」のほうが「HS vs. PTで得られるDEG数」よりも多そう。例2:ヒトは「オス vs. メス」でのDEG数は0に近いだろう。例3:RMM2が外れサンプルっぽいので、これを除去すれば生物種間比較時にDEG数が増えるだろう。

hoge8.png



Contents

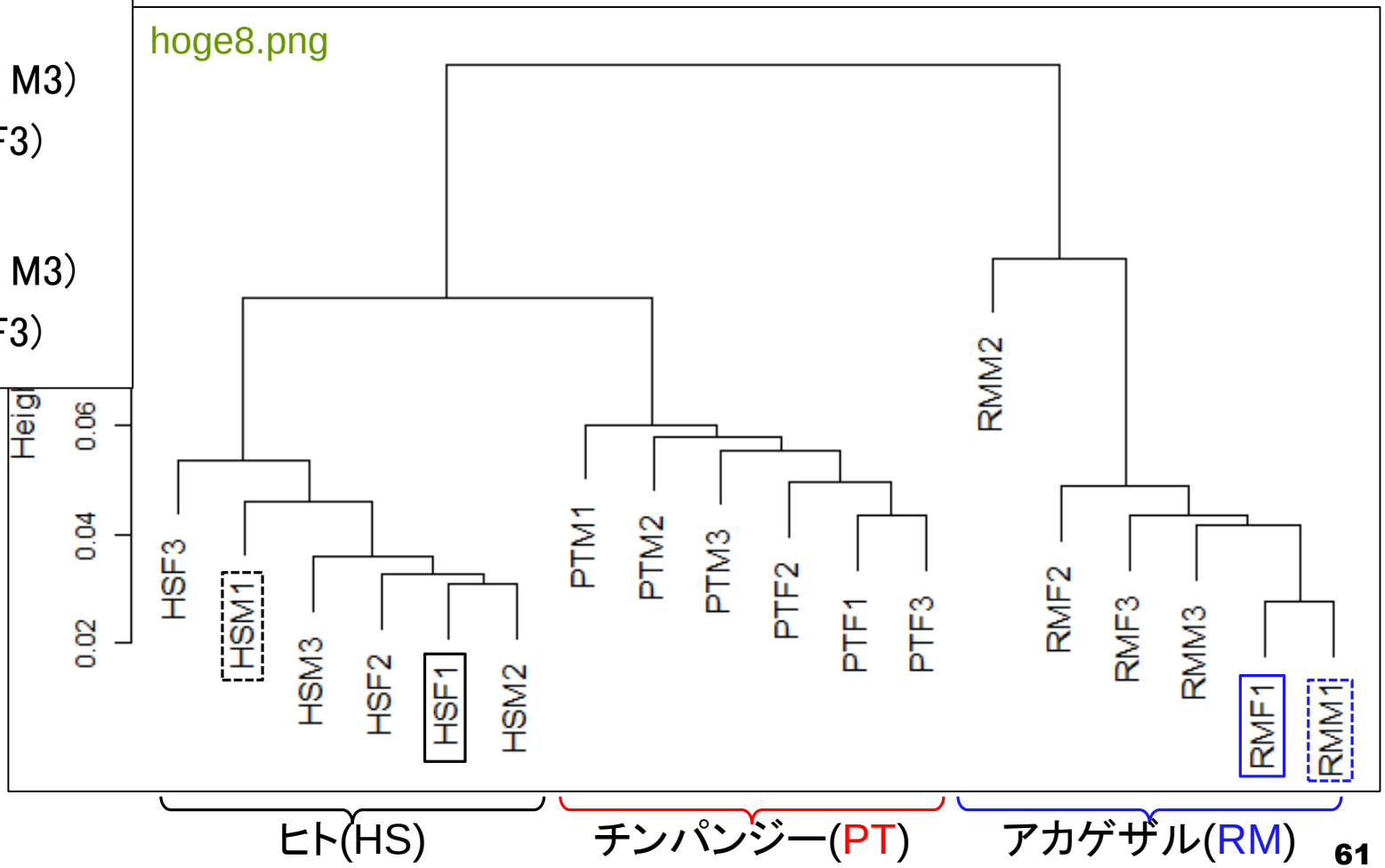
- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較）：発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

「HS vs. RM」の2群間比較をTCCパッケージ(Sun et al., 2013)で行ってみよう。

2群間比較

- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)

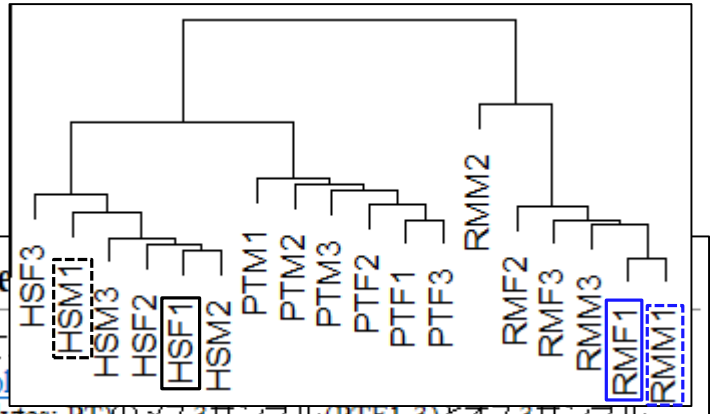
hoge8.png



TCC実行

「HS vs. RM」の2群間比較をTCCパッケージ(Sun et al., 2013)で行ってみよう。

- 解析 | シミュレーションカウントデータ | Biological rep. | 3群間 | 基礎 | TCC(Sun 2013) (last modified 20
- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013) (last modified 2015/07/05)推奨 NEW
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun 2013) ① modified 2015/0
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li 2013) (last modified 201 02/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | edgeR(Robinson 2010) (last modified 2014/07/24)
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応



解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Ble

Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて
サンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ(sampl
ス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus macaque; RM)のメス3サンプル(RMF1-3)とオス3サンプル(RMM1-3)の並びになっています。つまり、以下のような感じです。Fは Female(メス)、Mは Male(オス)を表します。

ヒト(1-6列目): HSF1, HSF2, HSF3, HSM1, HSM2, and HSM3
チンパンジー(7-12列目): PTF1, PTF2, PTF3, PTM1, PTM2, and PTM3
アカゲザル(13-18列目): RMF1, RMF2, RMF3, RMM1, RMM2, and RMM3

「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下をコピペ。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
in_f <- "sample_blekhman_18.txt"        #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge1.txt"                   #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1.png"                   #出力ファイル名を指定してout_f2に格納
param_subset <- c(1, 4, 13, 16)         #取り扱いたいサブセット情報を指定
param_G1 <- 2                           #G1群のサンプル数を指定
param_G2 <- 2                           #G2群のサンプル数を指定
param_FDR <- 0.05                       #DEG検出時のfalse discovery rate (FDR)閾値を指定
param_fig <- c(430, 350)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)
param_mar <- c(4, 4, 0, 0)              #下、左、上、右の順で余白を指定(単位は行)

#必要なパッケージをロード
library(TCC)                            #パッケージの読み込み
```

TCC実行

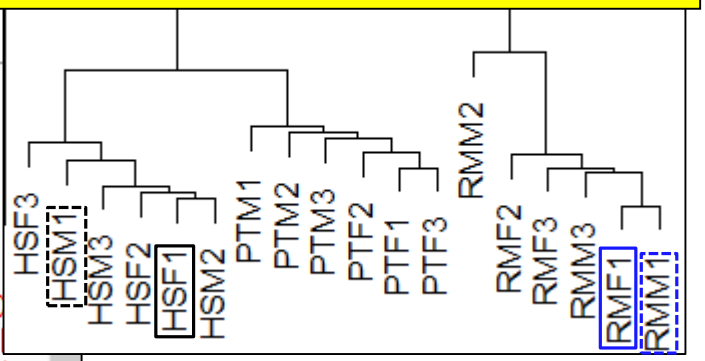
①ここで取得したいサブセットの列番号やグループ情報を指定。②発現変動解析に用いるサブセットは20,689 genes × 4 samplesのデータ。③正しくヒト vs. アカゲザルになっていることが分かる。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RM

1, 4, 13, 16 列目のデータのみ抽出しています。

```
in_f <- "sample_blekhman_18.txt"          #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge1.txt"                     #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1.png"                     #出力ファイル名を指定してout_f2に格納
param_subset <- c(1, 4, 13, 16)           #取り扱いたいサブセット情報を指定
param_G1 <- 2                             #G1群のサンプル数を指定
param_G2 <- 2                             #G2群のサンプル数を指定
param_FDR <- 0.05                         #DEG検出時のfalse discovery rate (FD
param_fig <- c(430, 350)                  #ファイル出力時の横幅と縦幅を指定(単位
param_mar <- c(4, 4, 0, 0)                #下、左、上、右の

#必要なパッケージをロード
library(TCC)                              #パッケージの読み

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep=

#前処理(サブセットの抽出とTCCクラスオブジェクトの作成)
data <-data[,param_subset]                #param_subsetで指
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を
tcc <- new("TCC", data, data.cl)          #TCCクラスオブジェ
dim(data)                                 #行数と列数を表示
head(data)                                #最初の6行分を表示
```

①



R Console

```
> #前処理(サブセットの抽出とTCCクラスオブ$
> data <-data[,param_subset]                #$
> data.cl <- c(rep(1, param_G1), rep(2, pa$
> tcc <- new("TCC", data, data.cl)          #$
> dim(data)                                 #$
[1] 20689     4
> head(data)                                #$
                 HSF1 HSM1 RMF1 RMM1
ENSG00000000003  329  121  511  424
ENSG00000000005    0    0    0    2
ENSG00000000419   81   39   67   49
ENSG00000000457   91  114   89  117
ENSG00000000460    6   15    4    7
ENSG00000000938   44   73   73   80
>
```

②

③

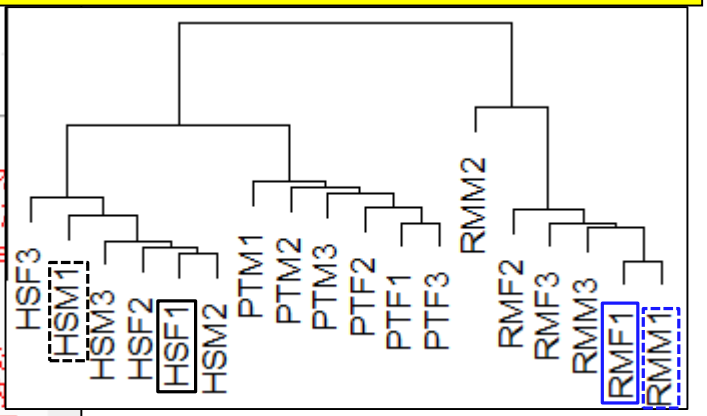
FDR

q < 0.05を満たす遺伝子数は2,488個。
FDR = 0.05なので、2,488*0.05 = 124.4個は偽物で残りの95%は本物だと判断する。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
#本番(正規化)
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="edger",#正規化
                       iteration=3, FDR=0.1, floorPDEG=0.05)#正規化を実行し
normalized <- getNormalizedData(tcc)    #正規化後のデータを取り出してnormalize

#本番(DEG検出)
tcc <- estimateDE(tcc, test.method="edger", FDR=param_FDR)#DEG検出を実行した
result <- getResult(tcc, sort=FALSE)    #p値などの結果をした結果をresultに格納
sum(tcc$stat$q.value < param_FDR)       #FDR < param_FDR

#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(tcc$count), normalized, result)#正
tmp <- tmp[order(tmp$rank),]            #発現変動順にソー
write.table(tmp, out_f1, sep="\t", append=F, quote=F, r

#ファイルに保存(M-A plot)
png(out_f2, pointsize=13, width=param_fig[1], height=pa
par(mar=param_mar)                      #余白を指定
plot(tcc, FDR=param_FDR, xlim=c(-2, 17), ylim=c(-10, 10
     cex=0.9, cex.lab=1.2,              #param_FDRで指定
     cex.axis=1.2, main="",             #param_FDRで指定
     xlab="A = (log2(G2) + log2(G1))/2",#param_FDRで指定
```



R Console

```
+                          iteration=3, FDR=$
TCC::INFO: Calculating normalization facto$
TCC::INFO: (iDEGES pipeline : tmm - [ edge$
TCC::INFO: Done.
> normalized <- getNormalizedData(tcc)    #$
>
> #本番(DEG検出)
> tcc <- estimateDE(tcc, test.method="edge$
TCC::INFO: Identifying DE genes using edge$
TCC::INFO: Done.
> result <- getResult(tcc, sort=FALSE)    #$
> sum(tcc$stat$q.value < param_FDR)        #$
[1] 2488
>
>
```

①

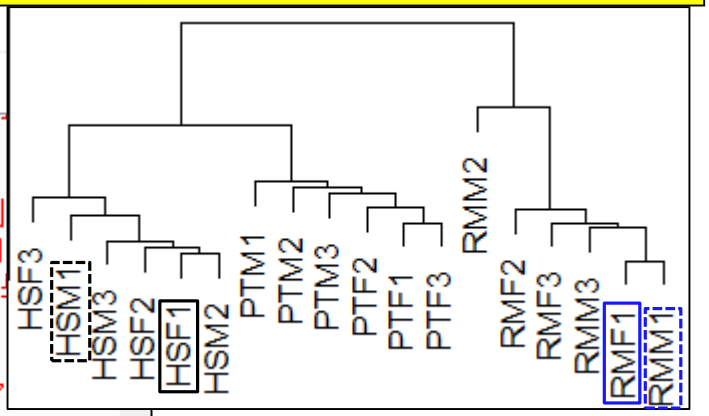
FDR

q < 0.30を満たす遺伝子数は4,786個。
FDR = 0.30なので、4,786*0.30 = 1,435.8個は偽物で残りの70%は本物だと判断する。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(tcc$count), normalized, result)#正規化後のデータの右側
tmp <- tmp[order(tmp$rank),]           #発現変動順にソートした結果をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中

#ファイルに保存(M-A plot)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファ
par(mar=param_mar)                     #余白を指定
plot(tcc, FDR=param_FDR, xlim=c(-2, 17), ylim=c(-10, 10
     cex=0.8, cex.lab=1.2,             #param_FDRで指定
     cex.axis=1.2, main="",            #param_FDRで指定
     xlab="A = (log2(G2) + log2(G1))/2",#param_FDRで指定
     ylab="M = log2(G2) - log2(G1)")   #param_FDRで指定
legend("topright", c(paste("DEG(FDR<", param_FDR, ")",
       col=c("magenta", "black"), pch=20, cex=1.2)#凡例
dev.off()                              #おまじない
sum(tcc$stat$q.value < 0.05)           #FDR < 0.05を満た
sum(tcc$stat$q.value < 0.10)           #FDR < 0.10を満た
sum(tcc$stat$q.value < 0.20)           #FDR < 0.20を満た
sum(tcc$stat$q.value < 0.30)           #FDR < 0.30を満た
```



R Console

```
+       ylab="M = log2(G2) - log2(G1)")    #$
> legend("topright", c(paste("DEG(FDR<", p$
+         col=c("magenta", "black"), pch=20$
> dev.off()                                 #$
null device
          1
> sum(tcc$stat$q.value < 0.05)              #$
[1] 2488
> sum(tcc$stat$q.value < 0.10)              #$
[1] 3122
> sum(tcc$stat$q.value < 0.20)              #$
[1] 4049
> sum(tcc$stat$q.value < 0.30)              #$
[1] 4786
>
```

①

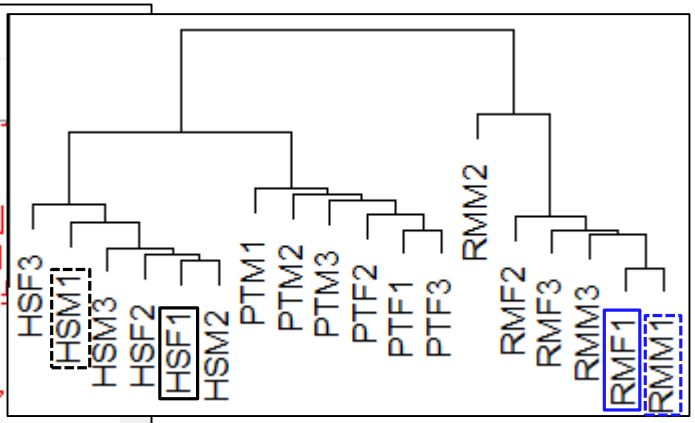
DEG数の見積もり

FDR閾値が比較的緩めのところを眺め、20,689 genes中3,300個程度が本物のDEGと判断する。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(tcc$count), normalized, result)#正規化後のデータの右側
tmp <- tmp[order(tmp$rank),]           #発現変動順にソートした結果をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中

#ファイルに保存(M-A plot)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファ
par(mar=param_mar)                     #余白を指定
plot(tcc, FDR=param_FDR, xlim=c(-2, 17), ylim=c(-10, 10),#param_FDRで指定した閾値
     cex=0.8, cex.lab=1.2,             #param_FDRで指定した閾値を満たすDEGをマゼンタ
     cex.axis=1.2, main="",            #param_FDRで指定した閾値を満たすDEGをマゼンタ
     xlab="A = (log2(G2) + log2(G1))/2",#param_FDRで指定した閾値を満たすDEGをマゼン
     ylab="M = log2(G2) - log2(G1)")   #param_FDRで指定した閾値を満たすDEGをマゼンタ
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を
       col=c("magenta", "black"), pch=20, cex=1.2)#凡例を作成
dev.off()                              #おまじない
sum(tcc$stat$q.value < 0.05)           #FDR < 0.05を満た
sum(tcc$stat$q.value < 0.10)           #FDR < 0.10を満た
sum(tcc$stat$q.value < 0.20)           #FDR < 0.20を満た
sum(tcc$stat$q.value < 0.30)           #FDR < 0.30を満た
```



R Console

```
> 2488*(1 - 0.05)
[1] 2363.6
> 3122*(1 - 0.10)
[1] 2809.8
> 4049*(1 - 0.20)
[1] 3239.2
> 4786*(1 - 0.30)
[1] 3350.2
>
```

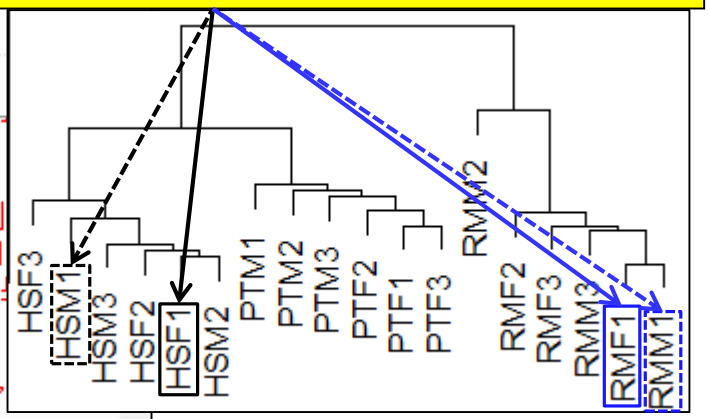
樹形図と一致

今比較しているのはHS vs. RM。クラスタリング結果からも、これらの発現プロファイルの類似度が低い（距離が遠い）ことがわかるので妥当

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(tcc$count), normalized, result)#正規化後のデータの右側
tmp <- tmp[order(tmp$rank),]           #発現変動順にソートした結果をtmpに格納
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中

#ファイルに保存(M-A plot)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファ
par(mar=param_mar)                     #余白を指定
plot(tcc, FDR=param_FDR, xlim=c(-2, 17), ylim=c(-10, 10),#param_FDRで指定した閾値
     cex=0.8, cex.lab=1.2,             #param_FDRで指定した閾値を満たすDEGをマゼンタ
     cex.axis=1.2, main="",            #param_FDRで指定した閾値を満たすDEGをマゼンタ
     xlab="A = (log2(G2) + log2(G1))/2",#param_FDRで指定した閾値を満たすDEGをマゼン
     ylab="M = log2(G2) - log2(G1)")   #param_FDRで指定した閾値を満たすDEGをマゼンタ
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を
       col=c("magenta", "black"), pch=20, cex=1.2)#凡例を作成
dev.off()                              #おまじない
sum(tcc$stat$q.value < 0.05)           #FDR < 0.05を満た
sum(tcc$stat$q.value < 0.10)           #FDR < 0.10を満た
sum(tcc$stat$q.value < 0.20)           #FDR < 0.20を満た
sum(tcc$stat$q.value < 0.30)           #FDR < 0.30を満た
```



R Console

```
> 2488*(1 - 0.05)
[1] 2363.6
> 3122*(1 - 0.10)
[1] 2809.8
> 4049*(1 - 0.20)
[1] 3239.2
> 4786*(1 - 0.30)
[1] 3350.2
>
```

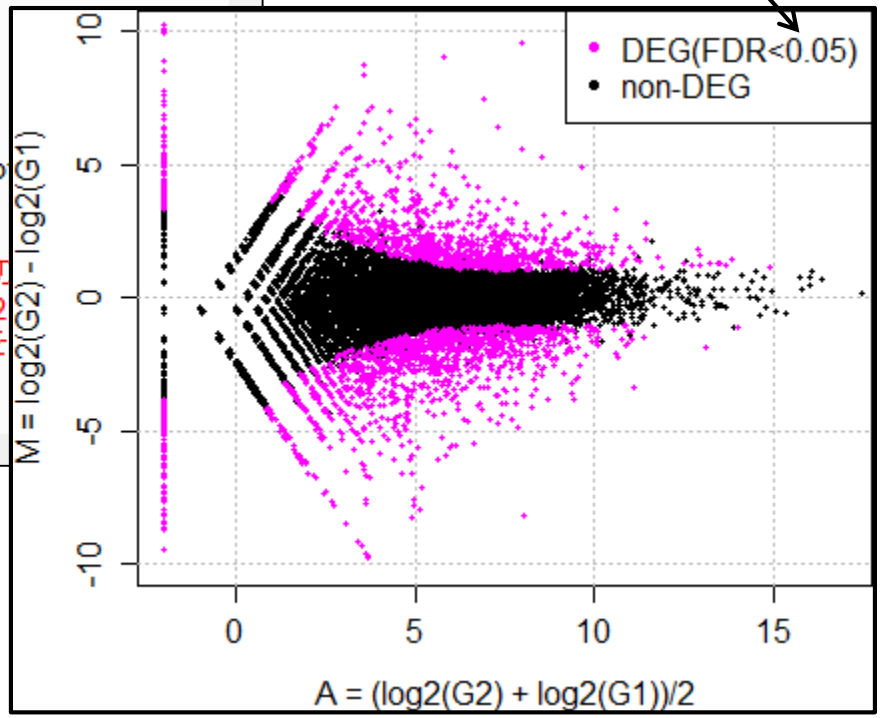
M-A plot

これがM-A plot。発現変動遺伝子（DEG）と判定されたものが多数存在することがわかる。param_FDRで指定した閾値(0.05)を満たす遺伝子群がマゼンタ色で表示されている。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とR

1, 4, 13, 16 列目のデータのみ抽出しています。

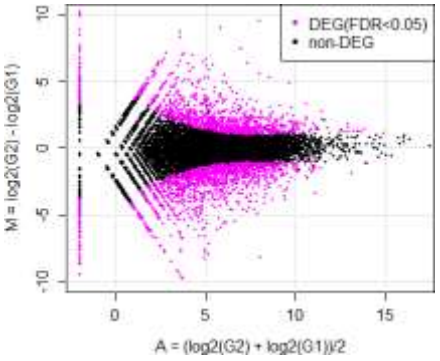
```
in_f <- "sample_blekhman_18.txt"          #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge1.txt"                     #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1.png"                     #出力ファイル名を指定してout_f2に格納
param_subset <- c(1, 4, 13, 16)           #取り扱いたいサブセット情報を指定
param_G1 <- 2                             #G1群のサンプル数を指定
param_G2 <- 2                             #G2群のサンプル数を指定
param_FDR <- 0.05                         #DEG検出時のfalse discovery rate (FDR)
param_fig <- c(430, 350)                  #ファイル出力時の横幅と縦幅を指定(単位はピ
param_mar <- c(4, 4, 0, 0)                #下、左、上、右の順で余白を指定(単位は行)

#必要なパッケージをロード
library(TCC)                              #パッケージの読み込み

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quo

#前処理(サブセットの抽出とTCCクラスオブジェクトの作成)
data <-data[,param_subset]                #param_subsetで指定した列の
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2
tcc <- new("TCC", data, data.cl)          #TCCクラスオブジェクトtccを
dim(data)                                 #行数と列数を表示
head(data)                                #最初の6行分を表示
```

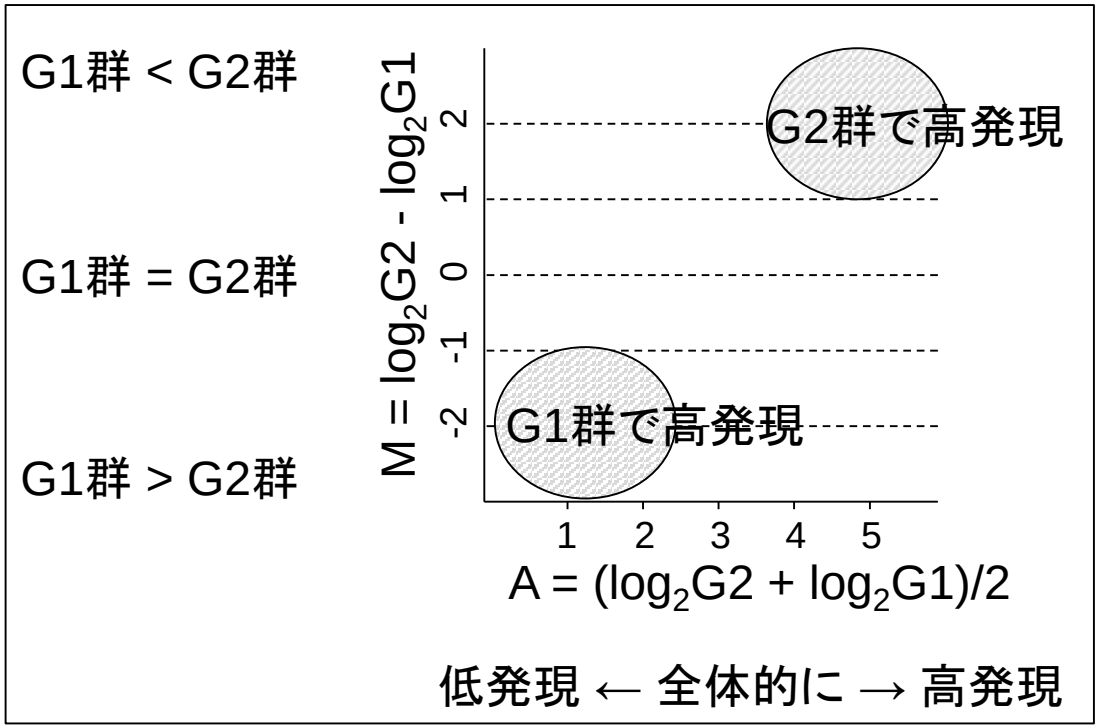


DEGが存在しないデータのM−A plotを眺めることで、縦軸の閾値のみに相当する**倍率変化**を用いたDEG同定の危険性が分かります

M-A plot

- 2群間比較用
- 横軸が全体的な発現レベル、縦軸がlog比からなるプロット
- 名前の由来は、おそらく対数の世界での縦軸が引き算（Minus）、横軸が平均（Average）





Dudoit et al., *Stat. Sinica*, **12**: 111-139, 2002

基本的にはこれが解析結果

DEG検出結果

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
in_f <- "sample_blekhman_18.txt"          #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge1.txt"                     #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1.png"                     #出力ファイル名を指定してout_f2に格納
param_subset <- c(1, 4, 13, 16)           #取り扱いたいサブセット情報を指定
param_G1 <- 2                             #G1群のサンプル数を指定
param_G2 <- 2                             #G2群のサンプル数を指定
param_FDR <- 0.05                         #DEG検出時のfalse discovery rate (FDR)
param_fig <- c(430, 350)                  #ファイル出力時の横幅と縦幅を指定(単位はピ
param_mar <- c(4, 4, 0, 0)                #下、左、上、右の順で余白を指定(単位は行)

#必要なパッケージをロード
library(TCC)                              #パッケージの読み込み

#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで
```

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.8	1477.0	ENSG00000208570	−2.04	11.29	4.41E−53	9.13E−49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.1	ENSG00000220191	5.79	9.06	4.54E−47	4.70E−43	2	1
ENSG00000106366	4421.9	4411.0	23.1	8.3	ENSG00000106366	8.04	−8.14	2.46E−45	1.70E−41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	−2.04	10.23	3.29E−44	1.70E−40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	−2.04	10.08	1.74E−43	7.21E−40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	−2.04	10.03	4.68E−42	1.61E−38	6	1
ENSG00000209007	0.0	0.0	615.2	479.6	ENSG00000209007	−2.04	9.92	1.24E−40	3.67E−37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	−9.69	1.52E−38	3.93E−35	8	1
ENSG00000156222	367.5	301.5	0.9	0.0	ENSG00000156222	3.61	−9.56	1.09E−36	2.51E−33	9	1
ENSG00000165272	404.8	438.9	2.6	0.8	ENSG00000165272	4.03	−7.59	5.45E−36	1.12E−32	10	1

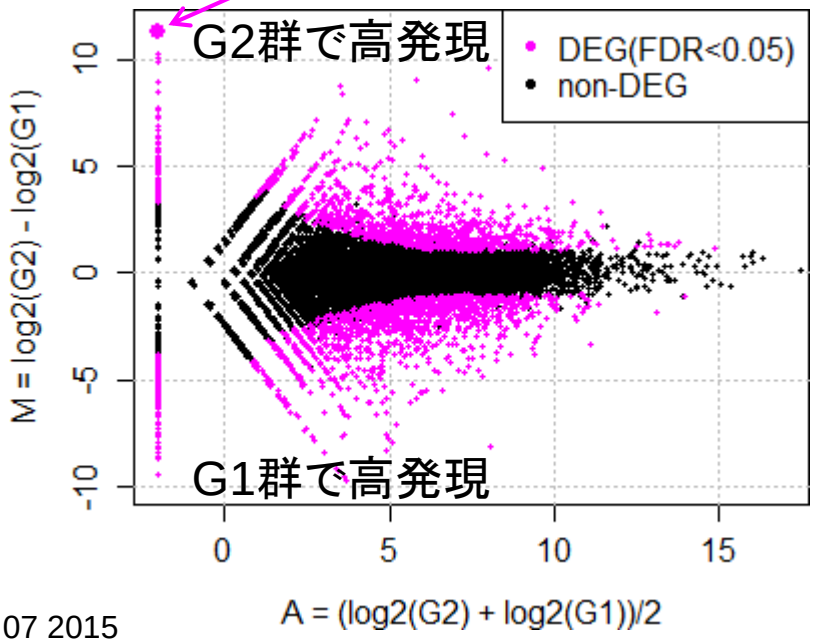
1位はRM群（G2群）で高発現のDEG

DEG検出結果

G1(HS)群　G2(RM)群

***p*-valueとその順位**

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.8	1477.0	ENSG00000208570	−2.04	11.29	4.41E−53	9.13E−49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.1	ENSG00000220191	5.79	9.06	4.54E−47	4.70E−43	2	1
ENSG00000106366	4421.9	4411.0	23.1	8.3	ENSG00000106366	8.04	−8.14	2.46E−45	1.70E−41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	−2.04	10.23	3.29E−44	1.70E−40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	−2.04	10.08	1.74E−43	7.21E−40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	−2.04	10.03	4.68E−42	1.61E−38	6	1
ENSG00000209007	0.0	0.0	615.2	479.6	ENSG00000209007	−2.04	9.92	1.24E−40	3.67E−37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	−9.69	1.52E−38	3.93E−35	8	1
ENSG00000156222	367.5	301.5	0.9	0.0	ENSG00000156222	3.61	−9.56	1.09E−36	2.51E−33	9	1
ENSG00000165272	404.8	428.9	2.6	0.8	ENSG00000165272	4.02	−7.59	5.45E−26	1.12E−22	10	1



M-A plotのA値とM値

***q*-value**

FDR閾値判定結果。***q*-value < 0.05**を満たすDEGが1、non-DEGが0。

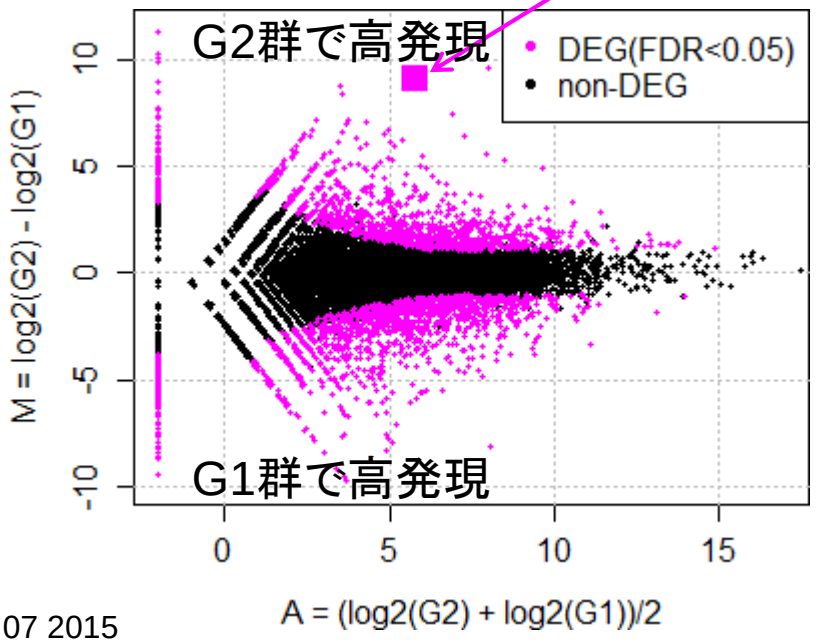
2位もRM群（G2群）で高発現のDEG

DEG検出結果

G1(HS)群　G2(RM)群

***p*-valueとその順位**

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.8	1477.0	ENSG00000208570	−2.04	11.29	4.41E−53	9.13E−49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.1	ENSG00000220191	5.79	9.06	4.54E−47	4.70E−43	2	1
ENSG00000106366	4421.9	4411.0	23.1	8.3	ENSG00000106366	8.04	−8.14	2.46E−45	1.70E−41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	−2.04	10.23	3.29E−44	1.70E−40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	−2.04	10.08	1.74E−43	7.21E−40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	−2.04	10.03	4.68E−42	1.61E−38	6	1
ENSG00000209007	0.0	0.0	615.2	479.6	ENSG00000209007	−2.04	9.92	1.24E−40	3.67E−37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	−9.69	1.52E−38	3.93E−35	8	1
ENSG00000156222	367.5	301.5	0.9	0.0	ENSG00000156222	3.61	−9.56	1.09E−36	2.51E−33	9	1
ENSG00000165272	404.8	428.0	2.6	0.8	ENSG00000165272	4.02	−7.59	5.45E−26	1.12E−22	10	1



M-A plotのA値とM値

***q*-value**

FDR閾値判定結果。***q*-value < 0.05**を満たすDEGが1、non-DEGが0。

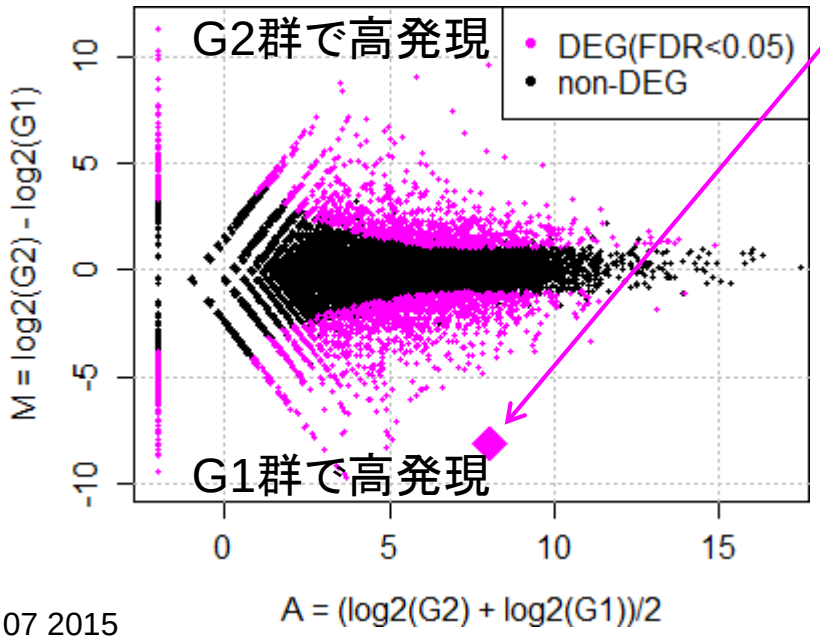
3位はHS群（G1群）で高発現のDEG

DEG検出結果

G1(HS)群　G2(RM)群

***p*-valueとその順位**

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.8	1477.0	ENSG00000208570	−2.04	11.29	4.41E−53	9.13E−49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.1	ENSG00000220191	5.79	9.06	4.54E−47	4.70E−43	2	1
ENSG00000106366	4421.9	4411.0	23.1	8.3	ENSG00000106366	8.04	−8.14	2.46E−45	1.70E−41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	−2.04	10.23	3.29E−44	1.70E−40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	−2.04	10.08	1.74E−43	7.21E−40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	−2.04	10.03	4.68E−42	1.61E−38	6	1
ENSG00000209007	0.0	0.0	615.2	479.6	ENSG00000209007	−2.04	9.92	1.24E−40	3.67E−37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	−9.69	1.52E−38	3.93E−35	8	1
ENSG00000156222	367.5	301.5	0.9	0.0	ENSG00000156222	3.61	−9.56	1.09E−36	2.51E−33	9	1
ENSG00000165272	404.8	428.9	2.6	0.8	ENSG00000165272	4.02	−7.59	5.45E−26	1.12E−22	10	1



M-A plotのA値とM値

***q*-value**

FDR閾値判定結果。***q*-value < 0.05**を満たすDEGが1、non-DEGが0。

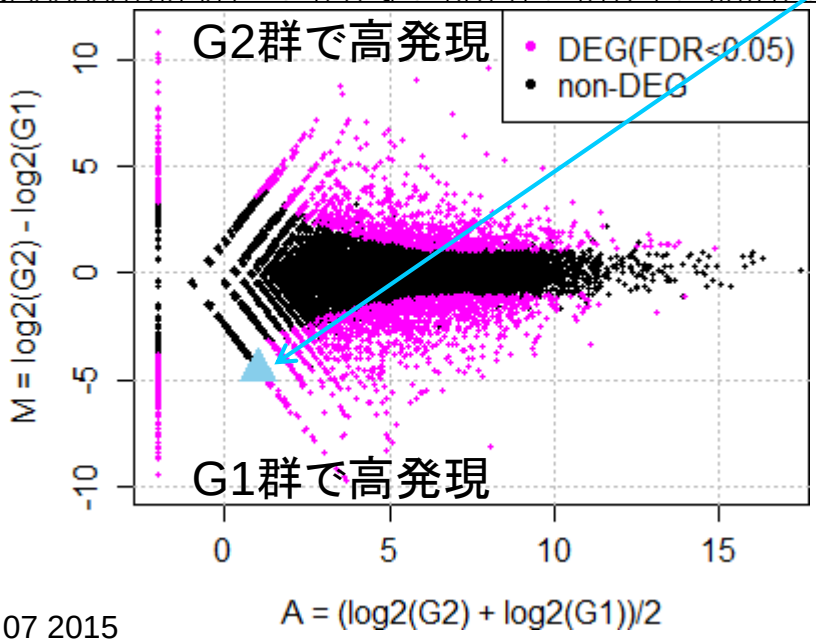
DEG検出結果

指定したFDR閾値（0.05）をギリギリ満たす2,488位の遺伝子

G1(HS)群　G2(RM)群

***p*-valueとその順位**

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000180672	9.0	8.9	0.9	1.7	ENSG00000180672	1.76	−2.82	0.00596	0.04967	2484	1
ENSG00000159899	161.7	89.1	47.9	60.7	ENSG00000159899	6.37	−1.21	0.00597	0.0497	2485	1
ENSG00000110442	108.5	103.1	214.0	219.4	ENSG00000110442	7.24	1.03	0.00599	0.04987	2486	1
ENSG00000105327	5.7	24.2	1.8	2.5	ENSG00000105327	2.50	−2.80	0.006	0.04989	2487	1
ENSG00000139445	17.0	2.5	0.0	0.8	ENSG00000139445	1.01	−4.55	0.006	0.04989	2488	1
ENSG00000105321	61.1	128.5	14.2	47.4	ENSG00000105321	5.76	−1.62	0.00602	0.05008	2489	0
ENSG00000118017	1.1	2.5	13.3	10.0	ENSG00000118017	2.21	2.66	0.00603	0.05009	2490	0
ENSG00000110917	768.8	591.6	1440.9	1334.8	ENSG00000110917	9.92	1.03	0.00603	0.05011	2491	0
ENSG00000119630	19.2	12.7	34.6	55.7	ENSG00000119630	4.75	1.50	0.00604	0.05011	2492	0
ENSG00000144567	431.9	402.0	810.5	868.6	ENSG00000144567	9.21	1.01	0.00605	0.05019	2493	0



M-A plotのA値とM値

***q*-value**

FDR閾値判定結果。***q*-value < 0.05**を満たすDEGが1、non-DEGが0。

例題2-5のコピペで作成しました

様々なM-A plot

- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 201
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | edgeR(Robinson_2010) (last modified 2
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | WAD(Kadota_2008) (last modified 201
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応
- 解析 | 発現変動 | 2群間 | 対応

①

解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) NEW

Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて、TCCを用いた様々な例題を示します。入力は全てサンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ(sample_blekhman_18.txt)です。ヒトHomo sapiens; HS)のメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus macaque; RM)のメス3サンプル(RMF1-3)とオス3サンプル(RMM1-3)の並びになっています。つまり、以下のような感じです。Fは Female(メス)、Mは Male(オス)を表します。

ヒト(1-6列目): HSF1, HSF2, HSF3, HSM1, HSM2, and HSM3

チンパンジー(7-12列目): PTF1, PTF2, PTF3, PTM1, PTM2, and PTM3

アカゲザル(13-18列目): RMF1, RMF2, RMF3, RMM1, RMM2, and RMM3

「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下をコピペ。

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル(G2群:RMF1とRMM1)の場合:

1, 4, 13, 16 列目のデータのみ抽出しています。

```
in_f <- "sample_blekhman_18.txt"        #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge1.txt"                   #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge1.png"                   #出力ファイル名を指定してout_f2に格納
param_subset <- c(1, 4, 13, 16)         #取り扱いたいサブセット情報を指定
param_G1 <- 2                           #G1群のサンプル数を指定
param_G2 <- 2                           #G2群のサンプル数を指定
param_FDR <- 0.05                       #DEG検出時のfalse discovery rate (FDR)閾値を指定
param_fig <- c(430, 350)                #ファイル出力時の横幅と縦幅を指定(単位はピクセル)
param_mar <- c(4, 4, 0, 0)              #下、左、上、右の順で余白を指定(単位は行)

#必要なパッケージをロード
library(TCC)                            #パッケージの読み込み
```

Contents

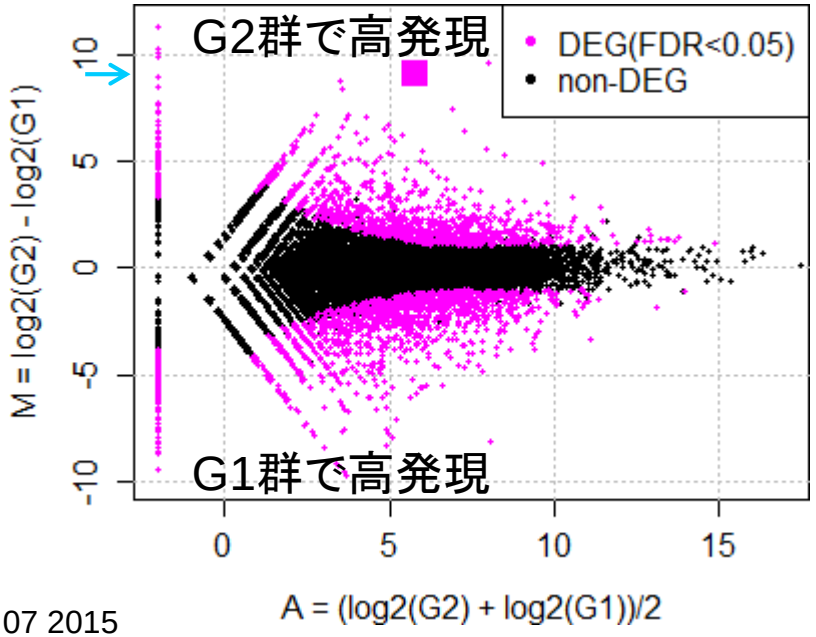
- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) : 発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

Tips：logの世界

M-A plotのM値は、倍率変化（log比）に相当（$2^{9.06}$ 倍G2群で高発現）。

G1(HS)群　G2(RM)群

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.8	1477.0	ENSG00000208570	−2.04	11.29	4.41E−53	9.13E−49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.1	ENSG00000220191	5.79	9.06	4.54E−47	4.70E−43	2	1
ENSG00000106366	4421.9	4411.0	23.1	8.3	ENSG00000106366	8.04	−8.14	2.46E−45	1.70E−41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	−2.04	10.23	3.29E−44	1.70E−40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	−2.04	10.08	1.74E−43	7.21E−40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	−2.04	10.03	4.68E−42	1.61E−38	6	1
ENSG00000209007	0.0	0.0	615.2	479.6	ENSG00000209007	−2.04					
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67					
ENSG00000156222	367.5	301.5	0.9	0.0	ENSG00000156222	3.61					
ENSG00000165272	404.8	428.0	2.6	0.8	ENSG00000165272	4.02					

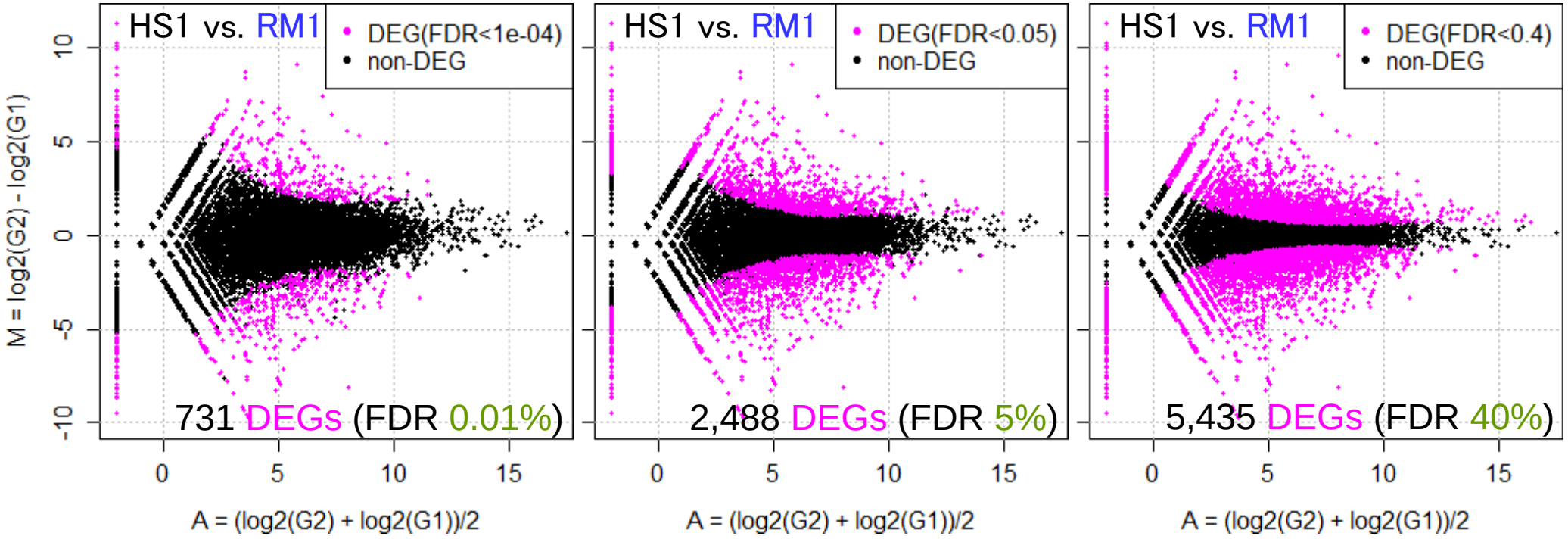


R Console

```
> (2.3 + 2.5)/2
[1] 2.4
> (1394.7 + 1171.1)/2
[1] 1282.9
> (log2(1282.9) + log2(2.4))/2
[1] 5.794114
> log2(1282.9) - log2(2.4)
[1] 9.062159
> 1282.9/2.4
[1] 534.5417
> log2(1282.9/2.4)
[1] 9.062159
> 2^9.062159
[1] 534.5418
>
```

分布やモデル

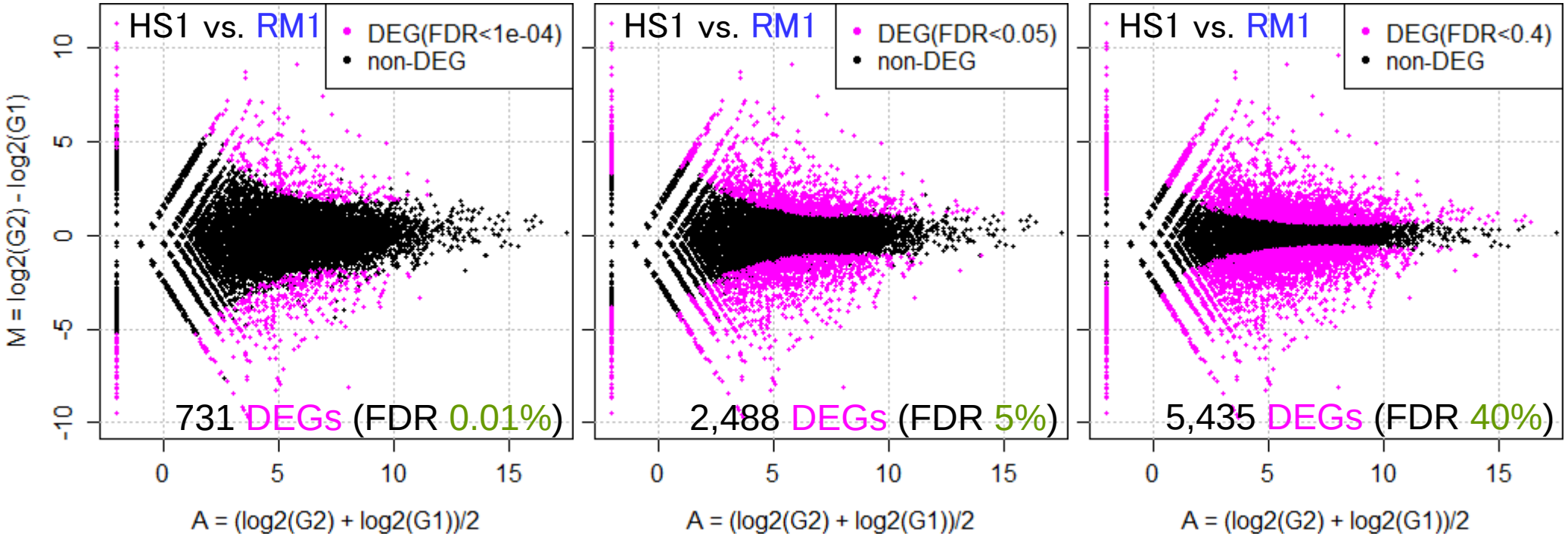
（当たり前だが）FDR閾値を緩めると得られるDEG数は増える傾向にあることがわかる。例題6のコピペで作成。



厳しめ ← FDR閾値 → 緩め

重要：黒の分布はnon-DEGの分布に相当

分布やモデル

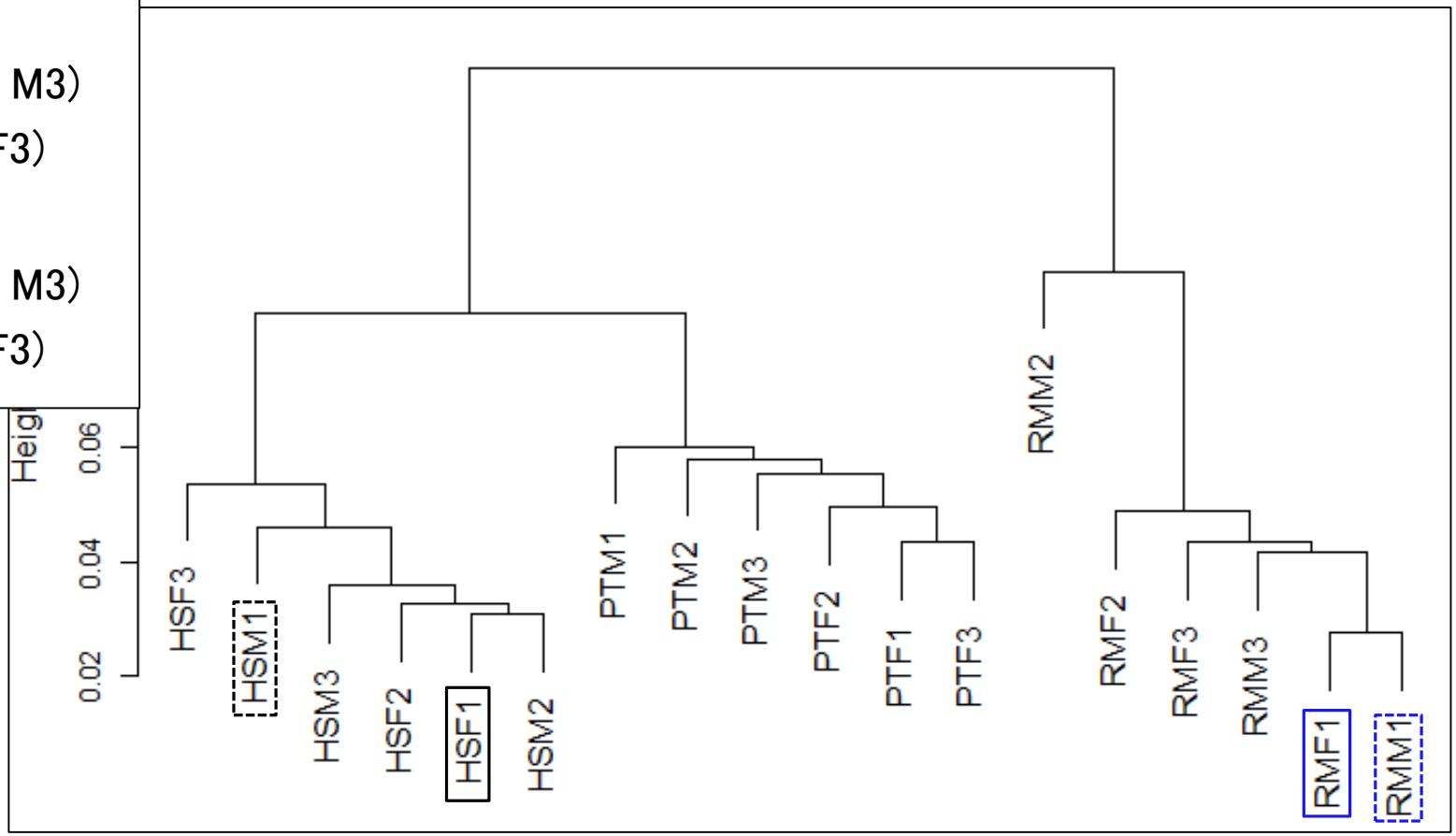


厳しめ ← FDR閾値 → 緩め

「HS vs. RM」の発現変動解析結果として、20,689 genes中3,300個程度が本物のDEGと判断した。

おさらい

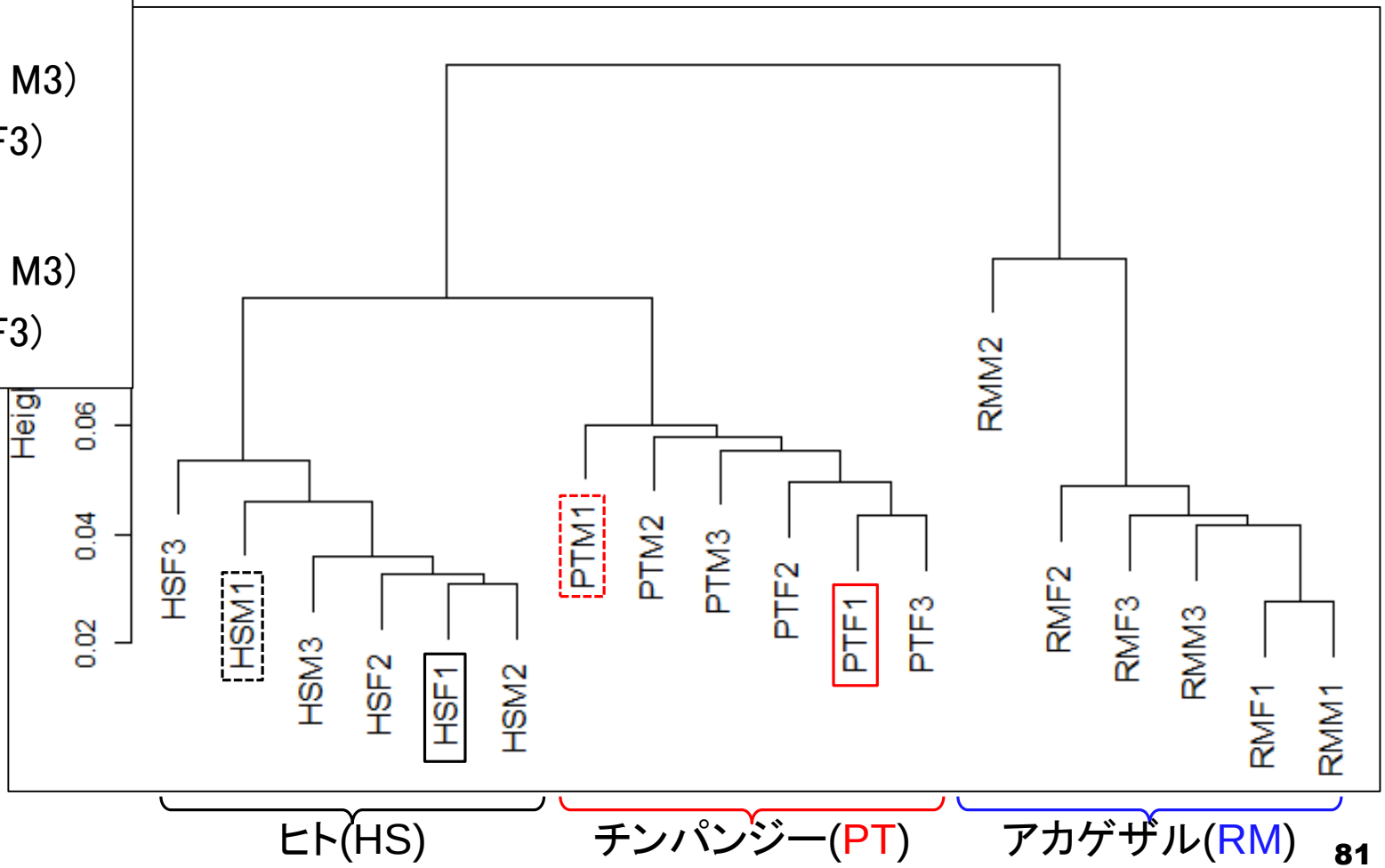
- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)



2群間比較

「HS vs. PT」のDEG同定を行う。ヒト(HS)とチンパンジー(PT)で明瞭にサブクラスターに分かれていることから、DEGは存在すると予想される。しかし、「HS vs. RM」(3,300個程度が本物のDEGと判断した)のときほどDEGは多くないだろうと予想できる。

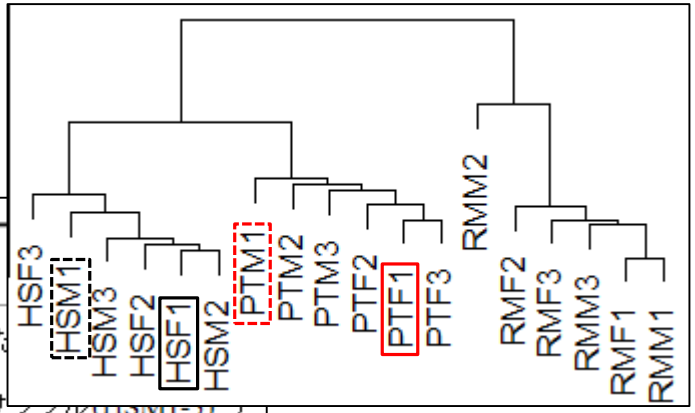
- ヒト(HS)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- チンパンジー(PT)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1, F2, F3)



TCC実行

「HS vs. PT」の2群間比較をTCCパッケージ(Sun et al., 2013)で行ってみよう。

- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) ①
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 201
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | edgeR(Robinson_2010) (last modified
- 解析 | 発現変動 | 2
- 解析 | 発現変動 | 2
- 解析 | 発現変動 | 2
- 解析 | 発現変動 | 2



解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデ
(Sun_2013) NEW

Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて、TCCを用いた
ます。入力は全てサンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ
(sample_blekhman_18.txt)です。ヒトHomo sapiens; HS)のメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3),
ンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus
macaque; RM)のメス
うな感じです。FIはFer
ヒト(1-6列目): HSF1, HSF
チンパンジー(7-12列目):
アカゲザル(13-18列目): F
「ファイル」-「ディレクトリの

1. ヒト2サンプル(G1群:H

1, 4, 13, 16 列目のデー

```
in_f <- "sample_b
out_f1 <- "hoge1.
out_f2 <- "hoge1.
param_subset <- c
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
```

②

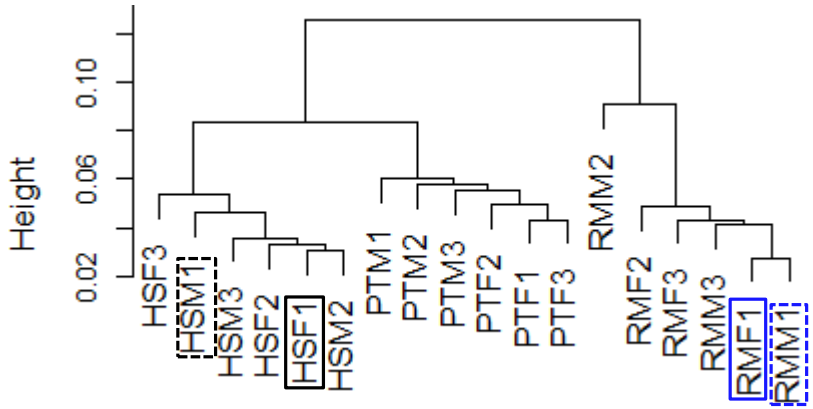
7. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:

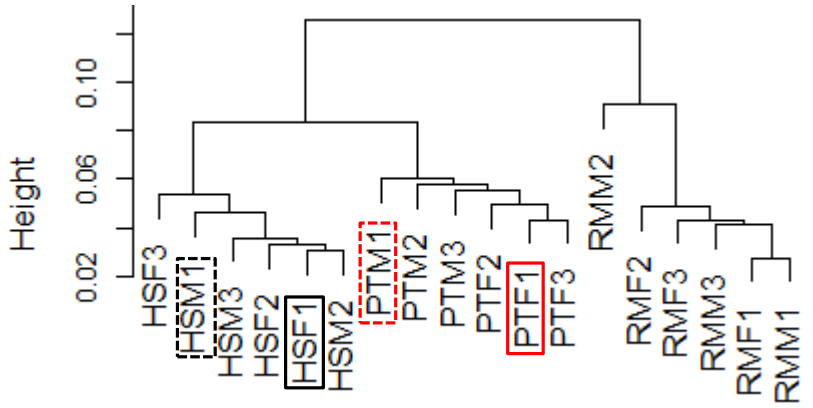
Blekhman et al., Genome Res., 2010の20,689 genes×18 samplesのカウントデータです。ヒトのメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーのメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザルのメス3サンプル(RMF1-3)とオス3サンプル(RMM1-3)の並びになっています。ここでは、1, 4, 7, 10 列目のデータのみ抽出して、ヒト2サンプル(G1群:HSF1とHSM1) vs. チンパンジー2サンプル(G2群:PTF1とPTM1)の2群間比較を行います。

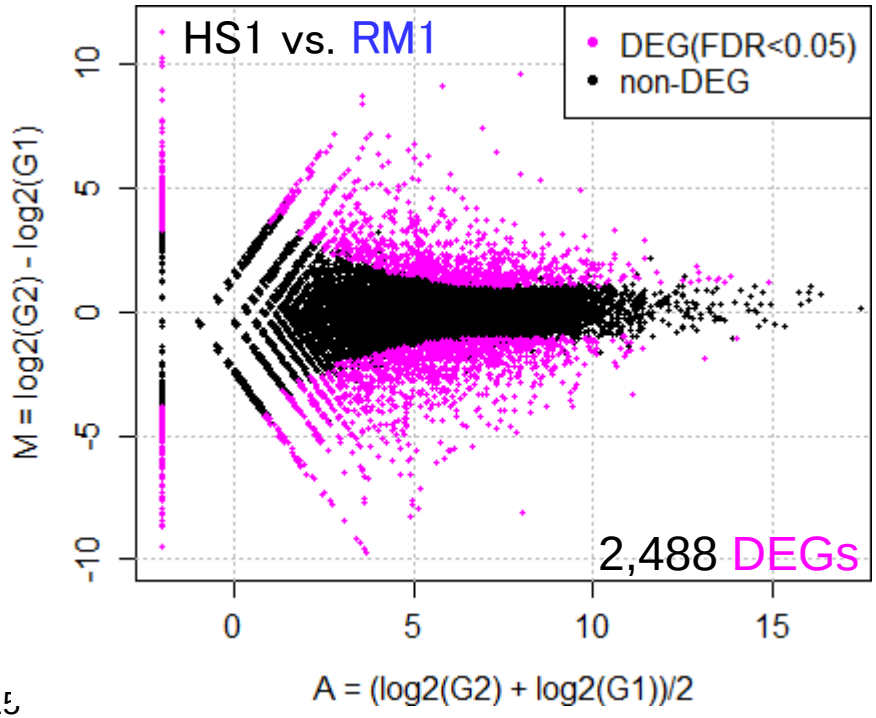
```
in_f <- "sample_blekhman_18.txt"          #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge7.txt"                     #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge7.png"                     #出力ファイル名を指定してout_f2に格納
param_subset <- c(1, 4, 7, 10)            #取り扱いたいサブセット情報を指定
param_G1 <- 2                             #G1群のサンプル数を指定
param_G2 <- 2                             #G2群のサンプル数を指定
param_FDR <- 0.05                         #DEG検出時のfalse discovery rate (
param_fig <- c(400, 310)                  #ファイル出力時の横幅と縦幅を指定(単
param_mar <- c(4, 4, 0, 0)                #下、左、上、右の順で余白を指定(単位

#必要なパッケージをロード                                    #パッケージの読み込み
library(TCC)
```

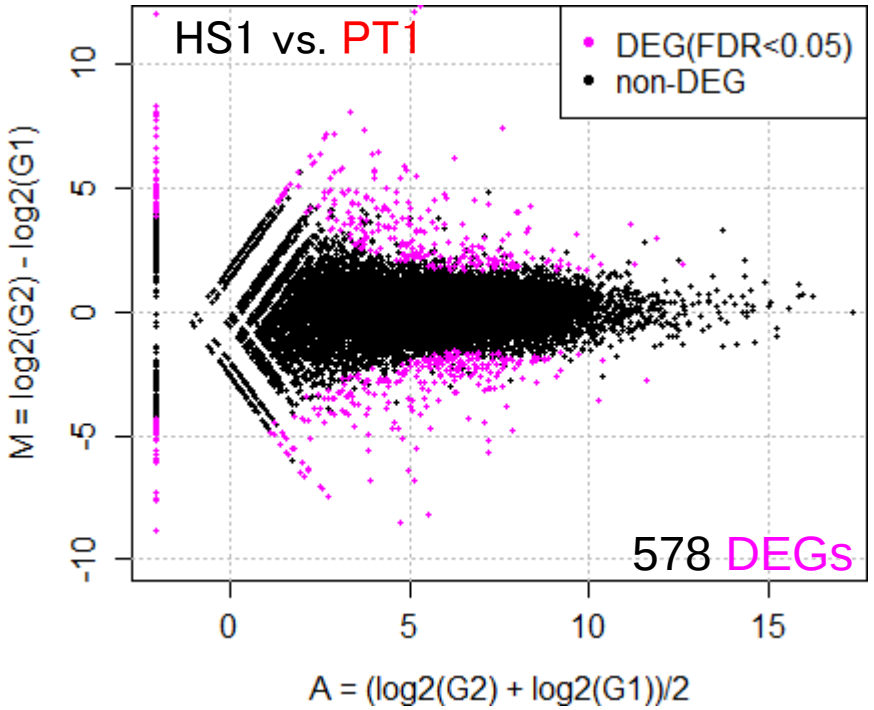
結果の比較

「HS vs. PT」は、「HS vs. RM」に比べて全体的に似ているのでDEG数は少なくなる。



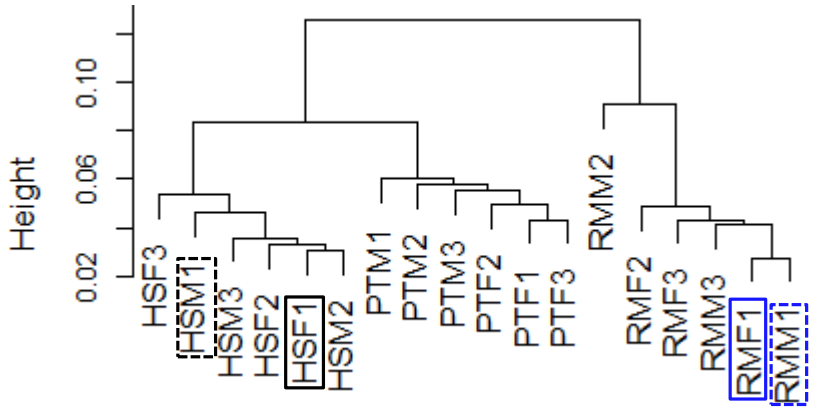


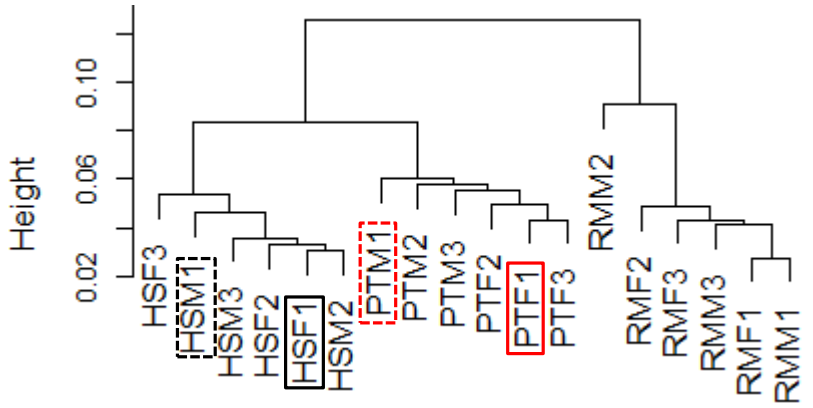


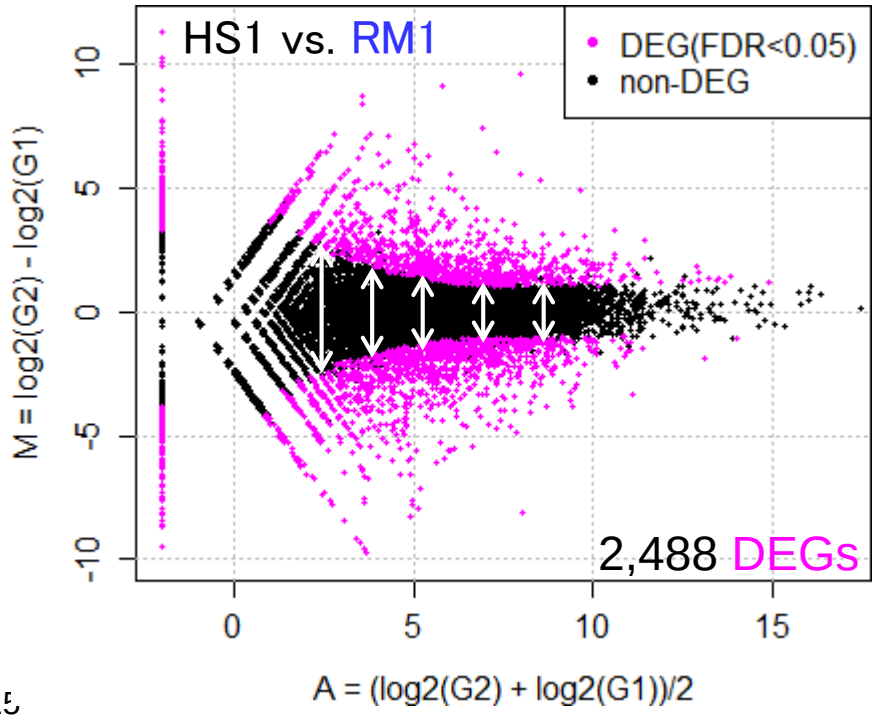


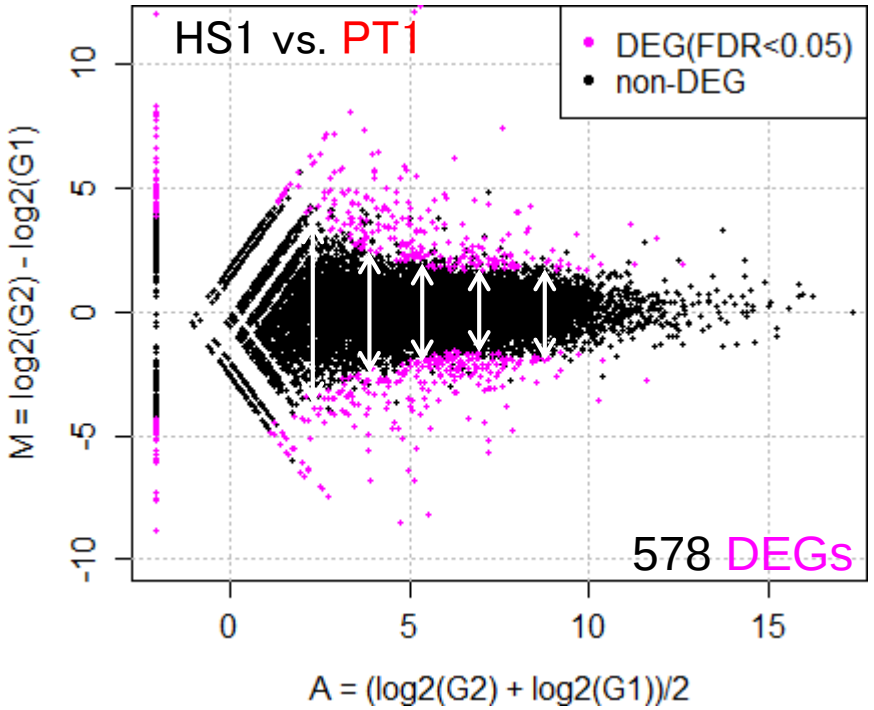
素朴な疑問

何故、白矢印で示すように「HS vs. PT」のnon-DEGの分布(黒の点の分布)は、「HS vs. RM」に比べて広がっているのか?





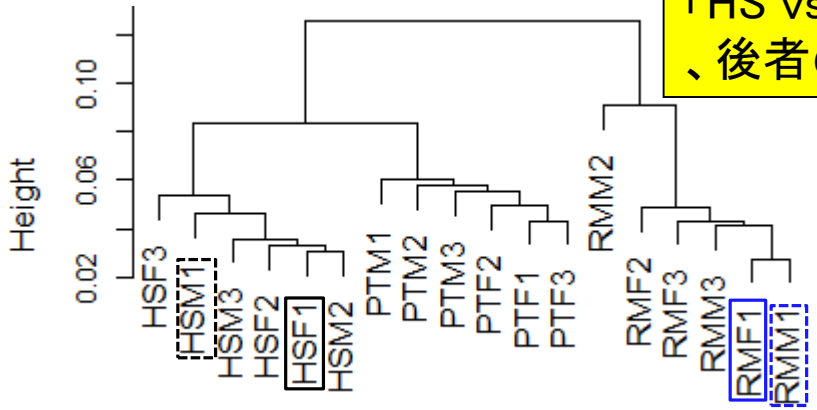


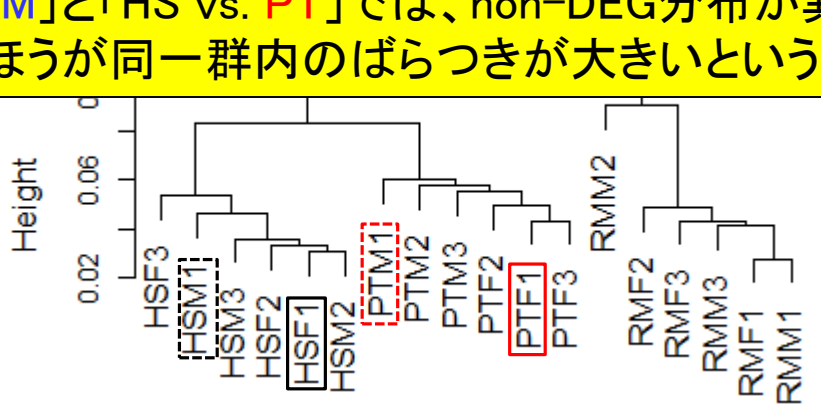


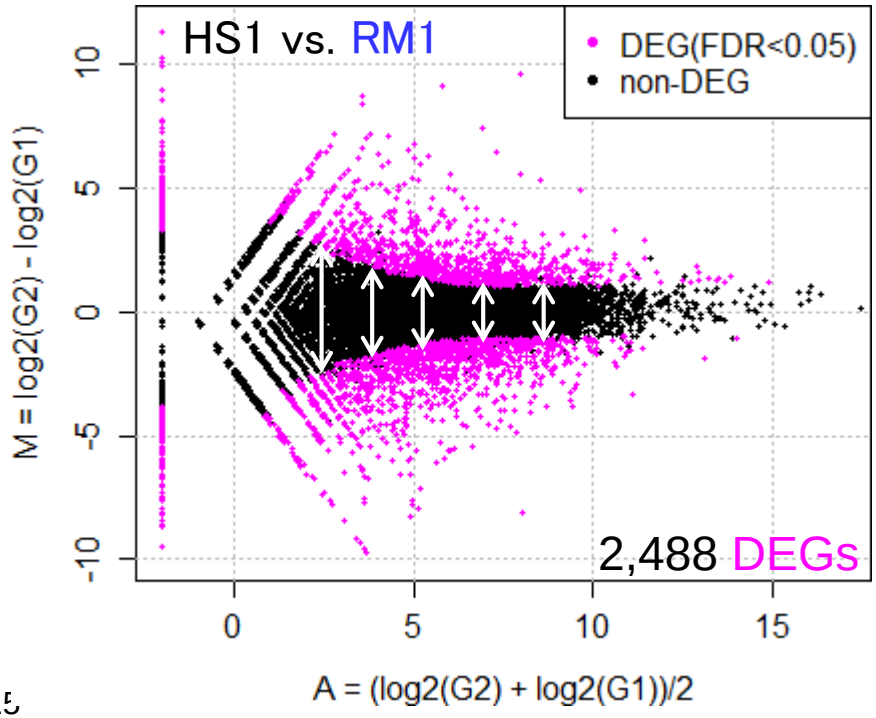
参考

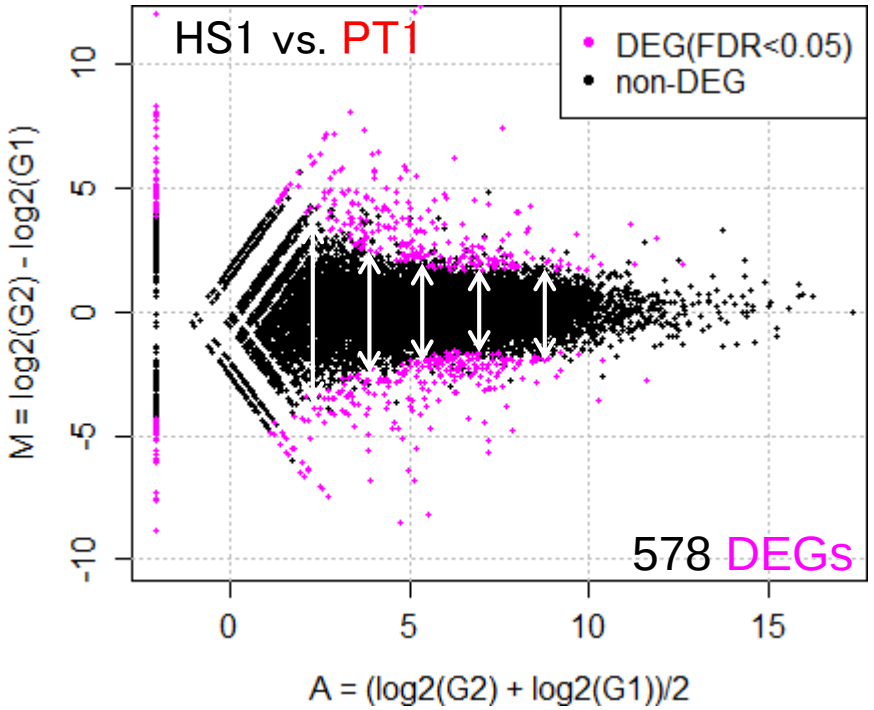
統計的手法とは

疑問に対する解答は、統計的手法の原理を再考すればよい。同一群内のばらつきの分布（non-DEG分布）以外のものがDEGと判定されるのが統計的手法の結果。つまり、「HS vs. RM」と「HS vs. PT」では、non-DEG分布が異なり、後者のほうが同一群内のばらつきが大きいということ。





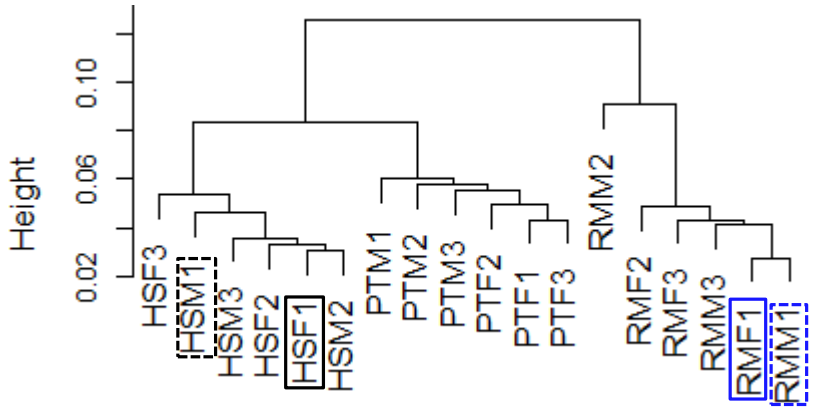


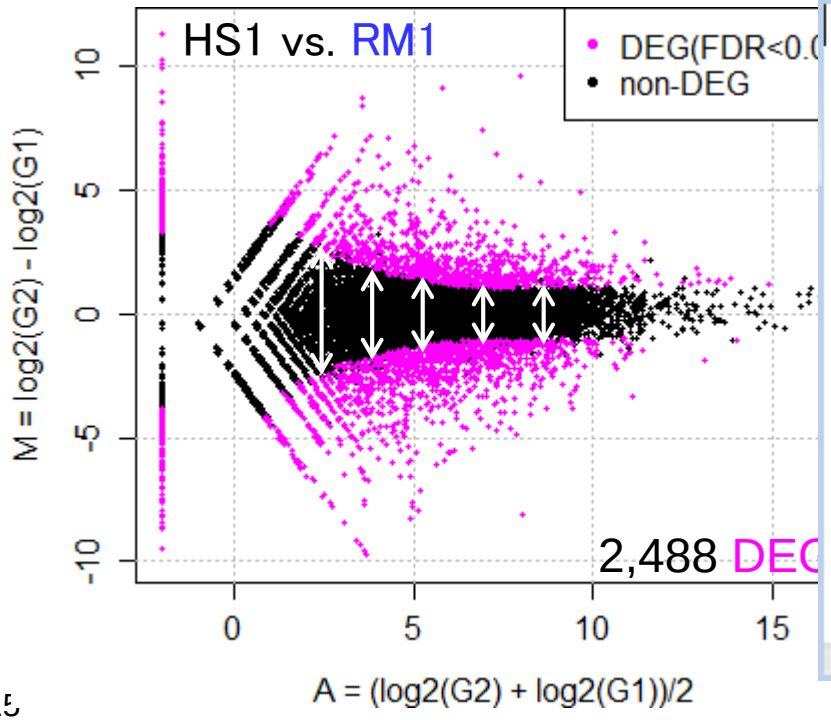


参考

サンプル間類似度

同一群内のばらつきは、サンプル間の類似度で大まかに把握可能。①HS群内(HSF1 vs. HSM1)のSpearman相関係数は0.950。②RM群内(RMF1 vs. RMM1)は0.972。③「HS vs. RM」の群間比較結果は、例えばHSM1 vs. RMM1の相関係数(0.880)が0.950と0.972よりも低いことからDEGの存在を予測可能。





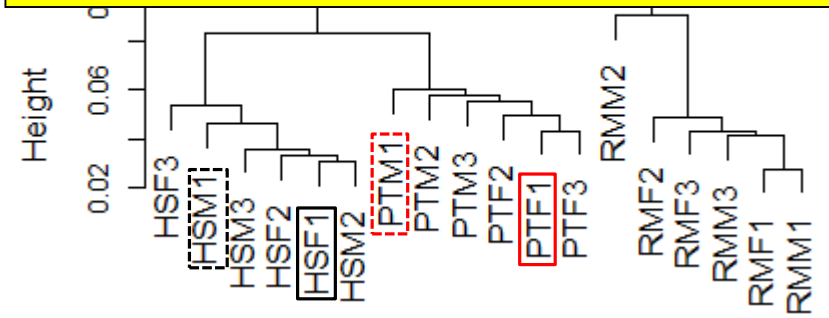
R Console

```
> in_f <- "sample_blekhman_18.txt"          #入$
> data <- read.table(in_f, header=TRUE, row.n$
> dim(data)
[1] 20689    18
> data <- unique(data)
> dim(data)
[1] 16561    18
> cor(data$HSM1, data$HSF1, method="spearman")
[1] 0.9502333  ①
> cor(data$RMM1, data$RMF1, method="spearman")
[1] 0.9724166  ②
> cor(data$HSM1, data$RMM1, method="spearman")
[1] 0.8799668  ③
>
```

参考

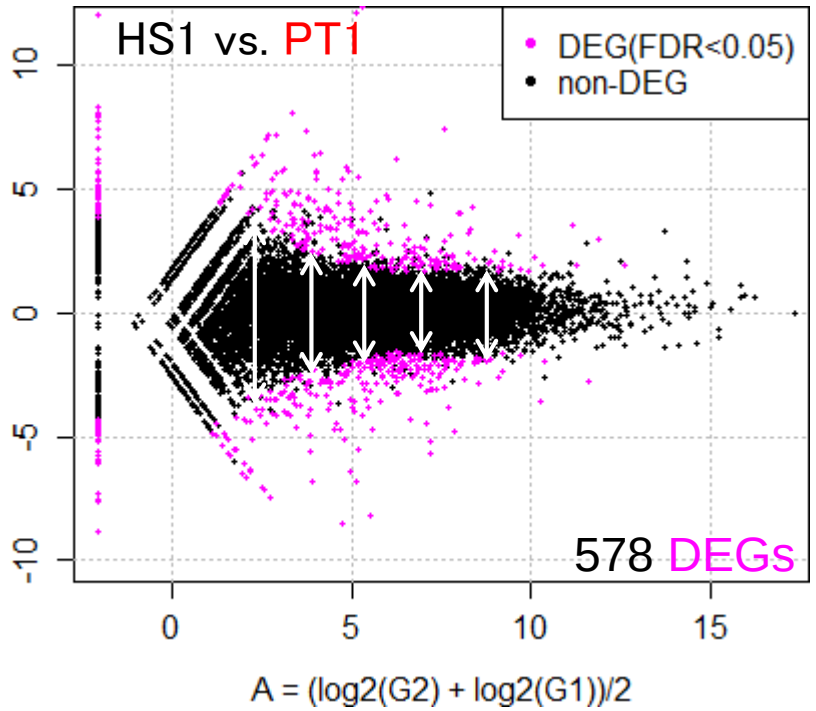
サンプル間類似度

①HS群内(HSF1 vs. HSM1)のSpearman相関係数は0.95。②PT群内(PTF1 vs. PTM1)は0.950。③「HS vs. PT」の群間比較結果は、例えばHSM1 vs. PTM1の相関係数(0.902)が0.950と0.949よりも低いことからDEGの存在を予測可能。



R Console

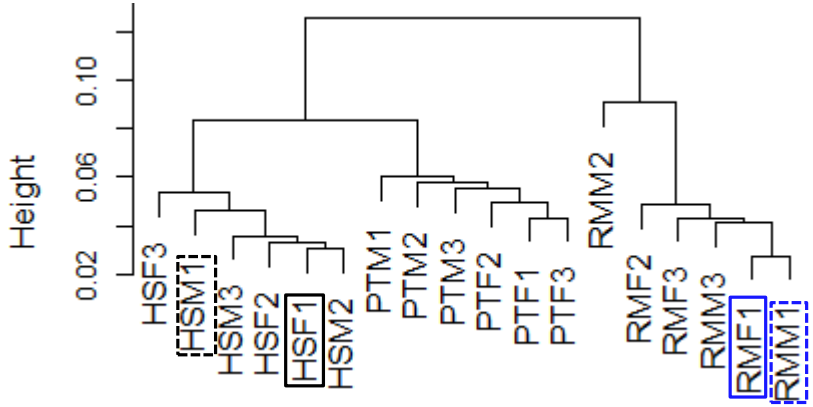
```
> in_f <- "sample_blekhman_18.txt"          #入$
> data <- read.table(in_f, header=TRUE, row.n$
> dim(data)
[1] 20689    18
> data <- unique(data)
> dim(data)
[1] 16561    18
> cor(data$HSM1, data$HSF1, method="spearman")
[1] 0.9502333 ①
> cor(data$PTM1, data$PTF1, method="spearman")
[1] 0.9489023 ②
> cor(data$HSM1, data$PTM1, method="spearman")
[1] 0.9019057 ③
>
```

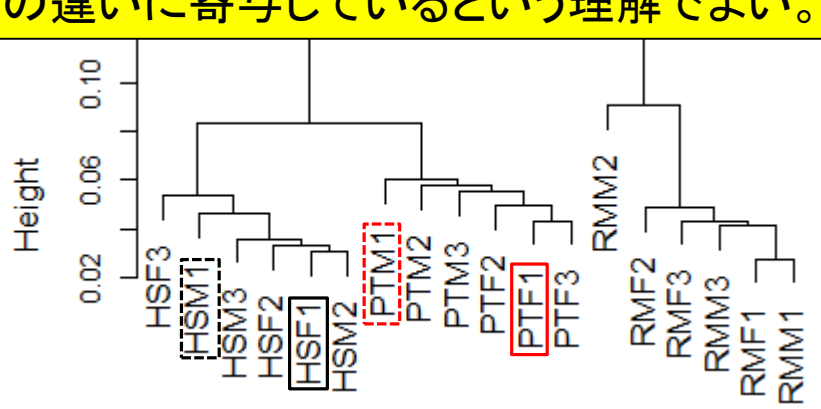


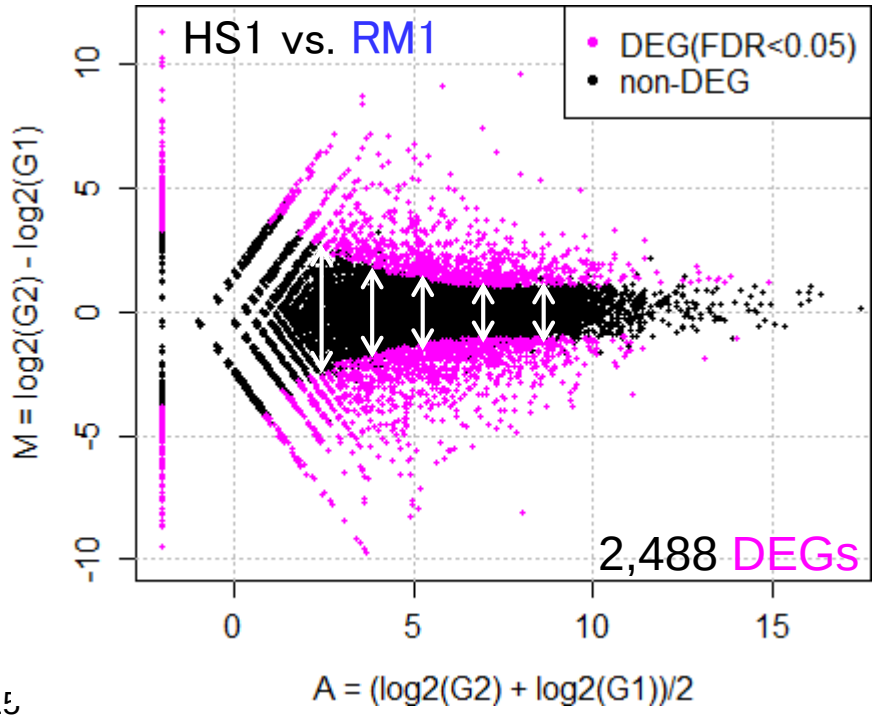
参考

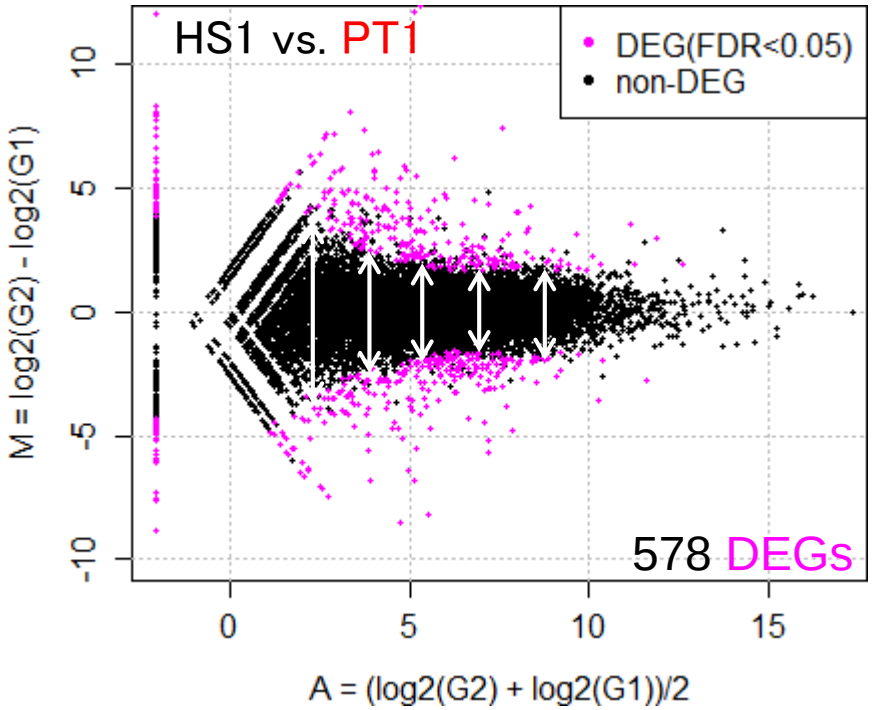
DEG検出結果の比較

RM群内(RMF1 vs. RMM1)のSpearman相関係数は0.972。一方、PT群内(PTF1 vs. PTM1)は0.949。大まかにいって、この差がnon-DEG分布の違いに寄与しているという理解でよい。









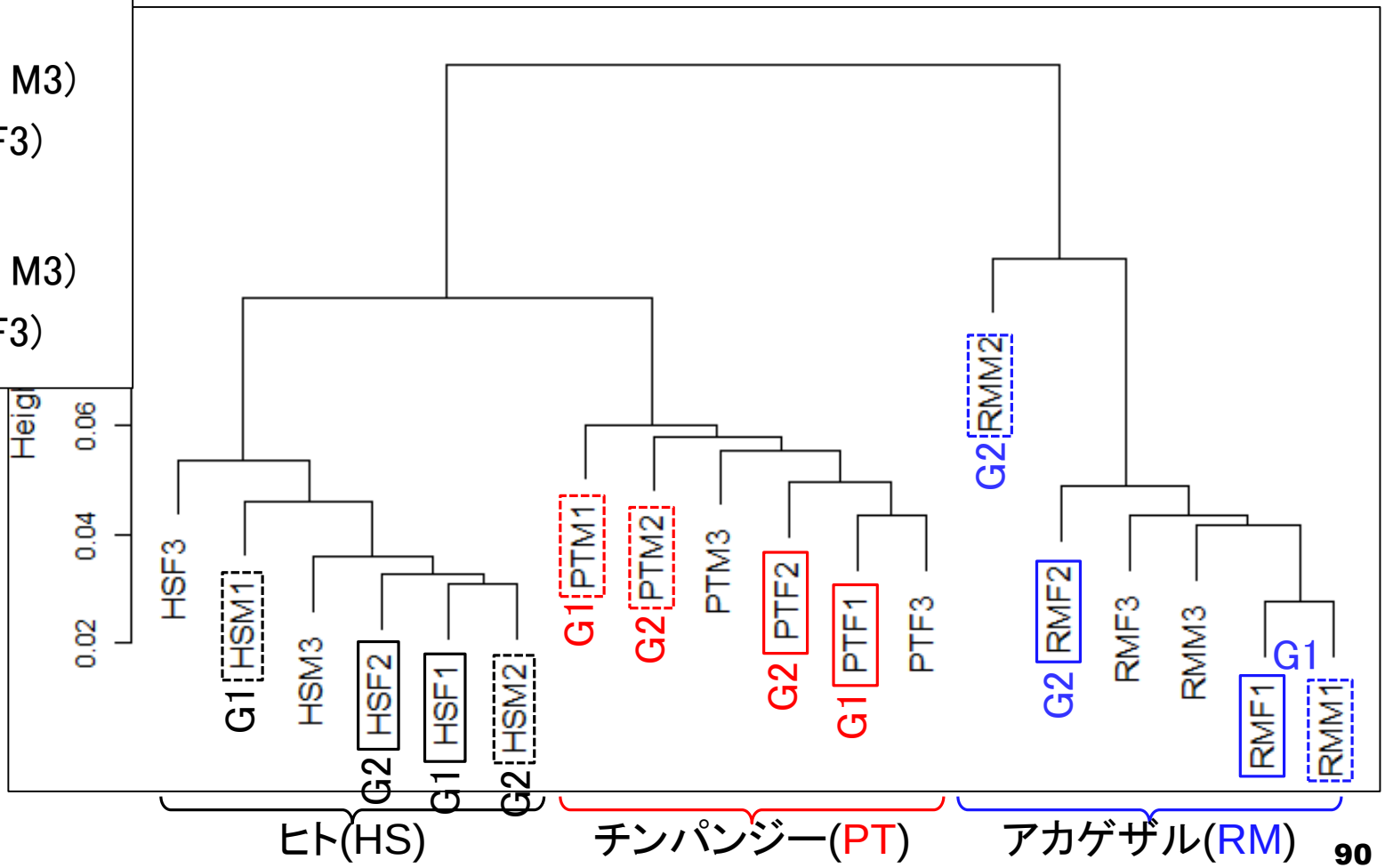
Contents

- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) : 発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

2群間比較

同一群内のばらつきの分布（non-DEG分布）を調べるべく、「G1群(M1とF1) vs. G2群(M2とF2)」の2群間比較を行ってみる。予想はDEGはあったとしてもごくわずか。

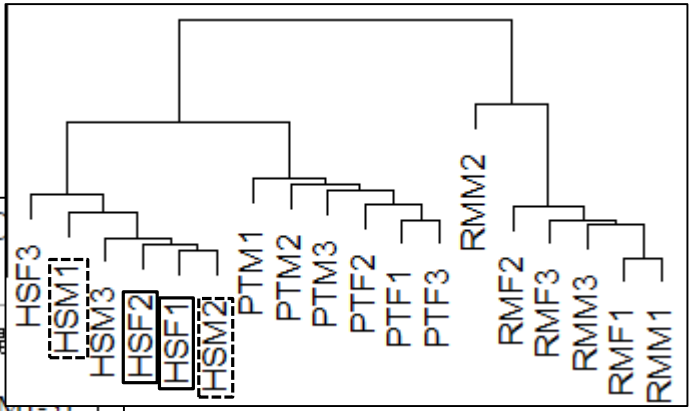
- ヒト(HS)
  - オス3匹(M1(G1), M2(G2), M3)
  - メス3匹(F1(G1), F2(G2), F3)
- チンパンジー(PT)
  - オス3匹(M1(G1), M2(G2), M3)
  - メス3匹(F1(G1), F2(G2), F3)
- アカゲザル(RM)
  - オス3匹(M1, M2, M3)
  - メス3匹(F1(G1), F2(G2), F3)



TCC実行

「G1群(HSM1とHSF1) vs. G2群(HSM2とHSF2)」の2群間比較結果。7 DEGs。

- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) ①
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 201



解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) NEW

Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて、TCCを用いた様々な例題を用意しています。入力は全てサンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ(sample_blekhman_18.txt)です。ヒトHomo sapiens; HS)のメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus macaque; RM)のメス3サンプル(RMF1-3)とオス3サンプル(RMM1-3)のRNA-seqデータのような感じ

ヒト(1-6列目

チンパンジー

アカゲザル(

「ファイル」

1. ヒト2サン

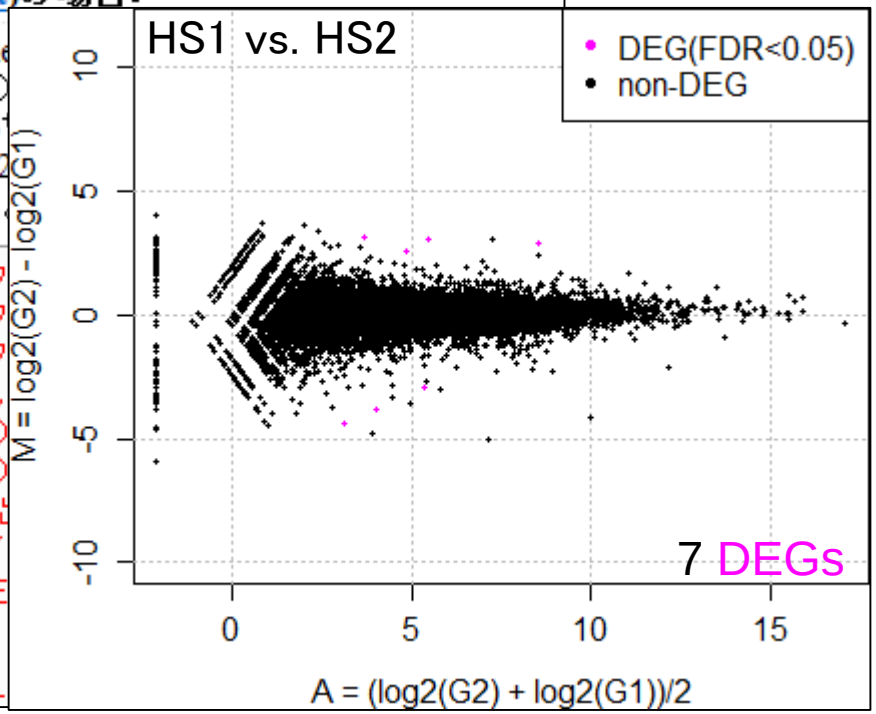
1, 4, 13, 1

```
in_f <-
out_f1
out_f2
param_s
param_G
param_G
param_F
```

② 8. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:

Blekhman et al., Genome Res., 2010の 20,689 genes×18 sample
サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーの
ブル(PTM1-3), アカゲザルのメス3サンプル(RMF1-3)とオス3
います。ここでは、1, 4, 2, 5 列目のデータのみ抽出して、ヒト2
ヒト22サンプル(G2群:HSF2とHSM2)の2群間比較を行います

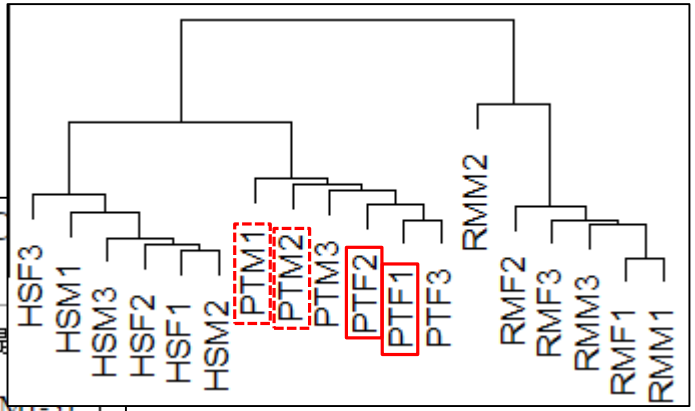
```
in_f <- "sample_blekhman_18.txt"          #入力フ
out_f1 <- "hoge8.txt"                     #出力フ
out_f2 <- "hoge8.png"                     #出力フ
param_subset <- c(1, 4, 2, 5)             #取り扱い
param_G1 <- 2                             #G1群の
param_G2 <- 2                             #G2群の
param_FDR <- 0.05                         #DEG検出
param_fig <- c(430, 350)                  #ファイ
param_mar <- c(4, 4, 0, 0)                #下、左

#必要なパッケージをロード
library(TCC)                              #パッケ
```



TCC実行

「G1群(PTM1とPTF1) vs. G2群(PTM2とPTF2)」の2群間比較結果。16 DEGs。

- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) ①
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 201



解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) NEW

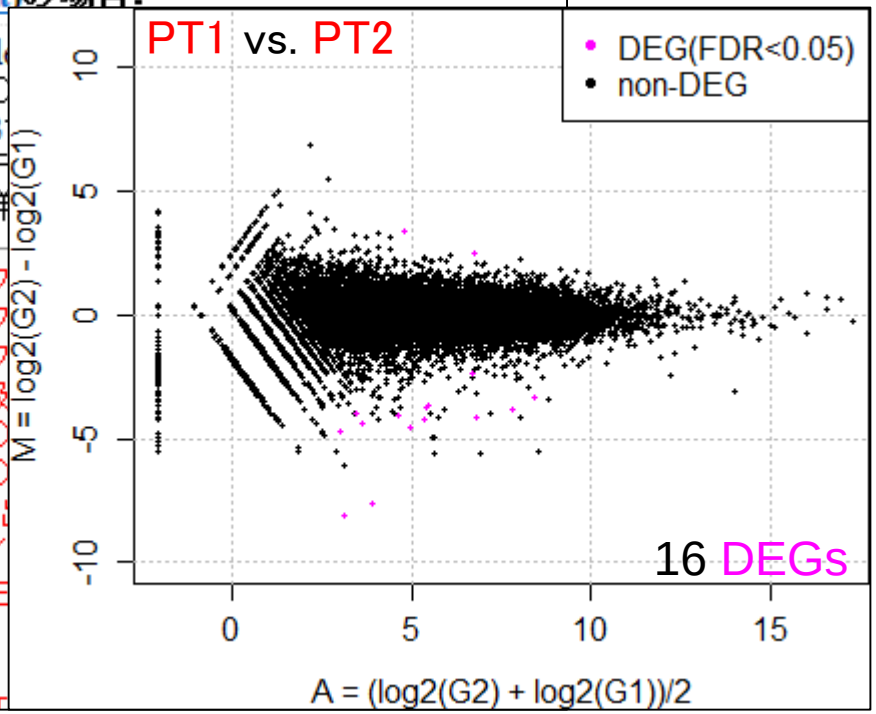
Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて、TCCを用いた様々な例題を用意しています。入力は全てサンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ(sample_blekhman_18.txt)です。ヒトHomo sapiens; HS)のメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus macaqu

②

9. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:

Blekhman et al., Genome Res., 2010の20,689 genes×18 sampl
サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーの
ブル(PTM1-3), アカゲザルのメス3サンプル(RMF1-3)とオス3
います。ここでは、7, 10, 8, 11列目のデータのみ抽出して、チ
PTM1) vs. チンパンジー2サンプル(G2群:PTF2とPTM2)の2群

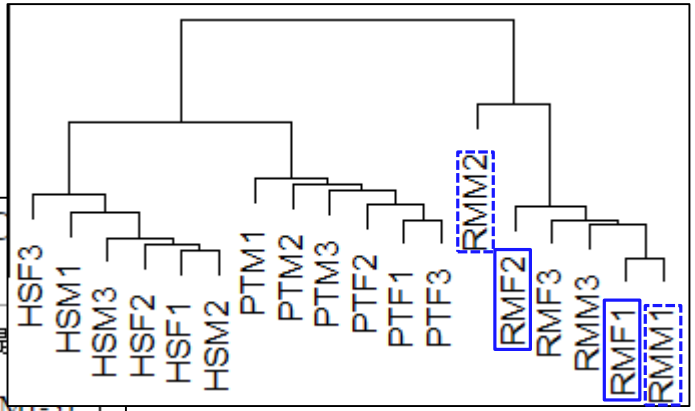
```
in_f <- "sample_blekhman_18.txt"          #入力フ
out_f1 <- "hoge9.txt"                     #出力フ
out_f2 <- "hoge9.png"                     #出力フ
param_subset <- c(7, 10, 8, 11)           #取り扱
param_G1 <- 2                             #G1群の
param_G2 <- 2                             #G2群の
param_FDR <- 0.05                         #DEG検出
param_fig <- c(430, 350)                  #ファイ
param_mar <- c(4, 4, 0, 0)                #下、左

#必要なパッケージをロード
library(TCC)                              #パッケ
```



TCC実行

「G1群(RMM1とRMF1) vs. G2群(RMM2とRMF2)」の2群間比較結果。24 DEGs。

- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) ①
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 201



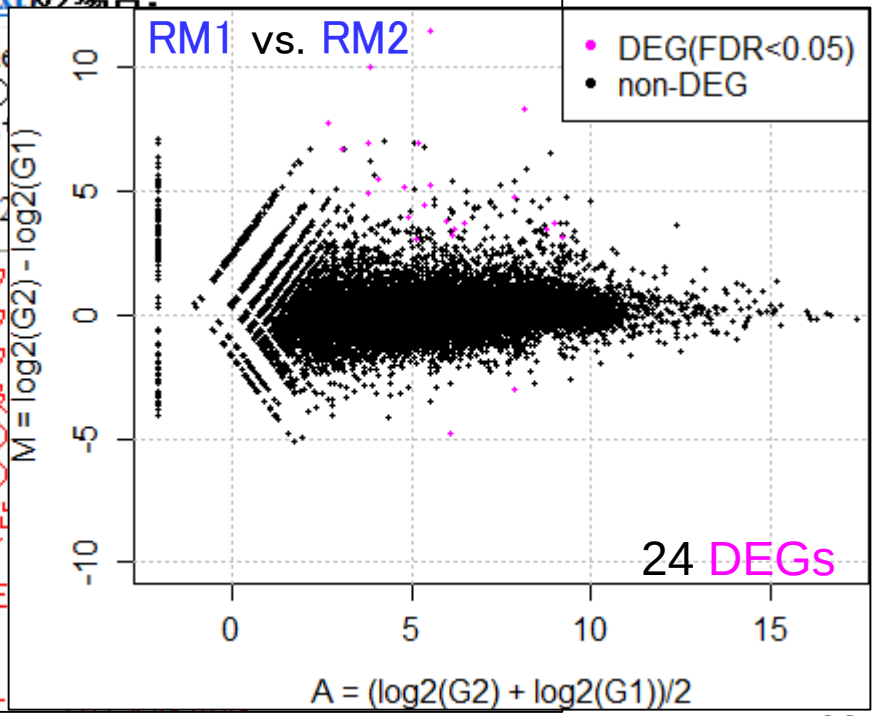
解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) NEW

Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて、TCCを用いた様々な例題を用意しています。入力は全てサンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ(sample_blekhman_18.txt)です。ヒトHomo sapiens; HS)のメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus macaque

② **10. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:**

Blekhman et al., Genome Res., 2010の20,689 genes×18 sample
サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーの
ブル(PTM1-3), アカゲザルのメス3サンプル(RMF1-3)とオス3
います。ここでは、13, 16, 14, 17列目のデータのみ抽出して、
とRMM1) vs. アカゲザル2サンプル(G2群:RMF2とRMM2)の

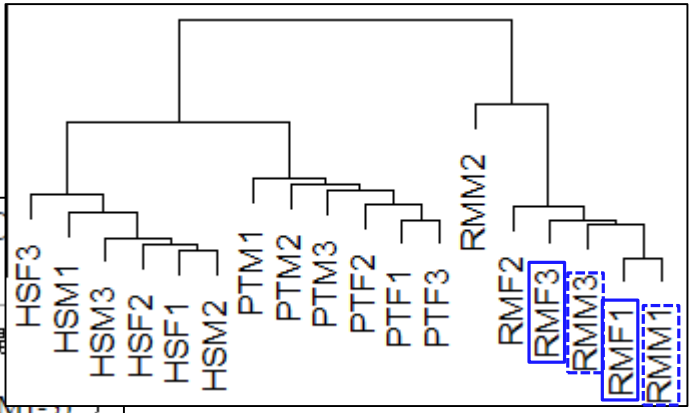
```
in_f <- "sample_blekhman_18.txt"          #入力フ
out_f1 <- "hoge10.txt"                    #出力フ
out_f2 <- "hoge10.png"                    #出力フ
param_subset <- c(13, 16, 14, 17)         #取り扱
param_G1 <- 2                             #G1群の
param_G2 <- 2                             #G2群の
param_FDR <- 0.05                         #DEG検出
param_fig <- c(430, 350)                  #ファイ
param_mar <- c(4, 4, 0, 0)                #下、左

#必要なパッケージをロード
library(TCC)                              #パッケ
```



TCC実行(おまけ)

「G1群(RMM1とRMF1) vs. G2群(RMM3とRMF3)」の2群間比較結果。202 DEGs。

- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) ①
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 201



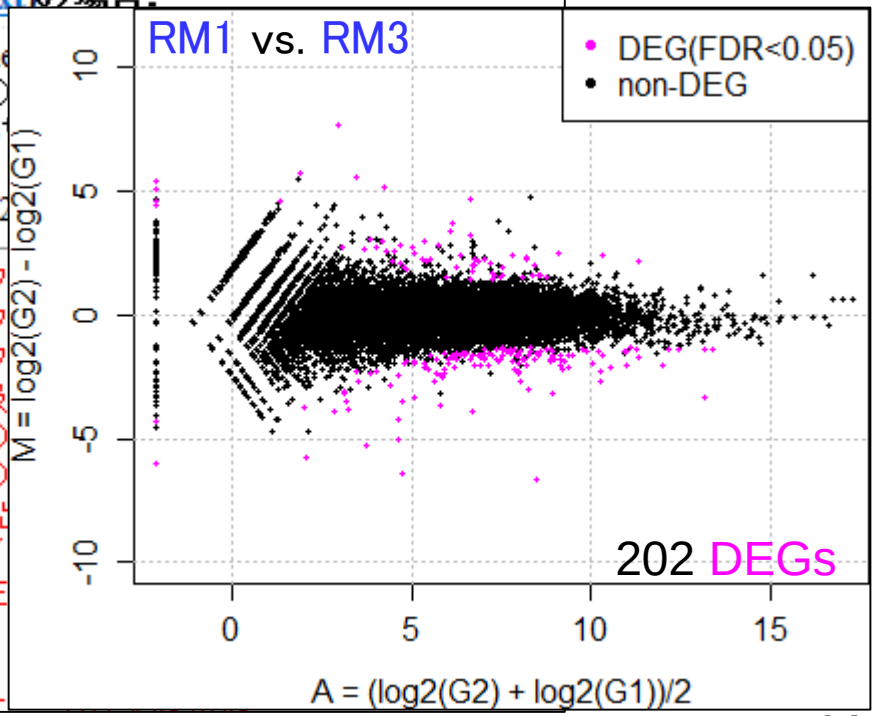
解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) NEW

Blekhman et al., Genome Res., 2010の公共カウントデータ解析に特化させて、TCCを用いた様々な例題を用意しています。入力は全てサンプルデータ42の20,689 genes×18 samplesのリアルカウントデータ(sample_blekhman_18.txt)です。ヒトHomo sapiens; HS)のメス3サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジー(Pan troglodytes; PT)のメス3サンプル(PTF1-3)とオス3サンプル(PTM1-3), アカゲザル(Rhesus macaque

② 11. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:

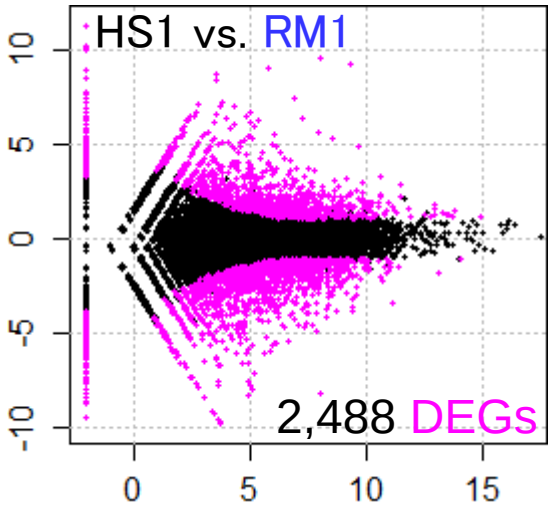
Blekhman et al., Genome Res., 2010の20,689 genes×18 sample
サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーの
ブル(PTM1-3), アカゲザルのメス3サンプル(RMF1-3)とオス3
います。ここでは、13, 16, 15, 18列目のデータのみ抽出して、
とRMM1) vs. アカゲザル2サンプル(G2群:RMF3とRMM3)の

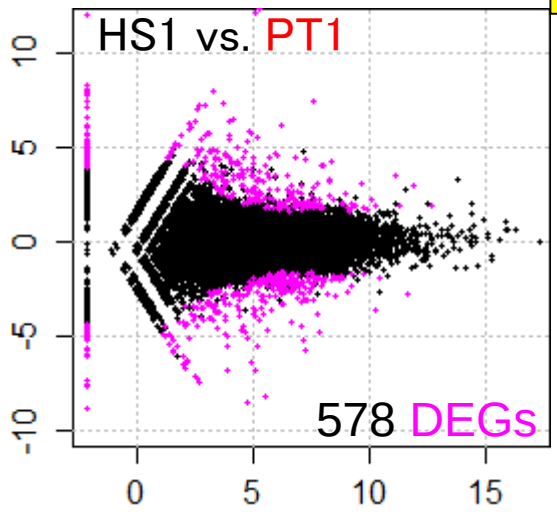
```
in_f <- "sample_blekhman_18.txt"          #入力フ
out_f1 <- "hoge11.txt"                    #出力フ
out_f2 <- "hoge11.png"                    #出力フ
param_subset <- c(13, 16, 15, 18)         #取り扱
param_G1 <- 2                             #G1群の
param_G2 <- 2                             #G2群の
param_FDR <- 0.05                         #DEG検出
param_fig <- c(430, 350)                  #ファイ
param_mar <- c(4, 4, 0, 0)                #下、左

#必要なパッケージをロード
library(TCC)                              #パッケ
```

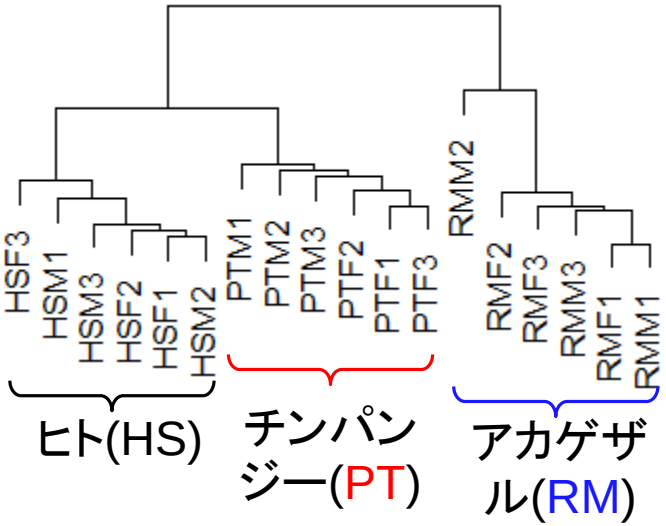


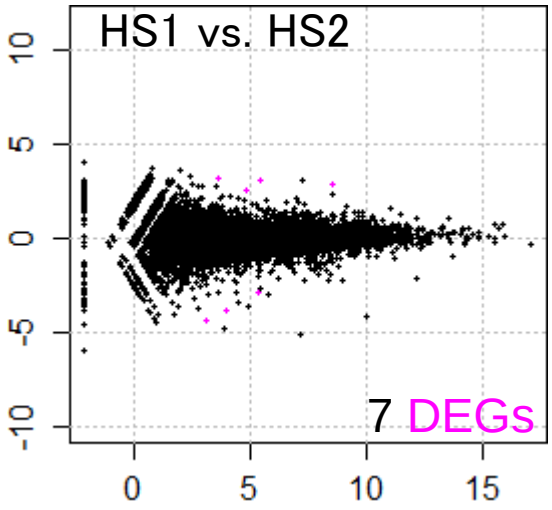
結果の比較

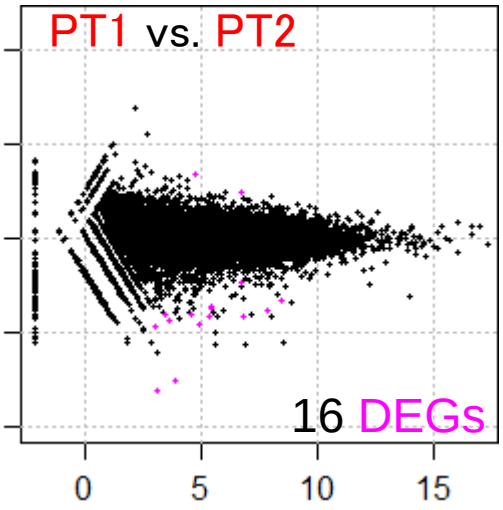
同一群(下段)の分布は、異なる群(上段)のnon-DEG分布とよく一致する。同一群内のばらつきの分布(non-DEG分布)以外のものがDEGと判定されるのが統計的手法の結果

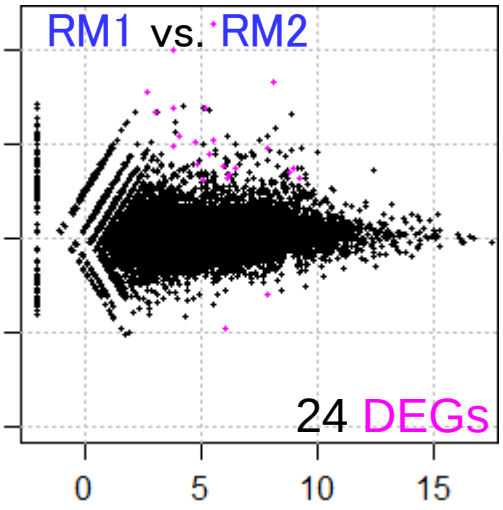


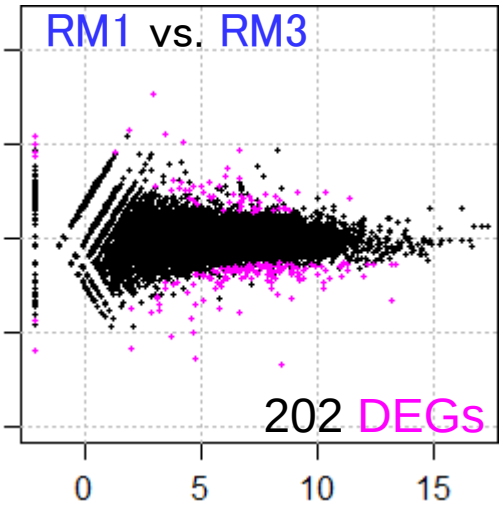










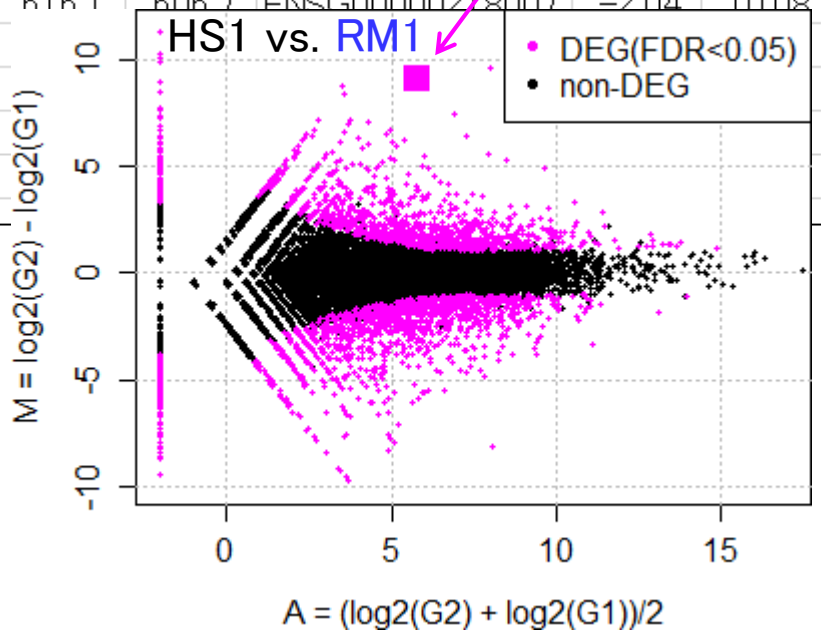


統計的手法とは

同一群内のばらつきの分布（non-DEG分布）から遠く離れたところに位置するものは、0に近い***p***-value

- 同一群内の遺伝子の**ばらつき**の程度を把握し、帰無仮説に従う分布の全体像を把握しておく（モデル構築）
  - non-DEGのばらつきの程度を把握しておくことと同義
- 実際に比較したい2群の遺伝子の**ばらつき**の程度がnon-DEG分布のどのあたりに位置するかを評価（検定）

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.8	1477.0	ENSG00000208570	−2.04	11.29	4.41E−53	9.13E−49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.1	ENSG00000220191	5.79	9.06	4.54E−47	4.70E−43	2	1
ENSG00000106366	4421.9	4411.0	23.1	8.3	ENSG00000106366	8.04	−8.14	2.46E−45	1.70E−41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	−2.04	10.23	3.29E−44	1.70E−40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	−2.04	10.08	1.74E−43	7.21E−40	5	1
ENSG00000070985	0.0	0.0						4.68E−42	1.61E−38	6	1
ENSG00000209007	0.0	0.0						1.24E−40	3.67E−37	7	1
ENSG00000182327	367.5	363.9						1.52E−38	3.93E−35	8	1
ENSG00000156222	367.5	301.5						1.09E−36	2.51E−33	9	1
ENSG00000165272	404.8	428.9						5.45E−36	1.12E−32	10	1

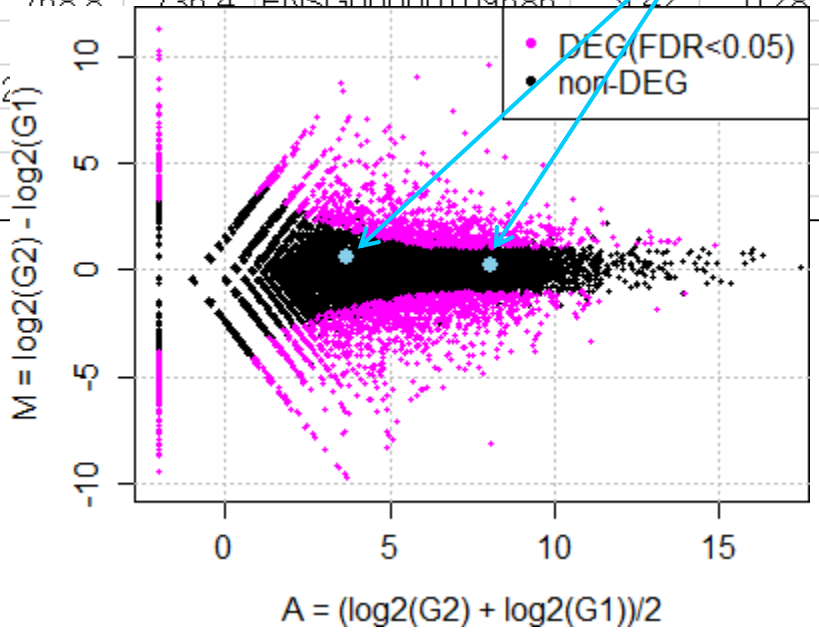


統計的手法とは

同一群内のばらつきの分布（non-DEG分布）のど真ん中に位置するものは、1に近い***p***−value

- 同一群内の遺伝子の**ばらつき**の程度を把握し、帰無仮説に従う分布の全体像を把握しておく（モデル構築）
  - non-DEGのばらつきの程度を把握しておくことと同義
- 実際に比較したい2群の遺伝子の**ばらつき**の程度がnon-DEG分布のどのあたりに位置するかを評価（検定）

rownames(tcc$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000047578	69.0	131.0	98.5	148.8	ENSG00000047578	6.80	0.31	0.4906	1	9727	0
ENSG00000115325	3.4	17.8	16.9	14.1	ENSG00000115325	3.68	0.55	0.49087	1	9728	0
ENSG00000122257	256.7	234.1	253.9	334.1	ENSG00000122257	8.07	0.26	0.49092	1	9729	0
ENSG00000090861	603.8	339.7	542.4	601.8	ENSG00000090861	9.02	0.28	0.491	1	9730	0
ENSG00000109686	451.1	792.6	768.8	736.4	ENSG00000109686	9.42	0.28	0.49109	1	9731	0
ENSG00000032389	53.1	36.9						0.49115	1	9732	0
ENSG00000125844	2299.7	3137.5						0.49127	1	9733	0
ENSG00000180190	52.0	28.0						0.4913	1	9734	0
ENSG00000100351	2.3	12.7						0.49134	1	9735	0
ENSG00000160554	72.5	122.6						0.49139	1	9736	0

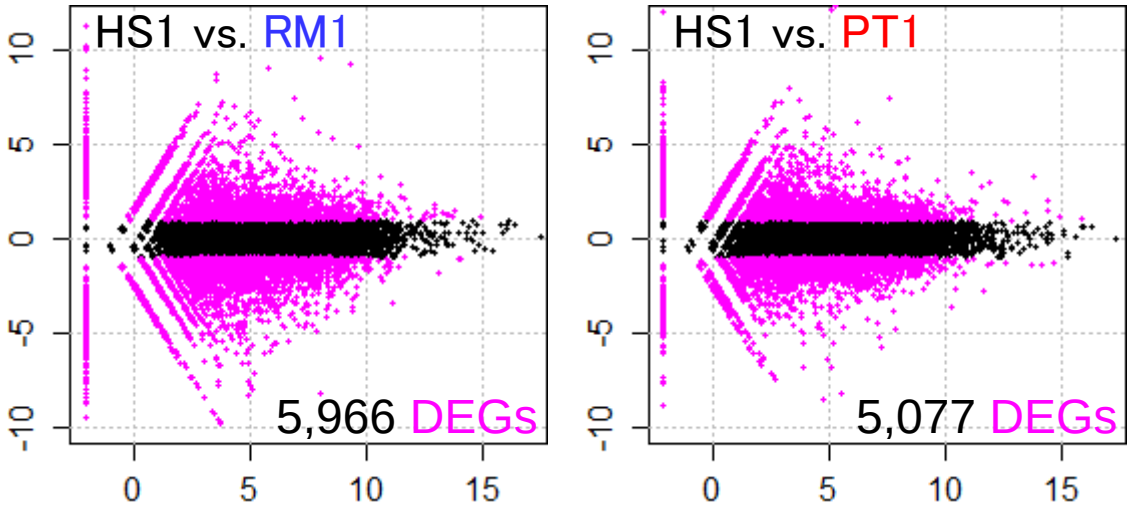


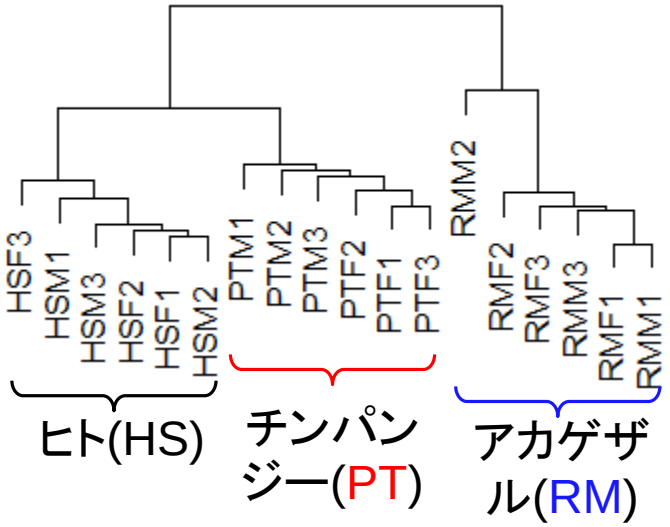
Contents

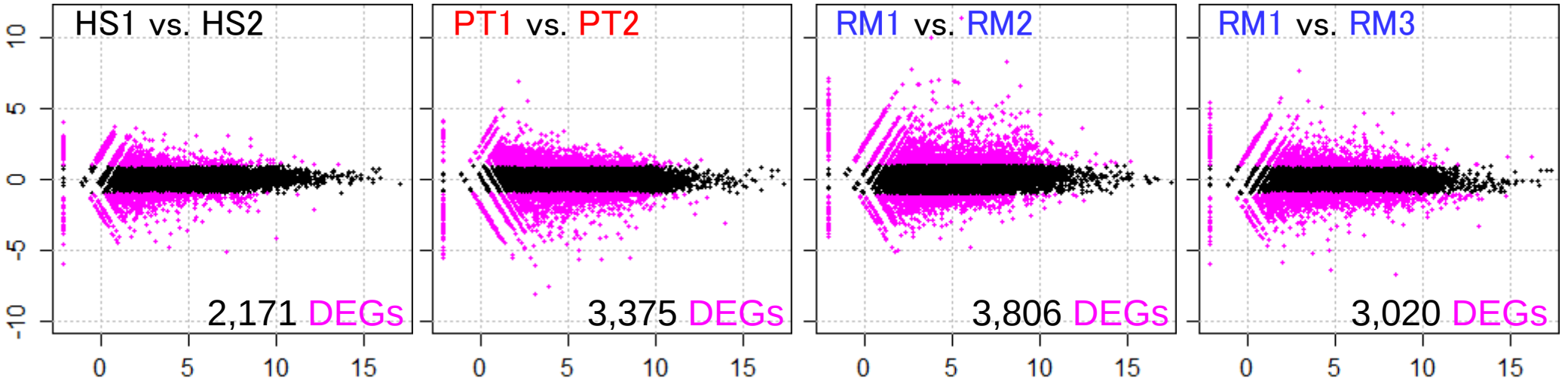
- 先週の課題について
  - 課題4とFastQC (ver. 0.11.3)のオプション
- マッピングからカウント情報取得
  - アノテーション情報なし
  - アノテーション情報あり（GFF形式ファイル読み込み）
- データ正規化
  - RPKMの基本的な考え方
  - 配列長とカウント数の関係：RPK, RPKM
- データ解析
  - サンプル間クラスタリング
  - 発現変動解析(2群間比較) : 発現変動遺伝子(DEG)が多数存在する場合
  - モデル、分布、統計的手法、2群間比較でDEGがそれほど多くない場合
  - 2群間比較でDEGがほとんどない同一群の場合
  - 倍率変化（2倍以上、1/2以下の発現変動）を用いた場合
  - 発現変動解析：3群間比較など

結果の比較(2倍変化)

倍率変化(fold-change; FC)でのDEG検出結果。同一群内比較でも多数の偽陽性が検出されている。例題13をベースに作成

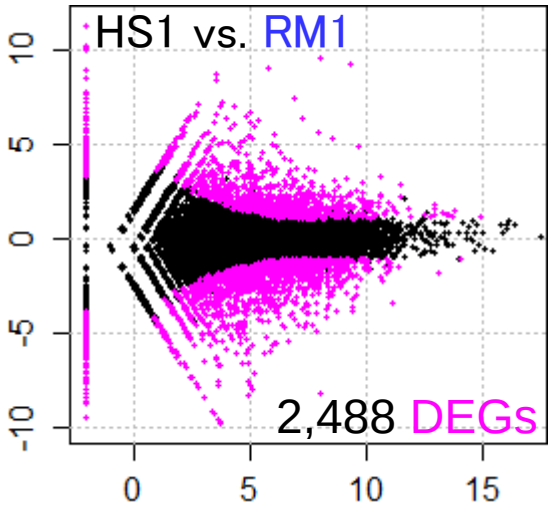


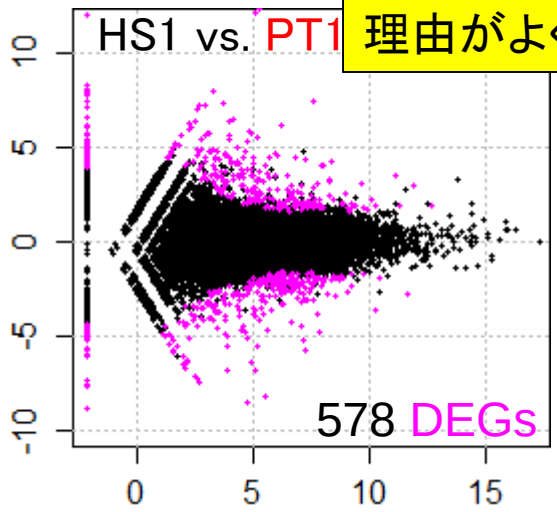


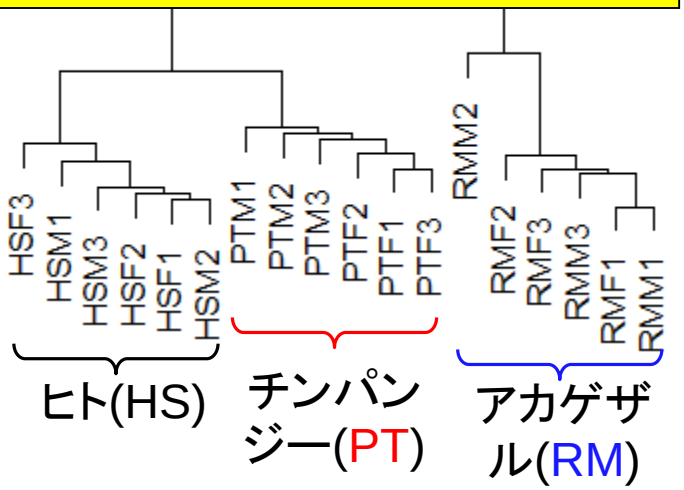


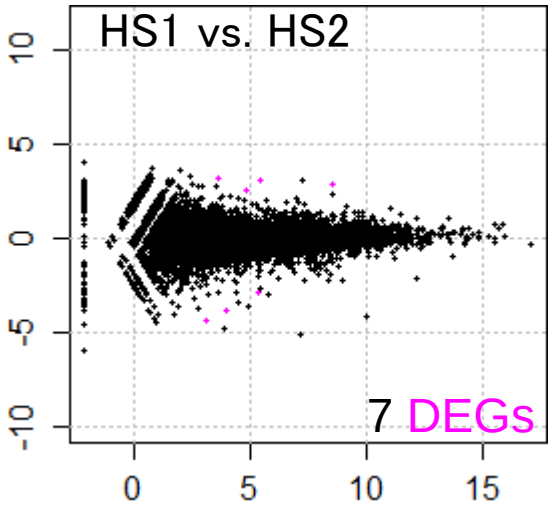
結果の比較(FDR)

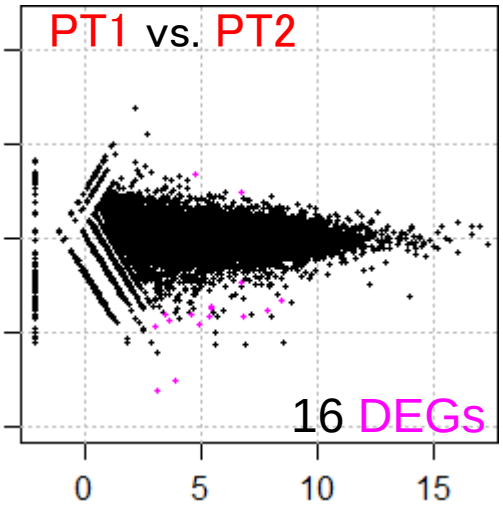
統計的手法(TCC)も多少偽陽性が存在するが、倍率変化(FC)ほど凶悪ではないことがわかる。また高発現側のDEGは、FCと比較的よく一致していることがわかる。先人がFCのみで比較的信頼性の高い結果を得てきた理由がよくわかる(高発現側を信頼するという経験則)。

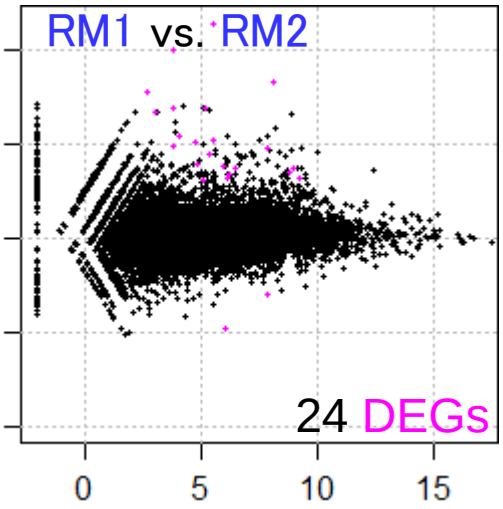


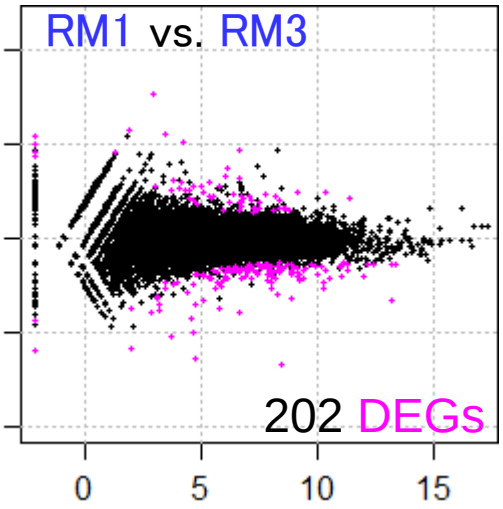












3群間比較

①発現パターンごとの分類もしたい場合に便利。②post-hoc test的なことをやりたいときの項目。③複製なしデータの場合。

- 解析 | 発現変動 | 3群間 | 対応なし | について (last modified 2015/02/10)
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 基礎 | DESeq2(Love_2014) (last modified 2015/02/04)
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 基礎 | TCC(Sun_2013) (last modified 2015/03/04)推奨
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 基礎 | EBSeq(Leng_2013) (last modified 2015/02/10)
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 基礎 | SAMseq(Li_2013) (last modified 2015/02/10)
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 基礎 | edgeR(Robinson_2010) (last modified 2015/02/03)
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 基礎 | DESeq(Anders_2010) (last modified 2014/03/13)
- 解析 | 発現変動 | 3群間 | 対応なし | 複製あり | 応用 | TCC(Sun_2013) (last modified 2015/01/29)推奨
- 解析 | 発現変動 | 3群間 | 対応なし | 複製なし | TCC(Sun_2013) (last modified 2015/07/07)推奨 NEW
- 解析 | 発現変動 | 5群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2014/08/22)推奨
- 解析 | 発現変動 | 時系列 | について (last modified 2014/12/19)
- 解析 | 発現変動 | 時系列 | Bayesian model-based clustering(Nascimento_2012) (last modified 2012/09/10)
- 解析 | 発現変動 | 時系列 | maSigPro(Nueda_2014) (last modified 2014/07/18)

シミュレーションデータ

他にも多くの解析用パッケージが存在し、このウェブページ上で紹介しきれていないものが多く存在します。また、バージョンアップなどに追い付いていない項目も多くあります。そのため、正しい手順で解析できているのかが不安な局面があるでしょう。そういうときはTCCパッケージ中のシミュレーションデータ作成関数を利用して、「これがDEG検出結果の上位に来ていないやり方はオカシイはず」というようなデータを自分で作成して検証するのです。

- 解析 | クラスタリング | 遺伝子間 | MBCluster.Seq (Si 2014) (last modified 201
- 解析 | シミュレーションカウントデータ | について (last modified 2015/01/25)
- 解析 | シミュレーションカウントデータ | Technical rep.(ポアソン分布) (last modified 2015/01/23)
- 解析 | シミュレーションカウントデータ | Biological rep. | 基礎編 (last modified 2015/01/23)
- 解析 | シミュレーションカウントデータ | Biological rep. | 2群間 | 基礎編 | TCC(Sun 2013) (last modified 2015/01/28)
- 解析 | シミュレーションカウントデータ | Biological rep. | 2群間 | 応用編 | TCC(Sun 2013) (last modified 2015/01/28)
- 解析 | シミュレーションカウントデータ | Biological rep. | 3群間 | 基礎編 | TCC(Sun 2013) modified 2015/01/28)
- 解析 | 発現変動 | について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2015/02/02)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013) (last modified 2015/02/26)推奨 NEW