

第1部：統計解析 ～トランスクリプトーム解析1～



東京大学・大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究プログラム

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/>

トランスクリプトーム解析1

①の内容は、②をほぼ全て含みます。
③の内容(リアルカウントデータのところ:スライド46-92あたり)も一部含みます。①の1/3程度は、新規内容予定

～ 平成27年度 ～


7/22(水):PC環境の構築

7/23(木):Linux基礎

7/24(金):シェルスクリプト

7/27(月):Perl

7/28(火):Python

7/29(水):データ解析環境R 

7/30(木):データ解析環境R

8/3(月):NGS解析(基礎)

8/4(火):NGS解析(ゲノムReseq、変異解析)

8/5(水):NGS解析(RNA-seq:代表的なパイプライン)

8/5(水):NGS解析(RNA-seq:統計解析) 

8/6(木):NGS解析(ChIP-seq)

8/26(水):予備日

8/27(木):予備日

8/28(金):予備日

～ 平成28年度 ～

7/19(火):PC環境の構築

7/20(水):統計解析(塩基配列解析系)

7/21(木):統計解析(発現解析系) 

7/22(金):統計解析(発現解析系)

7/25(月):NGS解析基礎

7/26(火):ゲノムReseq、変異解析

7/27(水):RNA-seq

7/28(木):ChIP-seq

8/1(月):Linux環境でのデータ解析1

8/2(火):Linux環境でのデータ解析2

8/3(水):ウェブツール、ロングリード

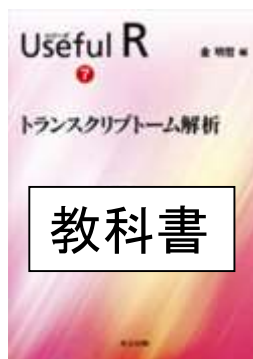
8/4(木):トランスクリプトーム解析系

Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)

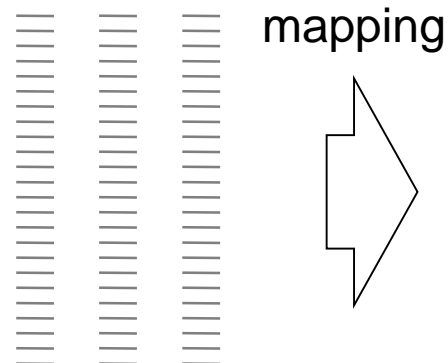


カウントデータ

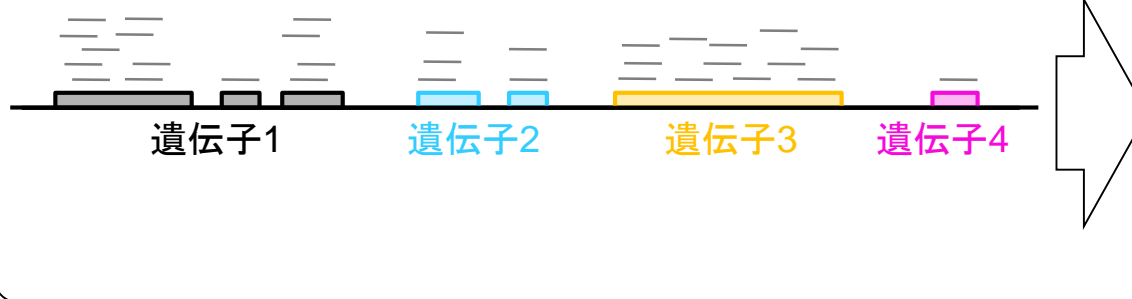


カウントデータとは、「マップされたリード数」をカウントしたデータのこと。以下の例では1サンプルなので1列分のデータしかないが、一般には複数サンプルのデータを取得し、サンプル間比較が行われるので複数の列からなる。それゆえ、数値ベクトルではなく**数値行列**

目的サンプルの
RNA-Seqデータ



リファレンス配列:ゲノム



	T1
遺伝子1	14
遺伝子2	5
遺伝子3	12
遺伝子4	1
遺伝子5	...
...	...

数値行列

①サンプルデータの、②例題41をコピペで実行し、③20,689 genes × 36 samplesのカウントデータファイル(sample_blekhman_36.txt)を得ておきましょう。やらないと後で困ります!



- (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 ① (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(次世代シーケンサ) | NGSハンズオン
- バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ) | 漢字
- 書籍
- 書籍
- 書籍

サンプルデータ **NEW**

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" #入力ファイル
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$R1L6, hoge$R6L4.PTM3,
  hoge$R1L7, hoge$R4L7.RMF3,
  hoge$R1L3, hoge$R4L3.RMM3)
```

Blekhman et al., *Genome Res.*, 20: 180-189, 2010

EXCELで概観

出力ファイル(sample_blekhman_36.txt)をEXCELで眺めるとこんな感じ。①で考えると、②はENSG00000000971という遺伝子領域上に2,262リードマップされたことを表す。③はENSG00000001460の遺伝子領域上に3リードマップされたことを表す。もしこの2つの配列長が同じなら、マップされたリード数が多い前者②の発現量が高いという理解でよい。スライドを見るだけ

	A	C	D	E	F	G	H	I	J	
1		R1L4.HSF1	R4L2.HSF1	R2L7.HSF2	R3L2.HSF2	R8L1.HSF3	R8L2.HSF3	R1L1.HSM1	R5L2.HSM1	R2L3
2	ENSG00000000003	172	157	147	153	78	90	60	61	2
3	ENSG00000000005	0	0	0	0	0	0	0	0	
4	ENSG00000000419	36	45	26	35	16	40	17	22	
5	ENSG00000000457	41	50	28	34	34	42	50	64	
6	ENSG00000000460	3	3	8	9	7	5	9	6	
7	ENSG00000000938	23	21	30	35	112	98	32	41	
8	ENSG00000000971	2262	2503	3473	3752	1665	1740	1726	1874	32
9	ENSG00000001036	155	142	118	133	79	110	99	101	
10	ENSG00000001084	323	307	377	360	151	155	155	181	4
11	ENSG00000001167	19	17	15	15	16	20	13	16	
12	ENSG00000001460	3	0	0	1	1	4	0	1	
13	ENSG00000001461	25	24	22	15	14	20	13	15	
14	ENSG00000001497	59	58	46	47	46	43	39	41	
15	ENSG00000001561	22	26	23	27	28	25	29	33	
16	ENSG00000001617	30	34	24	27	77	73	40	30	
17	ENSG00000001626	9	3	12	32	37	33	24	19	

データの正規化

①のサンプル内で、②は③より $2262/3 = 754$ 倍高発現と評価してはいけない。発現量の大小関係を比較したい場合は、長さで補正する必要がある。このあたりは④教科書のp132-137で述べている

sample_blekhman_36

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 アドイン 門田幸二

A1 :

	A	C	D	E	F	G	H	I	J
1	R1 L4.HSF1	R4L2.HSF1	R2L7.HSF2	R3L2.HSF2	R8L1.HSF3	R8L2.HSF3	R1 L1.HSM1	R5L2.HSM1	R2L3
2	ENSG000000000003	172	157	147	153	78	90	60	61
3	ENSG000000000005	0	0	0	0	0	0	0	0
4	ENSG000000000419	36	45	26	35	16	40	17	22
5	ENSG000000000457	41	50	28	34	34	42	50	64
6	ENSG000000000460	3	3	8	9	7	5	9	6
7	ENSG000000000938	23	21	30	35	112	98	32	41
8	ENSG000000000971	2262	2503	3473	3752	1665	1740	1726	1874
9	ENSG000000001036	155	142	119	122	79	110	99	101
10	ENSG000000001084	323	307	3					
11	ENSG000000001167	19	17						
12	ENSG000000001460	3	0						
13	ENSG000000001461	25	24						
14	ENSG000000001497	59	58						
15	ENSG000000001561	22	26						
16	ENSG000000001617	30	34						
17	ENSG000000001626	9	3						

sample_blekhman_36 (+)

準備完了

- 書籍 | トランスクリプトーム解析 | について (last modified 2014/05/12)
- 書籍 | トランスクリプトーム解析 | 2.3.1 RNA-seqデータ(FASTQファイル) (last modified 2016/02/09)
- 書籍 | トランスクリプトーム解析 | 2.3.2 リファレンス配列 (last modified 2014/04/16)
- 書籍 | トランスクリプトーム解析 | 2.3.3 アンテーション情報 (last modified 2014/04/17)
- 書籍 | トランスクリプトーム解析 | 2.3.4 マッピング(準備) (last modified 2014/06/20)
- 書籍 | トランスクリプトーム解析 | 2.3.5 マッピング(本番) (last modified 2014/06/21)
- 書籍 | トランスクリプトーム解析 | 2.3.6 カウントデータ取得 (last modified 2016/02/09)
- 書籍 | トランスクリプトーム解析 | 3.3.1 解析目的別留意点 (last modified 2014/04/20)
- 書籍 | トランスクリプトーム解析 | 3.3.2 データの正規化(基礎編) (last modified 2014/06/23)
- 書籍 | トランスクリプトーム解析 | 3.3.3 クラスターリング (last modified 2014/04/20)
- 書籍 | トランスクリプトーム解析 | 3.3.4 各種プロット (last modified 2014/04/27)
- 書籍 | トランスクリプトーム解析 | 4.3.1 シミュレーションデータ(負の二項分布) (last modified 2014/04/27)
- 書籍 | トランスクリプトーム解析 | 4.3.2 データの正規化(応用編) (last modified 2014/04/27)
- 書籍 | トランスクリプトーム解析 | 4.3.3 2群間比較 (last modified 2014/04/28)
- 書籍 | トランスクリプトーム解析 | 4.3.4 他の実験デザイン(3群間) (last modified 2014/04/28)

データの正規化

例えば、②と③の配列長がそれぞれ3000塩基、500塩基だったと仮定すると、②は③に対して $3,000/500 = 6$ 倍長いので、その分を補正してやる必要がある。④様々な表現方法があるが、発現量の比率(②/③)で考えると125.6667倍というのは不変

sample_blekh

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示

A1 :

	A	B	C	D	E	F	G	H	I	J
1		R1L4.HSF1	R4L2.HSF1	R2L7.HSF2	R3L2.HSF2	R8L1.HSF3	R8L2.HSF3	R1L1.HSM1	R5L2.HSM1	R2L3
2	ENSG000000000003	172	157	147	153	78	90	60	61	2
3	ENSG000000000005	0	0	0	0	0	0	0	0	
4	ENSG000000000419	36	45	26	35	16	40	17	22	
5	ENSG000000000457	41	50	28	34	34	42	50	64	
6	ENSG000000000460	3	3	8	9	7	5	9	6	
7	ENSG000000000938	23	21	30	35	112	98	32	41	
8	ENSG000000000971	2262	2503	3473	3752	1665	1740	1726	1874	32
9	ENSG000000001036	155	142	118	133	79	110	99	101	
10	ENSG000000001084	323	307	377	360	151	155	155	181	4
11	ENSG000000001167	19	17	15	15	16	20	13	16	
12	ENSG000000001460	3	0	0	1	1				
13	ENSG000000001461	25	24	22	15	14				
14	ENSG000000001497	59	58	46	47	46				
15	ENSG000000001561	22	26	23	27	28				
16	ENSG000000001617	30	34	24	27	77				
17	ENSG000000001626	9	3	12	32	37				

sample_blekhman_36

準備完了

R Console

```
> 2262/3
[1] 754
> (2262/6)/3
[1] 125.6667
> (2262/3000)/(3/500)
[1] 125.6667
> |
```


RPK補正のイントロ

④は「マップされたリード数(生のカウント数) × 1 / 配列長」に相当する。得られる数値は、塩基あたりのリード数(Reads per one base)ともいえる。これが長さ補正の基本形であるが、得られる数値(0.754や0.006)が小さすぎるのが難点

sample_blekhman_36

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 アド

A1 :

	A	B	C	D	E	F	G	H	I	J
1		R1L4.HSF1	R4L2.HSF1	R2L7.HSF2	R3L2.HSF2	R8L1.HSF3	R8L2.HSF3	R1L1.HSM1	R5L2.HSM1	R2L3
2	ENSG000000000003	172	157	147	153	78	90	60	61	2
3	ENSG000000000005	0	0	0	0	0	0	0	0	
4	ENSG000000000419	36	45	26	35	16	40	17	22	
5	ENSG000000000457	41	50	28	34	34	42	50	64	
6	ENSG000000000460	3	3	8	9	7	5	9	6	
7	ENSG000000000938	23	21	30	35	112				
8	ENSG000000000971	2262	2503	3473	3752	1665				
9	ENSG000000001036	155	142	118	133	79				
10	ENSG000000001084	323	307	377	360	151				
11	ENSG000000001167	19	17	15	15	16				
12	ENSG000000001460	3	0	0	1	1				
13	ENSG000000001461	25	24	22	15	14				
14	ENSG000000001497	59	58	46	47	46				
15	ENSG000000001561	22	26	23	27	28				
16	ENSG000000001617	30	34	24	27	77				
17	ENSG000000001626	9	3	12	32	37				

sample_blekhman_36

準備完了

R Console

```

> (2262/6)/3
[1] 125.6667
> (2262/3000)/(3/500)
[1] 125.6667
> 2262/3000
[1] 0.754
> 3/500
[1] 0.006
> 2262*(1000/3000)
[1] 754
> 3*(1000/500)
[1] 6
> |

```

RPK補正

④は「マップされたリード数(生のカウント数) × 1000 / 配列長」に相当する。得られる数値は、1000塩基あたりのリード数(Reads per one kilobase; RPK)ともいえる。配列長の異なる遺伝子間の発現レベルの大小関係を平等に比較すべく、「遺伝子が1000 bpだったときのリード数」とするのがRPKの考え方。RPK補正後の値は②が754、③が6となる

sample_blekhman_36

	A	C	D
1	R1 L4.HSF1	R4L2.HSF1	R2L7.HSF2
2	ENSG000000000003	172	157
3	ENSG000000000005	0	0
4	ENSG000000000419	36	45
5	ENSG000000000457	41	50
6	ENSG000000000460	3	3
7	ENSG000000000938	23	21
8	ENSG000000000971	2262	2503
9	ENSG000000001036	155	142
10	ENSG000000001084	323	307
11	ENSG000000001167	19	17
12	ENSG000000001460	3	0
13	ENSG000000001461	25	24
14	ENSG000000001497	59	58
15	ENSG000000001561	22	26
16	ENSG000000001617	30	34
17	ENSG000000001626	9	3

sample_blekhman_36

準備完了

R Console

```

> (2262/6) / 3
[1] 125.6667
> (2262/3000) / (3/500)
[1] 125.6667
> 2262/3000
[1] 0.754
> 3/500
[1] 0.006
> 2262 * (1000/3000)
[1] 754
> 3 * (1000/500)
[1] 6
>

```

RPM補正のイントロ

スライドを見るだけ。サンプル(列)ごとにマップされた総リード数を計算した結果。サンプル間比較の場合には、この総リード数を揃えるのが基本戦略。総リード数を100万(one million)に揃えるのが、RPM (Reads per million)補正

sample_blekhman_36.

ファイル ホーム 挿入 ページレイアウト 数式 テータ 校閲 表示 アドイン

B20692 : fx =SUM(B2:B20690)

	A	B	C	D	E	F	G	H	I
20677	ENSG00000221765	0	0	0	0	0	0	0	0
20678	ENSG00000221766	0	0	0	0	0	0	0	0
20679	ENSG00000221767	0	0	0	0	0	0	0	0
20680	ENSG00000221768	0	0	0	0	0	0	0	0
20681	ENSG00000221770	4	2	4	0	2	2	0	0
20682	ENSG00000221771	0	0	0	0	0	0	0	0
20683	ENSG00000221775	0	0	0	0	0	0	0	0
20684	ENSG00000221778	0	0	0	0	0	0	0	0
20685	ENSG00000221781	0	0	0	0	0	0	0	0
20686	ENSG00000221782	0	0	0	0	0	0	0	0
20687	ENSG00000221783	0	0	0	0	0	1	0	0
20688	ENSG00000221784	0	0	0	0	0	0	0	0
20689	ENSG00000221786	0	0	0	0	0	0	0	0
20690	ENSG00000221788	0	0	0	0	0	0	0	0
20691									
20692		1665987	1719125	1620189	1801009	1393867	1450604	1346515	1497738
20693									

sample_blekhman_36

準備完了

100%

RPM補正のイントロ

もし揃えずに、例えば①と②のサンプル間比較(発現変動遺伝子(DEG)検出)を行うと、①のほうが②に比べて全体的に $(1,801,009 / 1,346,515 =)$ 1.34倍高発現な状態であることを意味するので、①で高発現となるDEGが多く検出されるだろう。もちろんそれは間違い

sample_b

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表

B20692 : \times \checkmark f_x =SUM(B2:B20690)

	A	B	C	D	E	F	G	H	I
20677	ENSG00000221765	0	0	0	0	0	0	0	0
20678	ENSG00000221766	0	0	0	0	0	0	0	0
20679	ENSG00000221767	0	0	0	0	0	0	0	0
20680	ENSG00000221768	0	0	0	0	0	0	0	0
20681	ENSG00000221770	4	2	4	0	2	2	0	0
20682	ENSG00000221771	0	0	0	0	0	0	0	0
20683	ENSG00000221775	0	0	0	0	0	0	0	0
20684	ENSG00000221778	0	0	0	0	0	0	0	0
20685	ENSG00000221781	0	0	0	0	0	0	0	0
20686	ENSG00000221782	0	0	0	0	0	0	0	0
20687	ENSG00000221783	0	0	0	0	0	1	0	0
20688	ENSG00000221784	0	0	0	0	0	0	0	0
20689	ENSG00000221786	0	0	0	0	0	0	0	0
20690	ENSG00000221788	0	0	0	0	0	0	0	0
20691									
20692		1665987	1719125	1620189	1801009	1393867	1450604	1346515	1497738
20693									

sample_blekhman_36

準備完了

100%

RPM補正のイントロ

colSums関数で、列ごとの総リード数を一気に表示。
EXCELとR間で同じ値が得られていることがわかる
(①と②)。③RPM補正後のデータで同じ操作を実行
すると、全部100万になる(ここはまだ補正前の状態)

The screenshot shows an Excel spreadsheet with an R Console window overlaid. The spreadsheet has columns A and B. The R Console window shows the output of the `colSums` function. Red arrows point to specific values in both the spreadsheet and the R console output.

Excel Spreadsheet Data:

	A	B
20677	ENSG00000221765	0
20678	ENSG00000221766	0
20679	ENSG00000221767	0
20680	ENSG00000221768	0
20681	ENSG00000221770	4
20682	ENSG00000221771	0
20683	ENSG00000221775	0
20684	ENSG00000221778	0
20685	ENSG00000221781	0
20686	ENSG00000221782	0
20687	ENSG00000221783	0
20688	ENSG00000221784	0
20689	ENSG00000221786	0
20690	ENSG00000221788	0
20691		
20692		1665987
20693		

R Console Output:

```
> colSums (data)
R1L4.HSF1 R4L2.HSF1 R2L7.HSF2 R3L2.HSF2 R8L1.HSF3 R8L2.HSF3
1665987 1719125 1620189 1801009 1393867 1450604
R1L1.HSM1 R5L2.HSM1 R2L3.HSM2 R4L8.HSM2 R3L6.HSM3 R4L1.HSM3
1346515 1497738 2217235 2167994 1974228 1825373
R1L2.PTF1 R4L4.PTF1 R2L4.PTF2 R6L6.PTF2 R3L7.PTF3 R5L3.PTF3
2667264 2677771 1910402 1881431 1838275 1813918
R1L6.PTM1 R3L3.PTM1 R2L8.PTM2 R4L6.PTM2 R6L2.PTM3 R6L4.PTM3
1481536 1694688 1608138 1946512 1745188 1803555
R1L7.RMF1 R5L1.RMF1 R2L2.RMF2 R5L8.RMF2 R3L4.RMF3 R4L7.RMF3
2400660 2110806 2338433 1533906 2685655 2534595
R1L3.RMM1 R3L8.RMM1 R2L6.RMM2 R5L4.RMM2 R3L1.RMM3 R4L3.RMM3
2657274 2505941 1942296 1974502 2119496 2411707
> |
```

Red arrows in the image point to the following values:

- Arrow ①: Points to the value 1665987 in the R console output and the cell B20692 in the spreadsheet.
- Arrow ②: Points to the value 1497738 in the R console output and the cell B20680 in the spreadsheet.
- Arrow ③: Points to the value 4 in the R console output and the cell B20681 in the spreadsheet.

RPM補正

①入力は、20,689 genes × 36 samplesのカウントデータ。サンプル(列)ごとに総リード数は異なるので、②正規化係数nfは列ごとに異なる。③nfの中身。数値ベクトルnfの要素数は、列数と同じく36。全体のコピーはスライド17で行うが、②のnfオブジェクトの中身を見るために必要な部分までなど自由にコピー実行してよい

RPM正規化

「正規化」基礎 | [RPM or CPM \(総リード数補正\)](#) を samplesのカウントデータ ([sample_blekman_36.txt](#)) で

```
in_f <- "sample_blekman_36.txt"
out_f <- "hoge1.txt"
param1 <- 1000000
```

#入力ファイルの読み込み

```
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")
colSums(data)
```

#正規化

```
nf <- param1/colSums(data)
data <- sweep(data, 2, nf, "*")
colSums(data)
```

#ファイルに保存

```
tmp <- cbind(row.names(data), data)
write.table(tmp, out_f, sep="\t",
```

#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納
#補正後の総リード数を指定(RPMにしたい場合)

#正規化係数を計算した結果をnfに格納
#正規化係数を各列に掛けた結果をdata1に格納

R Console

```
> nf
R1L4.HSF1 R4L2.HSF1 R2L7.HSF2 R3L2.HSF2 R8L1.HSF3 R8L2.HSF3
0.6002448 0.5816913 0.6172119 0.5552443 0.7174286 0.6893680
R1L1.HSM1 R5L2.HSM1 R2L3.HSM2 R4L8.HSM2 R3L6.HSM3 R4L1.HSM3
0.7426579 0.6676735 0.4510122 0.4612559 0.5065271 0.5478332
R1L2.PTF1 R4L4.PTF1 R2L4.PTF2 R6L6.PTF2 R3L7.PTF3 R5L3.PTF3
0.3749160 0.3734449 0.5234500 0.5315103 0.5439882 0.5512928
R1L6.PTM1 R3L3.PTM1 R2L8.PTM2 R4L6.PTM2 R6L2.PTM3 R6L4.PTM3
0.6749752 0.5900791 0.6218372 0.5137394 0.5730042 0.5544605
R1L7.RMF1 R5L1.RMF1 R2L2.RMF2 R5L8.RMF2 R3L4.RMF3 R4L7.RMF3
0.4165521 0.4737527 0.4276368 0.6519304 0.3723486 0.3945404
R1L3.RMM1 R3L8.RMM1 R2L6.RMM2 R5L4.RMM2 R3L1.RMM3 R4L3.RMM3
0.3763255 0.3990517 0.5148546 0.5064568 0.4718103 0.4146441
> |
```

RPM補正

①nfベクトルの1番目の要素である、R1L4.HSF1サンプルの正規化係数(0.6002448)は、②1,000,000 / 1,665,987 = 0.6002448として計算している。ここで、③1,665,987はR1L4.HSF1サンプルの総リード数

RPM正規化

「正規化」基礎 | [RPM or CPM \(総リード数補正\)](#) をベースに作成。入力は、20,000 genes x 50 samplesのカウントデータ([sample blekman 36.txt](#))です。教科書p134。

```
in_f <- "sample_blekman_36.txt" #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt" #出力ファイル名を指定してout_fに格納
param1 <- 1000000 #補正後の総リード数を指定(RPMにしたい)
```

#入力ファイルの読み込み

```
data <- read.table(in_f, header=TRUE, as.is=TRUE)
colSums(data)
```

#本番(正規化)

```
nf <- param1/colSums(data)
data <- sweep(data, 2, nf, "*")
colSums(data)
```

#ファイルに保存

```
tmp <- cbind(rownames(data), data)
write.table(tmp, out_f, sep="\t", as.is=TRUE)
```

```
R Console
> nf
R1L4.HSF1 R4L2.HSF1 R2L7.HSF2 R3L2.HSF2 R8L1.HSF3 R8L2.HSF3
0.6002448 0.5816913 0.6172119 0.5552443 0.7174286 0.6893680
R1L1.HSM1 R5L2.HSM1 R2L3.HSM2 R4L8.HSM2 R3L6.HSM3 R4L1.HSM3
0.7426579 0.6676735 0.4510122 0.4612559 0.5065271 0.5478332
R1L2.PTF1 R4L4.PTF1 R2L4.PTF2 R6L6.PTF2 R3L7.PTF3 R5L3.PTF3
0.3749160 0.3734449 0.5234500 0.5315103 0.5439882 0.5512928
R1L6.PTM1 R3L3.PTM1 R2L8.PTM2 R4L6.PTM2 R6L2.PTM3 R6L4.PTM3
0.6749752 0.5900791 0.6218372 0.5137394 0.5730042 0.5544605
R1L7.RMF1 R5L1.RMF1 R2L2.RMF2 R5L8.RMF2 R3L4.RMF3 R4L7.RMF3
0.4165521 0.4737527 0.4276368 0.6519304 0.3723486 0.3945404
R1L3.RMM1 R3L8.RMM1 R2L6.RMM2 R5L4.RMM2 R3L1.RMM3 R4L3.RMM3
0.3763255 0.3990517 0.5148546 0.5064568 0.4718103 0.4146441
> 1000000/1665987
[1] 0.6002448
> param1/1665987
[1] 0.6002448
> |
```

RPM補正は、①入力ファイル情報に相当するdataの、②各列に対して、③正規化係数nfを、④掛けた結果を、再びdataオブジェクトに格納することで達成

RPM補正

• RPM正規化

「正規化|基礎|[RPM or CPM \(総リード数補正\)](#)」をベースに作成。入力は、20,689 genes×36 samplesのカウントデータ([sample blekhman 36.txt](#))です。教科書p134。

```

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 <- "hoge1.txt"                 #出力ファイル名を指定してout_fに格納
param1 <- 1000000                    #補正後の総リード数を指定(RPMにしたい場

#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_
colSums(data)                        #総リード数を表示

#本番(正規化)
nf <- param1/colSums(data)            #正規化係数を計算した結果をnfに格納
data <- sweep(data, 2, nf, "*")       #正規化係数を各列に掛けた結果をdata1に格納
colSums(data)                        #総リード数を表示

#ファイルに保存
tmp <- cbind(row.names(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

```


RPM補正

RPM補正後のdataオブジェクトに対して、colSums関数で各列の総リード数を表示。全部同じ100万(1e+06)になっていることがわかる

RPM正規化

「正規化」基礎 | [RPM or CPM \(総リード数補正\)](#) をベースに作成。入力は、20,689 genes×36 samplesのカウントデータ([sample blekhman 36.txt](#))です。教科書p134。

```
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 <- "hoge1.txt" #出力ファイル名を指定してout_fに格納
param1 <- 1000000 #補正後の総リード数を指定(RPMにしたい場
```

#入力ファイルの読み込み

```
data <- read.table(in_f, header=T, as.is=T)
colSums(data)
```

#本番(正規化)

```
nf <- param1/colSums(data)
data <- sweep(data, 2, nf, "*")
colSums(data)
```

#ファイルに保存

```
tmp <- cbind(rownames(data), data)
write.table(tmp, out_f, sep="\t",
```

```
R Console
> data <- sweep(data, 2, nf, "*") #正規化係数を各列$
> colSums(data) #総リード数を表示
R1L4.HSF1 R4L2.HSF1 R2L7.HSF2 R3L2.HSF2 R8L1.HSF3 R8L2.HSF3
1e+06 1e+06 1e+06 1e+06 1e+06 1e+06
R1L1.HSM1 R5L2.HSM1 R2L3.HSM2 R4L8.HSM2 R3L6.HSM3 R4L1.HSM3
1e+06 1e+06 1e+06 1e+06 1e+06 1e+06
R1L2.PTF1 R4L4.PTF1 R2L4.PTF2 R6L6.PTF2 R3L7.PTF3 R5L3.PTF3
1e+06 1e+06 1e+06 1e+06 1e+06 1e+06
R1L6.PTM1 R3L3.PTM1 R2L8.PTM2 R4L6.PTM2 R6L2.PTM3 R6L4.PTM3
1e+06 1e+06 1e+06 1e+06 1e+06 1e+06
R1L7.RMF1 R5L1.RMF1 R2L2.RMF2 R5L8.RMF2 R3L4.RMF3 R4L7.RMF3
1e+06 1e+06 1e+06 1e+06 1e+06 1e+06
R1L3.RMM1 R3L8.RMM1 R2L6.RMM2 R5L4.RMM2 R3L1.RMM3 R4L3.RMM3
1e+06 1e+06 1e+06 1e+06 1e+06 1e+06
>
> #ファイルに保存
> tmp <- cbind(rownames(data), data) #保存したい情報をt$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.$
> |
```

RPKM補正

赤枠部分の実行結果を表示。①RPM補正後のdata行列から、“ENSG00000000971”行、1列の情報を表示。②この列の正規化係数は0.6002448だった。③元のリードカウント値2262に0.6002448を掛けて確認してるだけ。④しつこくRPM値を確認してるだけ

RPKM正規化

行名が“ENSG00000000971”で、1列目のカウント値(2262)に対するRPM

#RPM補正部分

```
data["ENSG00000000971", 1]
nf[1]
2262 * nf[1]
2262 * (1000000/1665987)
```

```
#dataから、行名が"ENSG00000000971"
#RPM正規化係数nfの1番目の要素を表示
#"ENSG00000000971"の1列目のカウント
#"ENSG00000000971"の1列目のカウント
```

#RPK補正追加分

```
1000/3000 # "ENSG00000000971"の配列長が3000だ
data["ENSG00000000971", 1] * (1000/3000) #RPKM補正後の値を表示<
2262 * (1000000/1665987) * (1000/3000) #RPKM補正後の値を表示
```

```
R Console
> data["ENSG00000000971", 1] #d$
[1] 1357.754
> nf[1] #R$
R1L4.HSF1 #2
0.6002448
> 2262 * nf[1] #"$
R1L4.HSF1
1357.754
> 2262 * (1000000/1665987) #"$
[1] 1357.754
> |
```

RPKM補正

① “ENSG00000000971”の配列長が3000 bpだったときのRPK正規化係数は1000/3000。②オリジナルのリードカウント(2262)にRPM正規化係数とRPK正規化係数を掛けたものがRPKM値

RPKM正規化

行名が“ENSG00000000971”で、1列目のカウント値(2262)に対するRPKM値の計算例を示す。

#RPM補正部分

```
data["ENSG00000000971", 1]
```

```
nf[1]
```

```
2262 * nf[1]
```

```
2262 * (1000000/1665987)
```

#dataから、行名が“ENSG00000000971”

#RPM正規化係数nfの1番目の要素を表示

#“ENSG00000000971”の1列目のカウント

#“ENSG00000000971”の1列目のカウント

#RPK補正追加分

```
1000/3000
```

```
data["ENSG00000000971", 1] * (1000/3000) #RPKM補正後の値を表示<
```

```
2262 * (1000000/1665987) * (1000/3000) #RPKM補正後の値を表示
```

#“ENSG00000000971”の配列長が3000だ

```
R Console
> #RPK補正追加分
> 1000/3000                                #"$
[1] 0.3333333
> data["ENSG00000000971", 1] * (1000/3000) # $
[1] 452.5846
> 2262 * (1000000/1665987) * (1000/3000) #R$
[1] 452.5846
> |
```

①

②

Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



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ファイル。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。TCCパッケージを用いて、これらのサンプル間クラスタリングを行う。コピー

- 解析 | [発現量推定\(トランスクリプトーム配列を利用\)](#) (last modified 2014/02/05)
- 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\(Sun_2014\)](#) (last modified 2014/02/05)

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。
「ファイル名」→「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、以下をコピー。

1. 59 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="") #樹形図(デンドログラム)の表示
dev.off()
    
```

①出力は、hoge7.pngという名前のPNGファイル。②サイズは、700×400ピクセル。これは論文の図としても使えるレベル(実際我々の論文中でも使っている)

クラスタリング

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"
out_f <- "hoge7.png"
param_fig <- c(700, 400)
```

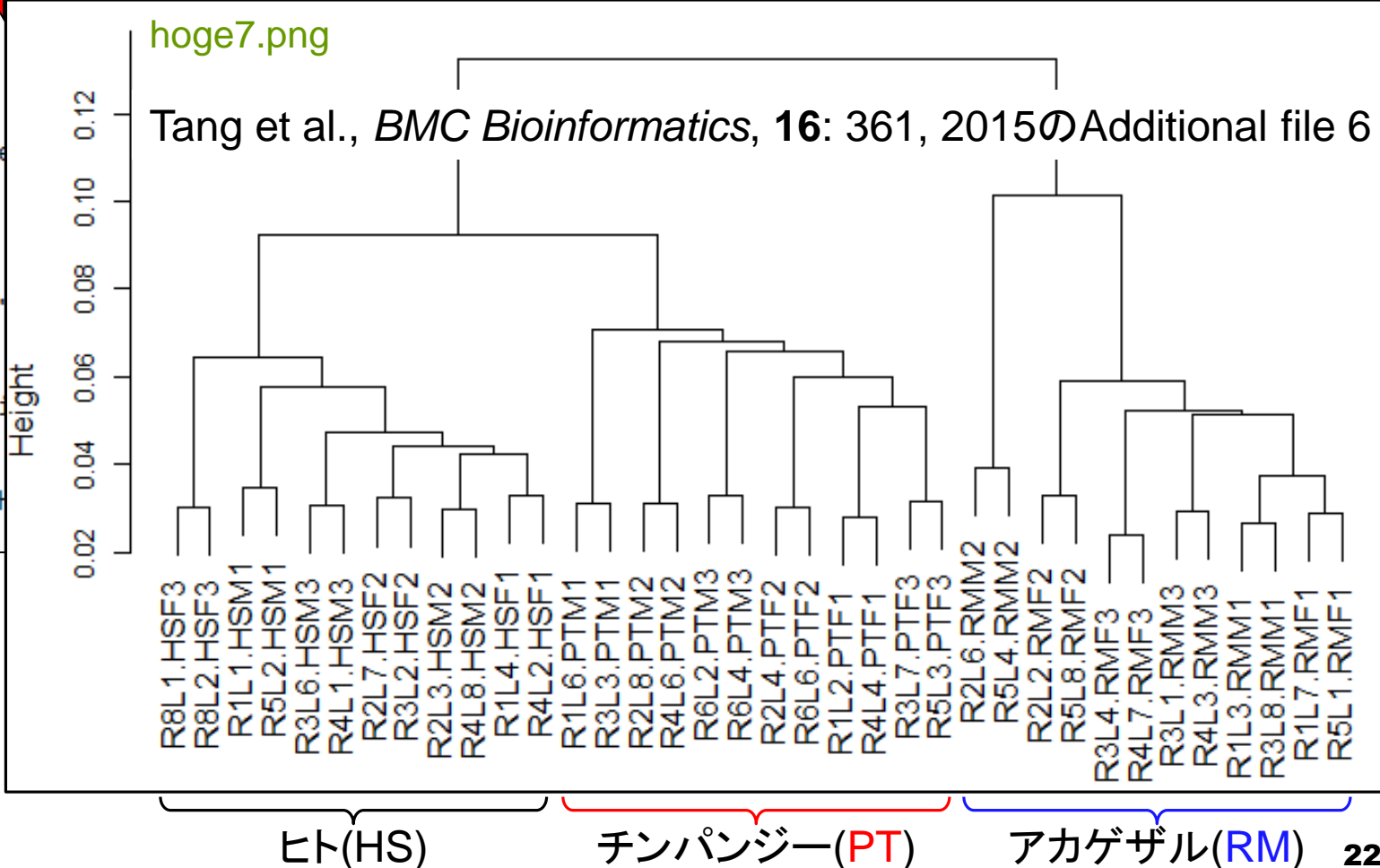
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)

```
#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, header=TRUE, as.is=TRUE)
dim(data)

#本番
out <- clusterSample(data, hclust.method="ward.D2")

#ファイルに保存
png(out_f, pointsize=13, width=700, height=400,
    par(mar=c(0, 4, 1, 0)))
plot(out, sub="", xlab="", ylab="Height",
    cex=1.3, main="", ylab="Height",
    dev.off())
```

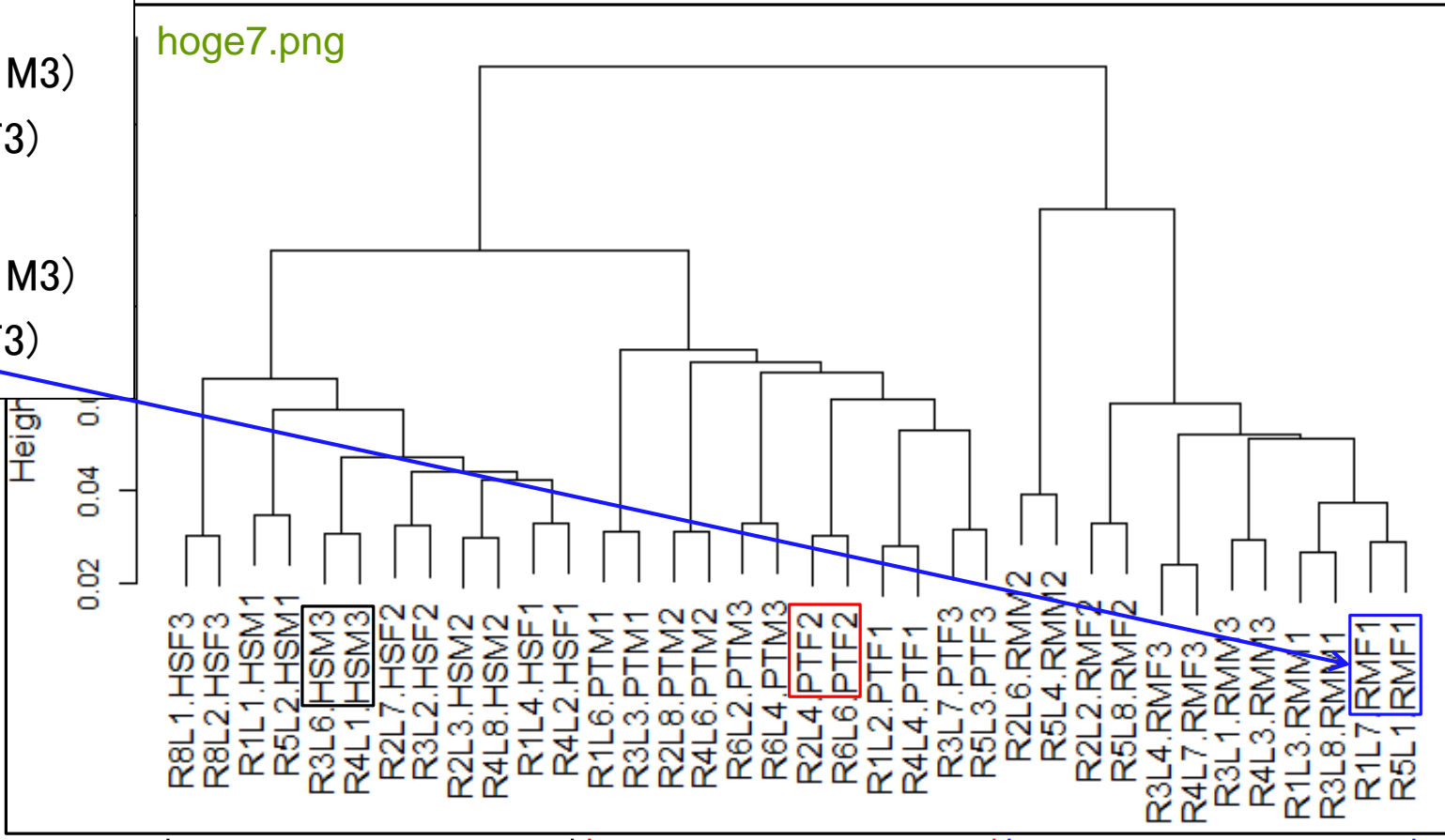


hoge7.png
Tang et al., *BMC Bioinformatics*, 16: 361, 2015のAdditional file 6

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。妥当ですよ

- 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)



ヒト(HS)

チンパンジー(PT)

アカゲザル(RM)

クラスタリング

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)」とは考えにくく、帰無仮説を棄却して「発現変動している(DEGである)」と判定することになる

ヒト(HS)

- オス3匹(M1, M2, M3)
- メス3匹(F1, F2, F3)

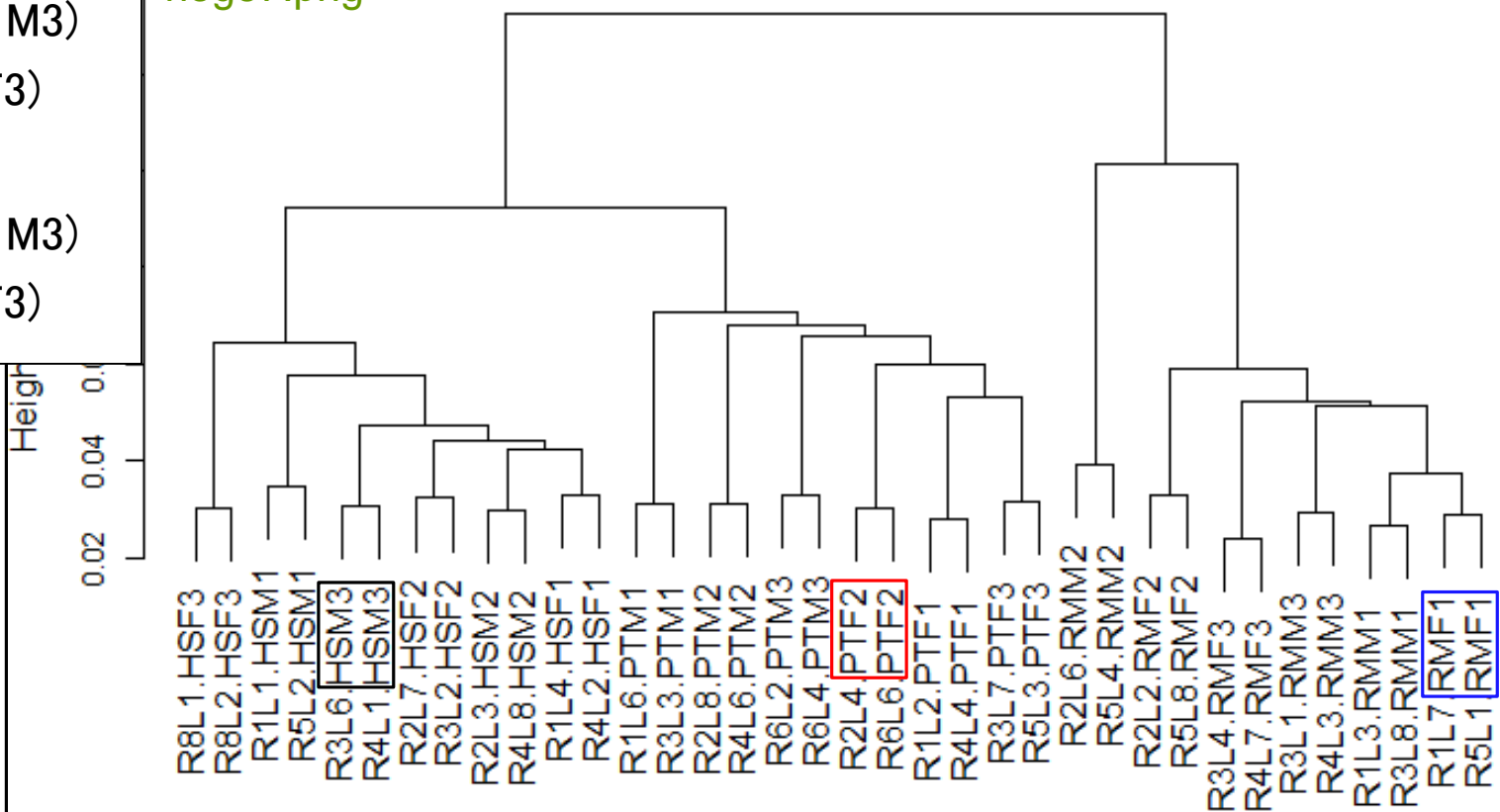
チンパンジー(PT)

- 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



ヒト(HS)

チンパンジー(PT)

アカゲザル(RM)

サブセット抽出と整形

①サンプルデータの、②例題42。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。③出力ファイルはsample_blekhman_18.txt。サンプル名部分は必要最小限の情報のみになっている

- ・ (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 ① (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(次世代シーケンサ) | NGSハンズオン講習会
- ・ バイオインフォマティクス人材育成カリキュラム(次世代シーケンサ) | 講習会
- ・ 書籍
- ・ 書籍
- ・ 書籍

サンプルデータ **NEW**

1. ② 42. [Blekhman et al., Genome Res., 2010](#)のリアルカウントデータです。
1つ前の例題41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"#入力ファイル名を指定してin_fに格納
in_f <- "suppTable1.xls" #出力ファイル名を指定してout_fに格納
out_f <- "sample_blekhman_18.txt"

#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
```

出力ファイル

出力ファイルは、20,689遺伝子×18サンプルの biological replicatesのみからなる、3生物種間比較用カウントデータ。ヒト(*Homo sapiens*; HS)、チンパンジー(*Pan troglodytes*; PT)、アカゲザル(*Rhesus macaque*; RM)。生物種ごとにメス3匹、オス3匹。雄雌を考慮しなければ biological replicates (生物学的な反復)は6

20,689 genes

	ヒト (<i>Homo sapiens</i> ; HS)						チンパンジー (<i>Pan troglodytes</i> ; PT)						アカゲザル (<i>Rhesus macaque</i> ; RM)					
	メス(Female)			オス(Male)			メス			オス			メス			オス		
	HSF1	HSF2	HSF3	HSM1	HSM2	HSM3	PTF1	PTF2	PTF3	PTM1	PTM2	PTM3	RMF1	RMF2	RMF3	RMM1	RMM2	RMM3
ENSG000000000003	329	300	168	121	421	359	574	429	386	409	685	428	511	464	480	424	1348	705
ENSG000000000005	0	0	0	0	1	0	1	4	1	0	1	1	0	1	2	2	0	0
ENSG000000000419	81	61	56	39	78	62	100	66	65	59	58	93	67	72	57	49	82	90
ENSG000000000457	91	62	76	114	73	95	131	229	87	274	239	149	89	69	118	117	114	163
ENSG000000000460	6	17	12	15	7	17	8	8	5	12	7	10	4	4	10	7	3	4
ENSG000000000938	44	65	210	73	43	65	84	104	76	198	31	58	73	28	54	80	34	72
ENSG000000000971	4765	7225	3405	3600	6383	5546	5382	8331	4335	2568	5019	2653	13566	9964	18247	14236	5196	11834
ENSG000000001036	297	251	189	200	234	249	305	301	313	254	151	331	292	106	379	201	88	140
ENSG000000001084	630	737	306	336	984	459	417	328	885	298	569	218	1062	786	1110	873	664	1752
ENSG000000001167	36	30	36	29	33	28	63	80	25	69	74	41	62	34	108	97	35	61
ENSG000000001460	3	1	5	1	4	2	0	1	1	1	1	3	1	1	1	0	1	3
ENSG000000001461	49	37	34	28	62	32	75	69	40	90	69	60	210	92	176	247	81	117
ENSG000000001487	117	93	88	80	131	110	125	98	75	108	130	131	138	95	187	137	158	172

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... (last modified 2014/02/05)

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」→「デスクトップの変更」で解析したいファイルを置いてあるデスクトップに移動し、以下をコピペ

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

1. 59

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

Neyret-
ンゲ

in_f
out_f
param

#必要
libra

#入力
data
dim(d

#本番
out <

```

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で指定したファイル
dim(data)                              #オブジェクトdataの行数と列数を表示

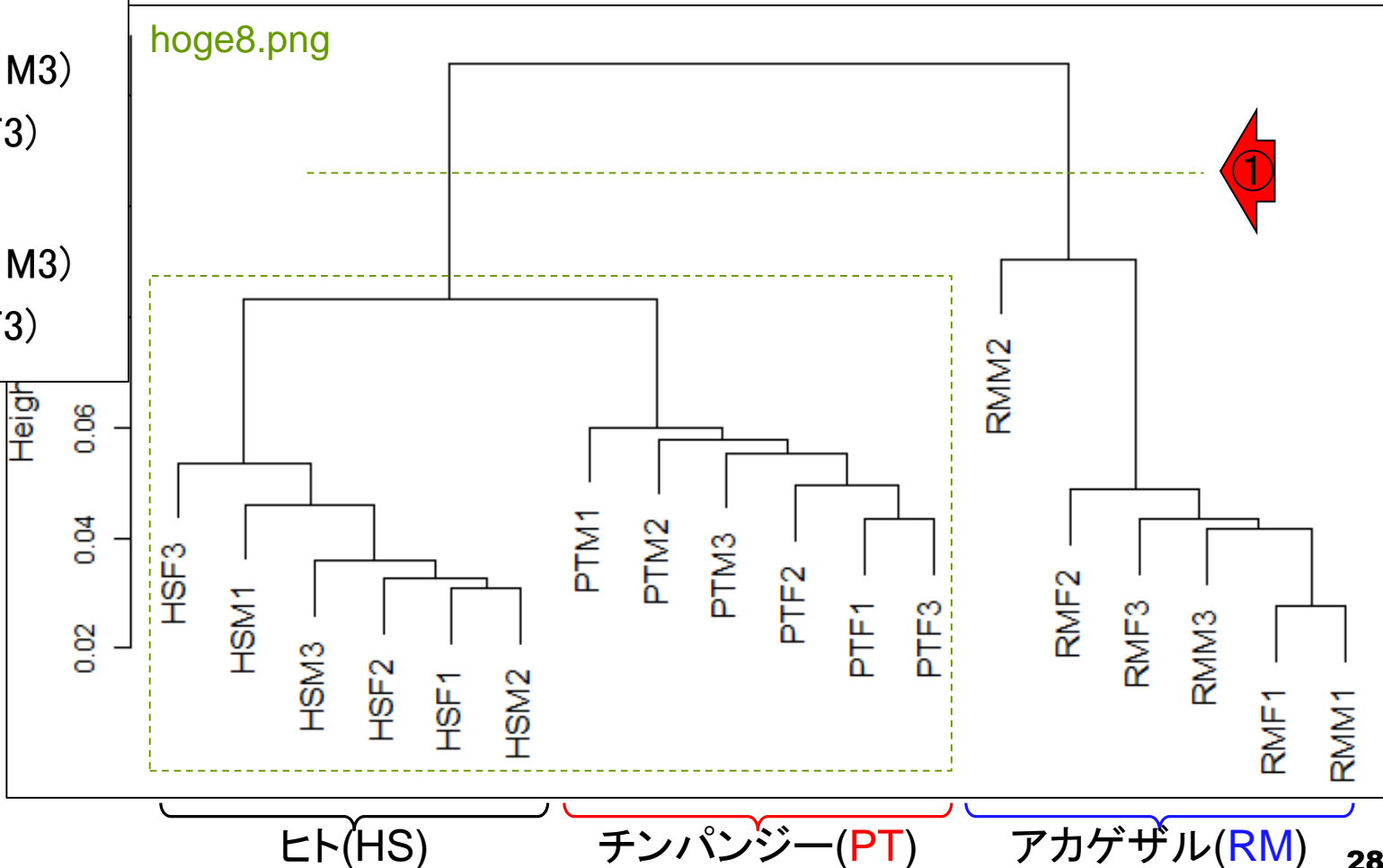
#本番
out <- clusterSample(data, dist.method="spearman",#クラスタリング実行結果をoutに格納
                    hclust.method="average", unique.pattern=TRUE)#クラスタリング実行結果をoutに格納

#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="", yaxp=1, 2, 1)#樹形図(デンドログラム)の表示
    
```

結果の解釈

3生物種間全体で眺める。①の部分で2つのグループに分けると…、ヒト(HS)とチンパンジー(PT)はよく似ている。2群間比較(発現変動遺伝子検出; DEG検出)を行ったときに、「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

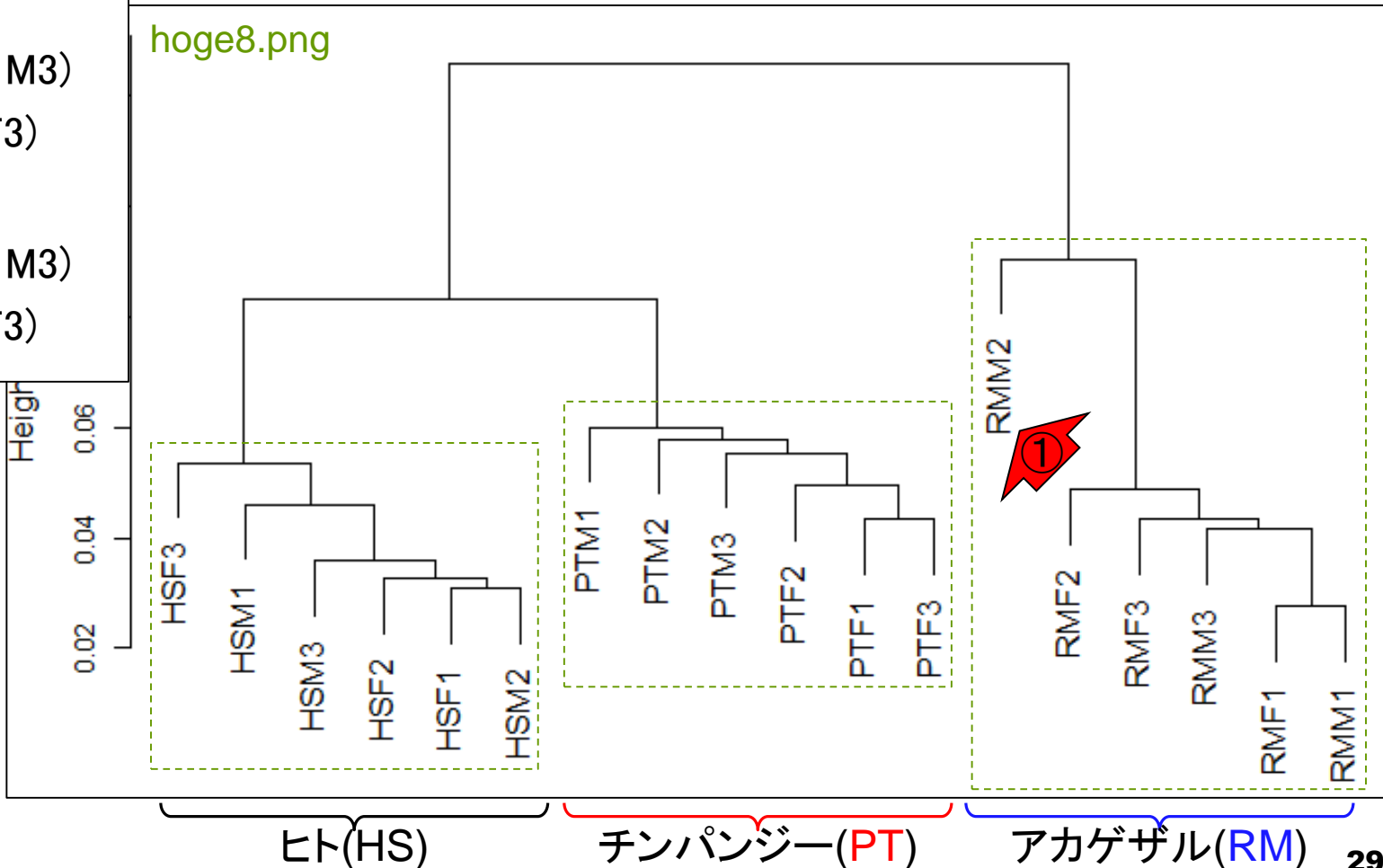
- ヒト(HS)
 - オス3匹(M1, M2, M3)
 - メス3匹(F1, F2, F3)
- チンパンジー(PT)
 - オス3匹(M1, M2, M3)
 - メス3匹(F1, F2, F3)
- アカゲザル(RM)
 - オス3匹(M1, M2, M3)
 - メス3匹(F1, F2, F3)



同一生物種でクラスターを形成している。①RMM2は「外れサンプル」っぽい

結果の解釈

- 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)



結果の解釈

①ヒト(HS)と②アカゲザル(RM)は、メスとオスのサンプルが入り混じっている。これらの生物種内で、「メス群 vs. オス群」の2群間比較を行ってもDEGはほとんど検出されないだろう

■ ヒト(HS)

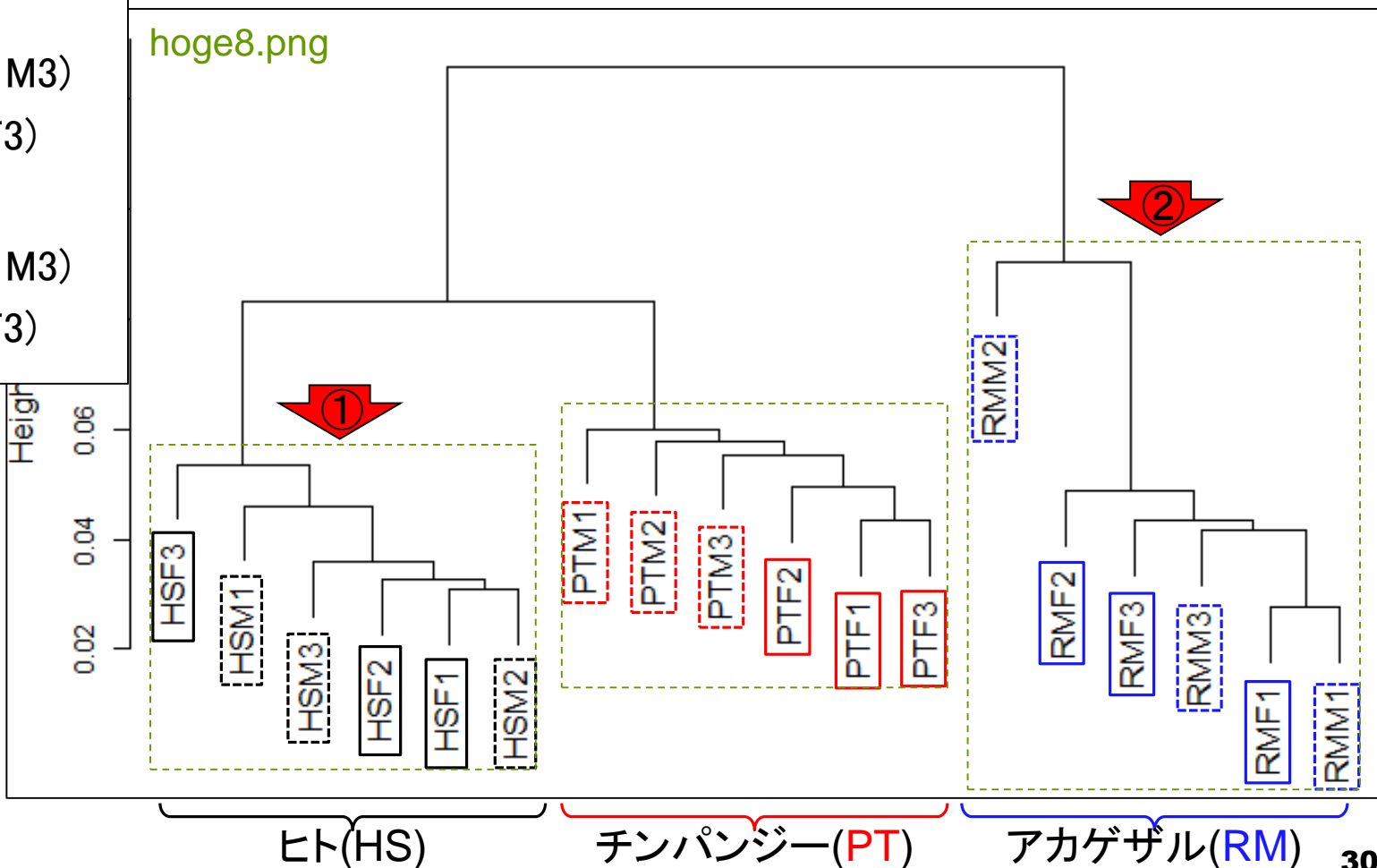
- オス3匹(M1, M2, M3)
- メス3匹(F1, F2, F3)

■ 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)

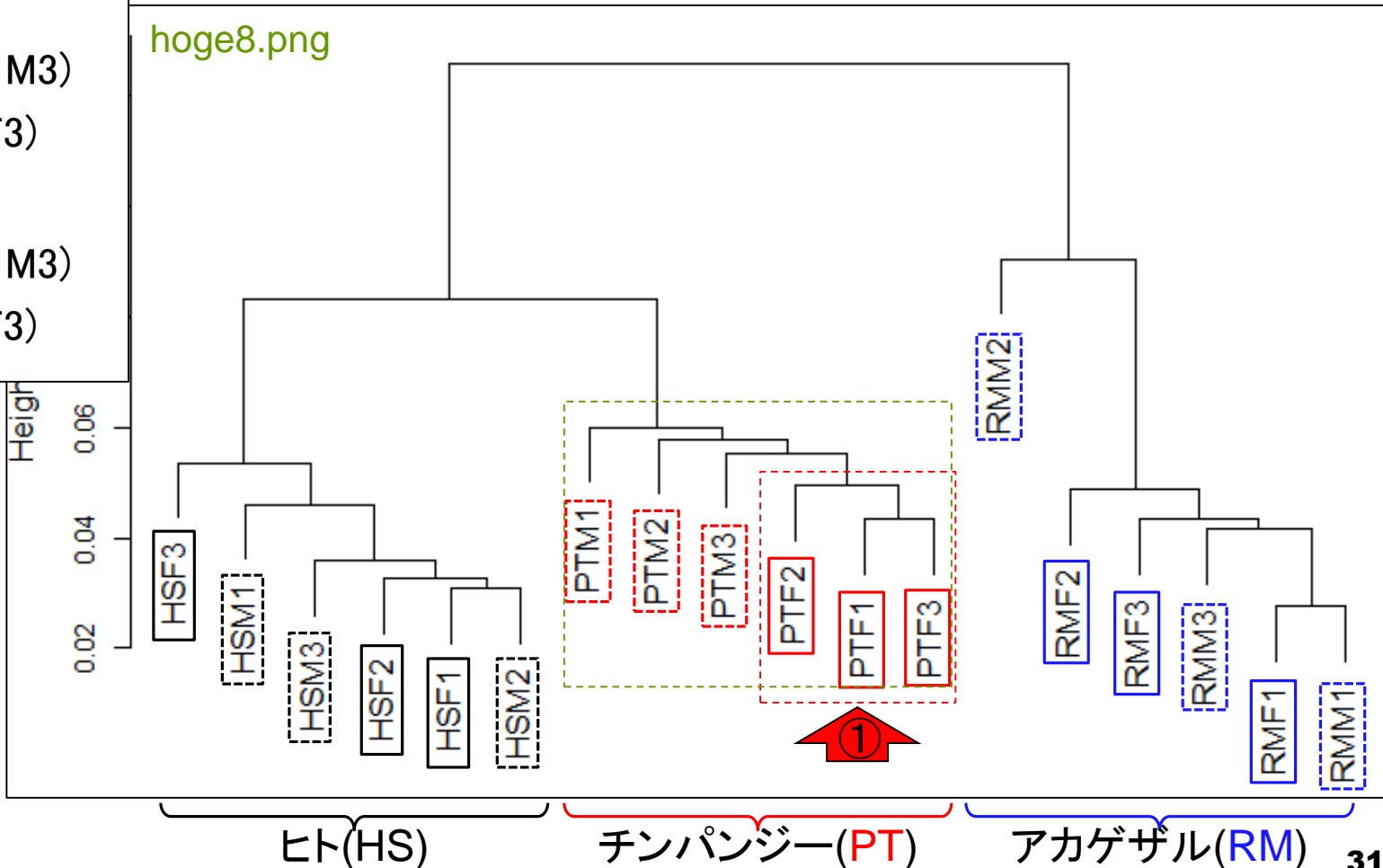
- オス3匹(M1, M2, M3)
- メス3匹(F1, F2, F3)



チンパンジー(PT)に限って言えば、①メス3匹がクラスターを形成しているので、「メス群 vs. オス群」の2群間比較結果として、多少なりともDEGが検出されるだろう

結果の解釈

- 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)



クラスタリングとDEG数の関係

BMC Bioinformatics. 2015 Nov 4;16:361. doi: 10.1186/s12859-015-0794-7.

Evaluation of methods for differential expression analysis on multi-group RNA-seq count data.

Tang M¹, Sun J², Shimizu K³, Kadota K⁴.

Author information

Abstract

BACKGROUND: RNA-seq is a powerful tool for measuring transcriptomes, especially for identifying differentially expressed genes or transcripts (DEGs) between sample groups. A number of methods have been developed for this task, and several evaluation studies have also been reported. However, those evaluations so far have been restricted to two-group comparisons. Accumulations of comparative studies for multi-group data are also desired.

METHODS: We compare 12 pipelines available in nine R packages for detecting differential expressions (DE) from multi-group RNA-seq count data, focusing on three-group data with or without replicates. We evaluate those pipelines on the basis of both simulation data and real count data.

RESULTS: As a result, the pipelines in the TCC package performed comparably to or better than other pipelines under various simulation scenarios. TCC implements a multi-step normalization strategy (called DEGES) that internally uses functions provided by other representative packages (edgeR, DESeq2, and so on). We found considerably different numbers of identified DEGs (18.5 ~ 45.7% of all genes) among the pipelines for the same real dataset but similar distributions of the classified expression patterns. We also found that DE results can roughly be estimated by the hierarchical dendrogram of sample clustering for the raw count data.

CONCLUSION: We confirmed the DEGES-based pipelines implemented in TCC performed well in a three-group comparison as well as a two-group comparison. We recommend using the DEGES-based pipeline that internally uses edgeR (here called the EEE-E pipeline) for count data with replicates (especially for small sample sizes). For data without replicates, the DEGES-based pipeline with DESeq2 (called SSS-S) can be recommended.



Tips: cex.lab

plot関数実行時に、cex.labオプションの数値を、例えば①2.0にすることで、デフォルトの2.0倍の大きさにできる。大きくする対象は、②で指定する軸ラベル情報の文字列(この場合はy軸のHeight)。③確かに大きくなっているが、左端が切れているので…

• Tips: cex.lab
cex.labは、この場合縦軸ラベル情報("Height")の文字の大きさをデフォルトの倍指定するオプション。例えばデフォルトの2.0倍にしたいときは、cex.lab=2.0とする

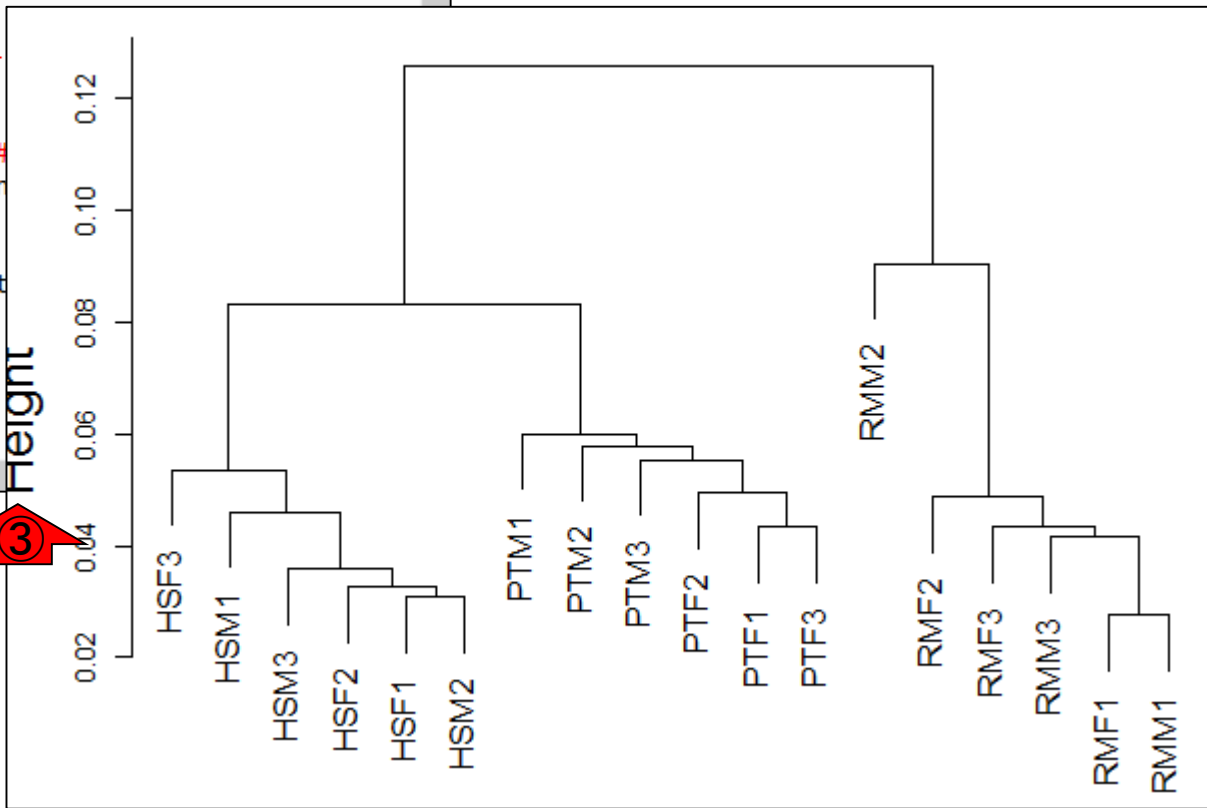
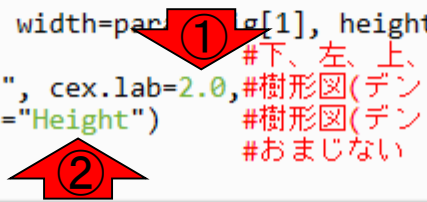
```
in_f <- "sample_blekman_18.txt" #入力ファイル名を指定してin_fに格納
out_f <- "hoge.png" #出力ファイル名を指定してout_fに格納
param_fig <- c(600, 400) #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#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, #オブジェクト
dim(data))

#本番
out <- clusterSample(data, dist.method="spearman", #
hclust.method="average", unique.pattern=1)

#ファイルに保存
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2], #下、左、上、
par(mar=c(0, 4, 1, 0)) #樹形図(デフォルト)
plot(out, sub="", xlab="", cex.lab=2.0, #樹形図(デフォルト)
cex=1.3, main="", ylab="Height") #おまじない
dev.off()
```



①左側の余白を4行分から5行分にする
ことで、Heightの文字が切れずに表示される

Tips: mar

• Tips: mar

marの左部分の余白を4から5行分に変更することで、"Height"の文字が切れなくなる。marはmargin(マージン)の意味です。

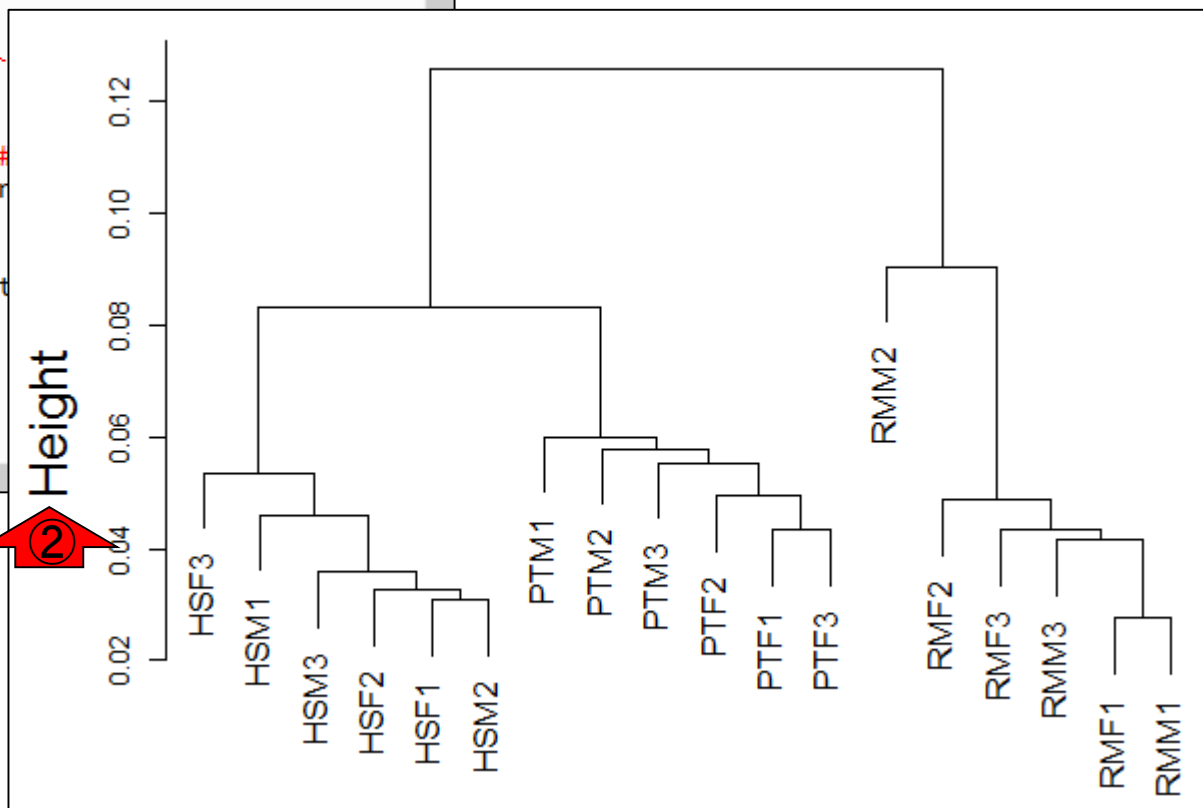
```
in_f <- "sample_blekhman_18.txt" #入力ファイル名を指定してin_fに格納  
out_f <- "hoge.png" #出力ファイル名を指定してout_fに格納  
param_fig <- c(600, 400) #ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

```
#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, #オブジェクト  
dim(data))
```

```
#本番  
out <- clusterSample(data, dist.method="spearman", #  
hclust.method="average", unique.patterns=TRUE)
```

```
#ファイルに保存  
png(out_f, width=param_fig[1], height=param_fig[2], #下、左、上、  
par(mar=c(0, 5, 1, 0)) #樹形図(デン  
plot(out, sub="", xlab="", cex.lab=2.0, #樹形図(デン  
cex=1.3, main="", ylab="Height") #おまじない  
dev.off())
```



Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



ReCount

17個のカウントデータを提供しているサイト。自分で1からマッピングなどを行わずに済むので便利。Technical replicatesのデータセットについては、biological replicatesにマージしたのも提供してくれている。①ページ下部に移動

The screenshot shows the ReCount website interface. At the top, there is a navigation bar with the ReCount logo and the Johns Hopkins Bloomberg School of Public Health logo. Below the navigation bar, the main content area contains a paragraph describing ReCount as an online resource for RNA-seq gene count datasets. To the right of the main content, there is a vertical sidebar with a scroll bar. A red arrow with the number '1' points to the bottom of the scroll bar, indicating the location of the navigation menu. The navigation menu includes links for Site Map, Home, News and Updates, Getting Started with ExpressionSets, Related Tools, and Related Publications. The Related Tools section lists Myrna: Cloud, differential gene expression. The Related Publications section lists a paper by Frazee et al. in BMC Bioinformatics.

ReCount
A multi-experiment resource of analysis-ready RNA-seq gene count datasets

JOHNS HOPKINS BLOOMBERG SCHOOL of PUBLIC HEALTH

ReCount is an online resource consisting of RNA-seq gene count datasets built using the raw data from 18 different studies. The raw sequencing data (.fastq files) were processed with [Myrna](#) to obtain tables of counts for each gene. For ease of statistical analysis, we combined each count table with sample phenotype data to form an R object of class [ExpressionSet](#). The count tables, ExpressionSets, and phenotype tables are ready to use and freely available here. By taking care of several preprocessing steps and combining many datasets into one easily-accessible website, we make finding and analyzing RNA-seq data considerably more straightforward.

All columns of the table below are sortable: clicking on the column title will alphabetize or order the column (keeping the rows properly aligned). The columns are as follows:

Study

With a few exceptions, the datasets are named for the first author of the paper from which the .fastq files were obtained. The Katz paper contained both mouse and human reads, so two separate datasets were created. The "maq" dataset was built from reads obtained from the [MicroArray Quality Control Project](#). The "modencodeworm" and "modencodefly" datasets were generated using reads from papers associated with the [modENCODE Consortium](#).

Site Map

- [Home](#)
- [News and Updates](#)
- [Getting Started with ExpressionSets](#)

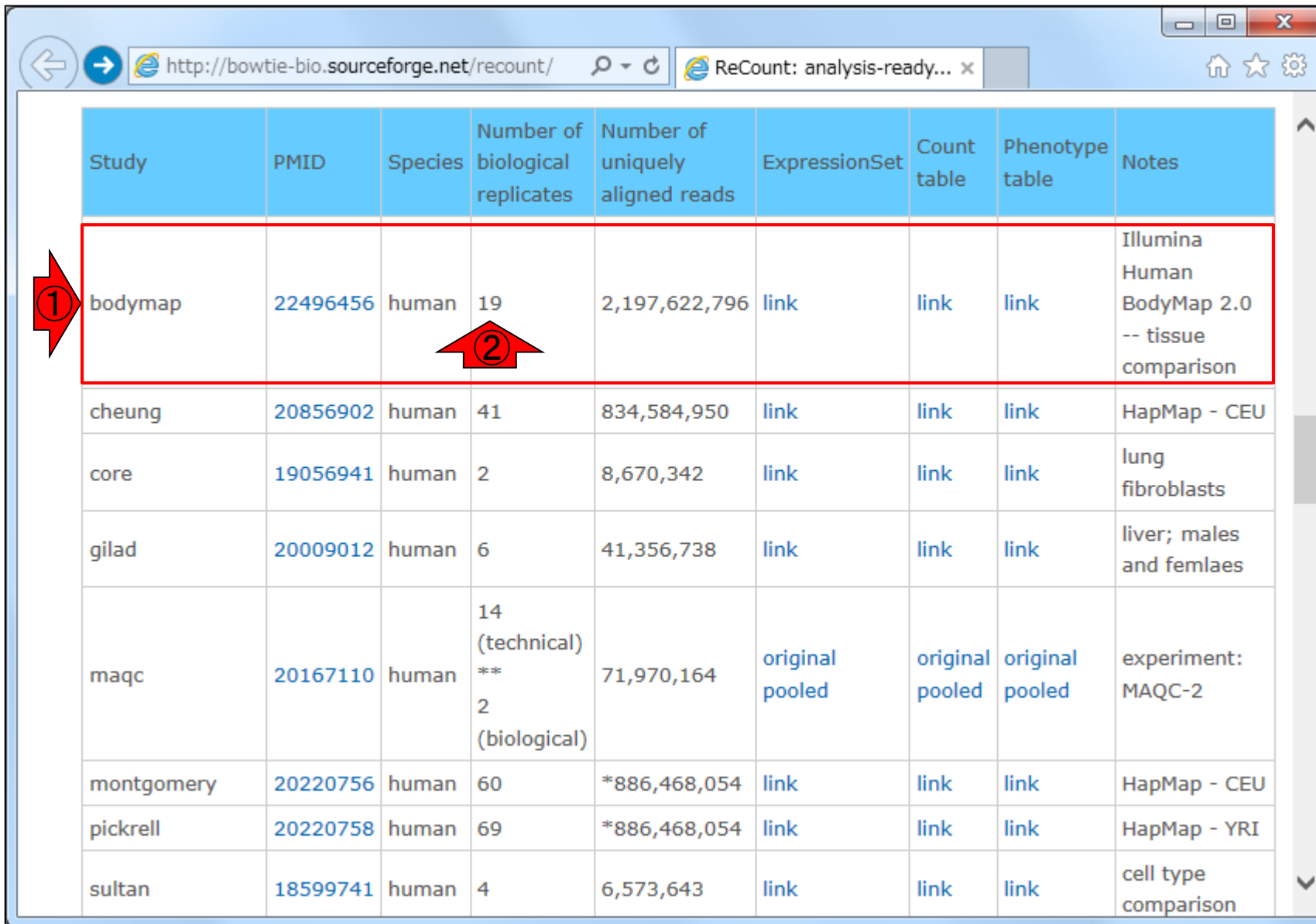
Related Tools

- [Myrna: Cloud, differential gene expression](#)

Related Publications

- Frazee AC, Langmead B, Leek JT. **ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets.** *BMC Bioinformatics* 12:449

ReCount



http://bowtie-bio.sourceforge.net/recount/ ReCount: analysis-ready...

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and femlaes
maq	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

ReCount

http://bowtie-bio.sourceforge.net/recount/ ReCount: analysis-ready...

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and femlaes
maq	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled			
montgomery	20220756	human	60	*886,468,054	link			
pickrell	20220758	human	69	*886,468,054	link			
sultan	18599741	human	4	6,573,643	link			

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="bodymap")
[1] "bodymap_count_table.txt"
[2] "bodymap_eset.RData"
[3] "bodymap_phenodata.txt"
> |
```

①3.3.3 クラスタリング、②p143-144の網掛け部分。手順の詳細な説明は教科書を参照のこと

bodymapデータ解析

- 書籍 | トランスクリプトーム解析 | について (last modified 2014/05/12)
- 書籍 | トランスクリプトーム解析 | 2.3.1 RNA-seqデータ(FASTQファイル) (last modified 2016)
- 書籍 | トランスクリプトーム解析 | 2.3.2 リファレンス配列 (last modified 2014/04/16)
- 書籍 | トランスクリプトーム解析 | 2.3.3 アンテーション情報 (last modified 2014/04/17)
- 書籍 | トランスクリプトーム解析 | 2.3.4 マッピング(準備) (last modified 2014/06/20)
- 書籍 | トランスクリプトーム解析 | 2.3.5 マッピング(本番) (last modified 2014/06/21)
- 書籍 | トランスクリプトーム解析 | 2.3.6 カウントデータ取得 (last modified 2016/02/09)
- 書籍 | トランスクリプトーム解析 | 3.3.1 解析目的別留意点 (last modified 2014/04/20)
- 書籍 | トランスクリプトーム解析 | 3.3.2 データの正規化(基礎編) (last modified 2014/06/23)
- 書籍 | トランスクリプトーム解析 | 3.3.3 クラスタリング (last modified 2014/04/20)
- 書籍 | トランスクリプトーム解析 | 3.3.4 各種プロット (last modified 2014/04/27)
- 書籍 | トランスクリプトーム解析 | 4.3.1 シミュレーションデータ(負の二項分布) (last modified

書籍 | トランスクリプトーム解析 | 3.3.3 クラスタリング

シリーズ Useful R 第7巻 トランスクリプトーム解析のp137-145のRコードです。ここではデスクトップ上に

「recount」というフォルダを作成し、そこで作業を行うという前提です。

p138:

書籍中では作業ディレクトリをデータベース(Frazer et al., BMC (bodymap count table.txt))であればどこでも構いません

```
getwd()
list.files()
```

② p143-144の網掛け部分:

```
in_f1 <- "bodymap_count_table.txt"
in_f2 <- "bodymap_phenodata.txt"
### ファイルの読み込み ###
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep=" ", quote="")
### dataオブジェクトの列名を変更 ###
colnames(data) <- phenotype$tissue.type
### フィルタリング(総カウント数が0でなく、ユニークな発現パターンをもつもののみ) ###
obj <- rowSums(data) != 0
hoge <- unique(data[obj,])
### クラスタリング ###
data.dist <- as.dist(1 - cor(hoge, method = "spearman"))
out <- hclust(data.dist, method = "average")
plot(out)
```

bodymapデータ解析

コピー実行結果。①全19サンプルの中にある3つのmixtureサンプルでクラスターを形成しており妥当。②肝臓(liver)はこのあたり

p143-144の網掛け部分:

```

in_f1 <- "bodymap_count_table.txt"
in_f2 <- "bodymap_phenodata.txt"
### ファイルの読み込み ###
data <- read.table(in_f1, header=1)
phenotype <- read.table(in_f2, header=1)
### dataオブジェクトの列名を変更 ###
colnames(data) <- phenotype$tissue
### フィルタリング(総カウント数が0でない) ###
obj <- rowSums(data) != 0
hoge <- unique(data[obj,])
### クラスタリング ###
data.dist <- as.dist(1 - cor(hoge))
out <- hclust(data.dist, method = "average")
plot(out)

```


TCCで実行すると...

(特に黒枠部分で示すように)TCCパッケージ中のclusterSample関数を利用するとこんな感じになります。手順は同じなので、結果も同じになる

・ TCCで実行すると...

「書籍 | トランスクリプトーム解析 | [3.3.3 クラスタリング](#)」の「p143-144の網掛け部分」を、「解析 | クラスタリング | [TCC\(Sun 2013\)](#)」のように書くと以下のような感じになります。

```
in_f1 <- "bodymap_count_table.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 <- "bodymap_phenodata.txt" #入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
out_f <- "hoge.png" #出力ファイル名を指定してout_fに格納
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)
```

#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

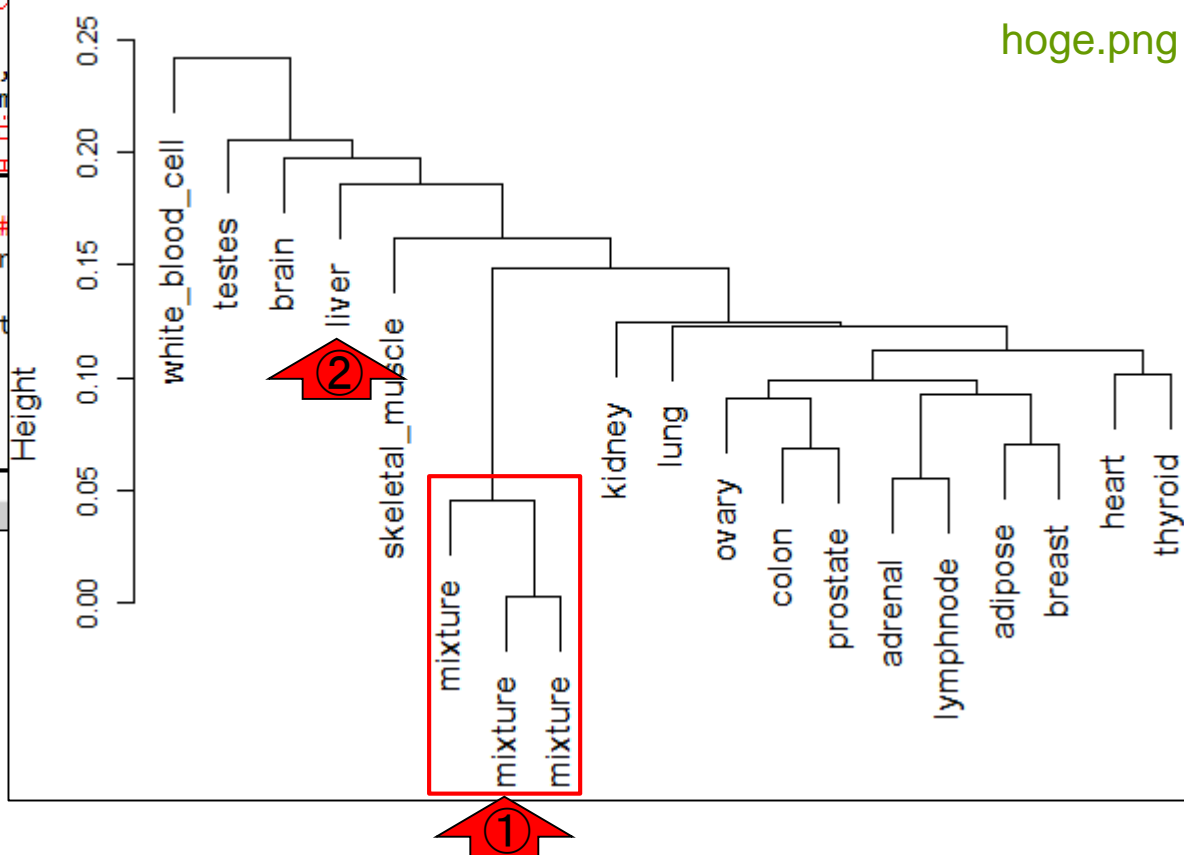
```
data <- read.table(in_f1, header=TRUE, row.names=1, #確認してるだ
phenotype <- read.table(in_f2, header=TRUE, row.names=1, #確認してるだ
colnames(data) <- phenotype$tissue.type #dataオブジェクト
```

#本番

```
out <- clusterSample(data, dist.method="spearman", #
hclust.method="average", unique.pattern=TRUE)
```

#ファイルに保存

```
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2], #下、左、上、
par(mar=c(0, 4, 1, 0)) #樹形図(デン
plot(out, sub="", xlab="", cex.lab=1.2, #樹形図(デン
cex=1.3, main="", ylab="Height") #おまじない
dev.off()
```



ReCountのヒトデータは

①bodymapを含む、ReCountで提供されているヒトデータは、②52,580遺伝子。③遺伝子ID(この場合はEnsembl gene ID)の最初の6個分を表示

・ TCCで実行すると...

「書籍 | トランスクリプトーム解析 | 3.3.3 クラスタリング」の「p143-144の網掛け部分」を、「解析 | クラスタリング | TCC(Sun 2013)」のように書くと以下のような感じになります。

```
in_f1 <- "bodymap_count_table.txt" #① #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 <- "bodymap_phenodata.txt" # #入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
out_f <- "hoge.png" # #出力ファイル名を指定してout_fに格納
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)
```

#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_f1, header=TRUE, row.names=1, sep=",")
```

```
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep=",") #確認してるだけ
```

```
colnames(data) <- phenotype$tissue.type #dataオブジェクトの列名をphenotypeのtissue.typeで置き換える
```

```
#本番
```

```
out <- clusterSample(data, dist.method="spearman", #クラスタリング方法指定
                      hclust.method="average", unique.pattern=1) #クラスタリング方法指定
```

```
#ファイルに保存
```

```
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2]) #ファイルに保存
```

```
par(mar=c(0, 4, 1, 0)) #下、左、上、右の余白を指定
```

```
plot(out, sub="", xlab="", cex.lab=1.2, #樹形図(デフォルト)
      + cex=1.3, main="", ylab="Height") #樹形図(デフォルト)
```

```
dev.off() #おまじ
```

```
windows #おまじ
```

```
> dim(data)
```

```
[1] 52580 19
```

```
> head(rownames(data))
```

```
[1] "ENSG000000000003" "ENSG000000000005"
```

```
[3] "ENSG000000000419" "ENSG000000000457"
```

```
[5] "ENSG000000000460" "ENSG000000000938"
```

```
> |
```

```
R Console
> #ファイルに保存
> png(out_f, pointsize=13, width=param_fig[1], h$
> par(mar=c(0, 4, 1, 0)) #下、左$
> plot(out, sub="", xlab="", cex.lab=1.2, #樹形図$
+ cex=1.3, main="", ylab="Height") #樹形図$
> dev.off() #おまじ$
windows
2
> dim(data)
[1] 52580 19
> head(rownames(data))
[1] "ENSG000000000003" "ENSG000000000005"
[3] "ENSG000000000419" "ENSG000000000457"
[5] "ENSG000000000460" "ENSG000000000938"
> |
```

Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



ReCount

次に、①giladという、②6サンプルのデータセットを解析する。③の記述内容から、肝臓(liver)サンプルで、「メス(female) 3サンプル vs. オス(male) 3サンプル」の比較を行っているデータなのだろうと妄想する

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
① gilad	20009012	human	② 6	41,356,738	link	link	link	③ liver; males and femlaes
maq	20167110	human	(technical) ** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

giladデータ解析

① Desktop - hogeフォルダ中の、gilad_*という2つのファイルは赤枠内のリンク先から取得

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and femlaes
maq	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled			
montgomery	20220756	human	60	*886,468,054	link	link	link	
pickrell	20220758	human	69	*886,468,054	link	link	link	
sultan	18599741	human	4	6,573,643	link	link	link	

```
R Console  
> getwd()  
[1] "C:/Users/kadota/Desktop/hoge"  
> list.files(pattern="gilad")  
[1] "gilad_count_table.txt"  
[2] "gilad_phenodata.txt"  
> |
```

giladデータ解析

①gilad_phenodata.txt中の形式は、さきほどのbodymap_phenodata.txtとは異なっている。②で異なる取り扱い方をしている点に注意。基本は一気にコピペでよい

giladデータ解析

giladデータのサンプル間クラスタリングを実行。phenotype情報は入力ファイルによってフォーマットが異なる点に注意。

```
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
out_f <- "hoge_gilad.png"
param_fig <- c(600, 400)

#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_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep=" ", quote="")

phenotype
colnames(data) <- paste(phenotype$gender, rownames(phenotype), sep="_")#dataオブジェクトの列
colnames(data)

#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
par(mar=c(0, 4, 1, 0))
```

① #入力ファイル名を指定してin_f1に格納(カウントデータ)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#パッケージの読み込み

#確認してるだけです

#確認してるだけです

#下、左、上、右の順で余白(行)を指定

②

giladデータ解析

giladデータ解析

giladデータのサンプル間クラスタリングを実行。phenotype情報は入力ファイルによってフォーマットが異なる点に注意。

```
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
out_f <- "hoge_gilad.png"
param_fig <- c(600, 400)
```

```
#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_f1)
phenotype <- read.table(in_f2)
phenotype
colnames(data)
colnames(data)
```

```
#本番
```

```
out <- clusterSam
      hclus
```

```
#ファイルに保存
```

```
png(out_f, points=param_fig)
par(mar=c(0, 4, 1, 0))
```

① #入力ファイル名を指定してin_f1に格納(カウントデータ)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#パッケージの読み込み

#確認してるだけです

```
> phenotype
```

```
      num.tech.reps gender
```

```
SRX014818and9
SRX014820and1
SRX014822and3
SRX014824and5
SRX014826and7
SRX014828and9
```

```
F
F
F
M
M
M
```

```
> colnames(data) <- paste(phenotype$gender, rownames(phenotype), sep="_")$
```

```
> colnames(data)
```

#確認してるだけです

```
[1] "F_SRX014818and9" "F_SRX014820and1" "F_SRX014822and3"
[4] "M_SRX014824and5" "M_SRX014826and7" "M_SRX014828and9"
> |
```

giladデータ解析

giladデータ解析

giladデータのサンプル間クラスタリングを実行。phenotype情報は入力ファイルによってフォーマットが異なる点に注意。

```
in_f1 <- "gilad_count_table.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 <- "gilad_phenodata.txt" #入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
out_f <- "hoge_gilad.png" #出力ファイル名を指定してout_fに格納
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)
```

#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_f1)
```

```
phenotype <- read.table(in_f2)
```

```
phenotype
```

```
colnames(data) <-
```

```
colnames(data)
```

#本番

```
out <- clusterSam
```

```
hclus
```

#ファイルに保存

```
png(out_f, points
```

```
par(mar=c(0, 4, 1
```

R Console

```
> phenotype
```

#確認してるだけです

```
num.tech.reps gender
```

num.tech.reps	gender
2	F
2	F
2	F
2	M
2	M
2	M

```
> colnames(data) <- paste(phenotype$gender, rownames(phenotype), sep="_")$
```

```
> colnames(data)
```

#確認してるだけです

```
[1] "F_SRX014818and9" "F_SRX014820and1" "F_SRX014822and3"
[4] "M_SRX014824and5" "M_SRX014826and7" "M_SRX014828and9"
```

```
> |
```


結果の解釈

① サンプル間クラスタリング結果。メス(Female)とオス(Male)サンプルが入り混じっていることが分かる。このことから「オス vs. メス」で2群間比較を行っても、おそらく何も見えてこない。例えば、発現変動遺伝子(Differentially Expressed Genes; DEGs)同定を行っても、DEG数はおそらくゼロ

• giladデータ解析
giladデータのサンプル間クラスタリングを実行。phenotype情報

```
in_f1 <- "gilad_count_table.txt"  
in_f2 <- "gilad_phenodata.txt"  
out_f <- "hoge_gilad.png"  
param_fig <- c(600, 400)
```



#入力ファイル名を指定してin_f1に格納(カウントデータ)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
#出力ファイル名を指定してout_fに格納
#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

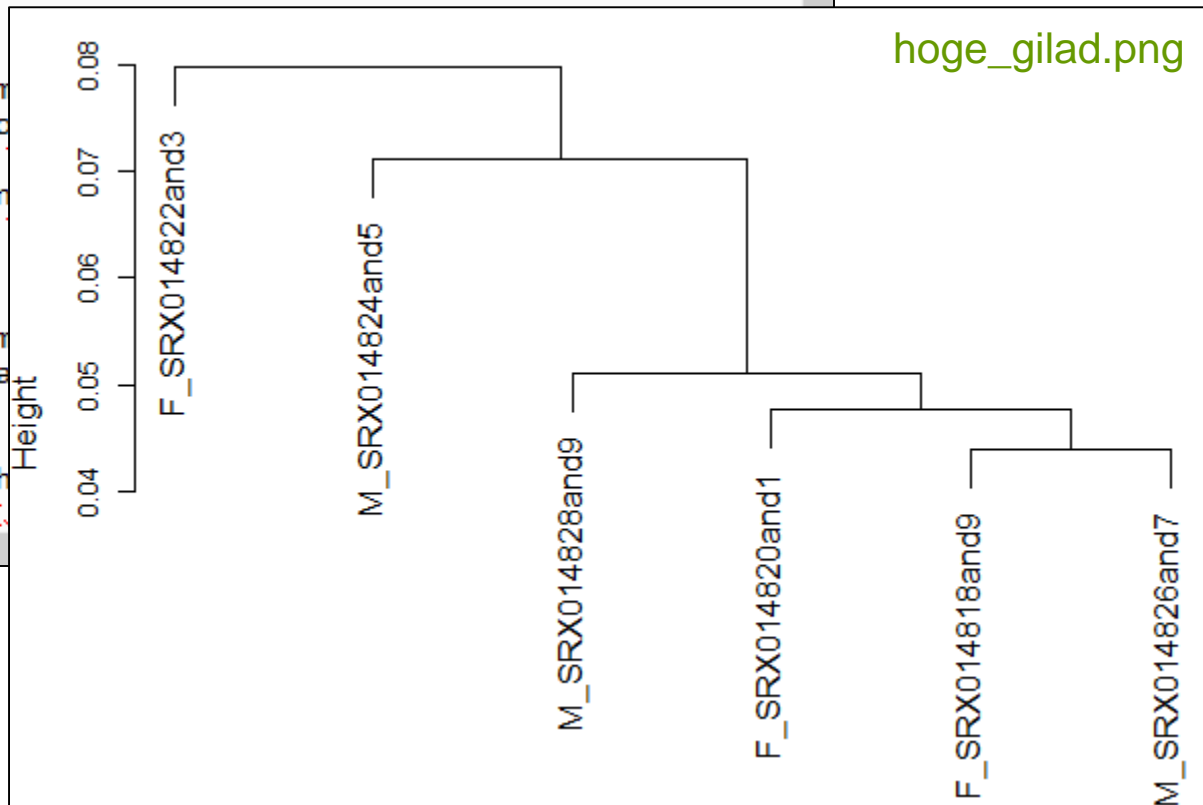
```
#入力ファイルの読み込み  
data <- read.table(in_f1, header=TRUE, row.names=colnames(data))  
phenotype <- read.table(in_f2, header=TRUE, row.names=colnames(phenotype))  
colnames(data) <- paste(phenotype$gender, rownames(data))  
colnames(phenotype) <- rownames(phenotype)
```

#確認し
#確認し

```
#本番  
out <- clusterSample(data, dist.method="spearm", hclust.method="average", unique.pars=TRUE)
```

```
#ファイルに保存  
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2],  
par(mar=c(0, 4, 1, 0)))
```

#下、左



縦軸に注目

一般に、クラスタリング時にはあまり①縦軸の距離 (Height)に注目されないが、②TCCのclusterSample関数は、サンプル間の距離Dを「1 - Spearman相関係数(r)」で定義している。この場合、値の取りうる範囲は $0 \leq D \leq 2$ であり、D=0が完全に同じ発現パターン、D=2が完全に逆の発現パターンとなる

• giladデータ解析
giladデータのサンプル間クラスタリングを実行。phenotype情報

```
in_f1 <- "gilad_count_table.txt" #入力ファイル名を指定してin_f1に格納
in_f2 <- "gilad_phenodata.txt" #入力ファイル名を指定してin_f2に格納
out_f <- "hoge_gilad.png" #出力ファイル名を指定してout_fに格納
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)
```

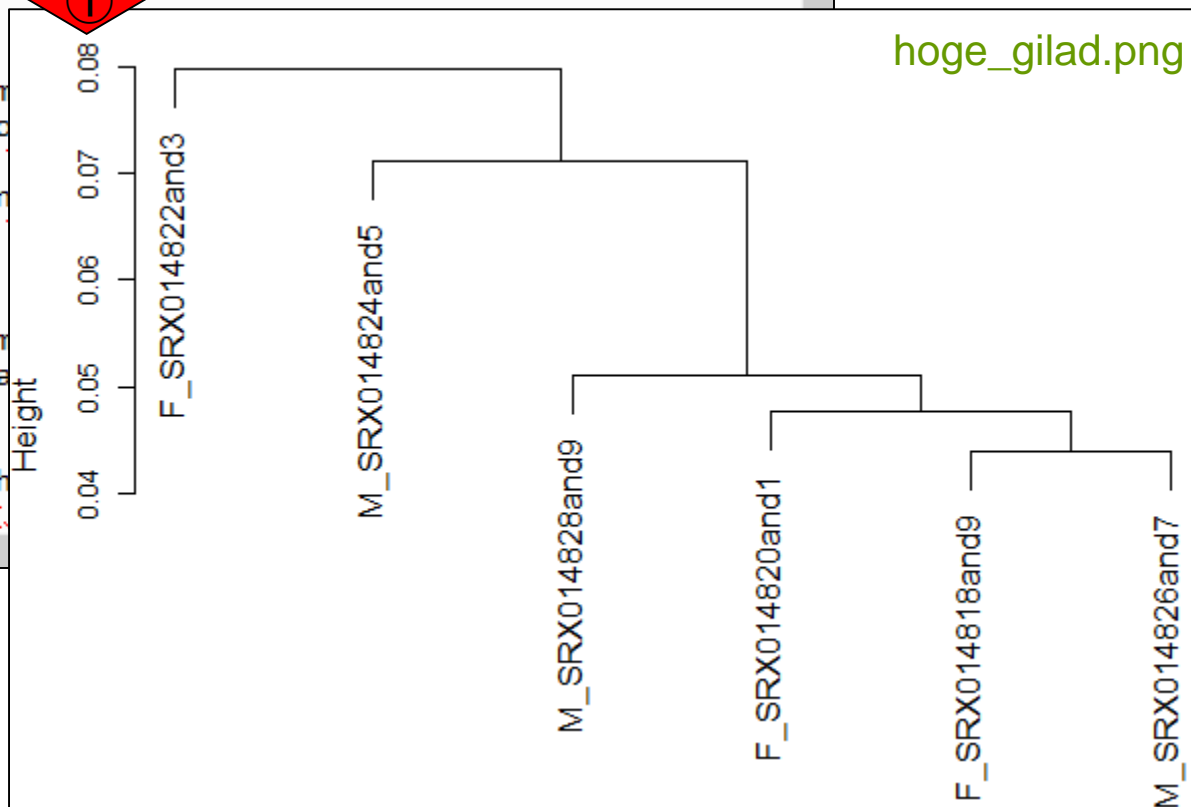
```
#必要なパッケージをロード
library(TCC)
```

```
#パッケージを読み込み
```

```
#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=phenotype) #確認し
phenotype <- read.table(in_f2, header=TRUE, row.names=phenotype) #確認し
colnames(data) <- paste(phenotype$gender, rownames(data)) #確認し
colnames(data)
```

```
#本番
out <- clusterSample(data, dist.method="spearman", hclust.method="average", unique.pairs=TRUE)
```

```
#ファイルに保存
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2], par(mar=c(0, 4, 1, 0))) #下、左
```



縦軸に注目

①ヒト肝臓(liver)のメス(F) vs. オス(M)のgiladデータセットの場合、②(サンプル間の平均)距離の最大値は0.08であり、全体的に似たサンプル同士であることがわかる、というのが次のスライド以降で示すbodymapデータを合わせたマージ後のデータとの比較でわかる

giladデータ解析

giladデータのサンプル間クラスタリングを実行、phenotype情報を入力

```
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
out_f <- "hoge_gilad.png"
param_fig <- c(600, 400)
```

```
#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_f1, header=TRUE, row.names="")
phenotype <- read.table(in_f2, header=TRUE, row.names="")
phenotype
colnames(data) <- paste(phenotype$gender, rownames(phenotype))
colnames(data)
```

#本番

```
out <- clusterSample(data, dist.method="spearm", hclust.method="average", unique.pars=1)
```

#ファイルに保存

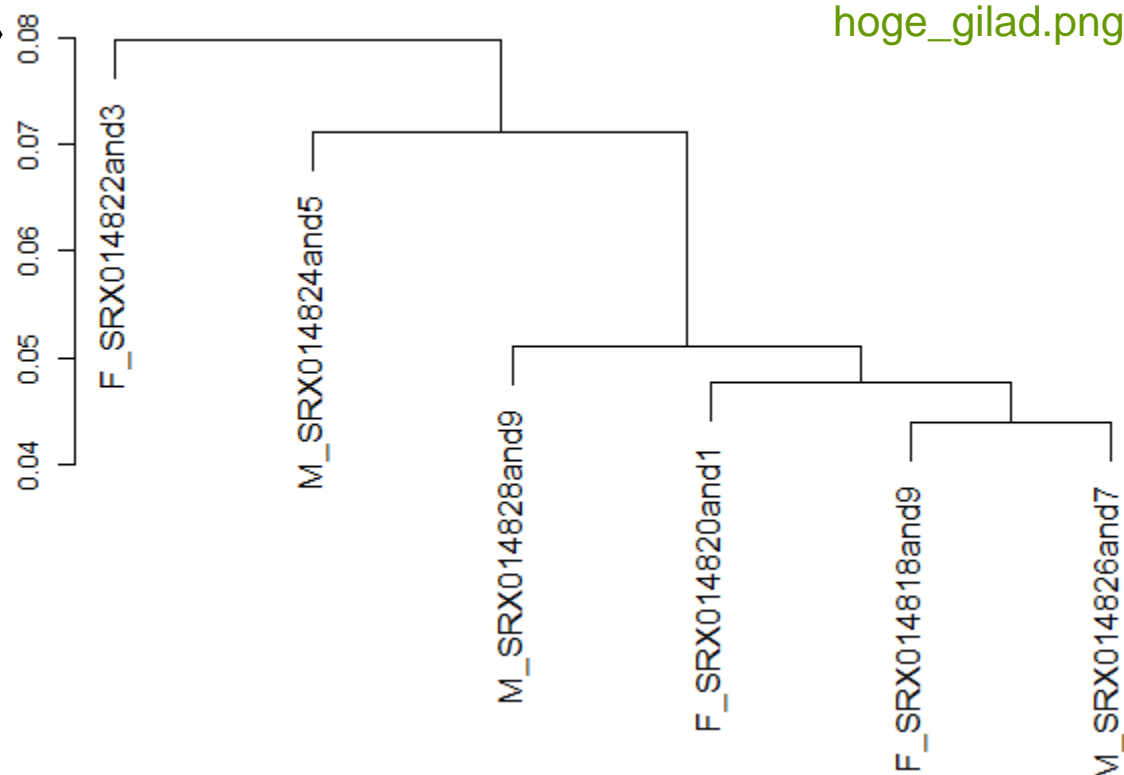
```
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])
par(mar=c(0, 4, 1, 0))
```

① #入力ファイル名を指定してin_f1に格納(カウントデータ)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)

#パッケージの読み込み

②

Height



Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



ReCountのヒトデータは

ほぼおさらい。①giladを含む、ReCountで提供されているヒトデータは、②52,580遺伝子。③遺伝子ID(この場合はEnsembl gene ID)の最初の6個分を表示

• giladデータ解析

giladデータのサンプル間クラスタリングを実行、phenotype情報は入力ファイルによってフォーマットが異なる点に注意。

```
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
out_f <- "hoge_gilad.png"
param_fig <- c(600, 400)
```

```
#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_f1, header=TRUE)
phenotype <- read.table(in_f2, header=
phenotype
colnames(data) <- paste(phenotype$gen
colnames(data)
```

```
#本番
```

```
out <- clusterSample(data, dist.metho
hclust.method="average",
```

```
#ファイルに保存
```

```
png(out_f, pointsize=13, width=par
par(mar=c(0, 4, 1, 0))
```

```
#入力ファイル名を指定してin_f1に格納(カウントデータ)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

```
#パッケージの読み込み
```

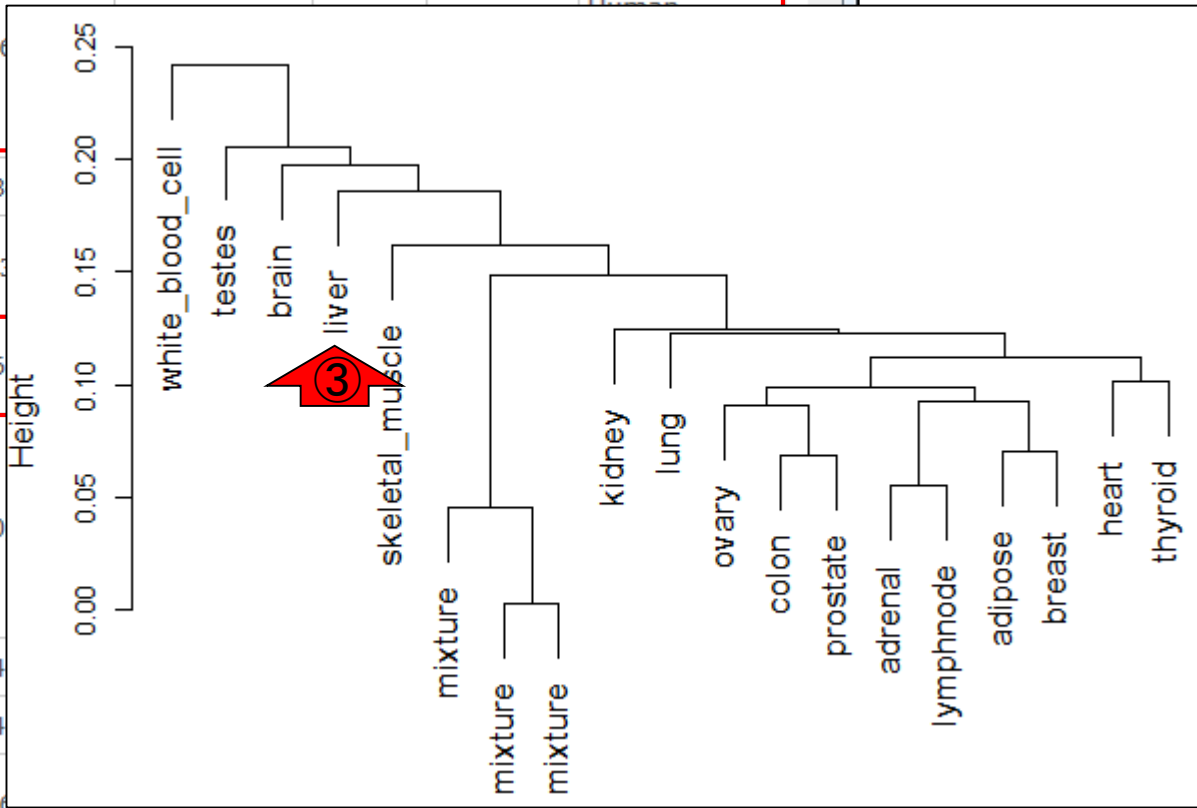
R Console

```
> #ファイルに保存
> png(out_f, pointsize=13, width=param_fig[1], height=par$
> par(mar=c(0, 4, 1, 0)) #下、左、上、右$
> plot(out, sub="", xlab="", cex.lab=1.2, #樹形図(デンドロ$
+ cex=1.3, main="", ylab="Height") #樹形図(デンドロ$
> dev.off() #おまじない
null device
1
> dim(data)
[1] 52580 6
> head(rownames(data))
[1] "ENSG000000000003" "ENSG000000000005" "ENSG000000000419"
[4] "ENSG000000000457" "ENSG000000000460" "ENSG000000000938"
> |
```

ReCount

①19サンプルのbodymapデータセットと、②肝臓6サンプルのgiladデータセットをマージ(合併;この場合は列方向で結合)してクラスタリング。③bodymapデータ中には肝臓(liver)サンプルがあるので、②giladデータ中のliverサンプルがどのあたりに位置するのかを眺める

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,6				Illumina Human
cheung	20856902	human	41	834,58				
core	19056941	human	2	8,670,3				
gilad	20009012	human	6	41,356				
maq	20167110	human	14 (technical) ** 2 (biological)	71,970				
montgomery	20220756	human	60	*886,4				
pickrell	20220758	human	69	*886,4				
sultan	18599741	human	4	6,573,6				



bodymap + gilad

①出力ファイルはhoge_merge.png。②bodymapデータ読み込み部分。③phenotype行列中のtissue.type列の情報をサンプル名として利用。④最終的にbodymapデータはdata_bodymapとして取り扱うようにしている。まずはコード全体をコピー実行しておき、各部の説明を聞くのでよい

bodymap + gilad

2つのデータセットを読み込んで、マージ(列方向で結合)して、

```
out_f <- "hoge_merge.png"
param_fig <- c(700, 400)
```



#出力ファイル名を指定してout_fに格納
#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(bodymap)
```

```
in_f1 <- "bodymap_count_table.txt"
in_f2 <- "bodymap_phenodata.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)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)

```
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep=" ", quote="")
```

#確認してるだけです
#dataオブジェクトの列名を変更

```
phenotype
colnames(data) <- phenotype$tissue.type
```

#確認してるだけです
#行数と列数を表示

```
colnames(data)
dim(data)
```

```
data_bodymap <- data
```

#dataをdata_bodymapに格納



```
#入力ファイルの読み込み(gilad)
```

```
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.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)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)

```
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep=" ", quote="")
```



bodymap + gilad

bodymap + gilad

2つのデータセットを読み込んで、マージ(列方向で結合)して、サンプル間クラスタリングを実行。

```

out_f <- "hoge_merge.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(bodymap)
in_f1 <- "bodymap_count_table.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 <- "bodymap_phenodata.txt"    #入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")
phenotype

```

```

colnames(data) <- phenotype$tissue
colnames(data)
dim(data)
data_bodymap <- data

```

```

#入力ファイルの読み込み(gilad)
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep="\t", quote="")

```

```

ERS025091      caucasian
> colnames(data) <- phenotype$tissue.type #dataオブジェクトの$
#確認してるだけです
> colnames(data)
[1] "adipose"      "adrenal"      "brain"
[4] "breast"       "colon"        "heart"
[7] "kidney"       "liver"        "lung"
[10] "lymphnode"    "mixture"      "mixture"
[13] "mixture"      "ovary"        "prostate"
[16] "skeletal_muscle" "testes"      "thyroid"
[19] "white_blood_cell"
> dim(data)
#行数と列数を表示
[1] 52580      19
> data_bodymap <- data
#dataをdata_bodymap$
> |

```



bodymap + gilad

①giladデータ読み込み部分。②bodymapのときはサンプル名の作り方が異なっている点に注意。③最終的にgiladデータはdata_giladとして扱うようにしている。④giladデータの行数は52580、列数は6

bodymap + gilad

2つのデータセットを読み込んで、マージ(列方向で結合)して、サンプル間比較分析を実行。

```
#入力ファイルの読み込み(gilad)
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")
phenotype <- read.table(in_f2, header=TRUE, row.names=1, sep=" ", quote="")
phenotype
colnames(data) <- paste(phenotype$gender, rownames(phenotype), sep="_")
colnames(data)
dim(data)
data_gilad <- data
```

#入力ファイル名を指定してin_f1に格納(カウントデータ)
#入力ファイル名を指定してin_f2に格納(サンプルラベル情報)
#確認してるだけです
#確認してるだけです
#行数と列数を表示
#dataをdata_giladに格納

①
②
③

#本番

```
data <- cbind(data_bodymap, data_gilad)
colnames(data)
dim(data)
out <- clusterSample(data, dist.
  hclust.method="average")
```

#ファイルに保存

```
png(out_f, pointsize=13, width=1000, height=1000)
par(mar=c(0, 4, 1, 0))
plot(out, sub="", xlab="", cex.l=1.5)
```

R Console

```
SRX014818and9      2      F
SRX014820and1      2      F
SRX014822and3      2      F
SRX014824and5      2      M
SRX014826and7      2      M
SRX014828and9      2      M
> colnames(data) <- paste(phenotype$gender, rownames(phenotype), sep="_")
> colnames(data)
[1] "F_SRX014818and9" "F_SRX014820and1" "F_SRX014822and3"
[4] "M_SRX014824and5" "M_SRX014826and7" "M_SRX014828and9"
#確認してるだけです
#行数と列数を表示
> dim(data)
[1] 52580      6
> data_gilad <- data
#dataをdata_giladに格納
> |
```

④

bodymap + gilad

①マージ(列方向で結合)して得られたdataオブジェクトを、②眺めているところ。③19 + 6 = 25列

bodymap + gilad

2つのデータセットを読み込んで、マージ(列方向で結合)して、サンプル間クラスタリングを実行。

```
#本番
data <- cbind(data_bodymap, data_gilad)
colnames(data)
dim(data)
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
par(mar=c(0, 4, 1, 0))
plot(out, sub="", xlab="", cex.l
     cex=1.3, main="", ylab="Height
dev.off()
```

① 2つのデータセットを列方向で結合(cbind)

#確認してるだけです
#行数と列数を表示

②

②

```
R Console
> colnames(data) #確認してるだけです
[1] "adipose"      "adrenal"      "brain"
[4] "breast"       "colon"        "heart"
[7] "kidney"       "liver"        "lung"
[10] "lymphnode"   "mixture"      "mixture"
[13] "mixture"     "ovary"        "prostate"
[16] "skeletal_muscle" "testes"      "thyroid"
[19] "white_blood_cell" "F_SRX014818and9" "F_SRX014820and1"
[22] "F_SRX014822and3" "M_SRX014824and5" "M_SRX014826and7"
[25] "M_SRX014828and9"

> dim(data) #行数と列数を表示
[1] 52580 25

> out <- clusterSample(data, dist.method="spearman", #クラスタ$
+                       hclust.method="average", unique.pattern=TRUE) #$
> |
```

③

bodymap + gilad

①出力ファイルはhoge_merge.png。②giladデータは、単独でクラスターを形成している。③bodymapサンプルで最も近いのはliverであり、極めて妥当

bodymap + gilad

2つのデータセットを読み込んで、マージ(列方向で結合)して、サンプル間クラスタリングを実行。

```
out_f <- "hoge_merge.png"
param_fig <- c(700, 400)
```



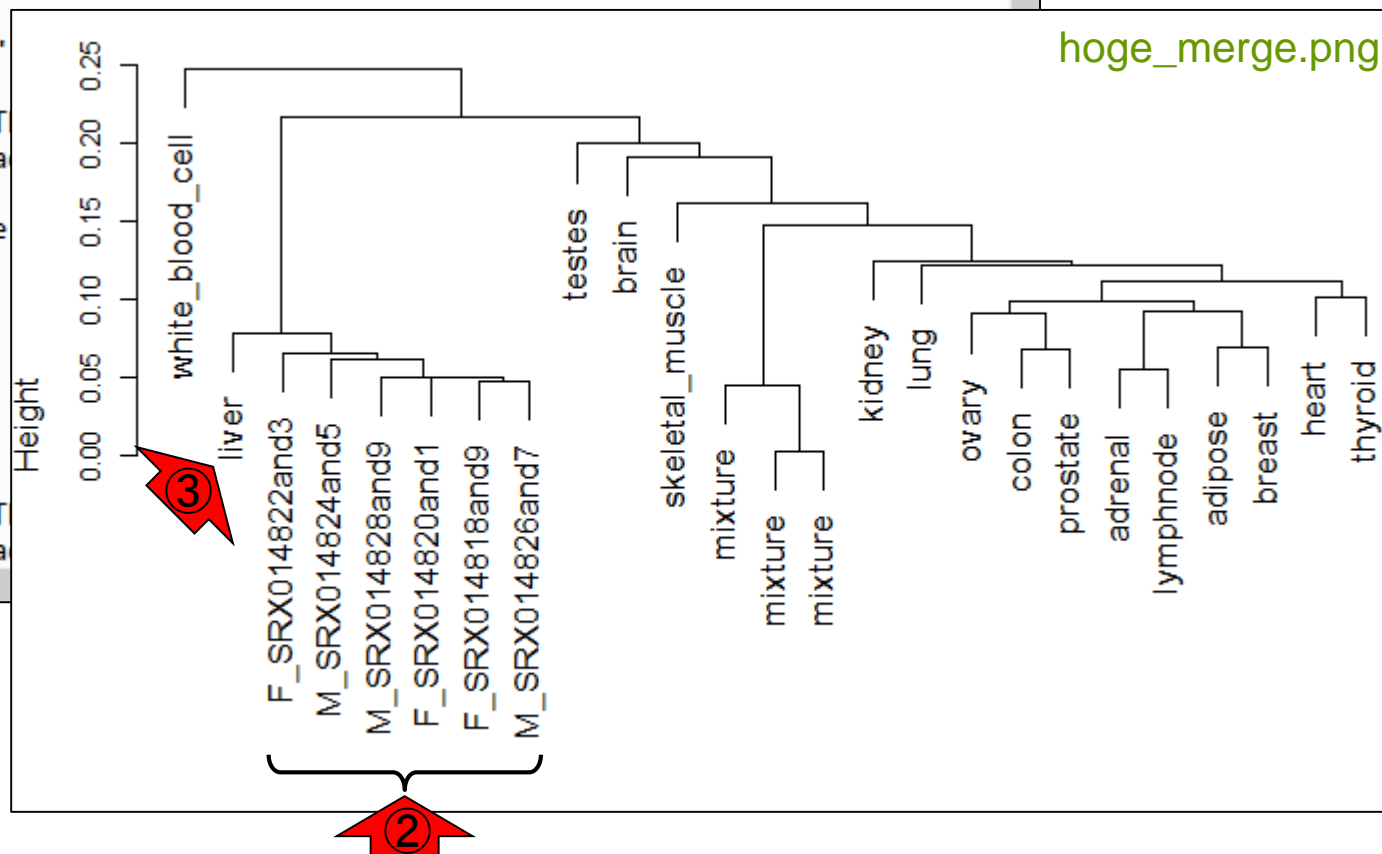
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)

```
#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(bodymap)
in_f1 <- "bodymap_count_table.txt"
in_f2 <- "bodymap_phenodata.txt"
data <- read.table(in_f1, header=T)
phenotype <- read.table(in_f2, header=T)
colnames(data) <- phenotype$tissue
dim(data)
```

```
#入力ファイルの読み込み(gilad)
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
data <- read.table(in_f1, header=T)
phenotype <- read.table(in_f2, header=T)
```



bodymap + gilad

①giladデータのみの、②(サンプル間の平均)距離の最大値は約0.07。③全体では約0.25。
①のgiladデータは、同じ肝臓(liver)サンプルでメス(Female) vs. オス(Male)のデータなので、よく考えてみると当たり前と言えれば当たり前

- bodymap + gilad
2つのデータセットを読み込んで、マージ(列方向で結合)して、サンプル間クラス

```
out_f <- "hoge_merge.png"  
param_fig <- c(700, 400)
```

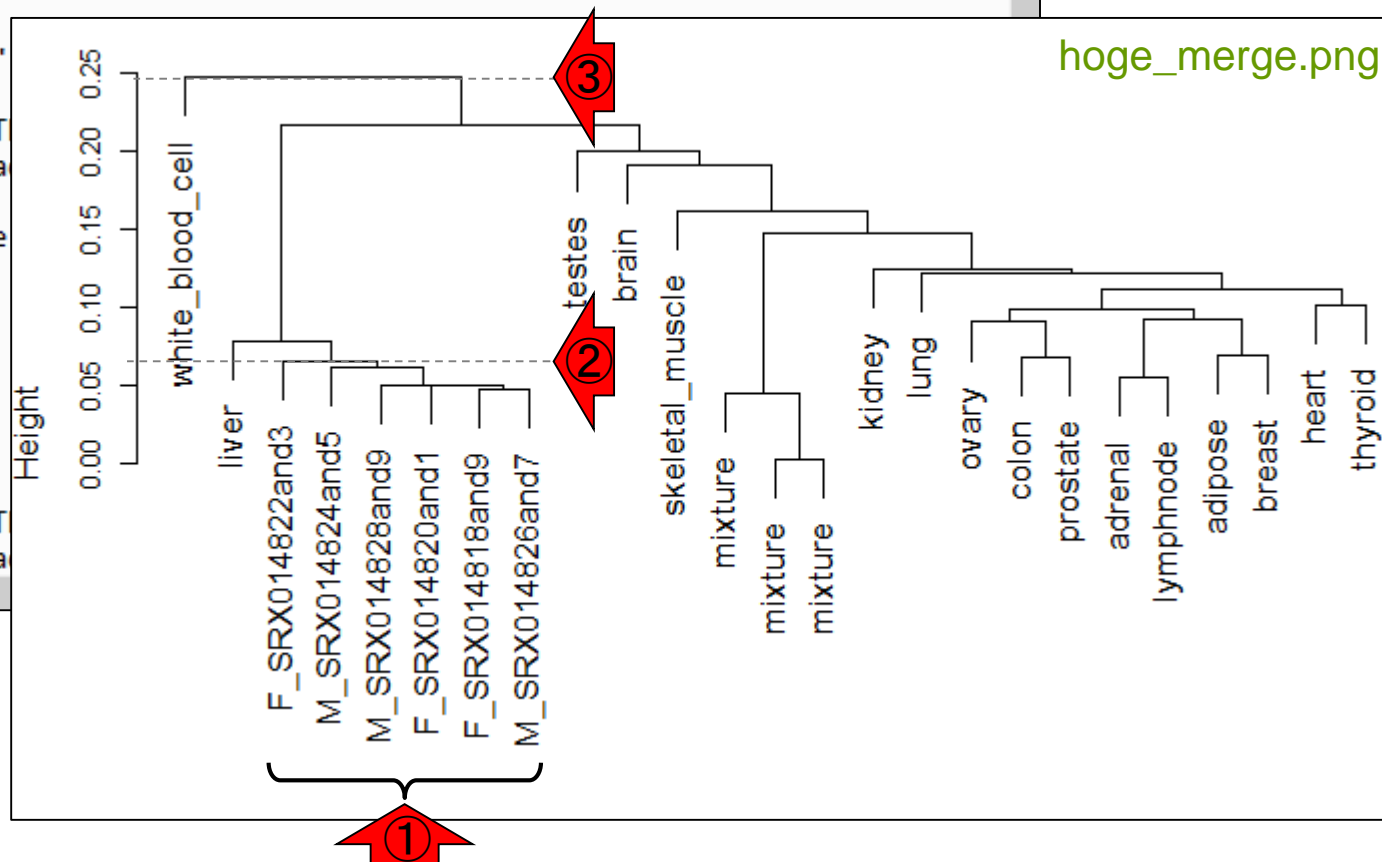
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)

```
#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(bodymap)  
in_f1 <- "bodymap_count_table.txt"  
in_f2 <- "bodymap_phenodata.txt"  
data <- read.table(in_f1, header=T)  
phenotype <- read.table(in_f2, header=T)  
phenotype  
colnames(data) <- phenotype$tissue  
colnames(data)  
dim(data)  
data_bodymap <- data
```

```
#入力ファイルの読み込み(gilad)  
in_f1 <- "gilad_count_table.txt"  
in_f2 <- "gilad_phenodata.txt"  
data <- read.table(in_f1, header=T)  
phenotype <- read.table(in_f2, header=T)
```



bodymap + gilad

赤枠のgiladデータのみの結果を眺めているだけでは、(原著論文の主目的は把握していないが…)2群間比較を目的としていた場合は発現変動遺伝子(DEG)がないことがほぼ明白。ゆえに、多少の誤解を承知の上で言えばhopeless。しかし(1-r)で定義した距離Dの値をよく眺めれば、似たサンプル内での議論だからだと納得できる

bodymap + gilad

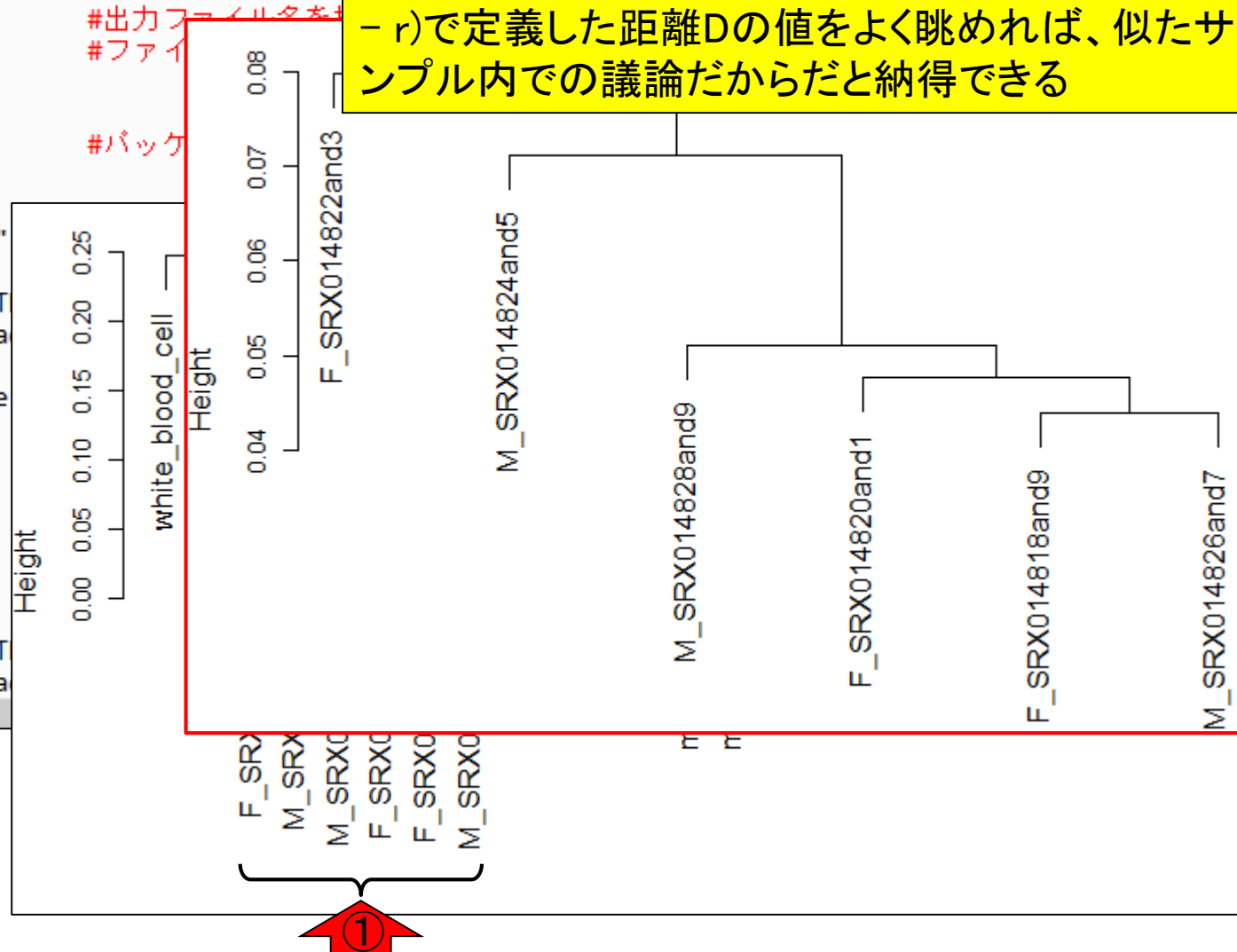
2つのデータセットを読み込んで、マージ(列方向で結合)して、サンプル間

```
out_f <- "hoge_merge.png"
param_fig <- c(700, 400)

#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(bodymap)
in_f1 <- "bodymap_count_table.txt"
in_f2 <- "bodymap_phenodata.txt"
data <- read.table(in_f1, header=T)
phenotype <- read.table(in_f2, header=T)
colnames(data) <- phenotype$tissue
dim(data)
data_bodymap <- data

#入力ファイルの読み込み(gilad)
in_f1 <- "gilad_count_table.txt"
in_f2 <- "gilad_phenodata.txt"
data <- read.table(in_f1, header=T)
phenotype <- read.table(in_f2, header=T)
```



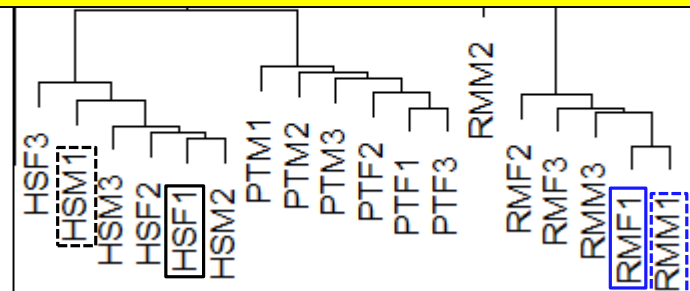
Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



解析データ

解析データは、スライド25で作成した20,689遺伝子 × 18サンプルのsample_blekhman_18.txt。クラスタリング結果からDEGが多く検出されると予想されるヒト(HS) vs. アカゲザル(RM)の2群間比較を行う。①雌雄差の影響を排除すべく、各群からメスとオス1匹ずつの、2 vs. 2の反復あり解析とする



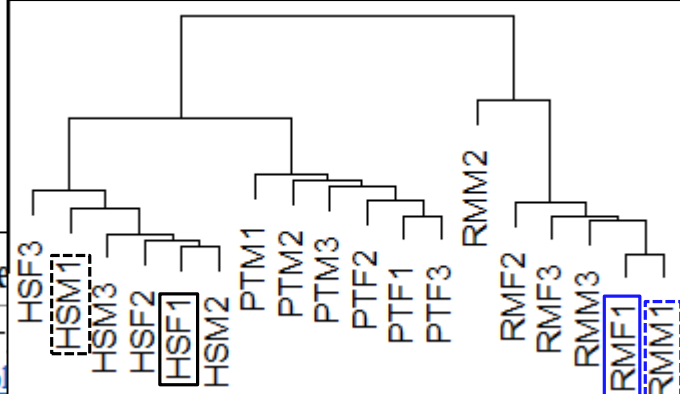
20,689 genes

	ヒト(HS)						チンパンジー(PT)						アカゲザル(RM)					
	メス(Female)			オス(Male)			メス			オス			メス			オス		
	HSF1	HSF2	HSF3	HSM1	HSM2	HSM3	PTF1	PTF2	PTF3	PTM1	PTM2	PTM3	RMF1	RMF2	RMF3	RMM1	RMM2	RMM3
ENSG000000000003	300	168	421	359	574	429	386	409	685	428	464	480	1348	705				
ENSG000000000005	0	0	1	0	1	4	1	0	1	1	1	1	1	2	0	0		
ENSG000000000419	81	61	56	39	78	62	100	66	65	59	58	93	67	72	57	49	82	90
ENSG000000000457	91	62	76	114	73	95	131	229	87	274	239	149	89	69	118	117	114	163
ENSG000000000460	6	17	12	15	7	17	8	8	5	12	7	10	4	4	10	7	3	4
ENSG000000000938	44	65	210	73	43	65	84	104	76	198	31	58	73	28	54	80	34	72
ENSG000000000971	4765	7225	3405	3600	6383	5546	5382	8331	4335	2568	5019	2653	13566	9964	18247	14236	5196	11834
ENSG000000001036	297	251	189	200	234	249	305	301	313	254	151	331	292	106	379	201	88	140
ENSG000000001084	630	737	306	336	984	459	417	328	885	298	569	218	1062	786	1110	873	664	1752
ENSG000000001167	36	30	36	29	33	28	63	80	25	69	74	41	62	34	108	97	35	61
ENSG000000001460	3	1	5	1	4	2	0	1	1	1	1	3	1	1	1	0	1	3
ENSG000000001461	49	37	34	28	62	32	75	69	40	90	69	60	210	92	176	247	81	117
ENSG000000001487	117	93	88	80	131	110	125	98	75	108	130	131	138	95	187	137	158	172

HS vs. RM

「HS vs. RM」の2群間比較をTCCで行う。
 ②例題1。まずはコード全体をコピー実行しておき、各部の説明を聞くのでよい

- 解析 | 発現変動 | 1について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | 1について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love 2014) (last modified 2015/11/15)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun 2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li 2013) (last modified 2015/02/07)
- 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 (sample

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ー。

②

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"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_subset <- c(1, 4, 13, 16)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)
```

#必要なパッケージをロード

```
#入力ファイル名を指定してin_f1に格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#取り扱いたいサブセット情報を指定
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)閾値を指定
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
#下、左、上、右の順で余白を指定(単位は行)
```

#パッケージの読み込み

サブセット抽出

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)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px)
param_mar <- c(4, 4, 0, 0) #下、左、上、右の順番で余白を指定(単位はpx)
```

```
#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=",")
```

```
#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)
```

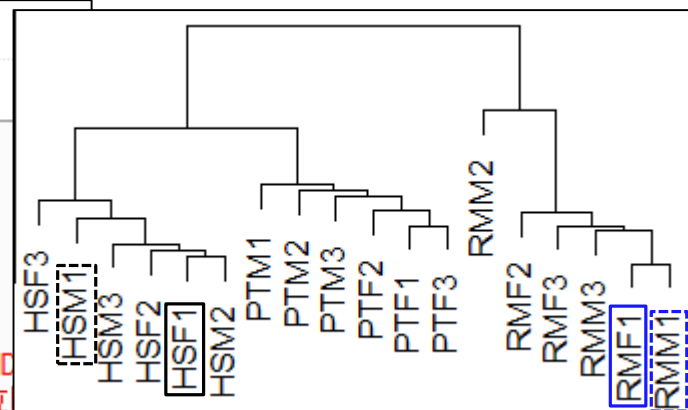
```
data <- data[,param_subset] #param_subsetで指定した列のみを抽出
```

```
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2でラベル
```

```
tcc <- new("TCC", data, data.cl) #TCCクラスオブジェクトの作成
```

```
dim(data) #行数と列数を表示
```

```
head(data) #最初の6行分を表示
```



```
R Console
> #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] # $
> data.cl <- c(rep(1, param_G1), rep(2, param_G2)) # $
> tcc <- new("TCC", data, data.cl) # $
> dim(data) # $
[1] 20689 4
> head(data) # $
           HSF1 HSM1 RMF1 RMM1
ENSG000000000003 329 121 511 424
ENSG000000000005 0 0 0 2
ENSG000000000419 81 39 67 49
ENSG000000000457 91 114 89 117
ENSG000000000460 6 15 4 7
ENSG000000000938 44 73 73 80
>
```

サブセット抽出

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:RMF1とRMM1)

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"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_subset <- c(1, 4, 13, 16)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)
```



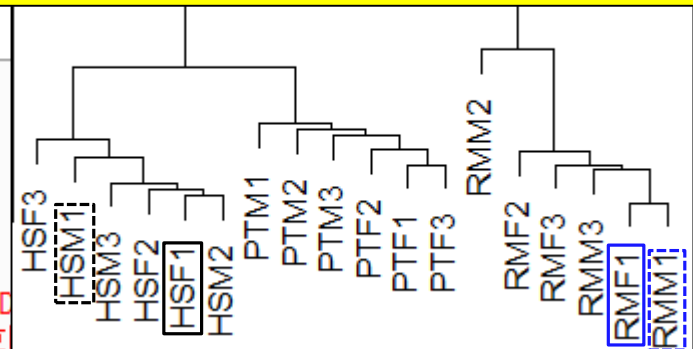
#入力ファイル名を指定してin_fに格納
 #出力ファイル名を指定してout_f1に格納
 #出力ファイル名を指定してout_f2に格納
 #取り扱いたいサブセット情報を指定
 #G1群のサンプル数を指定
 #G2群のサンプル数を指定
 #DEG検出時のfalse discovery rate (FDR)を指定
 #ファイル出力時の横幅と縦幅を指定(単位はpx)
 #下、左、上、右の順番で余白を指定(単位はpx)

```
#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, header=TRUE, row.names=1, sep="\t")

#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]
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
tcc <- new("TCC", data, data.cl)
dim(data)
head(data)
```

#パッケージの読み込み
 #param_subsetで指定したサブセットを抽出
 #G1群を1、G2群を2で指定
 #TCCクラスオブジェクトを作成
 #行数と列数を表示
 #最初の6行分を表示



```
R Console
> #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]
> data.cl <- c(rep(1, param_G1), rep(2, param_G2))
> tcc <- new("TCC", data, data.cl)
> dim(data)
[1] 20689 4
> head(data)
      HSF1 HSM1 RMF1 RMM1
ENSG000000000003 329 121 511 424
ENSG000000000005 0 0 0 2
ENSG000000000419 81 39 67 49
ENSG000000000457 91 114 89 117
ENSG000000000460 6 15 4 7
ENSG000000000938 44 73 73 80
```



サブセット抽出

入力ファイル(sample_blekhman_18.txt)を眺めるなどして、①該当サンプルの列の位置を把握していることが前提

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"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_subset <- c(1, 4, 13, 16)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)
```



```
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#取り扱いたいサブセット情報を指定
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)を指定
#ファイル出力時の横幅と縦幅を指定(単位は行)
#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)
```

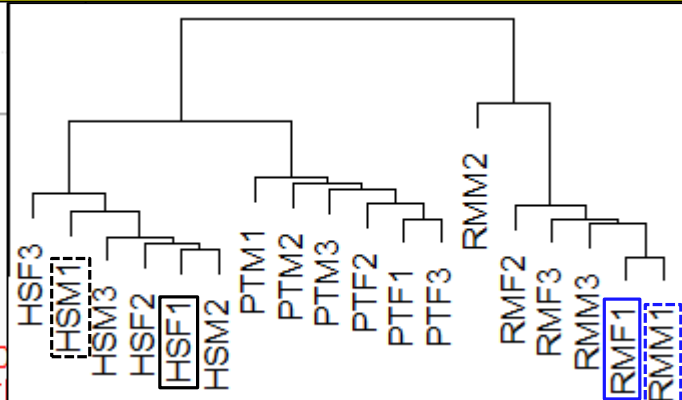
```
#パッケージの読み込み
```

```
#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
```

```
#前処理
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
data	HSF1	HSF2	HSF3	HSM1	HSM2	HSM3	PTF1	PTF2	PTF3	PTM1	PTM2	PTM3	RMF1	RMF2	RMF3	RMM1	RMM2	RMM3	
data	ENSG000000000003	329	300	168	121	421	359	574	429	386	409	685	428	511	464	480	424	1348	705
tcc	ENSG000000000005	0	0	0	0	1	0	1	4	1	0	1	1	0	1	2	2	0	0
dim	ENSG000000000419	81	61	56	39	78	62	100	66	65	59	58	93	67	72	57	49	82	90
head	ENSG000000000457	91	62	76	114	73	95	131	229	87	274	239	149	89	69	118	117	114	163
	ENSG000000000460	6	17	12	15	7	17	8	8	5	12	7	10	4	4	10	7	3	4
	ENSG000000000938	44	65	210	73	43	65	84	104	76	198	31	58	73	28	54	80	34	72
	ENSG000000000971	4765	7225	3405	3600	6383	5546	5382	8331	4335	2568	5019	2653	13566	9064	18247	14236	5196	11834



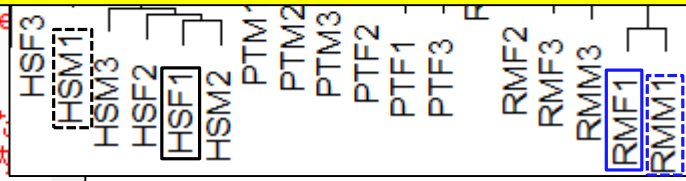
FDR

① $q < 0.05$ を満たす遺伝子数は2,489個。False discovery rate (FDR) = 0.05は、この閾値を満たす2,489個を発現変動遺伝子(Differentially Expressed Genes; **DEGs**)とみなすと、 $2,489 * 0.05 = 124.45$ 個は偽物であることを意味する。有意水準(false positive rate; FPR)5%と同じような位置づけであり、FDR5%というのは、「許容する偽物(non-DEG)混入割合」に相当する。詳細は2015.05.26の講義資料を参照のこと

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サンプル
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", iteration=3, FDR=0.1)
normalized <- getNormalizedData(tcc) #正規化後のデータを取り出してnormalized

#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を満たす遺伝子数を表示
```



```
#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(tcc$count), normalized, result) #正規化後のデータとDEG検出結果を結合
tmp <- tmp[order(tmp$rank),] #発現変動順にソート
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=FALSE)

#ファイルに保存(M-A plot)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])
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
      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 factors
TCC::INFO: (iDEGES pipeline : tmm - [ edgeR ]
TCC::INFO: Done.
> normalized <- getNormalizedData(tcc) #FDR < param_FDRを満たす遺伝子数を表示
>
> #本番 (DEG検出)
> tcc <- estimateDE(tcc, test.method="edgeR", FDR=param_FDR) #DEG検出を実行した
TCC::INFO: Identifying DE genes using edgeR
TCC::INFO: Done.
> 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を満たす遺伝子数を表示
[1] 2489
>
> |
```



FDR

ちなみに平成27年度講習会時は①2,488個でした。記憶が定かではありませんが、確かR ver. 3.1.3だったと思います。TCCのバージョンも対応して古いので、バージョンによって結果が微妙に異なる一例です。論文にバージョン番号も記載すべし!

1. ヒト2サンプル(G1群:HSF1とHSM1) vs. アカゲザル2サ

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)

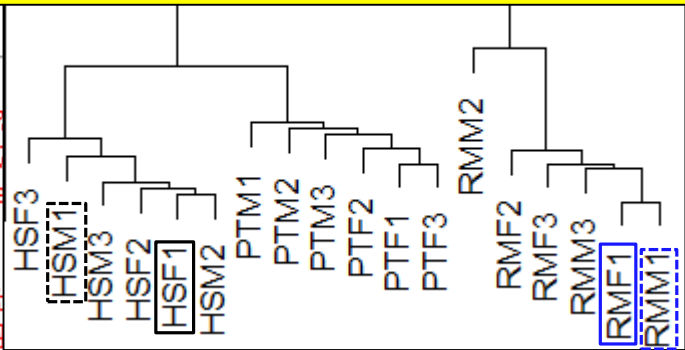
```
tcc <- estimateDE(tcc, test.method="edger", FDR=param_FDR) #DEG検出を実行した
result <- getResult(tcc, sort=FALSE) #p値などの結果をした結果をresultに格納
sum(tcc$stat$q.value < param_FDR) #FDR < param_FDRを満たす遺伝子数を表示
```

#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),] #発現変動順にソート
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
>
> |
```



FDR

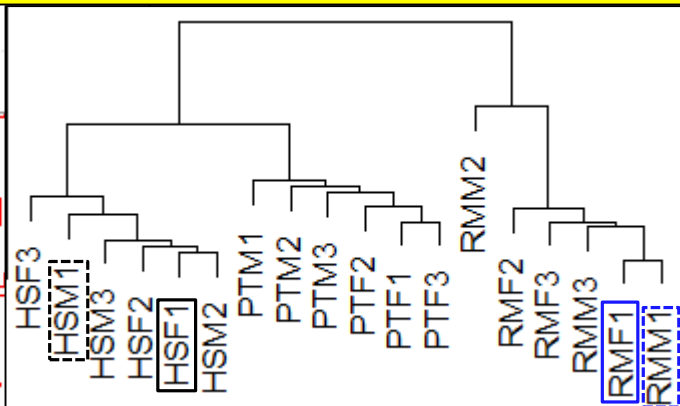
① $q < 0.30$ を満たす遺伝子数は4,785個。
 FDR = 0.30なので、 $4,785 * 0.30 = 1,435.5$ 個
 は偽物で残りの70%は本物だと判断する

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])#出力ファイル
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
      cex=0.8, cex.lab=1.2, #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, ")"), #凡例
                             col=c("magenta", "black"), pch=20),
      dev.off() #おまじない
sum(tcc$stat$q.value < 0.05) #FDR < 0.05を満たす遺伝子数
sum(tcc$stat$q.value < 0.10) #FDR < 0.10を満たす遺伝子数
sum(tcc$stat$q.value < 0.20) #FDR < 0.20を満たす遺伝子数
sum(tcc$stat$q.value < 0.30) #FDR < 0.30を満たす遺伝子数
```



```
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] 2489
> sum(tcc$stat$q.value < 0.10) # $
[1] 3121
> sum(tcc$stat$q.value < 0.20) # $
[1] 4049
> sum(tcc$stat$q.value < 0.30) # $
[1] 4785
> |
```



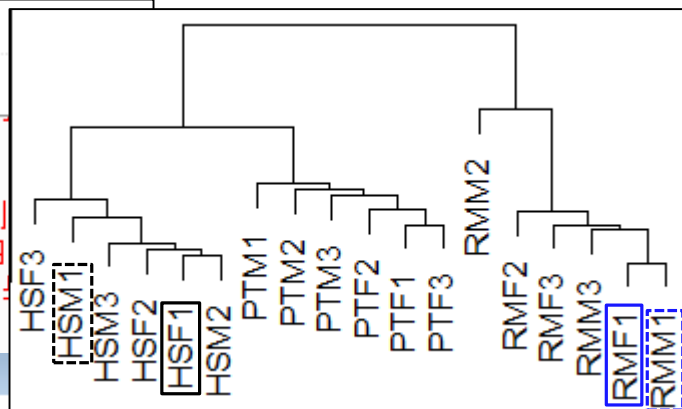
DEG数の見積もり

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:

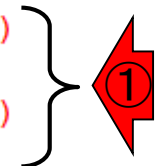
1, 4, 13, 16 列目のデータのみ抽出しています。

```
#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])
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を満たすDEG数
sum(tcc$stat$q.value < 0.10) #FDR < 0.10を満たすDEG数
sum(tcc$stat$q.value < 0.20) #FDR < 0.20を満たすDEG数
sum(tcc$stat$q.value < 0.30) #FDR < 0.30を満たすDEG数
```



```
R Console
> sum(tcc$stat$q.value < 0.05) # $
[1] 2489
> sum(tcc$stat$q.value < 0.10) # $
[1] 3121
> sum(tcc$stat$q.value < 0.20) # $
[1] 4049
> sum(tcc$stat$q.value < 0.30) # $
[1] 4785
> 2489*(1 - 0.05)
[1] 2364.55
> 3121*(1 - 0.10)
[1] 2808.9
> 4049*(1 - 0.20)
[1] 3239.2
> 4785*(1 - 0.30)
[1] 3349.5
> |
```



樹形図と一致

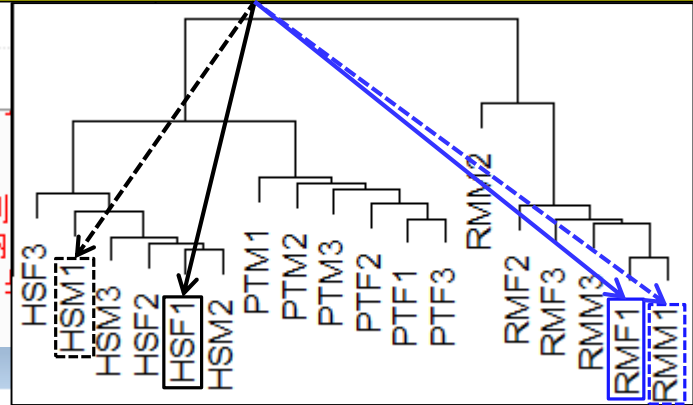
今比較しているのはHS vs. RM。クラスタリング結果からも、これらの発現プロファイルの類似度が低い(距離が遠い)ので妥当

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])
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を満たす遺伝子の数
sum(tcc$stat$q.value < 0.10) #FDR < 0.10を満たす遺伝子の数
sum(tcc$stat$q.value < 0.20) #FDR < 0.20を満たす遺伝子の数
sum(tcc$stat$q.value < 0.30) #FDR < 0.30を満たす遺伝子の数
```



```
R Console
> sum(tcc$stat$q.value < 0.05) # $
[1] 2489
> sum(tcc$stat$q.value < 0.10) # $
[1] 3121
> sum(tcc$stat$q.value < 0.20) # $
[1] 4049
> sum(tcc$stat$q.value < 0.30) # $
[1] 4785
> 2489*(1 - 0.05)
[1] 2364.55
> 3121*(1 - 0.10)
[1] 2808.9
> 4049*(1 - 0.20)
[1] 3239.2
> 4785*(1 - 0.30)
[1] 3349.5
> |
```


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F

1, 4, 13, 16 列目のデータのみ抽出しています。

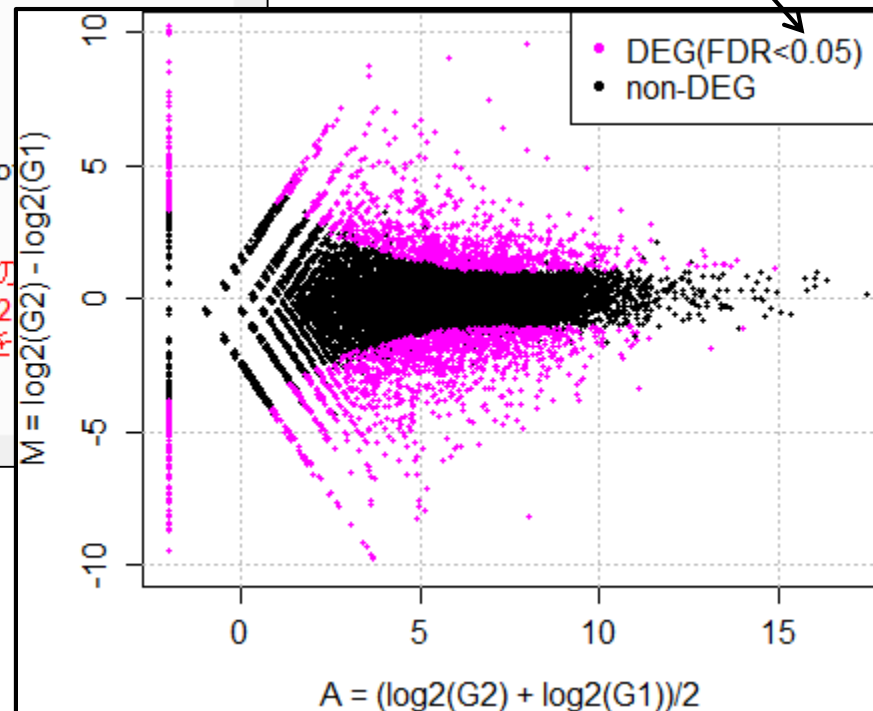
```
in_f <- "sample_blekhman_18.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_subset <- c(1, 4, 13, 16)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)
```

#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#取り扱いたいサブセット情報を指定
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
#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]
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)
dim(data)
head(data)
#param_subsetで指定した列の
#G1群を1, G2群を2
#TCCクラスオブジェクトtccを
#行数と列数を表示
#最初の6行分を表示
```



表記法...

①本当は「FDR < 0.05」という表記法は不正確であり、「q-value < 0.05、5% FDR、FDR = 5%」などと書くのが正解。「有意水準(significance level) $\alpha < 5\%$ 」と言わずに、「p-value < 0.05、a significance level of 5%、 $\alpha = 0.05$ 」などという表現が一般になされるのと同じです。数年前から放置してましたが、スライド110から徐々に修正...

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:HSF1とHSM1)

1, 4, 13, 16 列目のデータのみ抽出しています。

```
in_f <- "sample_blekman_18.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_subset <- c(1, 4, 13, 16)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)
```

```
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#取り扱いたいサブセット情報を指定
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)を指定
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
#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
```

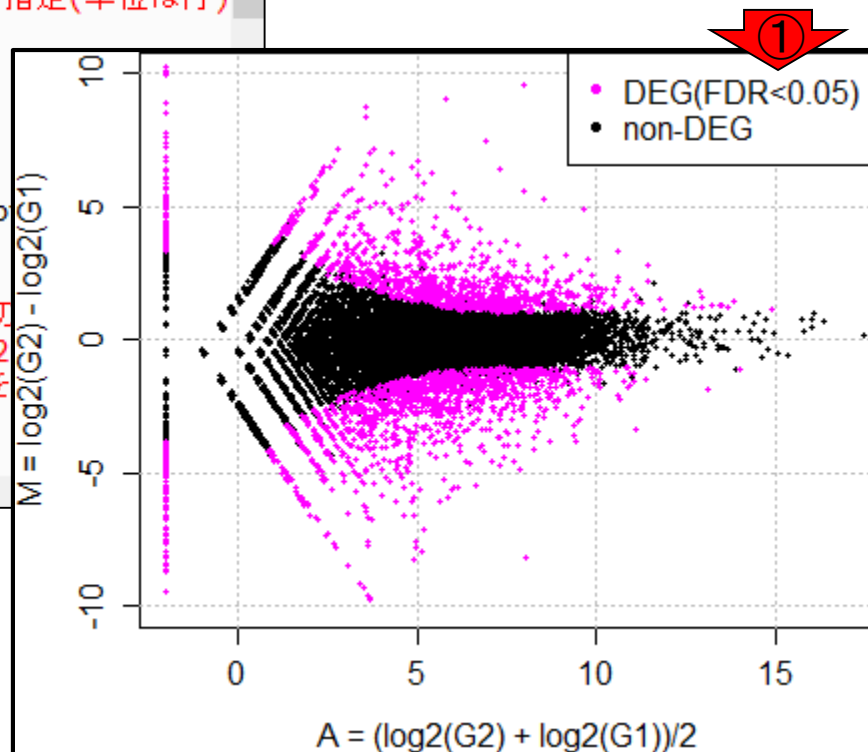
```
#入力ファイルの読み込み
```

```
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")
```

```
#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]
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
tcc <- new("TCC", data, data.cl)
dim(data)
head(data)
```

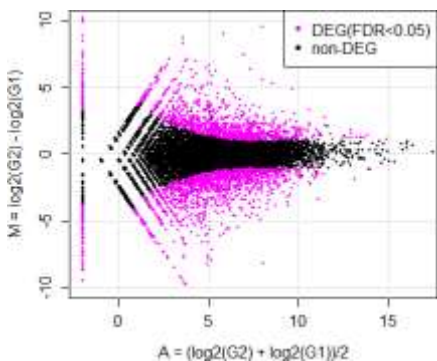
```
#param_subsetで指定した列の抽出
#G1群を1、G2群を2で指定
#TCCクラスオブジェクトtccを作成
#行数と列数を表示
#最初の6行分を表示
```



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

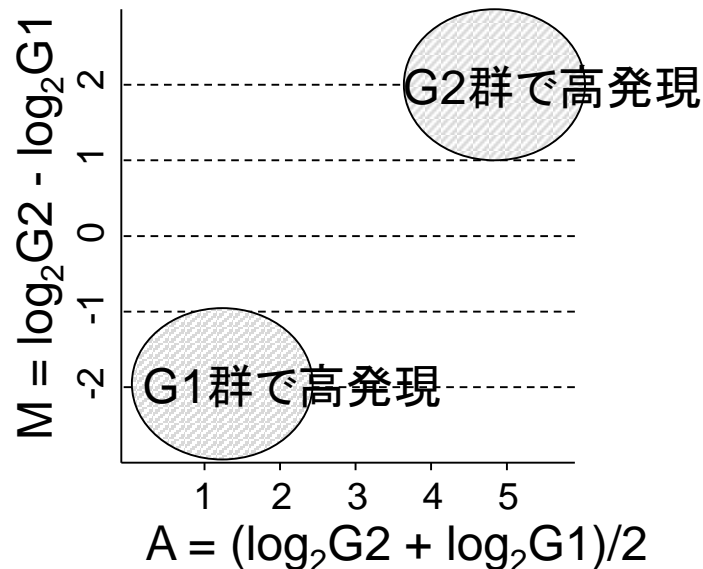
- 2群間比較用
- 横軸が全体的な発現レベル、縦軸がlog比からなるプロット
- 名前の由来は、おそらく対数の世界での縦軸が引き算 (Minus)、横軸が平均 (Average)



G1群 < G2群

G1群 = G2群

G1群 > G2群



低発現 ← 全体的に → 高発現

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"
out_f1 <- "hogel1.txt"
out_f2 <- "hogel1.png"
param_subset <- c(1, 4, 13, 16)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)

#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
```

#入力ファイル名を指定してin_fに格納
 #出力ファイル名を指定してout_f1に格納
 #出力ファイル名を指定してout_f2に格納
 #取り扱いたいサブセット情報を指定
 #G1群のサンプル数を指定
 #G2群のサンプル数を指定
 #DEG検出時のfalse discovery rate (FDR)
 #ファイル出力時の横幅と縦幅を指定(単位はピクセル)
 #下、左、上、右の順で余白を指定(単位は行)

#パッケージの読み込み



rownames(tcc\$count)	HSF1	HSM1	RMF1	RMM1	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
ENSG00000208570	0.0	0.0	1346.7	1476.9	ENSG00000208570	-2.04	11.29	4.45E-53	9.21E-49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.0	ENSG00000220191	5.80	9.06	4.58E-47	4.74E-43	2	1
ENSG00000106366	4422.0	4411.6	23.1	8.3	ENSG00000106366	8.04	-8.14	2.42E-45	1.67E-41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	-2.04	10.23	3.32E-44	1.72E-40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	-2.04	10.08	1.77E-43	7.32E-40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	-2.04	10.03	4.70E-42	1.62E-38	6	1
ENSG00000209007	0.0	0.0	615.2	479.5	ENSG00000209007	-2.04	9.92	1.25E-40	3.70E-37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	-9.69	1.53E-38	3.97E-35	8	1
ENSG00000156222	367.5	301.6	0.9	0.0	ENSG00000156222	3.61	-9.56	1.05E-36	2.42E-33	9	1
ENSG00000165272	404.8	420.0	2.6	0.8	ENSG00000165272	4.02	7.50	5.50E-36	1.14E-32	10	1

DEG検出結果1位

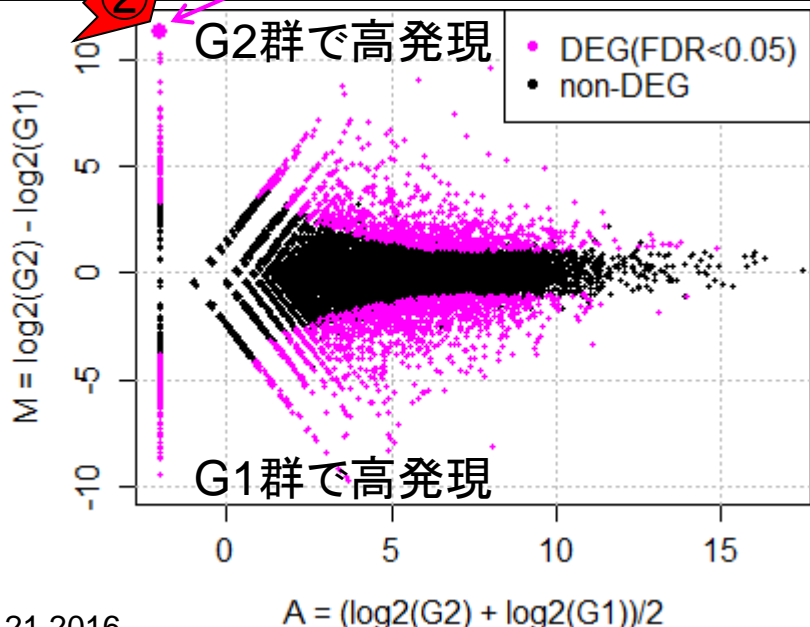
①1位はRM群(G2群)で高発現のDEG。②特定のプロットをハイライトさせるべく、③の例題2のコピペで図を作成。スライドを見るだけ



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	1346.7	1476.9	ENSG00000208570	-2.04	11.29	4.45E-53	9.21E-49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.0	ENSG00000220191	5.80	9.06	4.58E-47	4.74E-43	2	1
ENSG00000106366	4422.0	4411.6	23.1	8.3	ENSG00000106366	8.04	-8.14	2.42E-45	1.67E-41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	-2.04	10.23	3.32E-44	1.72E-40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	-2.04	10.08	1.77E-43	7.32E-40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	-2.04	10.03	4.70E-42	1.62E-38	6	1
ENSG00000209007	0.0	0.0	615.2	479.5	ENSG00000209007	-2.04	9.92	1.25E-40	3.70E-37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	-9.69	1.53E-38	3.97E-35	8	1
ENSG00000156222	367.5	301.6	0.9	0.0	ENSG00000156222	3.61	-9.56	1.05E-36	2.42E-33	9	1
ENSG00000165272	404.9	420.0	2.6	0.9	ENSG00000165272	4.02	7.50	5.50E-36	1.14E-32	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2位

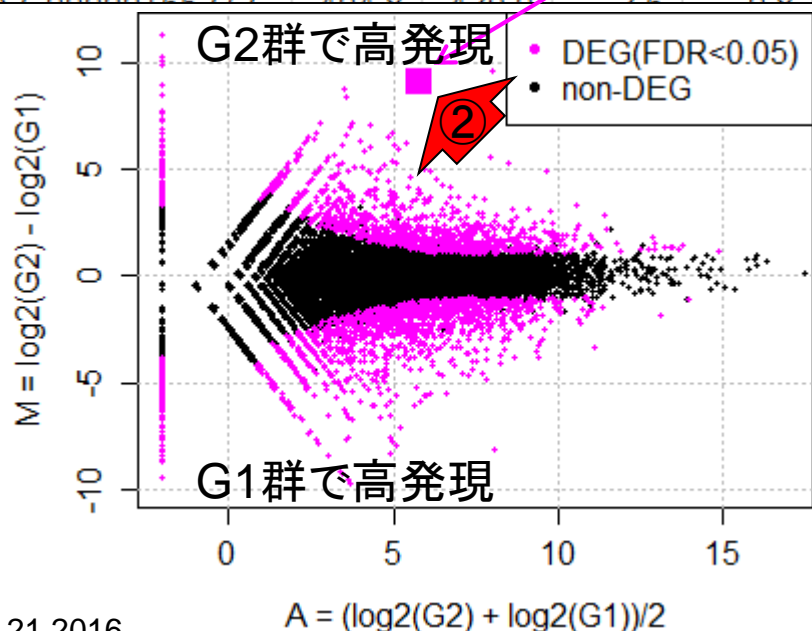
①2位もRM群(G2群)で高発現のDEG。②特定のプロットをハイライトさせるべく、③の例題3のコピペで図を作成。スライドを見るだけ



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	1346.7	1476.9	ENSG00000208570	-2.04	11.29	4.45E-53	9.21E-49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.0	ENSG00000220191	5.80	9.06	4.58E-47	4.74E-43	2	1
ENSG00000106366	4422.0	4411.6	23.1	8.3	ENSG00000106366	8.04	-8.14	2.42E-45	1.67E-41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	-2.04	10.23	3.32E-44	1.72E-40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	-2.04	10.08	1.77E-43	7.32E-40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	-2.04	10.03	4.70E-42	1.62E-38	6	1
ENSG00000209007	0.0	0.0	615.2	479.5	ENSG00000209007	-2.04	9.92	1.25E-40	3.70E-37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	-9.69	1.53E-38	3.97E-35	8	1
ENSG00000156222	367.5	301.6	0.9	0.0	ENSG00000156222	3.61	-9.56	1.05E-36	2.42E-33	9	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3位

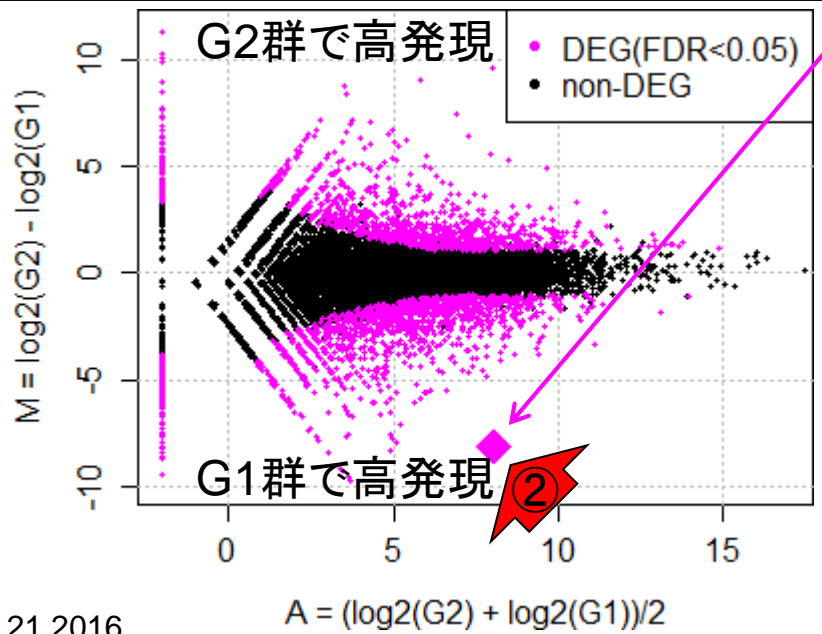
①3位はHS群(G1群)で高発現のDEG。②特定のプロットをハイライトさせるべく、③の例題4のコピペで図を作成。スライドを見るだけ



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	1346.7	1476.9	ENSG00000208570	-2.04	11.29	4.45E-53	9.21E-49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.0	ENSG00000220191	5.80	9.06	4.58E-47	4.74E-43	2	1
ENSG00000106366	4422.0	4411.6	23.1	8.3	ENSG00000106366	8.04	-8.14	2.42E-45	1.67E-41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	-2.04	10.23	3.32E-44	1.72E-40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	-2.04	10.08	1.77E-43	7.32E-40	5	1
ENSG00000070985	0.0	0.0	528.2	650.8	ENSG00000070985	-2.04	10.03	4.70E-42	1.62E-38	6	1
ENSG00000209007	0.0	0.0	615.2	479.5	ENSG00000209007	-2.04	9.92	1.25E-40	3.70E-37	7	1
ENSG00000182327	367.5	363.9	0.9	0.0	ENSG00000182327	3.67	-9.69	1.53E-38	3.97E-35	8	1
ENSG00000156222	367.5	301.6	0.9	0.0	ENSG00000156222	3.61	-9.56	1.05E-36	2.42E-33	9	1



M-A plotのA値とM値

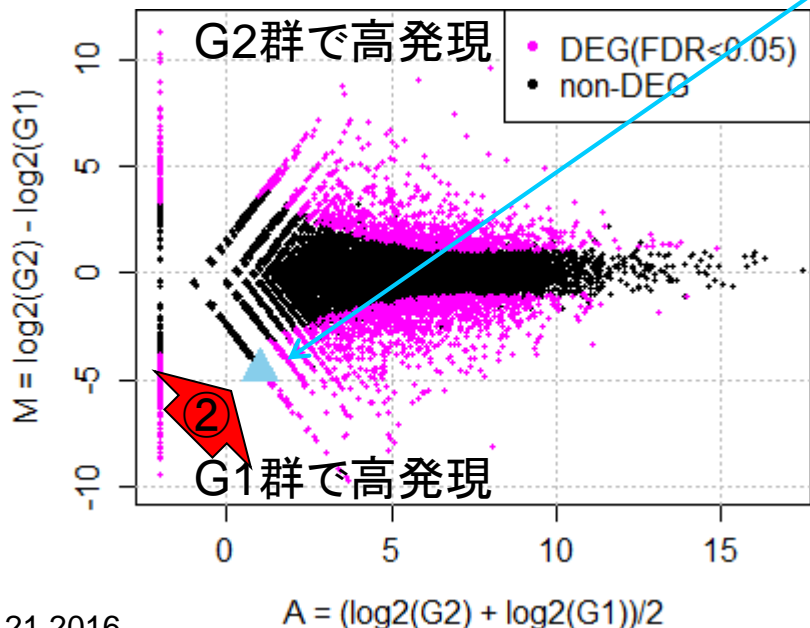
q-value

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2,489位^③

指定したFDR閾値(0.05)をギリギリ満たす2,489位の遺伝子。②をハイライトさせるべく、③の例題5のコピペで図を作成

rownames(tcc\$count)	G1(HS)群		G2(RM)群		gene_id	a.value	m.value	p-valueとその順位			
	HSF1	HSM1	RMF1	RMM1				p.value	q.value	rank	estimatedDEG
ENSG00000145687	9.0	8.9	0.9	1.7	ENSG00000145687	1.76	-2.82	0.00597	0.04967	2485	1
ENSG00000180672	9.0	8.9	0.9	1.7	ENSG00000180672	1.76	-2.82	0.00597	0.04967	2486	1
ENSG00000110442	108.5	103.1	213.9	219.4	ENSG00000110442	7.24	1.03	0.006	0.04988	2487	1
ENSG00000105327	5.7	24.2	1.8	2.5	ENSG00000105327	2.50	-2.80	0.00601	0.04998	2488	1
ENSG00000139445	17.0	2.5	0.0	0.8	ENSG00000139445	1.01	-4.55	0.00601	0.04998	2489	1
ENSG00000105321	61.1	128.5	14.2	47.4	ENSG00000105321	5.76	-1.62	0.00602	0.05005	2490	0
ENSG00000118017	1.1	2.5	13.3	10.0	ENSG00000118017	2.21	2.66	0.00603	0.05012	2491	0
ENSG00000119630	19.2	12.7	34.6	55.7	ENSG00000119630	4.75	1.50	0.00604	0.05013	2492	0
ENSG00000110917	768.8	591.7	1440.8	1334.8	ENSG00000110917	9.92	1.03	0.00604	0.05014	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。

Contents

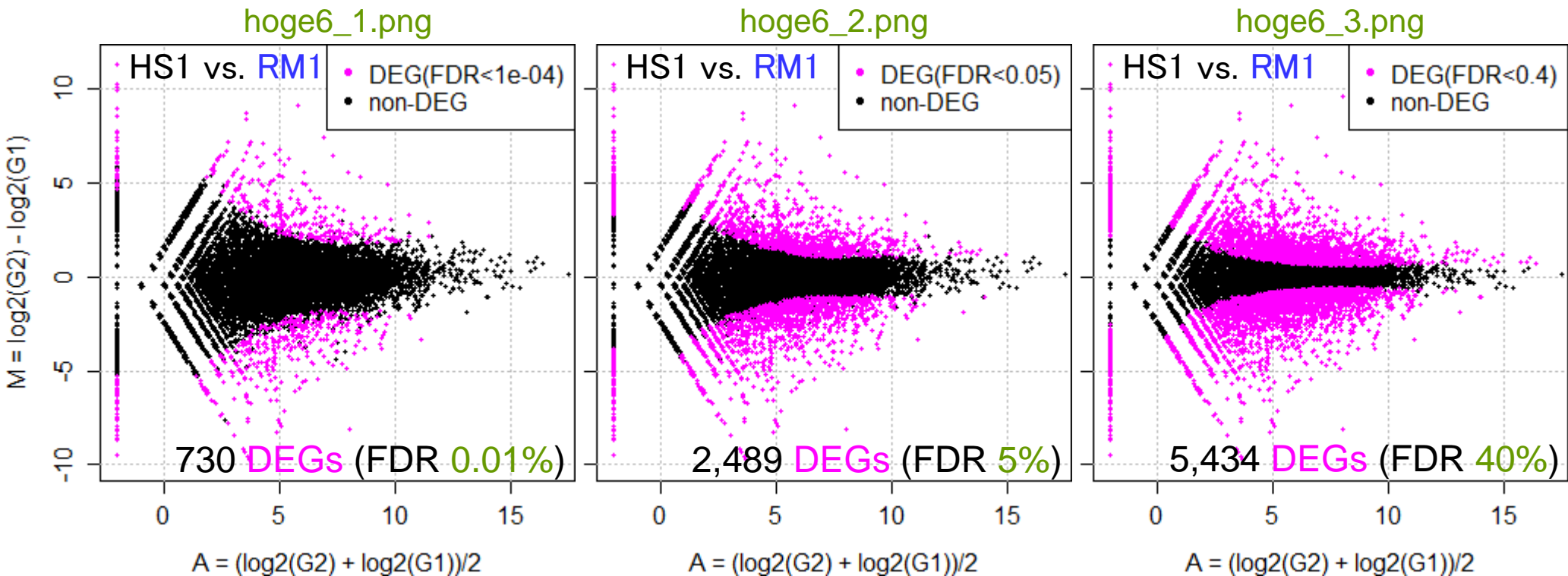
- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(DEGがほとんどない同一群の場合)
 - 倍率変化(2倍以上、1/2以下の発現変動)を用いた場合
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



分布やモデル

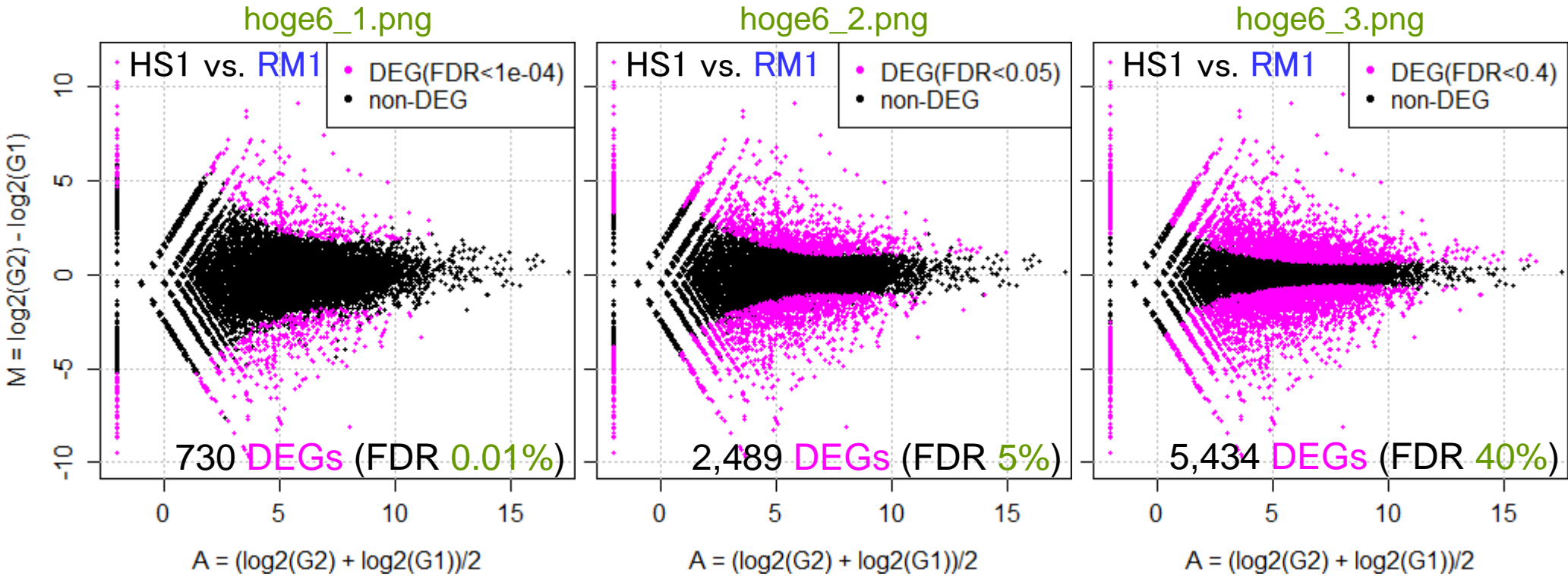


(当たり前だが) FDR閾値を緩めると得られるDEG数は増える傾向にある。①の例題6のコピペで図を作成



厳しい ← FDR閾値 → 緩い
 少ない ← DEG数 → 多い

分布やモデル



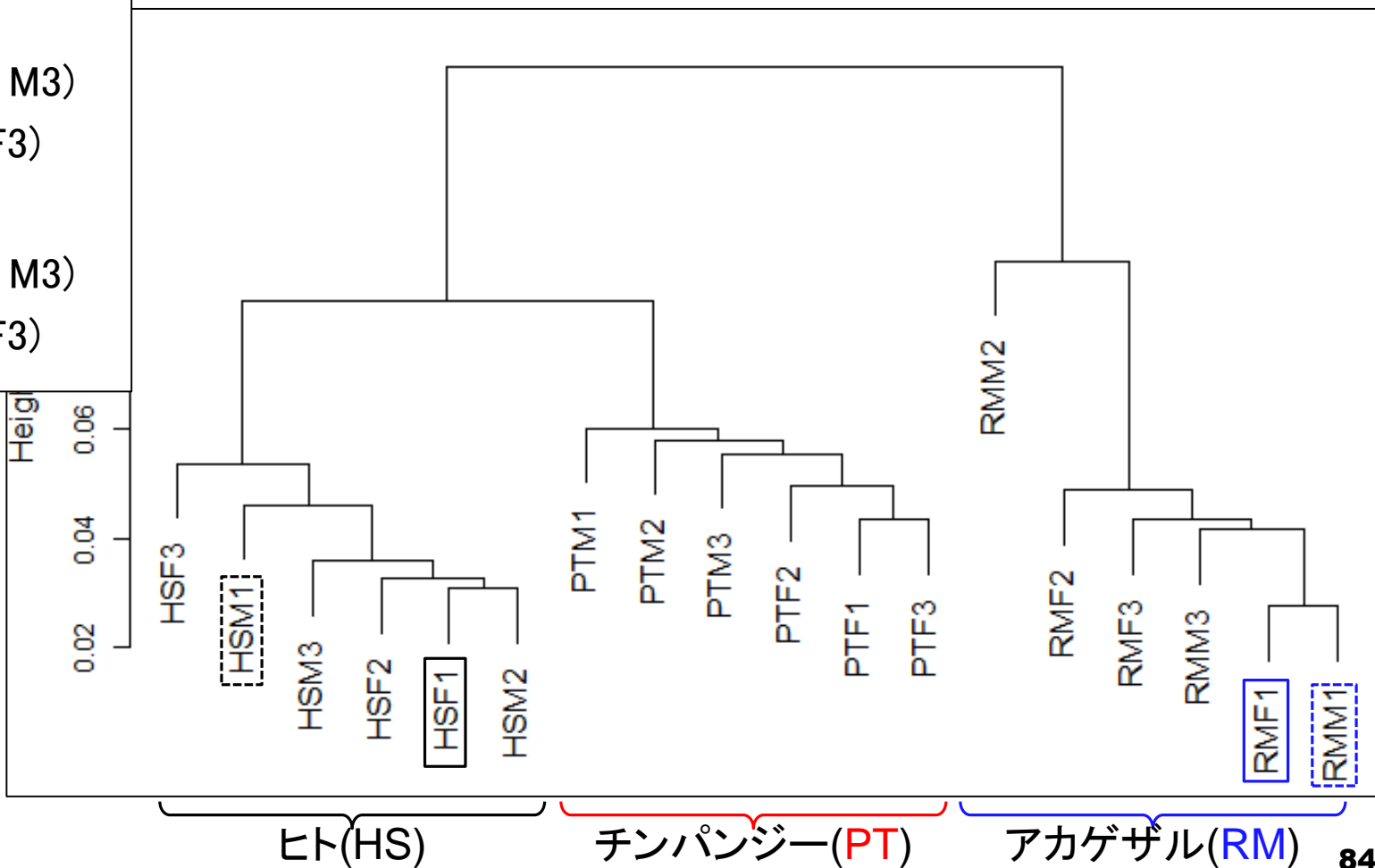
厳しい ← FDR閾値 → 緩い
 少ない ← DEG数 → 多い



「HS1 vs. RM1」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)



HS vs. PT

「HS vs. PT」のDEG同定を行う。①ヒト(HS)と②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)

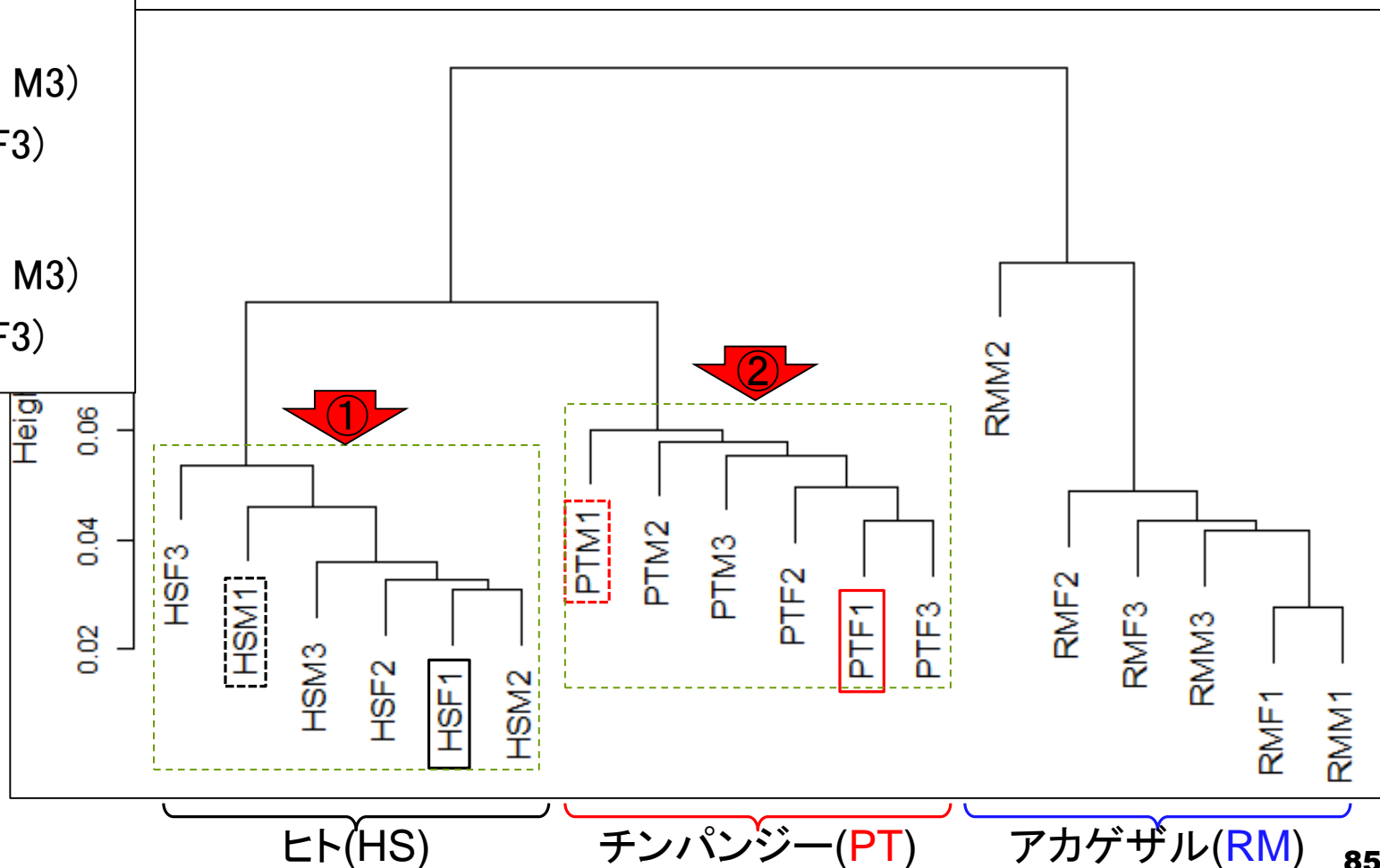
- オス3匹(M1, M2, M3)
- メス3匹(F1, F2, F3)

■ 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)

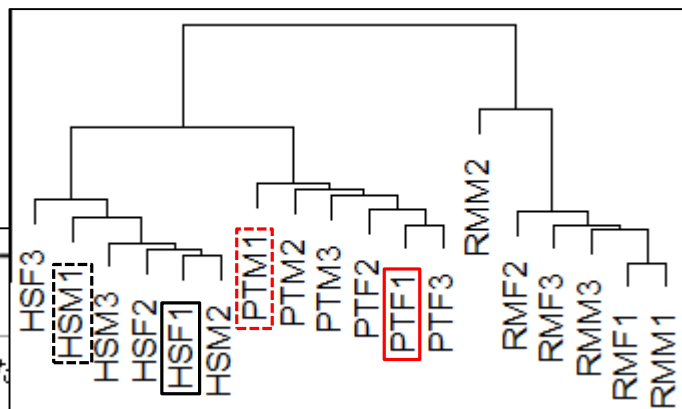


HS vs. PT

- 解析 | 発現変動 | 1について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | 1について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love_2014) (last modified 2015/11/15)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li_2013) (last modified 2015/02/07)

解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ (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を用いた解析を行います。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(Cebus imellanus; 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。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。



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。

1. ヒト2サンプル(G1群:HSF1とHSM1)

1, 4, 13, 16 列目のデータ

```
in_f <- "sample_blekhman_18.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_subset <- c(1, 4, 7, 10)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(400, 310)
param_mar <- c(4, 4, 0, 0)
```

```
in_f <- "sample_blekhman_18.txt"
out_f1 <- "hoge7.txt"
out_f2 <- "hoge7.png"
param_subset <- c(1, 4, 7, 10)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(400, 310)
param_mar <- c(4, 4, 0, 0)
```

#必要なパッケージをロード

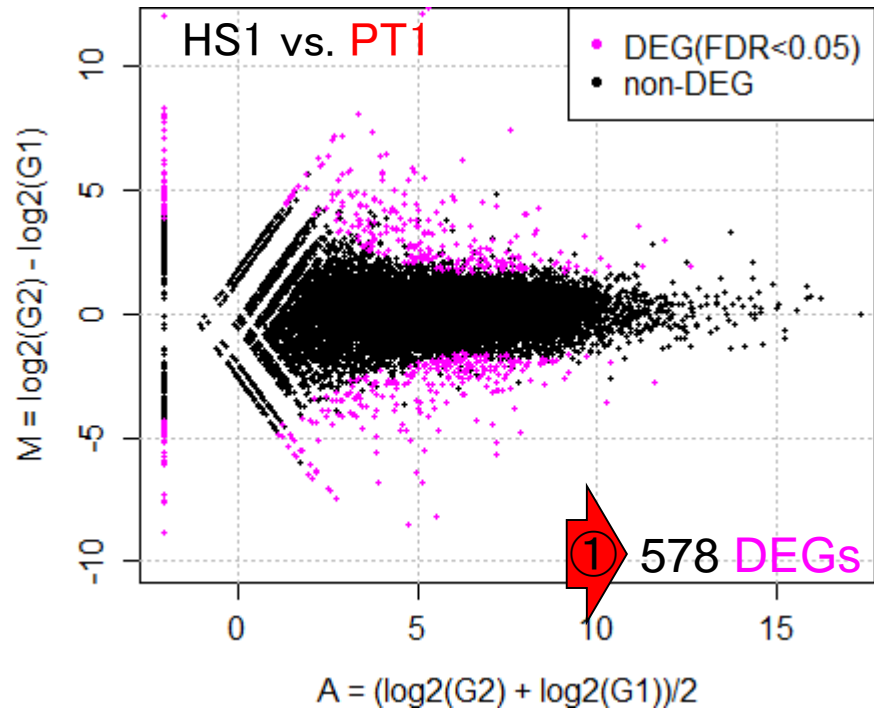
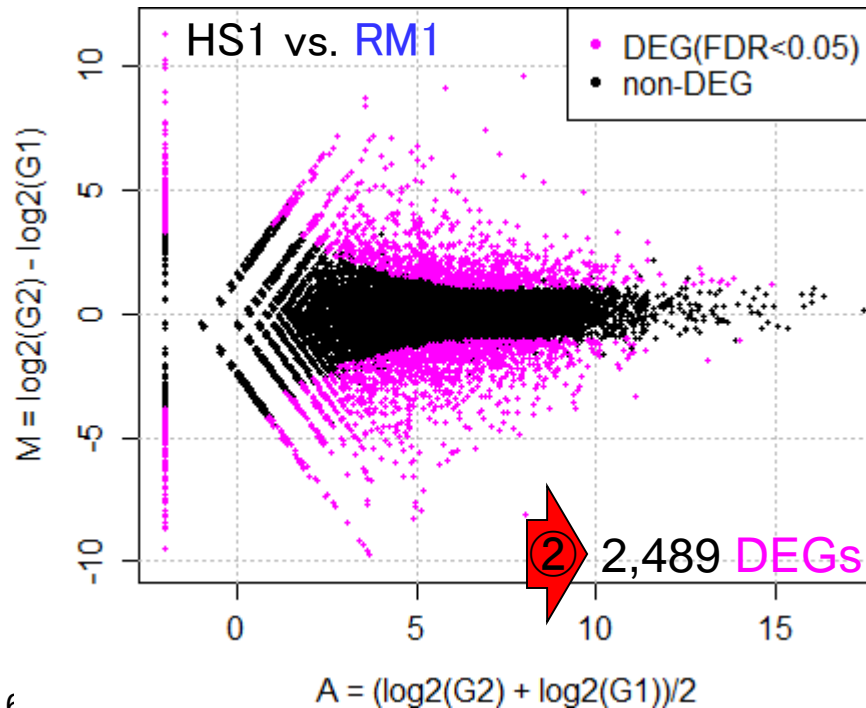
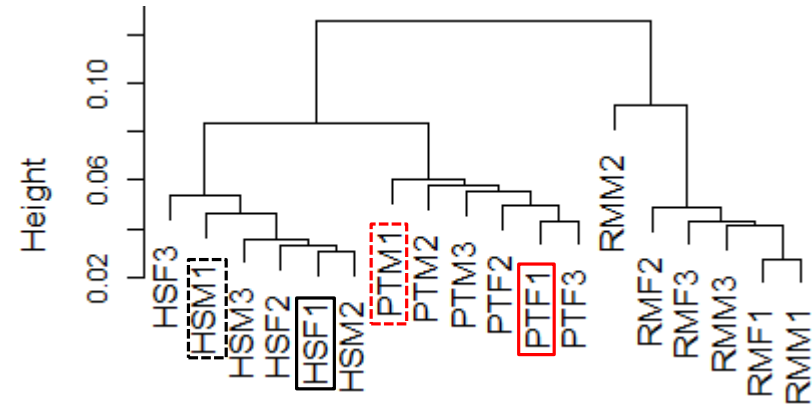
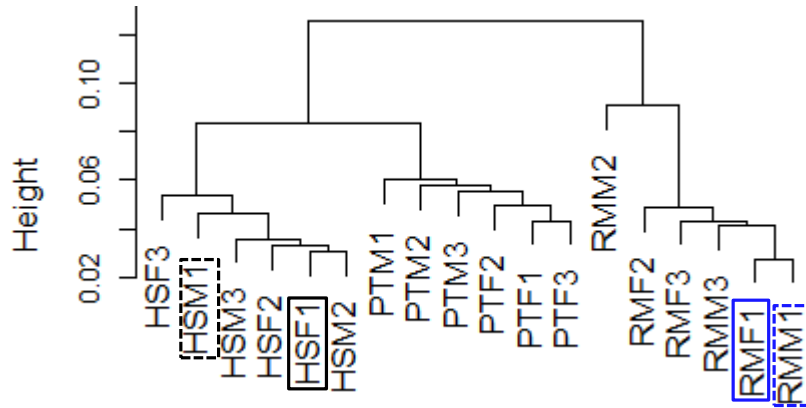
```
library(TCC)
```

#入力ファイル名を指定してin_fに格納
 #出力ファイル名を指定してout_f1に格納
 #出力ファイル名を指定してout_f2に格納
 #取り扱いたいサブセット情報を指定
 #G1群のサンプル数を指定
 #G2群のサンプル数を指定
 #DEG検出時のfalse discovery rate (FDR)
 #ファイル出力時の横幅と縦幅を指定(単位:ピクセル)
 #下、左、上、右の順で余白を指定(単位:ピクセル)

#パッケージの読み込み

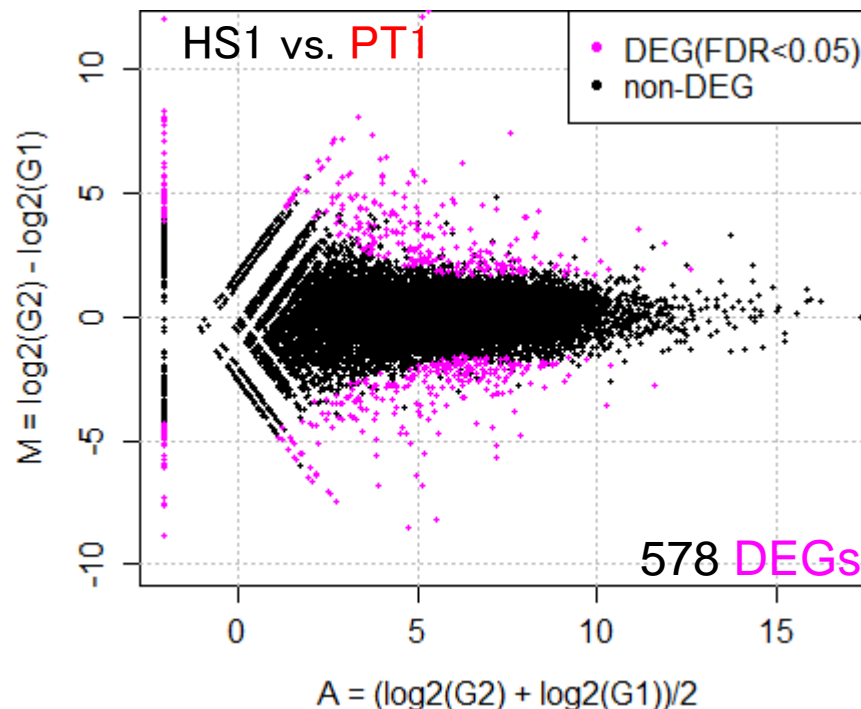
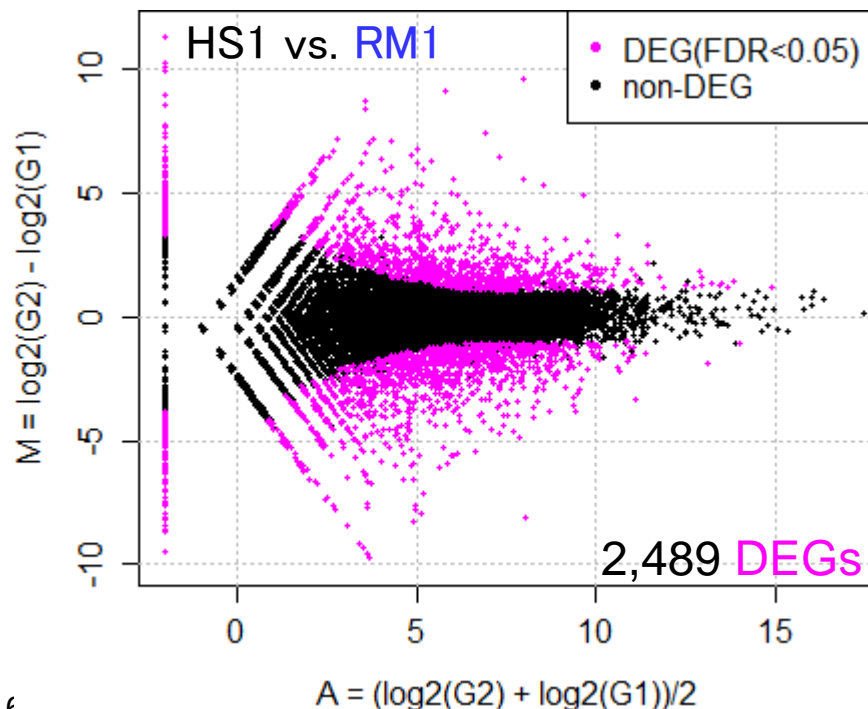
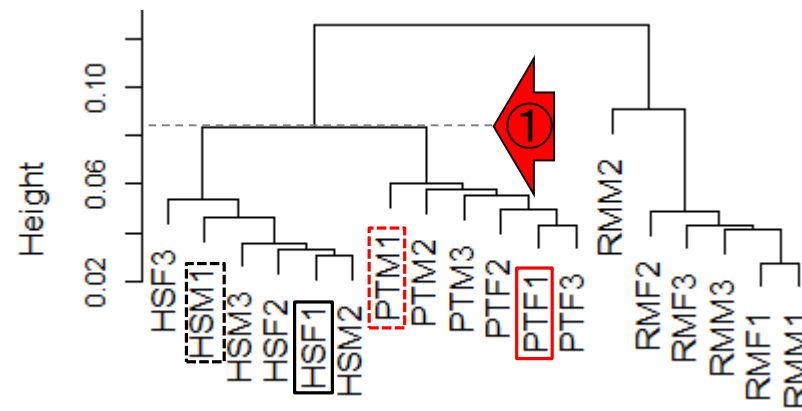
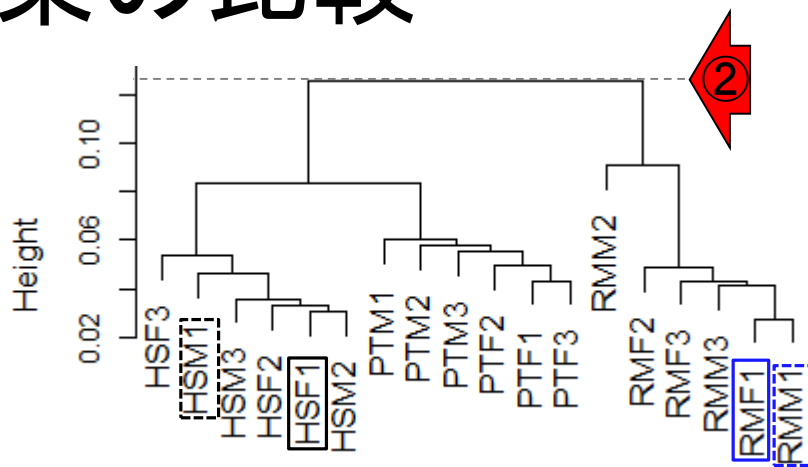
①「HS vs. PT」は、②「HS vs. RM」よりもDEGが少ない

結果の比較



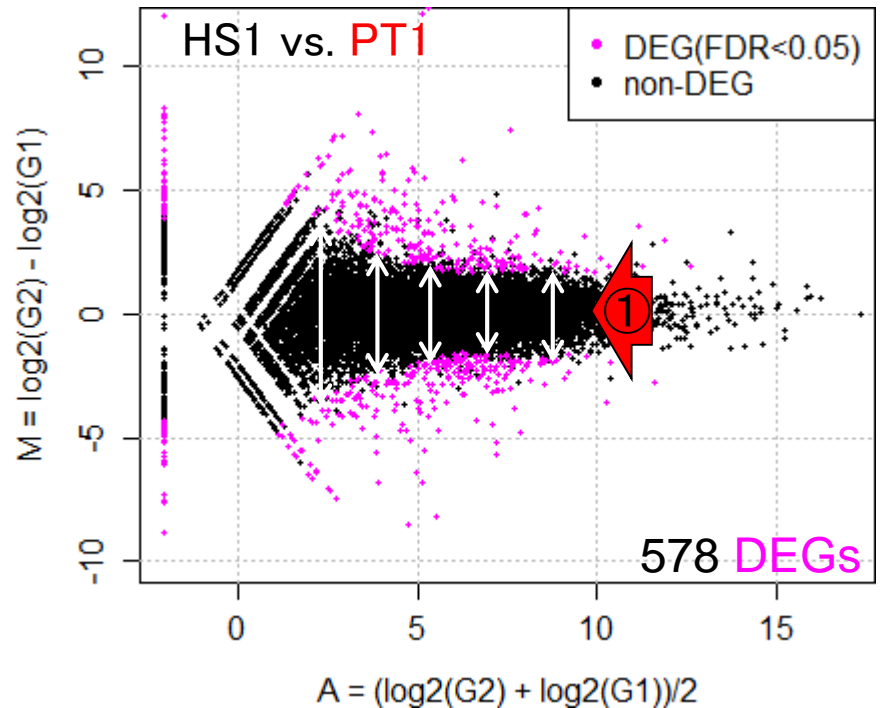
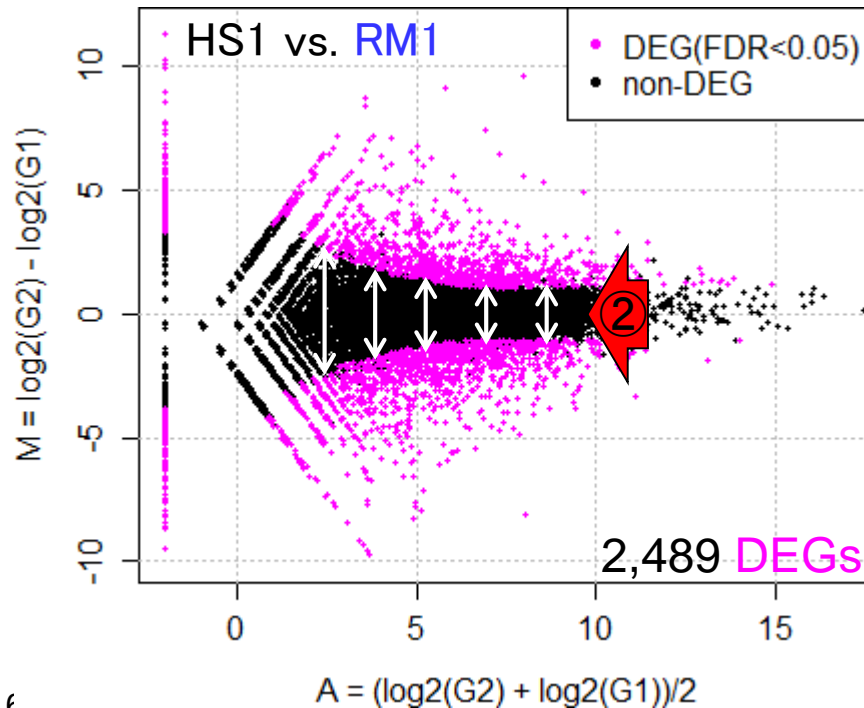
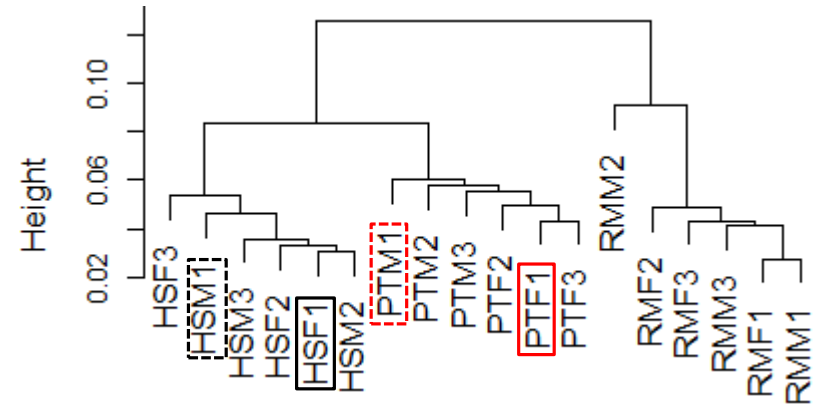
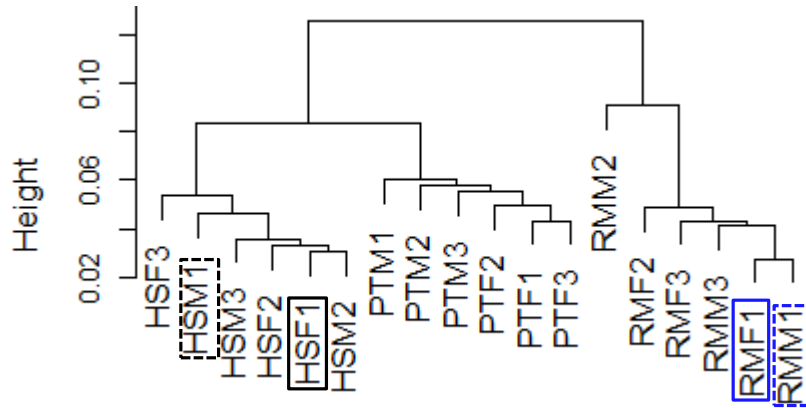
結果の比較

理由の1つは、②「HS vs. RM」に比べて、①「HS vs. PT」間の全体的な類似度が高い(距離が近い)ため



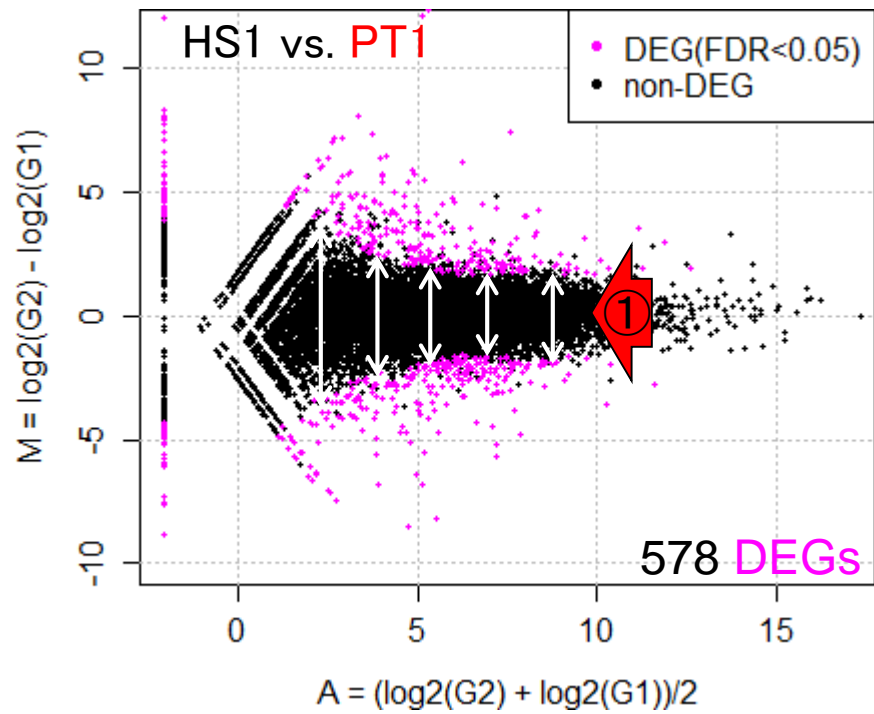
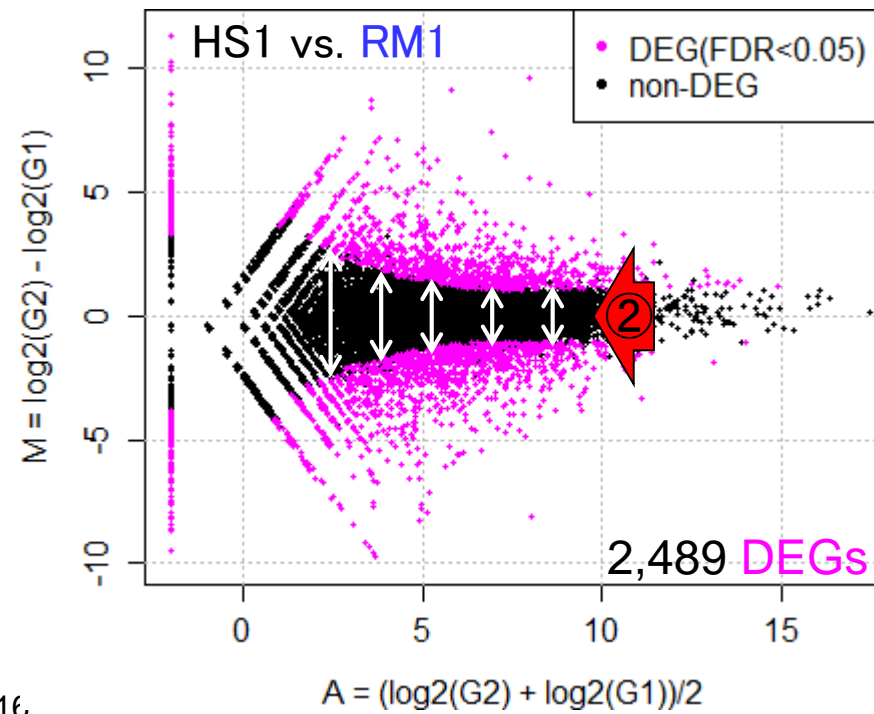
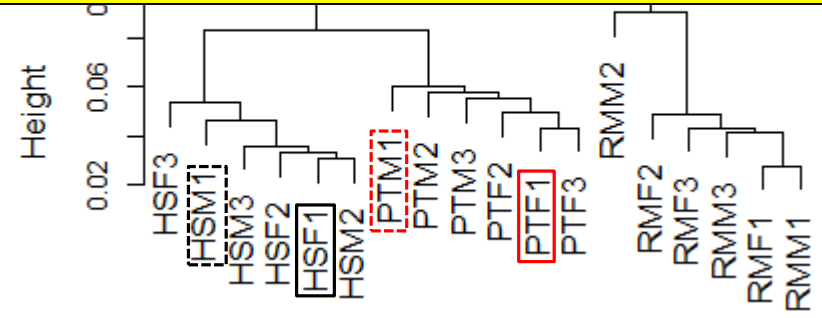
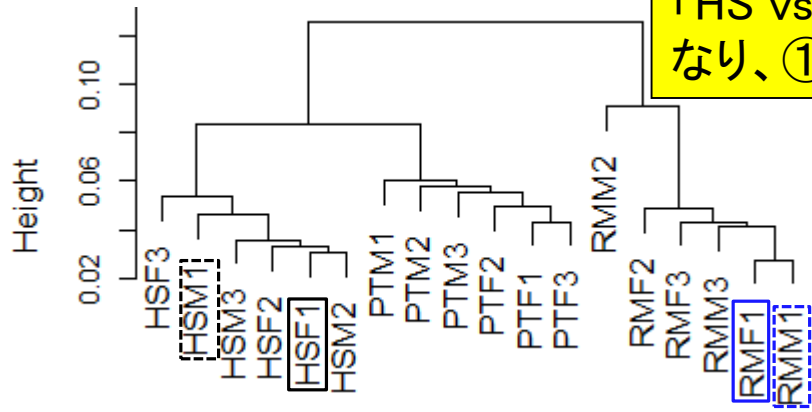
素朴な疑問

何故、白矢印で示すように①「HS vs. PT」のnon-DEGの分布(黒の点の分布)は、②「HS vs. RM」に比べて広がっているのか?



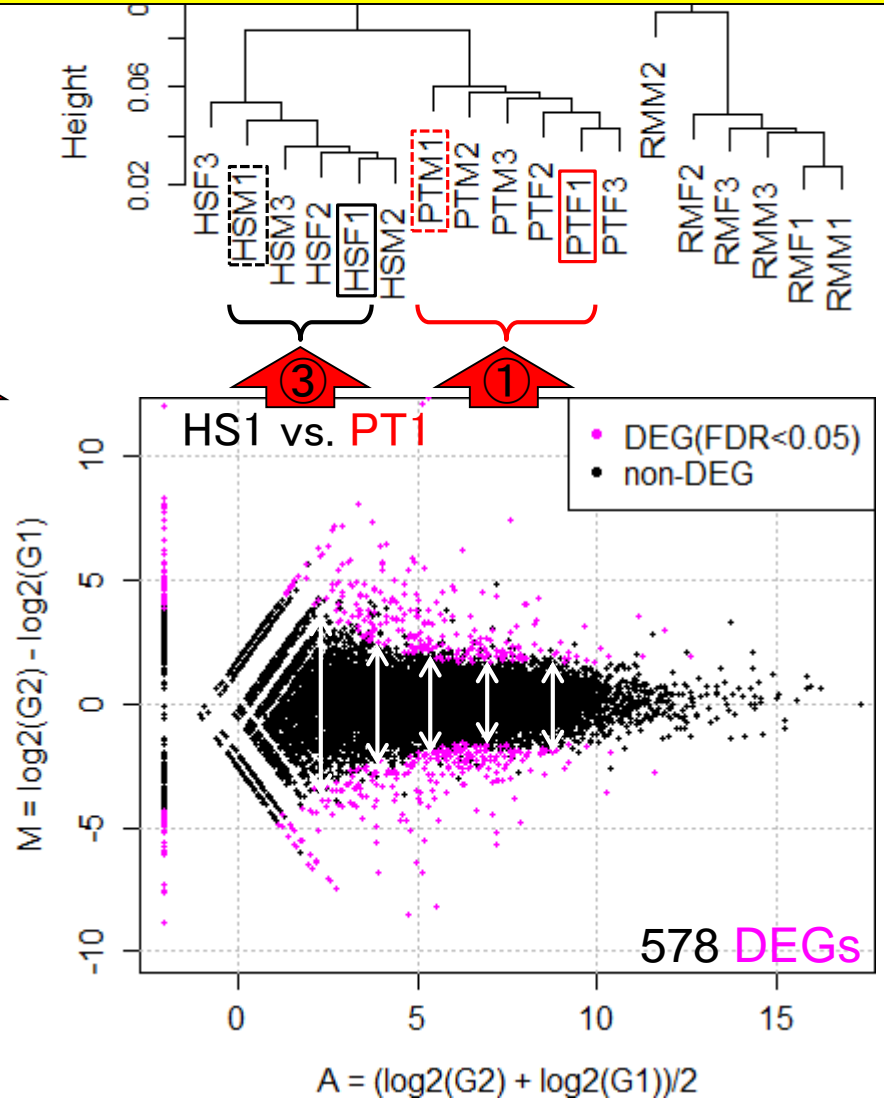
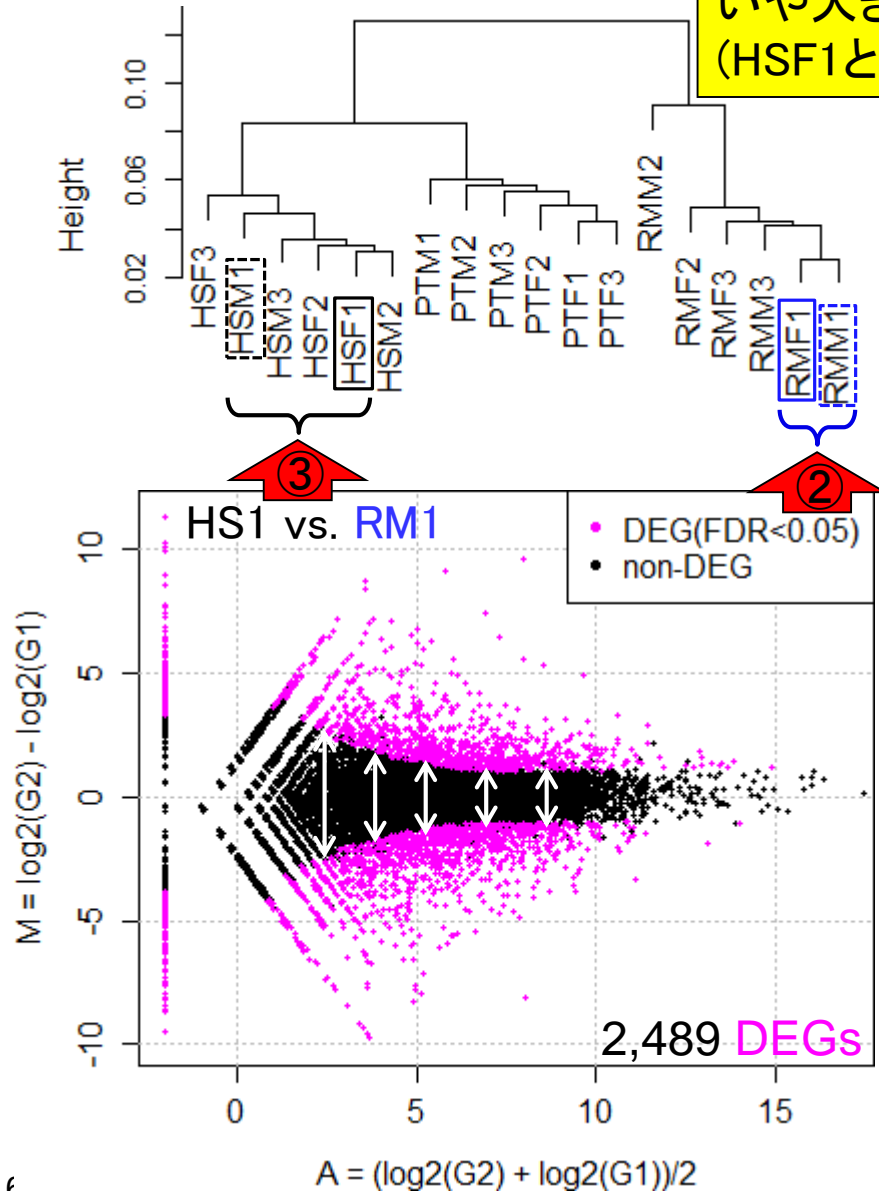
統計的手法とは

疑問に対する解答は、統計的手法の手順を再考すればよい。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、①「HS vs. **PT**」と②「HS vs. **RM**」とでは、non-DEG分布が異なり、①のほうが同一群内のばらつきが大きいということ



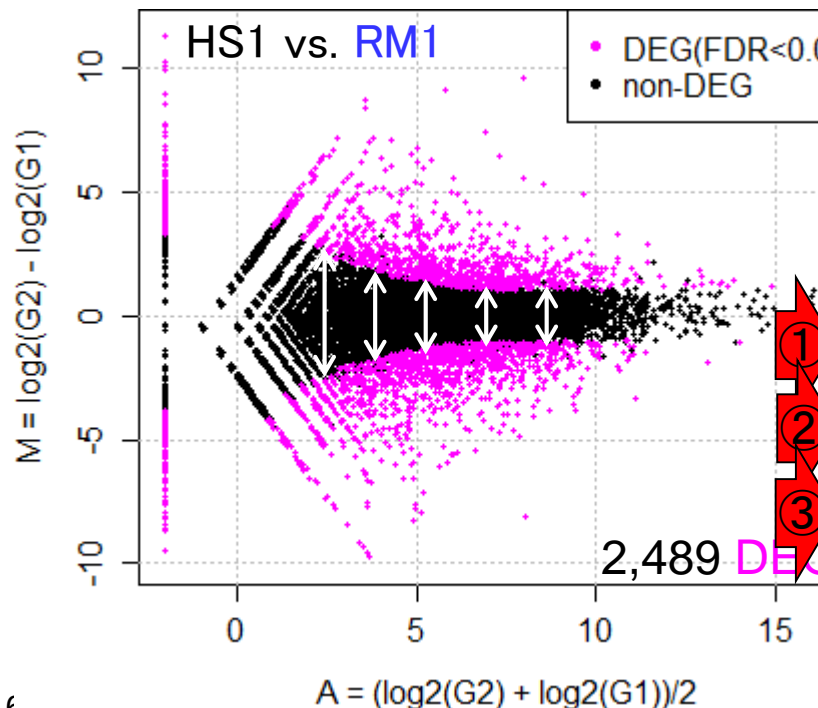
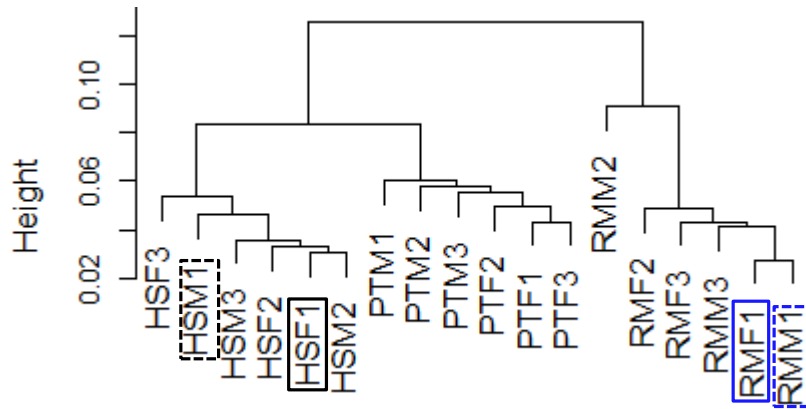
統計的手法とは

①PTの同一群内(PTF1とPTM1)は、②RMの同一群内(RMF1とRMM1)に比べて類似度が低い(距離が遠い or バラツキが大きい)。それがそのまま黒のnon-DEG分布の違いや大きさとなって表れている。尚、③HSの同一群内(HSF1とHSM1)のバラツキの分布(non-DEG分布)は同じ



サンプル間類似度RM

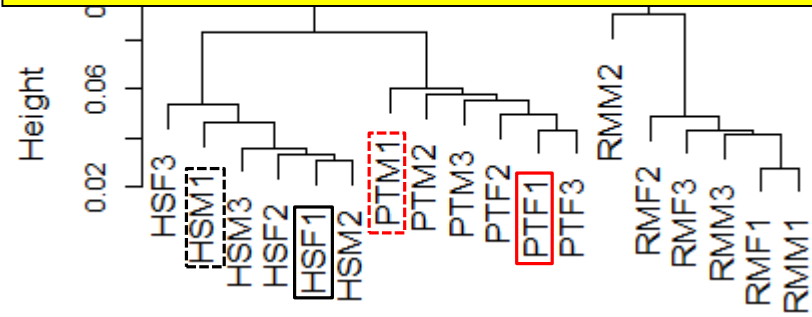
同一群内のばらつきは、サンプル間の類似度で大まかに把握可能。①HS群内(HSF1 vs. HSM1)のSpearman相関係数は0.950。②RM群内(RMF1 vs. RMM1)は0.972。③「HS vs. RM」の群間比較結果は、例えばHSM1 vs. RMM1の相関係数(0.880)が0.950と0.972よりも低いことからDEGの存在を予測可能



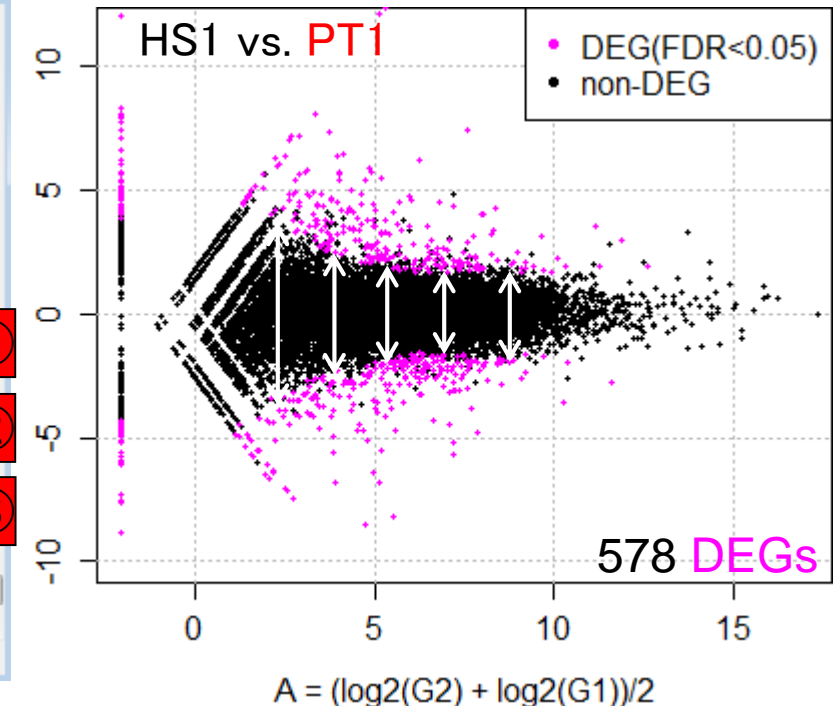
```
R Console
> in_f <- "sample_blekhman_18.txt" #入$
> data <- read.table(in_f, header=TRUE, row.n$
> dim(data)
[1] 20689 18
> data <- unique(data)
> dim(data)
[1] 16561 18
> cor(data$HSM1, data$HSF1, method="spearman")
[1] 0.9502333
> cor(data$RMM1, data$RMF1, method="spearman")
[1] 0.9724166
> cor(data$HSM1,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

サンプル間類似度PT

①HS群内(HSF1 vs. HSM1)のSpearman相関係数は0.950。②PT群内(PTF1 vs. PTM1)は0.949。「HS vs. PT」の群間比較結果は、例えば③HSM1 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



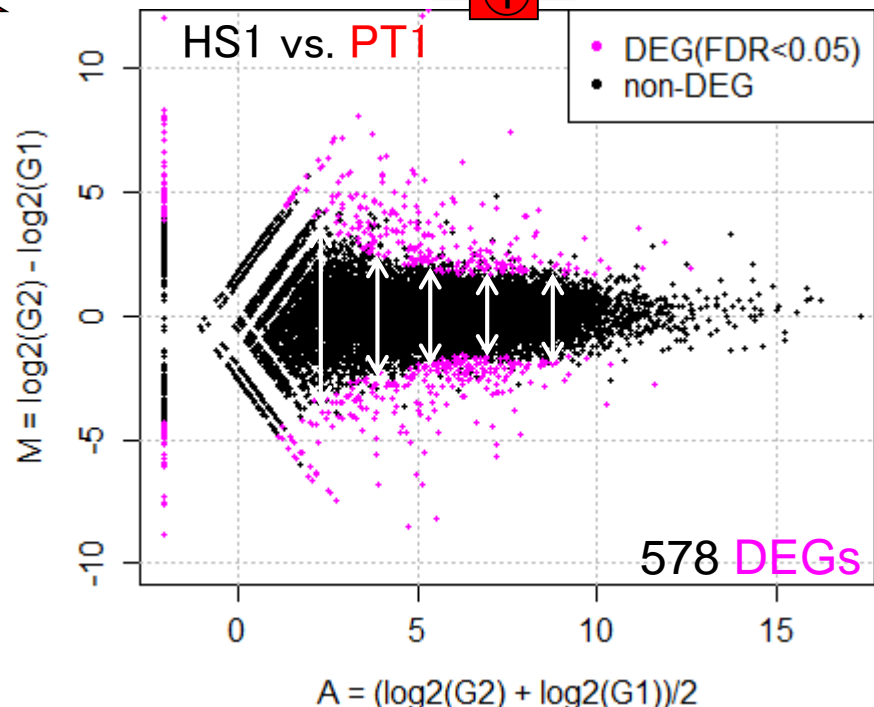
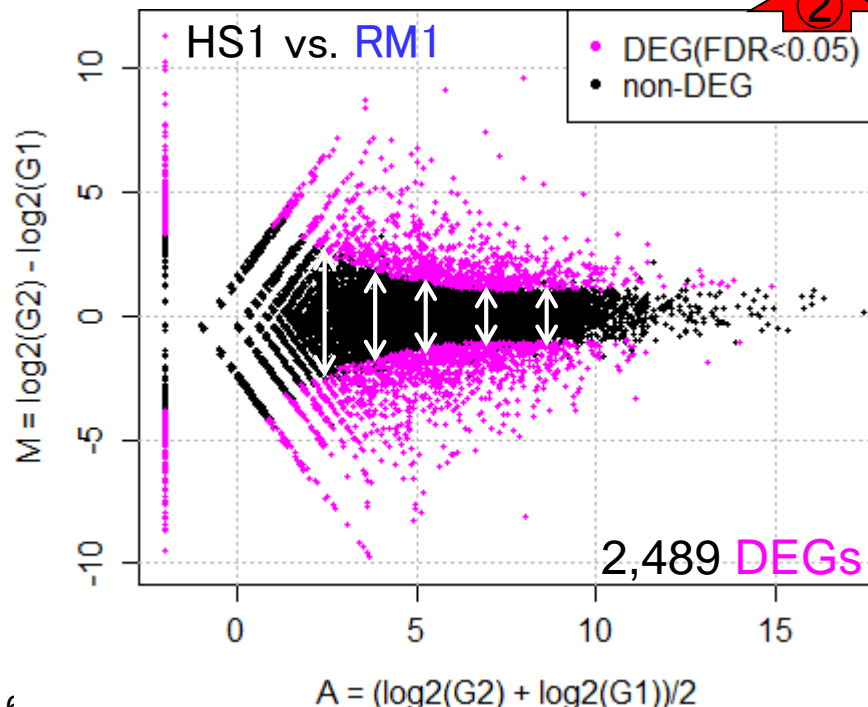
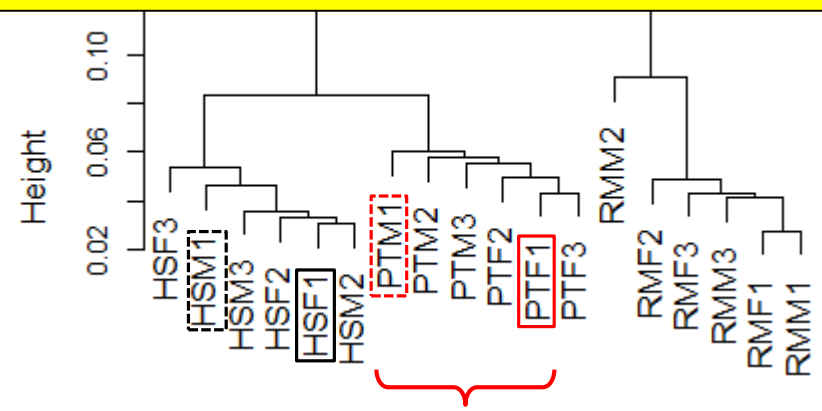
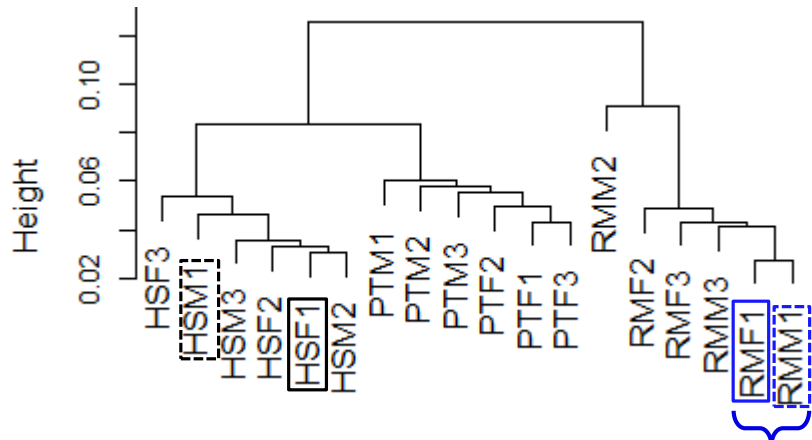
```
R Console
> in_f <- "sample_blekhman_18.txt" #入$
> data <- read.table(in_f, header=TRUE, row.n$
> dim(data)
[1] 20689 18
> data <- unique(data)
> dim(data)
[1] 16561 18
> cor(data$HSM1, data$HSF1, method="spearman")
[1] 0.9502333
> cor(data$PTM1, data$PTF1, method="spearman")
[1] 0.9489023
> cor(data$HSM1,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



参考

DEG検出結果の比較

①PT群内(PTF1 vs. PTM1)は0.949。一方、②RM群内(RMF1 vs. RMM1)のSpearman相関係数は0.972。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

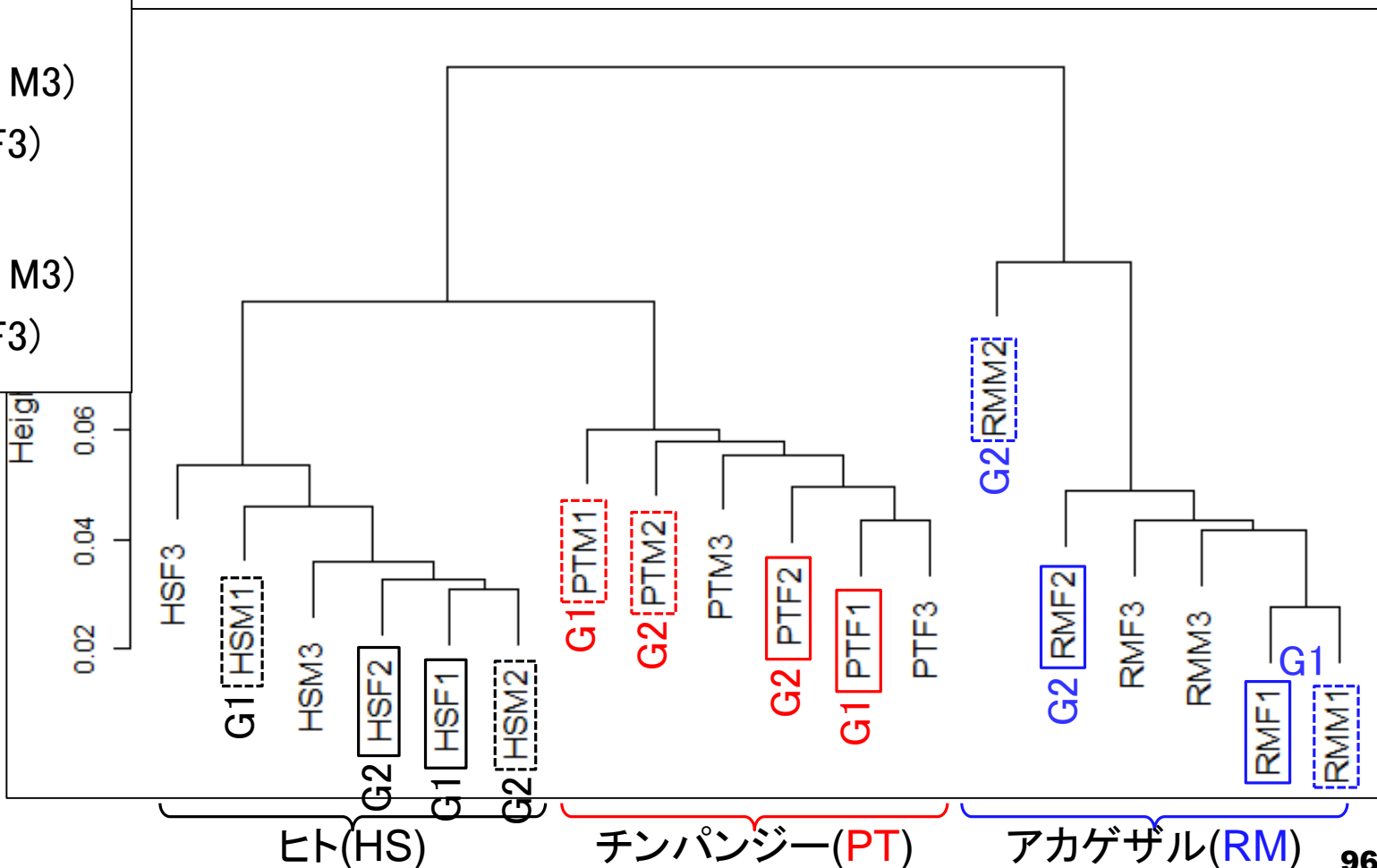
- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(DEGがほとんどない同一群の場合)
 - 倍率変化(2倍以上、1/2以下の発現変動)を用いた場合
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



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少数

2群間比較

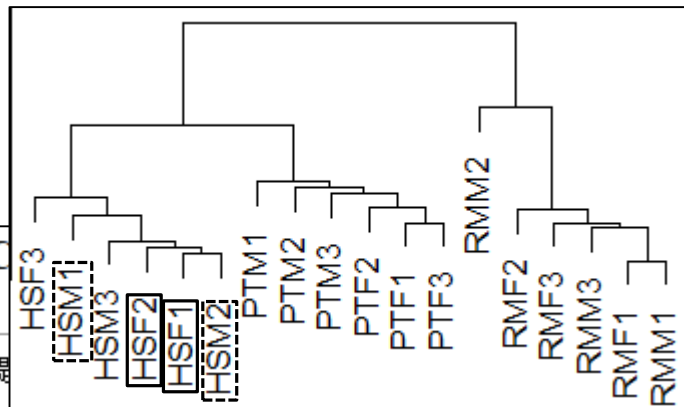
- ヒト(HS)
 - オス3匹 (M1, M2, M3)
 - メス3匹 (F1, F2, F3)
- チンパンジー(PT)
 - オス3匹 (M1, M2, M3)
 - メス3匹 (F1, F2, F3)
- アカゲザル(RM)
 - オス3匹 (M1, M2, M3)
 - メス3匹 (F1, F2, F3)



HS1 vs. HS2

- 解析 | 発現変動 | 1について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | 1について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love 2014) (last modified 2015/11/15)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun 2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li 2013) (last modified 2015/02/07)

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

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

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
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
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。

1. ヒト2サン

1, 4, 13, 16

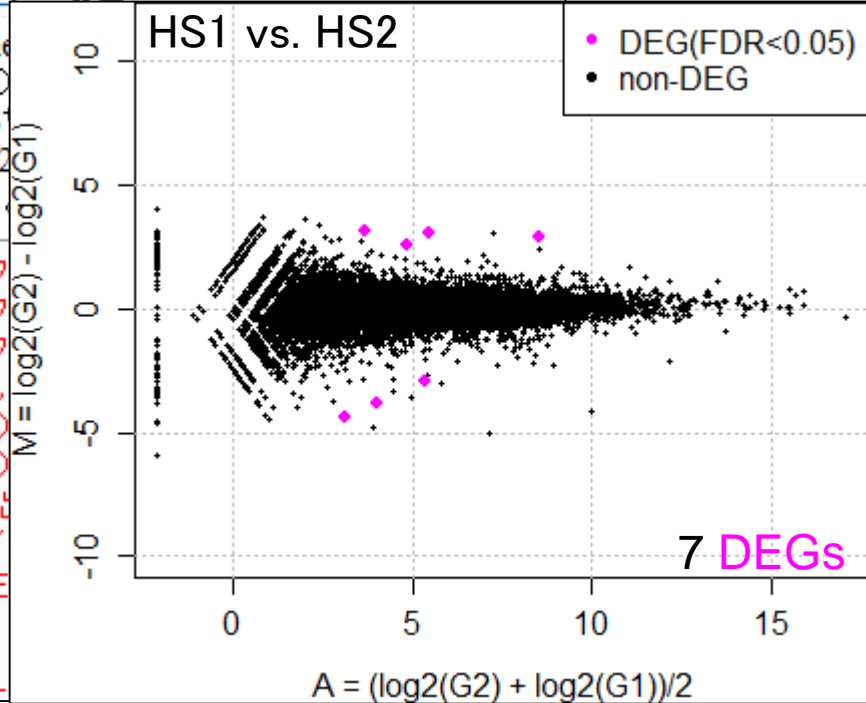
```
in_f <-
out_f1
out_f2
param_s
param_G
param_G
param_F
param_f
```

```
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
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

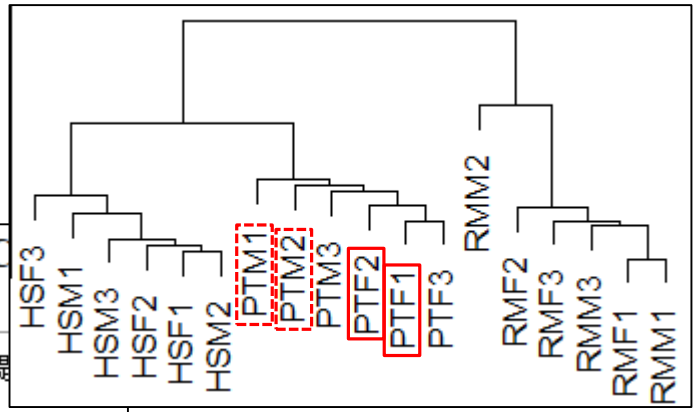
```
library(TCC)
```

#パッケ



PT1 vs. PT2

- 解析 | 発現変動 | 1について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | 1について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love 2014) (last modified 2015/11/15)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun 2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li 2013) (last modified 2015/02/07)



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があります。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

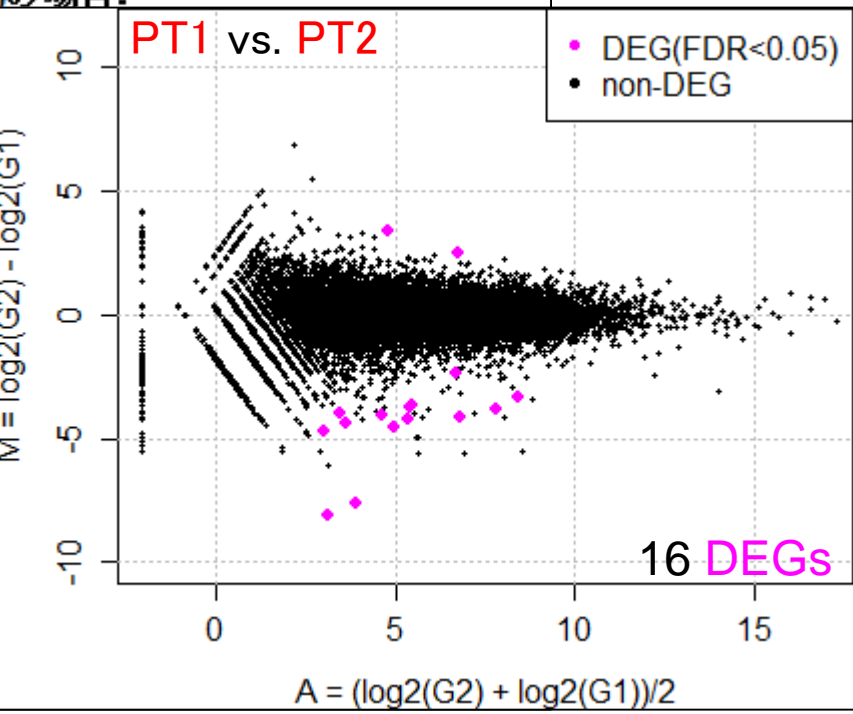
9. サンプルデータ42のリアルデータ(sample_blekhman_18.txt)の場合:

Blekhman et al. Genome Res., 2010の20,689 genes×18 sample サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーの 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)があります。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

```
1. ヒト2サン
1, 4, 13, 16
in_f <-
out_f1
out_f2
param_s
param_G
param_G
param_G
param_F
param_f
```

```
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
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
```

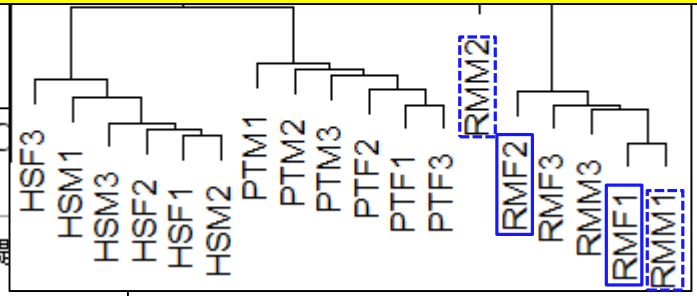


RM1 vs. RM2

②例題10。「G1群(RMF1とRMF1) vs. G2群(RMM2とRMF2)」の2群間比較結果。22 DEGs。G1群(RMF1とRMF1)内の類似度は高いが、G2群(RMM2とRMF2)内の類似度が低いので少なめのDEG数になったのだろうと解釈する

- 解析 | 発現変動 | 1について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | 1について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love 2014) (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun_2013) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun_2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li 2013) (last modified 2015/02/07)

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があります。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

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

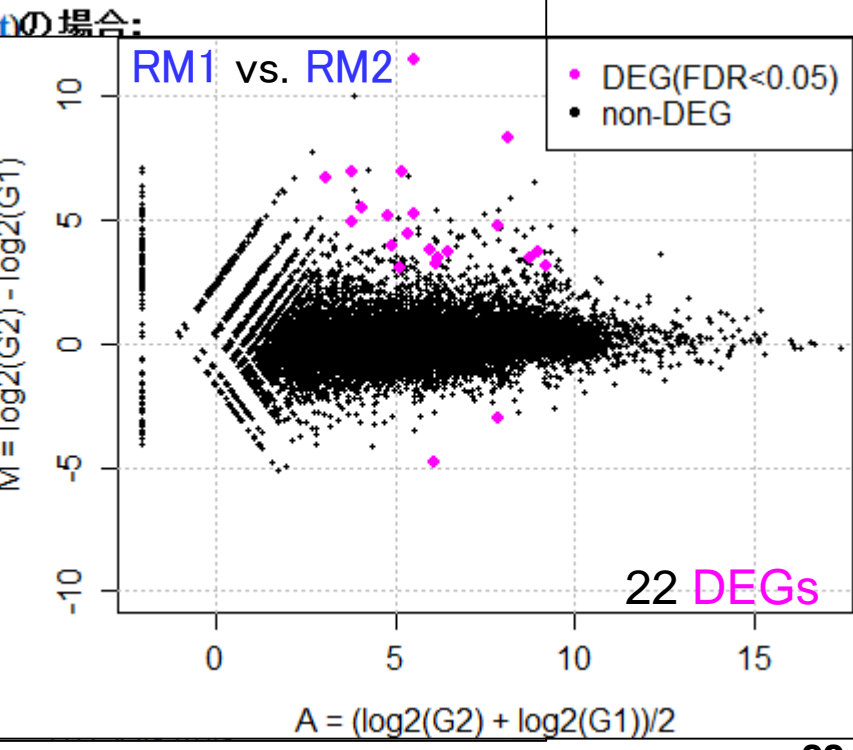
Blekhman et al., Genome Res., 2010の20,689 genes×18 sample サンプル(HSF1-3)とオス3サンプル(HSM1-3), チンパンジーのサンプル(PTM1-3), アカゲザルのメス3サンプル(RMF1-3)とオス3サンプル(RMM1-3)があります。ここでは、13, 16, 14, 17列目のデータのみ抽出して、G1群(RMF1とRMF1) vs. アカゲザル2サンプル(G2群:RMF2とRMM2)の2群間比較結果

1. ヒト2サンプル
1, 4, 13, 16

```

in_f <- "sample_blekhman_18.txt" #入力ファイル
out_f1 <- "hoge10.txt" #出力ファイル
out_f2 <- "hoge10.png" #出力ファイル
param_subset <- c(13, 16, 14, 17) #取り扱ったサンプルID
param_G1 <- 2 #G1群のサンプル数
param_G2 <- 2 #G2群のサンプル数
param_FDR <- 0.05 #DEG検出率
param_fig <- c(430, 350) #ファイギュアサイズ
param_mar <- c(4, 4, 0, 0) #下、左マージン
    
```

#必要なパッケージをロード
library(TCC) #パッケージ

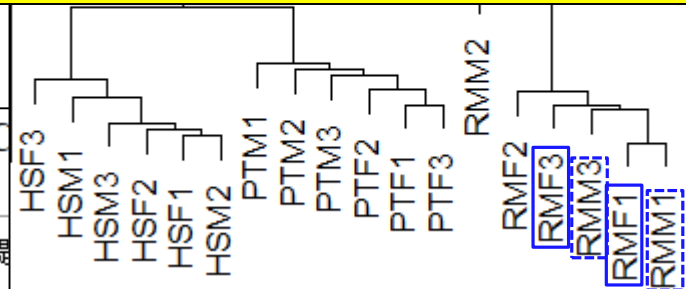


RM1 vs. RM3

②例題11。「G1群(RMM1とRMF1) vs. G2群(RMM3とRMF3)」の2群間比較結果。202 DEGs。G1群(RMM1とRMF1)内の類似度が高く、G2群(RMM3とRMF3)内の類似度もそこそこ高いのでDEG数が増えたと解釈すればよい

- 解析 | 発現変動 | 1について (last modified 2014/07/10)
- 解析 | 発現変動 | 2群間 | 対応なし | 1について (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | DESeq2(Love 2014) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | Blekhmanデータ | TCC(Sun 2013) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | SAMseq(Li 2013) (last modified 2015/02/07)

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があります。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

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)があります。ここでは、13, 16, 15, 18列目のデータのみ抽出して、ヒトとアカゲザル2サンプル(G1群:RMM1とRMF1) vs. アカゲザル2サンプル(G2群:RMF3とRMM3)の2群間比較結果をプロットしています。

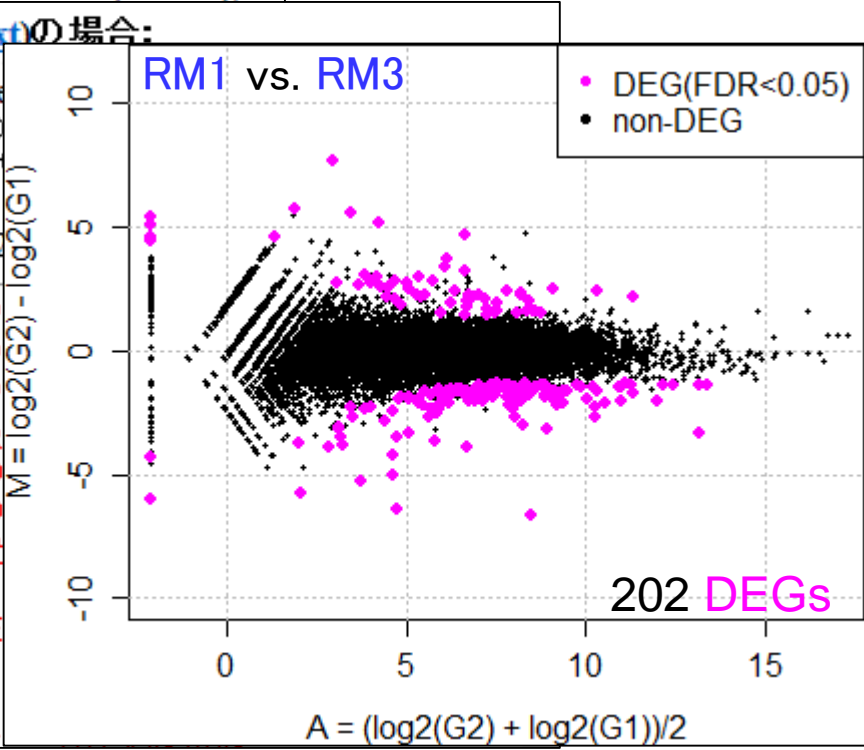
1. ヒト2サンプル

1, 4, 13, 16

```
in_f <- "sample_blekhman_18.txt"
out_f1 <- "hoge11.txt"
out_f2 <- "hoge11.png"
param_subset <- c(13, 16, 15, 18)
param_G1 <- 2
param_G2 <- 2
param_FDR <- 0.05
param_fig <- c(430, 350)
param_mar <- c(4, 4, 0, 0)
```

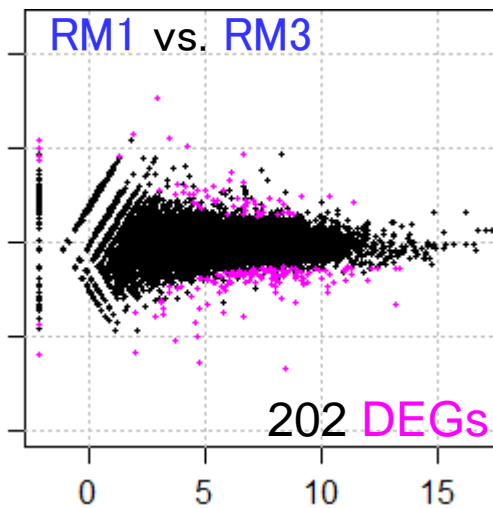
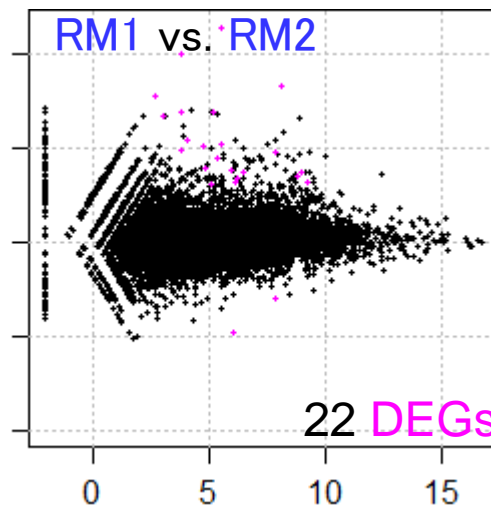
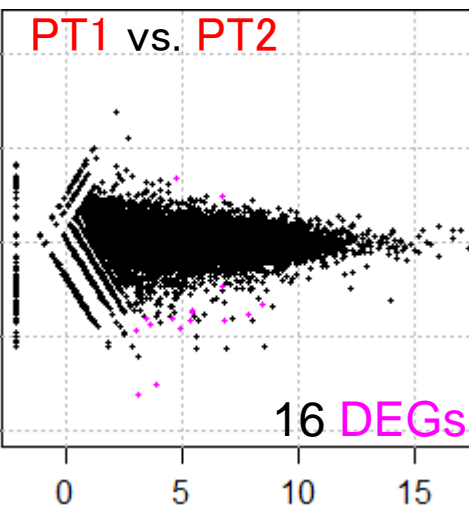
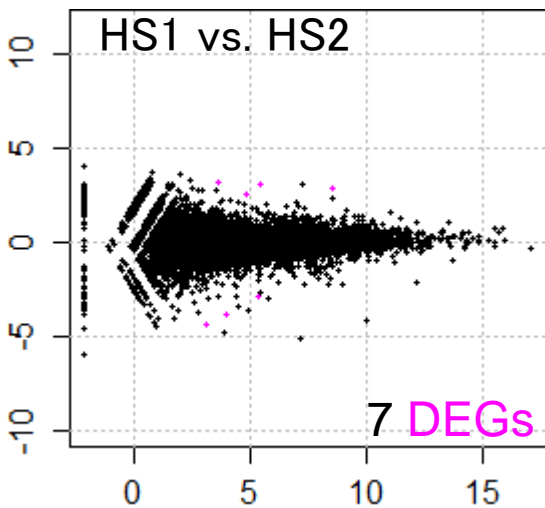
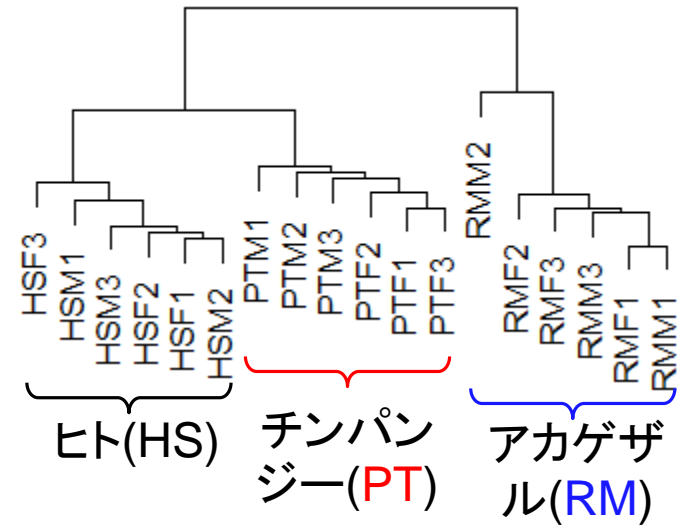
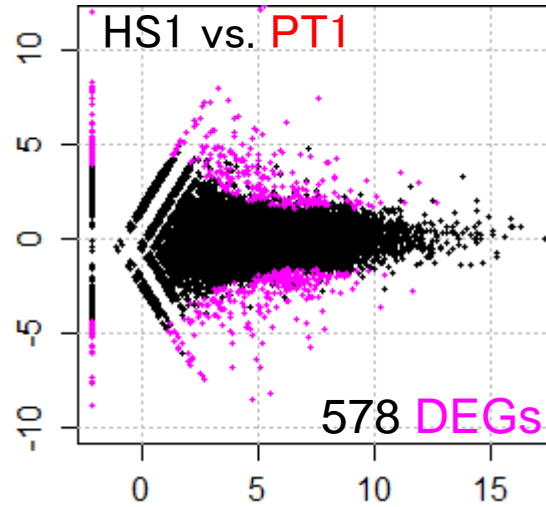
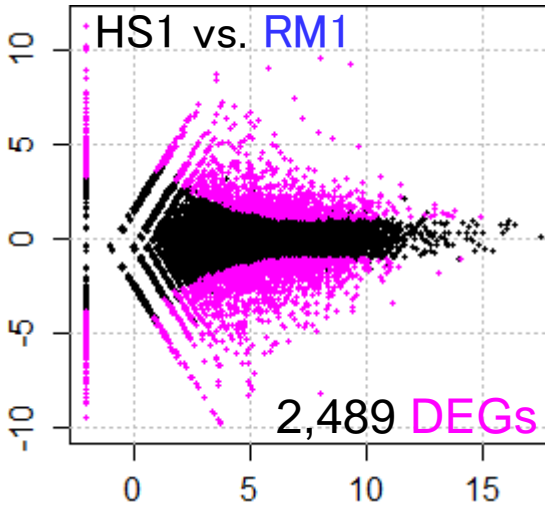
```
in_f <- "sample_blekhman_18.txt" #入力ファイル
out_f1 <- "hoge11.txt" #出力ファイル
out_f2 <- "hoge11.png" #出力ファイル
param_subset <- c(13, 16, 15, 18) #取り扱うサンプルID
param_G1 <- 2 #G1群のサンプル数
param_G2 <- 2 #G2群のサンプル数
param_FDR <- 0.05 #DEG検出率
param_fig <- c(430, 350) #ファイギュアサイズ
param_mar <- c(4, 4, 0, 0) #下、左
```

#必要なパッケージをロード
library(TCC) #パッケージ



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



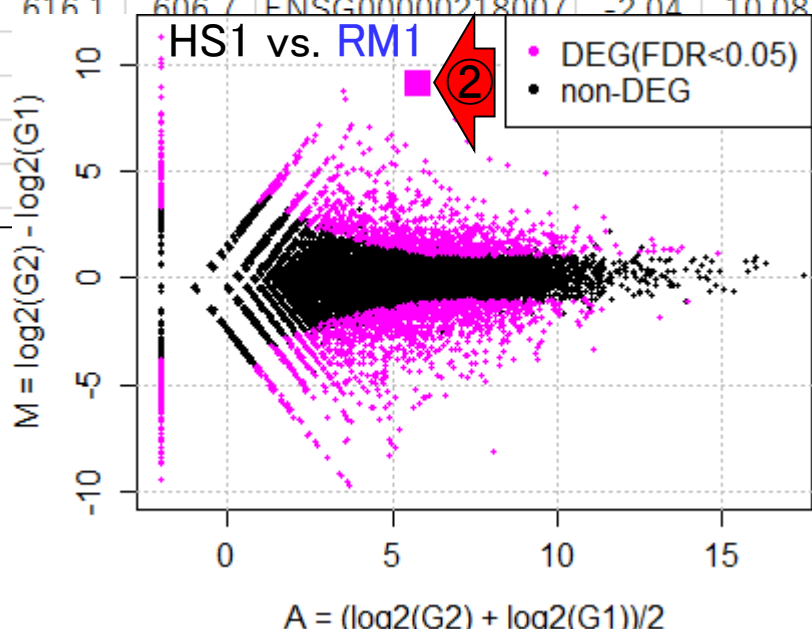
統計的手法とは



①の例題3実行結果の、②第2位で説明。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	1346.7	1476.9	ENSG00000208570	-2.04	11.29	4.5E-53	9.21E-49	1	1
ENSG00000220191	2.3	2.5	1394.7	1171.0	ENSG00000220191	5.80	9.06	1.2E-47	4.74E-43	2	1
ENSG00000106366	4422.0	4411.6	23.1	8.3	ENSG00000106366	8.04	-8.14	2.7E-45	1.67E-41	3	1
ENSG00000209449	0.0	0.0	644.5	713.1	ENSG00000209449	-2.04	10.23	3.32E-44	1.72E-40	4	1
ENSG00000218007	0.0	0.0	616.1	606.7	ENSG00000218007	-2.04	10.08	1.77E-43	7.32E-40	5	1
ENSG00000070985	0.0	0.0						4.70E-42	1.62E-38	6	1
ENSG00000209007	0.0	0.0						1.25E-40	3.70E-37	7	1
ENSG00000182327	367.5	363.9						1.53E-38	3.97E-35	8	1
ENSG00000156222	367.5	301.6						1.05E-36	2.42E-33	9	1
ENSG00000165272	404.8	420.0						5.50E-36	1.14E-33	10	1

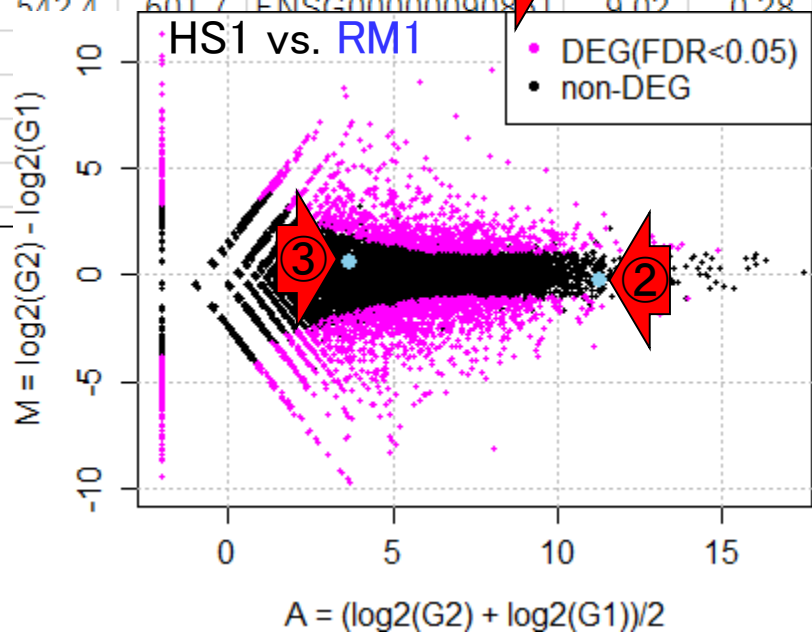


①の例題12実行結果の、q-valueが1に近い②と③で説明。同一群内のばらつきの分布(non-DEG分布)のど真ん中に位置するものは、1に近いp-value

統計的手法とは

- 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	131.1	98.5	148.8	ENSG00000047578	6.80	0.31	0.49066	1	9727	0
ENSG00000122257	256.7	234.1	253.9	334.1	ENSG00000122257	8.07	0.26	0.49094	1	9728	0
ENSG00000125844	2299.7	3137.8	2113.7	2429.3	ENSG00000125844	11.28	-0.26	0.49101	1	9729	0
ENSG00000115325	3.4	17.8	16.9	14.1	ENSG00000115325	3.68	0.55	0.49102	1	9730	0
ENSG00000090861	603.8	339.7	512.4	601.7	ENSG00000090861	9.02	0.28	0.49114	1	9731	0
ENSG00000032389	53.1	36.9						0.49122	1	9732	0
ENSG00000180190	52.0	28.0						0.49136	1	9733	0
ENSG00000109686	451.1	792.7						0.49141	1	9734	0
ENSG00000100351	2.3	12.7						0.49143	1	9735	0
ENSG00000169567	504.7	112.0						0.49146	1	9736	0



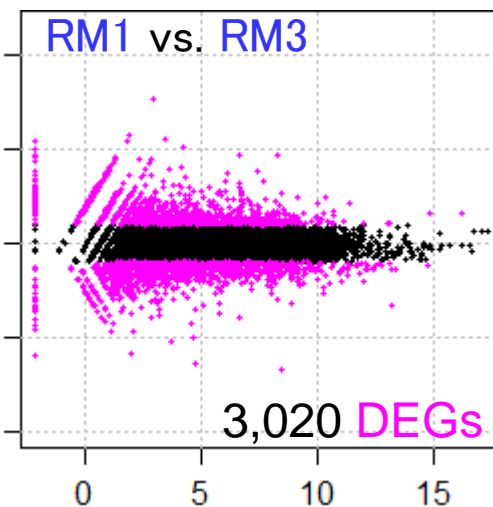
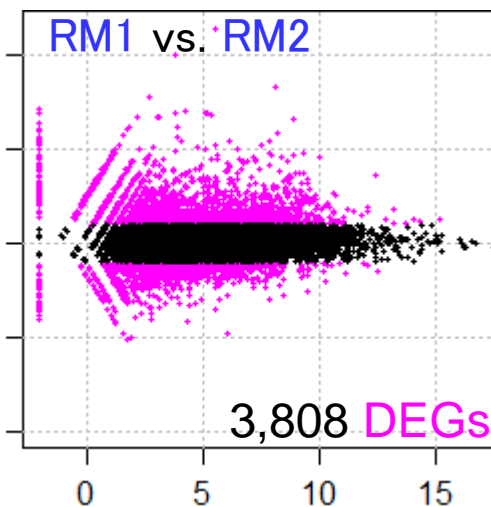
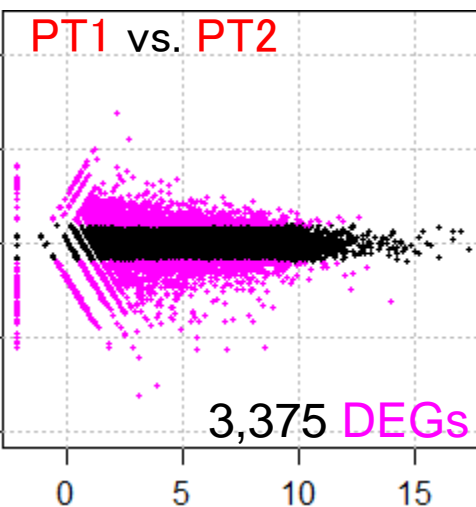
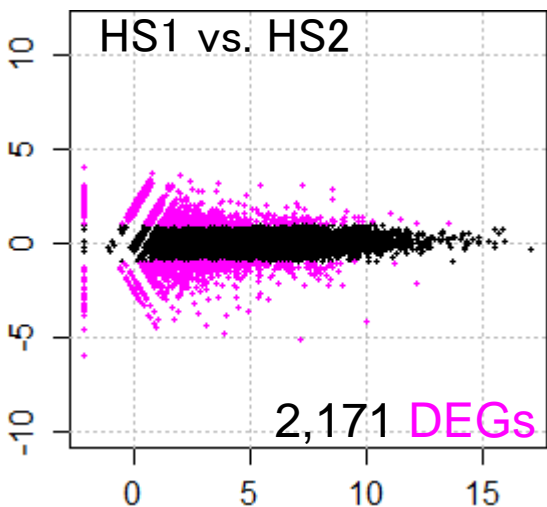
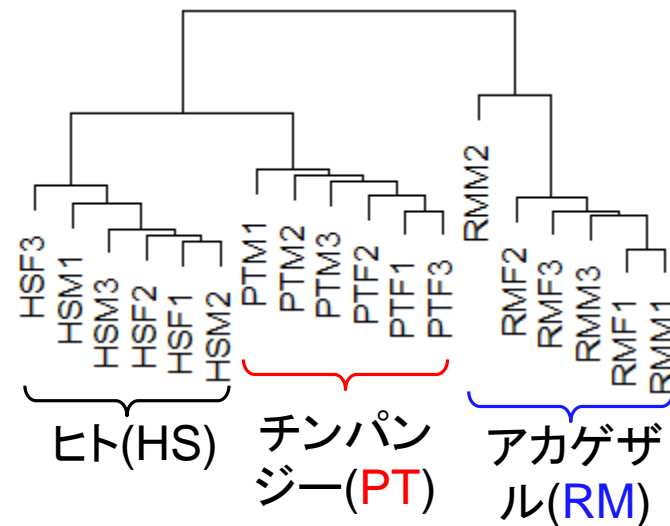
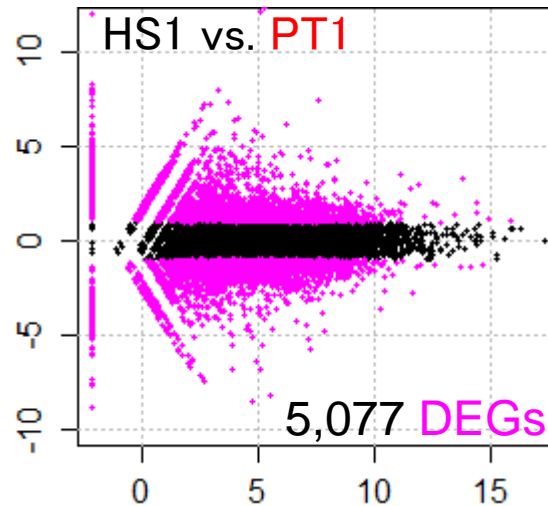
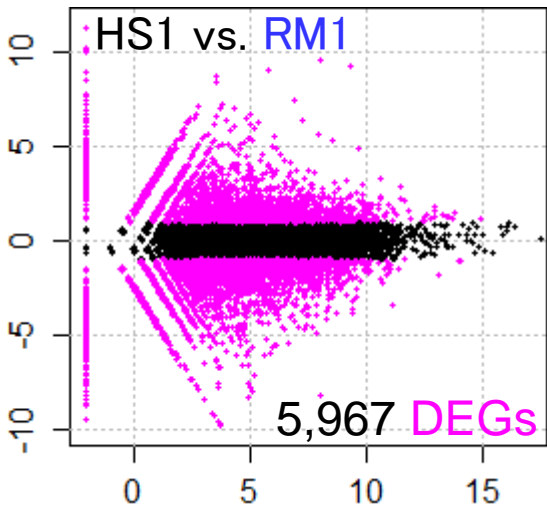
Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



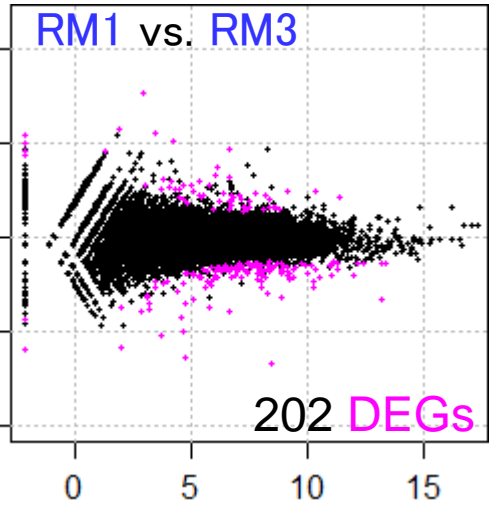
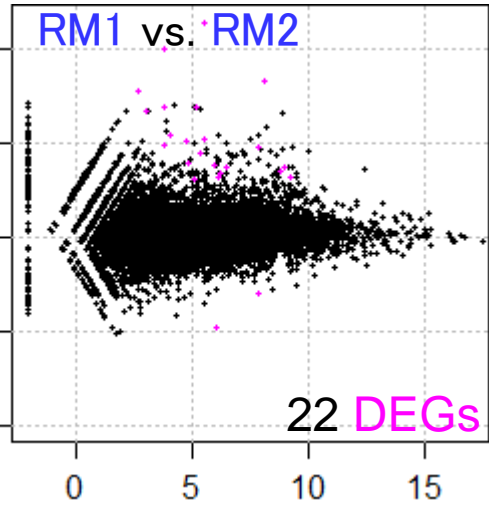
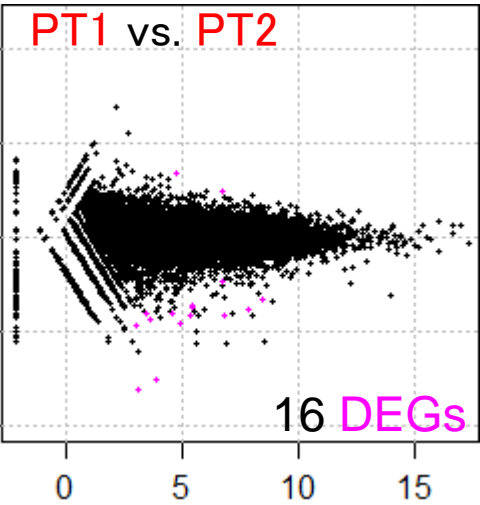
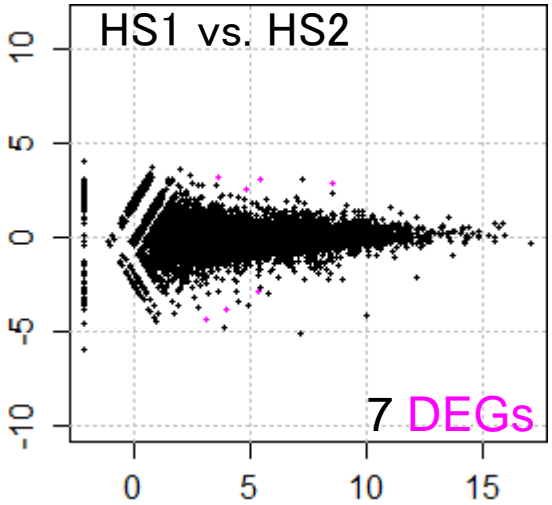
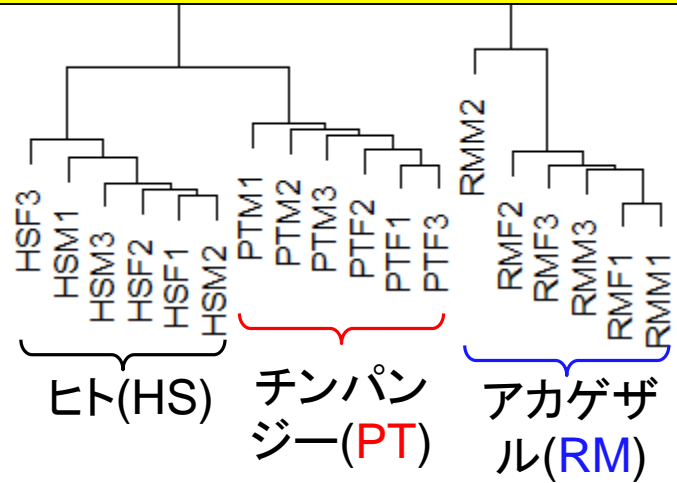
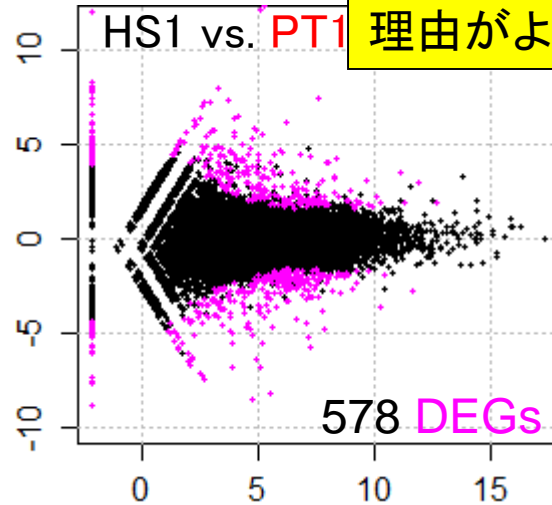
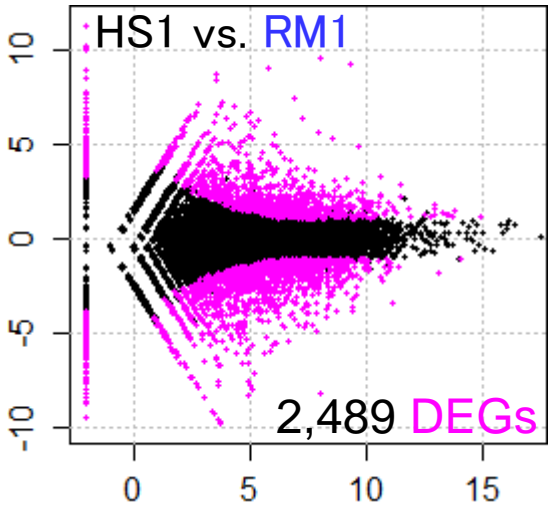
結果の比較(倍率変化^①)

倍率変化(fold-change; FC)でのDEG検出結果。下段の同一群内比較でも多数の偽陽性が検出されている。①の例題13をベースに作成。DEG数はヒトによって若干異なるかも…



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)

結果の比較(FDR)



Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



giladデータでDEG同定

①Desktop - hogeフォルダ中の、②giladカウントデータファイルを入力とします。③サンプルラベル情報ファイルを予め眺めておくことで、最初の3サンプルがメス(F)で、残りの3サンプルがオス(M)だということを把握

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	
cheung	20856902	human	41	834,584,950	link	link	link	
core	19056941	human	2	8,670,342	link	link	link	
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and femlaes
maq	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled			
montgomery	20220756	human	60	*886,468,054	link	link	link	
pickrell	20220758	human	69	*886,468,054	link	link	link	
sultan	18599741	human	4	6,573,643	link	link	link	

```
sample.id num.tech.reps gender
SRX014818and9 2 F
SRX014820and1 2 F
SRX014822and3 2 F
SRX014824and5 2 M
SRX014826and7 2 M
SRX014828and9 2 M
```

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="gilad")
[1] "gilad_count_table.txt"
[2] "gilad_phenodata.txt"
> |
```

giladデータでDEG同定

コピー。①FDR 30% (許容する偽物混入率が30%という閾値)でも、0個。念のため、②もう少し緩めの閾値も眺め、このデータセットにはDEGはないとの確定診断を下す

• giladデータでDEG同定

ReCountのgiladデータ([gilad_count_table.txt](#))で反復あり2群間比較を行う。「解析 | 実行変動 | 2群間 | 対応なし | 複製あり | [TCC\(Sun 2013\)](#)」例題1をベースに作成。M-A plot周辺はオプションなどを多少変更しています。

```
in_f <- "gilad_count_table.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_G1 <- 3 #G1群のサンプル数を指定
param_G2 <- 3 #G2群のサンプル数を指定
param_FDR <- 0.05 #DEG検出時のfalse discovery rateを指定
param_fig <- c(430, 350) #ファイル出力時の横縦サイズを指定
```

```
#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")
```

```
#前処理(TCCクラスオブジェクトの作成)
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2で指定
tcc <- new("TCC", data, data.cl) #TCCクラスオブジェクトの作成
```

```
#本番(正規化)
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="tmm",
                      iteration=3, FDR=0.1, floorPDEG=0.001)
```

```
R Console
> dev.off() $
null device
1
> sum(tcc$stat$q.value < 0.05) $
[1] 0
> sum(tcc$stat$q.value < 0.10) $
[1] 0
> sum(tcc$stat$q.value < 0.20) $
[1] 0
> sum(tcc$stat$q.value < 0.30) ① $
[1] 0
> sum(tcc$stat$q.value < 0.50) ② $
[1] 0
> sum(tcc$stat$q.value < 0.70) $
[1] 1
> |
```

giladデータでDEG同定

①M-A plot。デフォルトは②FDR = 0.05。確かにDEGのプロットはない。③以前眺めたサンプル間クラスタリング結果でもFemaleとMaleが混在しており妥当

• giladデータでDEG同定

ReCountのgiladデータ(gilad_count_table.txt)で反復あり2群間比較を行う。「解析 | 発現変動 | 2群間 | 対応なし | 複製あり | TCC(Sun 2013)」例題1をベースに作成。M-A plot周辺はオプションなどを多少変更しています。

```
in_f <- "gilad_count_table.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_G1 <- 3
param_G2 <- 3
param_FDR <- 0.05
param_fig <-
```

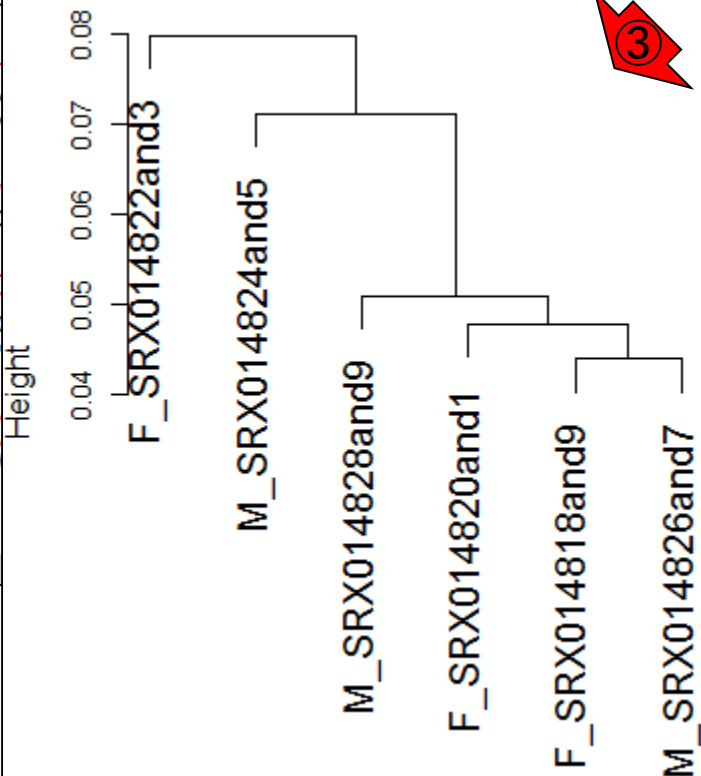
```
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)
#M-A plotの横幅と縦幅を指定(単位はcm)
```

```
#必要なパッケージを読み込み
library(TCC)

#入力ファイルを読み込み
data <- read.csv(in_f)

#前処理(TCC)
data.cl <- cluster(data)
tcc <- new(tcc)
tcc$clusters <- data.cl

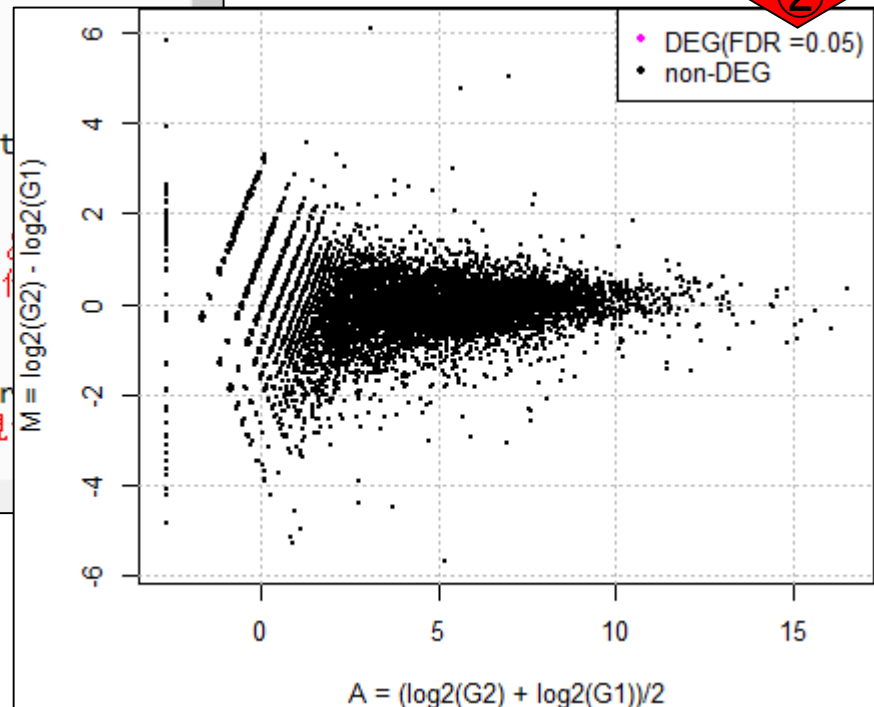
#本番(正規化)
tcc <- calc(tcc)
```



```
read.csv(in_f, sep="\t", quote="\"", as.is=T)

#群を1、G2群を2としてプロジェクトtccを作成
tcc <- new(tcc, G1=param_G1, G2=param_G2)

#M-A plotの生成
tcc$M.A <- M.A(tcc, param_FDR, param_fig, method="edger", DEG=0.05) #正規化済み
```



giladデータでDEG同定

①コード下部。②FDR = 0.05になるようにひっそりと修正しています。③これはTCCパッケージで提供しているplot関数ですが、デフォルトのplot関数でなくても、main, xlab, ylabなどのオプションは変更可能だという例

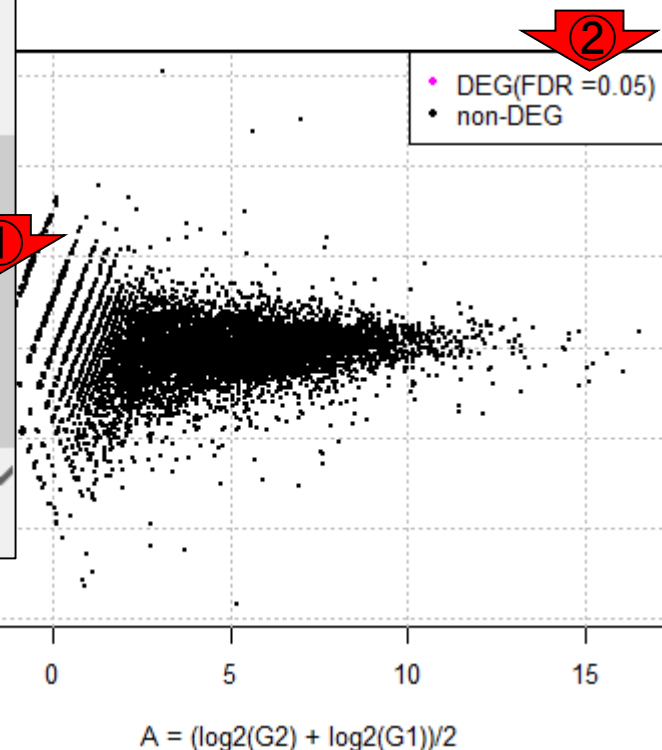
giladデータでDEG同定

ReCountのgiladデータ([gilad_count_table.txt](#))で反復あり2群間比較を行う。「解析」タブに対応なし | 複製あり | [TCC\(Sun 2013\)](#) | 例題1をベースに作成。M-A plot周辺はオプションを変更しています。

```
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を満たす遺伝子数を表示

#ファイルに保存(テキストファイル)
tmp <- cbind(rownames(tcc$count), tcc$count, result)#入力データの右側にDEG検出結果
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中身

#ファイルに保存(M-A plot)
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイル
par(mar=c(4, 4, 0, 0)) #余白を指定
plot(tcc, FDR=param_FDR, main="", #param_FDRで指定した閾値を満たすDEGをマゼンタ色にして描画
      xlab="A = (log2(G2) + log2(G1))/2", #閾値を満たすDEGをマゼンタ色にして描画
      ylab="M = log2(G2) - log2(G1)") #閾値を満たすDEGをマゼンタ色にして描画
legend("topright", c(paste("DEG(FDR =", param_FDR, ")"), "non-DEG"),
      col=c("magenta", "black"), pch=20)#凡例を作成
dev.off() #おまじない
sum(tcc$stat$q.value < 0.05) #FDR = 0.05 (q-value < 0.05)を満たす遺伝子数
sum(tcc$stat$q.value < 0.10) #FDR = 0.10 (q-value < 0.10)を満たす遺伝子数
sum(tcc$stat$q.value < 0.20) #FDR = 0.20 (q-value < 0.20)を満たす遺伝子数
sum(tcc$stat$q.value < 0.30) #FDR = 0.30 (q-value < 0.30)を満たす遺伝子数
```



Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



maqcデータでDEG同定

①オリジナルのmaqcデータは計14サンプル。
 ②最初の7列分がbrain、③残りの7列分がUHR (Universal Human Reference)

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phen table	
bodymap	22496456	human	19	2,197,622,796	link	link	link	
cheung	20856902	human	41	834,584,950	link	link	link	
core	19056941	human	2	8,670,342	link	link	link	
gilad	20009012	human	6	41,356,738	link	link	link	
maqc	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

```

sample.id num.tech.reps tissue
SRX016359.1 1 brain
SRX016359.2 1 brain
SRX016359.3 1 brain
SRX016359.4 1 brain
SRX016359.6 1 brain
SRX016359.7 1 brain
SRX016359.8 1 brain
SRX016367.1 1 UHR
SRX016367.2 1 UHR
SRX016367.3 1 UHR
SRX016367.4 1 UHR
SRX016367.6 1 UHR
SRX016367.7 1 UHR
SRX016367.8 1 UHR
    
```



Technical replicates

①オリジナルのmaqは、同一個体サンプルを分割して得られたtechnical replicatesのデータである。例えば、②最初の7列分のbrainデータは、同じ個人(例えば門田)の脳を7つに分割して測定した結果である

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phen table	sample.id	num.tech.reps	tissue
bodymap	22496456	human	19	2,197,622,796	link	link	link	SRX016359.1	1	brain
								SRX016359.2	1	brain
								SRX016359.3	1	brain
								SRX016359.4	1	brain
								SRX016359.6	1	brain
								SRX016359.7	1	brain
								SRX016359.8	1	brain
cheung	20856902	human	41	834,584,950	link	link	link	SRX016367.1	1	UHR
								SRX016367.2	1	UHR
								SRX016367.3	1	UHR
								SRX016367.4	1	UHR
								SRX016367.6	1	UHR
								SRX016367.7	1	UHR
								SRX016367.8	1	UHR
								SRX016367.8	1	UHR
maq	20167110	human	14 (technical) ** 2 (biological)	2,197,164	original pooled	original pooled	original pooled	experiment: MAQC-2		
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU		
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI		
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison		



同一個体 vs. 別個体

もし②が全て別個体由来のサンプルであったなら、異なる7人から脳組織を採取したbiological replicatesのデータということになる。実際には同一個体由来のtechnical replicatesデータなので、このサンプル間の類似度は非常に高い

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phen table	
bodymap	22496456	human	19	2,197,622,796	link	link	link	
cheung	20856902	human	41	834,584,950	link	link	link	
core	19056941	human	2	8,670,342	link	link	link	
gilad	20009012	human	6	41,356,738	link	link	link	
maqc	20167110	human	14 (technical) ** 2 (biological)	7,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

```

sample.id num.tech.reps tissue
SRX016359.1 1 brain
SRX016359.2 1 brain
SRX016359.3 1 brain
SRX016359.4 1 brain
SRX016359.6 1 brain
SRX016359.7 1 brain
SRX016359.8 1 brain
SRX016367.1 1 UHR
SRX016367.2 1 UHR
SRX016367.3 1 UHR
SRX016367.4 1 UHR
SRX016367.6 1 UHR
SRX016367.7 1 UHR
SRX016367.8 1 UHR
    
```



クラスタリング (maqc)

①カウントデータファイルと②phenotype情報
ファイルを読み込んでサンプル間クラスタリング

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622				Illumina Human
cheung	20856902	human	41	834,584,9				
core	19056941	human	2	8,670,342				
gilad	20009012	human	6	41,356,738	link	link	link	and femlaes
maqc	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled	original	original	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

```
R Console  
> getwd()  
[1] "C:/Users/kadota/Desktop/hoge"  
> list.files(pattern="maqc")  
[1] "maqc_count_table.txt" "maqc_phenodata.txt"  
> |
```



クラスタリング (maqc)

①コピー実行結果ファイル。②(同一)群内の類似度は非常に高く(距離が近い;バラツキが小さい)、③群間の相対的な類似度は非常に低い(距離が遠い)ことがわかる

クラスタリング (maqc)

ReCountのmaqcデータ([maqc_count_table.txt](#)と[maqc_phenodata.txt](#))を入力としてサンプル間クラスタリング。

```
in_f1 <- "maqc_count_table.txt" #入力ファイル名を指定してin_f1に格納(カウント)
in_f2 <- "maqc_phenodata.txt" #入力ファイル名を指定してin_f2に格納(サンプル)
out_f <- "hoge.png" #出力ファイル名を指定してout_fに格納
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)
```

```
#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
```

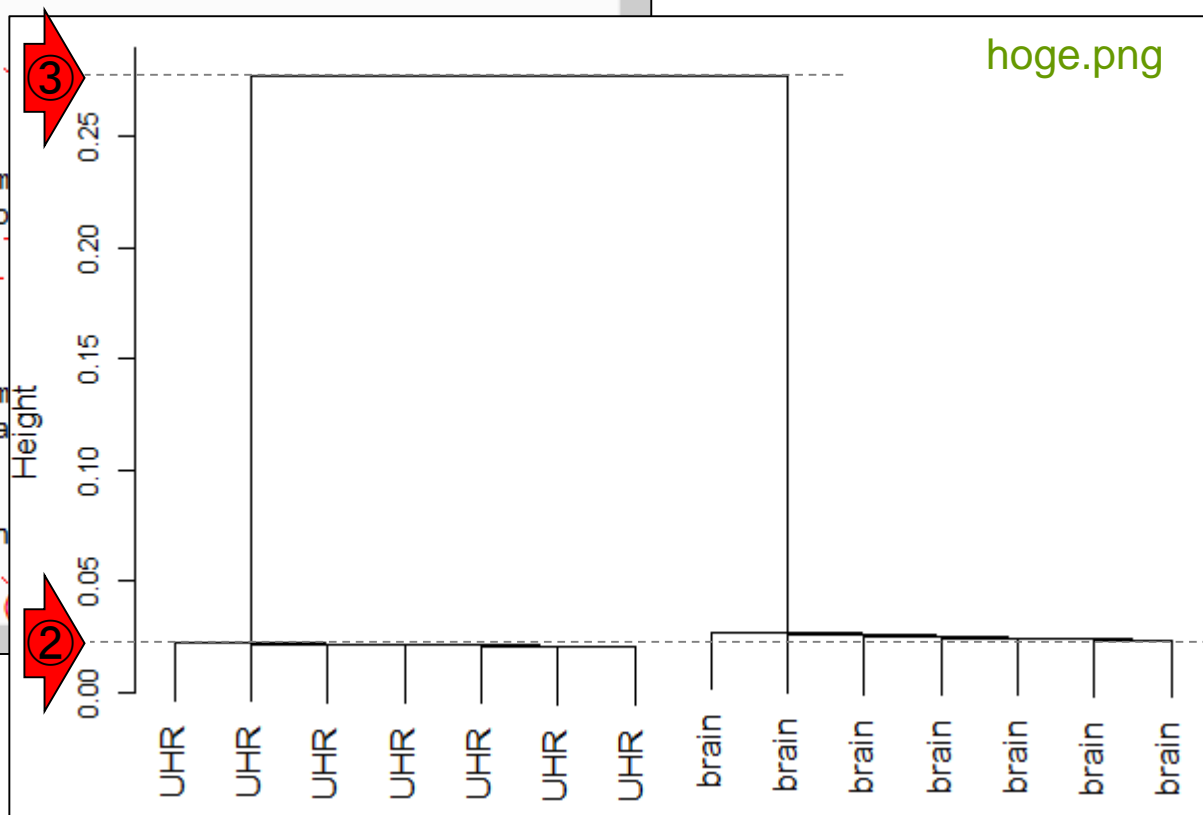
```
data <- read.table(in_f1, header=TRUE, row.names="phenotype")
phenotype <- read.table(in_f2, header=TRUE, row.names="phenotype")
colnames(data) <- phenotype$tissue
```

```
#本番
```

```
out <- clusterSample(data, dist.method="spearm", hclust.method="average", unique.p)
```

```
#ファイルに保存
```

```
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])
par(mar=c(0, 4, 1, 0))
plot(out, sub="", xlab="", cex.lab=1.2, #樹形図)
```



クラスタリング (blekhman)

スライドを見るだけ。①Blekhmanデータにもtechnical replicatesデータがあったことを思い出そう。同一サンプル名のは②高い類似度のあたりでsame terminal branchを形成していることがわかる

クラスタリング (blekhman)

「解析 | クラスタリング | サンプル間 | [TCC\(Sun 2013\)](#)」の例題7をベースに作成。入力は、20, samplesのカウントデータ([sample blekhman 36.txt](#))です。縦軸の範囲を[0, 0.28]にするやり方下さいm()m(「〇〇氏提供情報」とさせていただきます。)

```
in_f <- "sample_blekhman_36.txt"
out_f <- "hoge7.png"
param_fig <- c(700, 400)
param_yrange <- c(0, 0.28)
```

```
#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, header=TRUE,
dim(data))
```

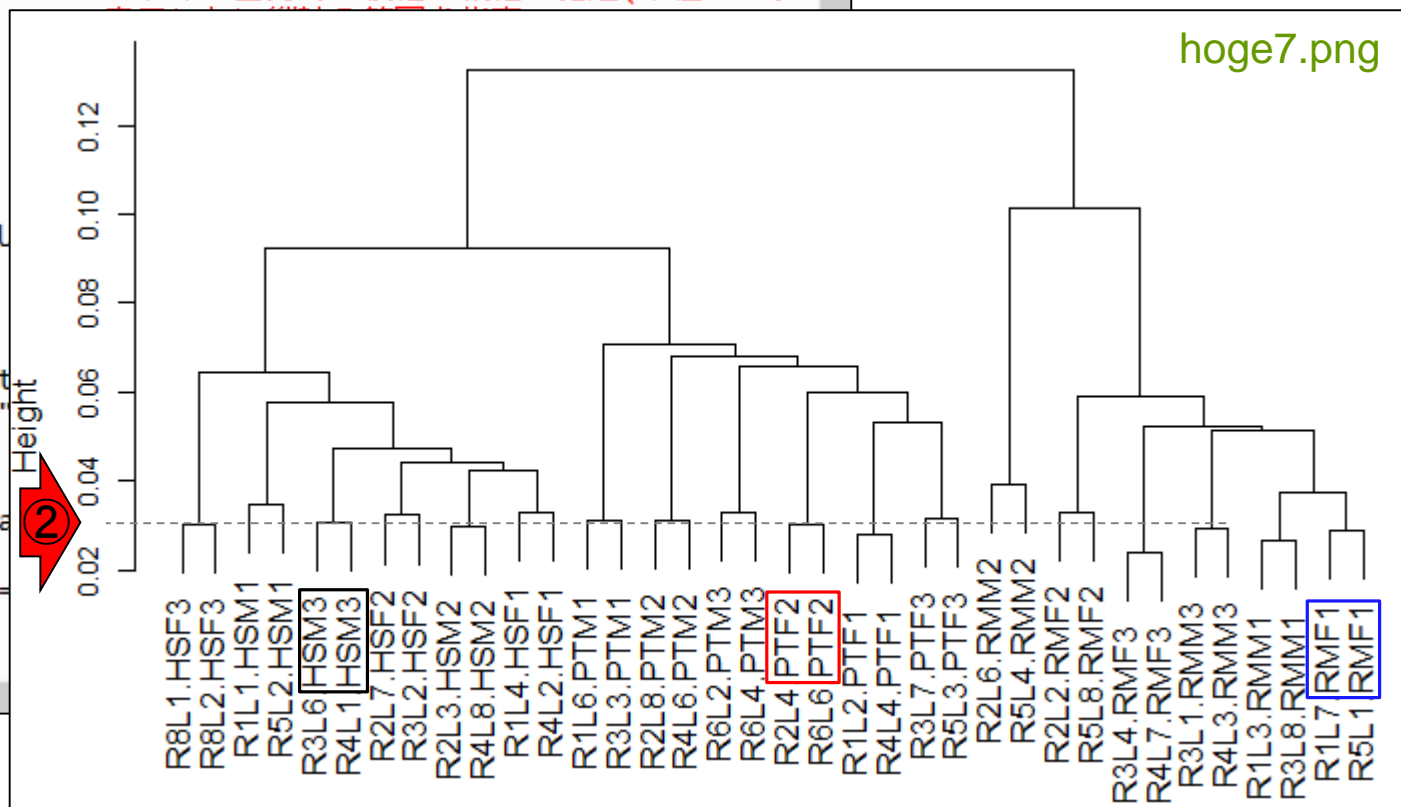
```
#本番
```

```
out <- clusterSample(data, dist.met
hclust.method="average"
```

```
#ファイルに保存
```

```
png(out_f, pointsize=13, width=para
par(mar=c(0, 4, 1, 0))
plot(out, sub="", xlab="", cex.lab=
cex=1.3, main="", ylab="Height",
ylim=param_yrange)
```

① #入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_fに格納
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)



クラスタリング (maqc)

①maqcデータでDEG同定すると、大量にDEGが得られることは、②と③の関係から容易に想像がつく

クラスタリング (maqc)

ReCountのmaqcデータ([maqc_count_table.txt](#)と[maqc_phenodata.txt](#))を入力としてサンプル間クラスタリング。

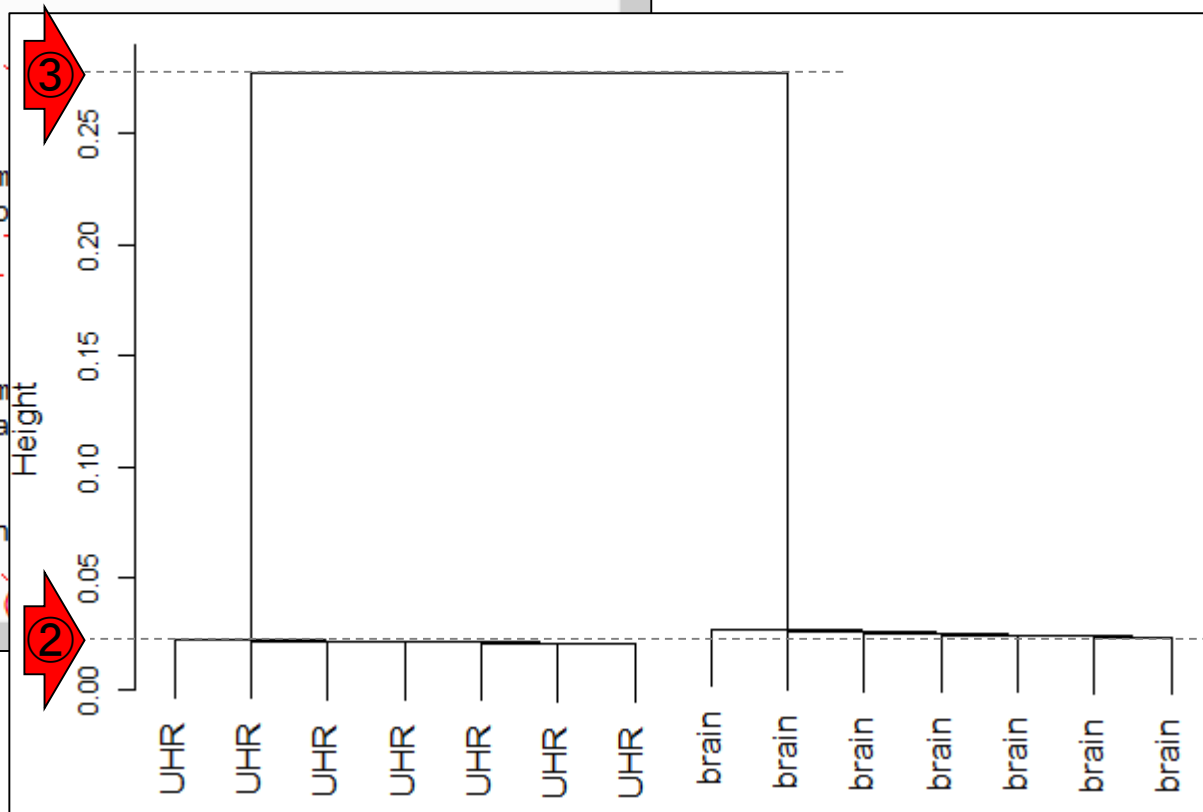
```
in_f1 <- "maqc_count_table.txt" ① #入力ファイル名を指定してin_f1に格納(カウント)
in_f2 <- "maqc_phenodata.txt"    #入力ファイル名を指定してin_f2に格納(サンプル)
out_f <- "hoge.png"             #出力ファイル名を指定してout_fに格納
param_fig <- c(600, 400)        #ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

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

```
#入力ファイルの読み込み
data <- read.table(in_f1, header=TRUE, row.names=colnames(data))
phenotype <- read.table(in_f2, header=TRUE, row.names=colnames(data)) #確認し
colnames(data) <- phenotype$tissue #dataオ
```

```
#本番
out <- clusterSample(data, dist.method="spearm", hclust.method="average", unique.pa
```

```
#ファイルに保存
png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2], #下、左、
par(mar=c(0, 4, 1, 0)) #樹形図(
```



maqcデータでDEG同定

①maqcデータでDEG同定した結果。②FDR 5%で7,618個なので、 $7,618 \times (1 - 0.05) = 7,237.1$ 個が本物のDEGと判断する

• maqcデータでDEG同定

ReCountのmaqcデータ([maqc_count_table.txt](#))で反復あり2群間比較を行う。「解析 | 発現変動 | 2群間比較」の「複製あり | TCC(Sun 2013)」例題1をベースに作成。M-A plot周辺はオプションなどを多少変更しています。

```
in_f <- "maqc_count_table.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_G1 <- 7
param_G2 <- 7
param_FDR <- 0.05
param_fig <- c(430, 350)
```

#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, header=TRUE, row.names=1, sep="")
```

#前処理(TCCクラスオブジェクトの作成)

```
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の作成
```

#本番(正規化)

```
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="tmm",
iteration=3, FDR=0.1, floorPDEG=0)
```

```
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)閾値を指定
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

#パッケージの読み込み

```
R Console
> sum(tcc$stat$q.value < 0.05)
[1] 7618
> sum(tcc$stat$q.value < 0.10)
[1] 7843
> sum(tcc$stat$q.value < 0.20)
[1] 8114
> sum(tcc$stat$q.value < 0.30)
[1] 8338
> dim(data)
[1] 52580 14
> |
```


maqcデータでDEG同定

①FDR閾値を緩めても、意外にDEG数が増えないことに気づく。
②入力データ中の、③総遺伝子数は52,580個もあるのに…である

maqcデータでDEG同定

ReCountのmaqcデータ([maqc_count_table.txt](#))で反復あり2群間比較を行う。「解析 | 発現変動 | 2群間比較」の「複製あり | TCC(Sun 2013)」例題1をベースに作成。M-A plot周辺はオプションなどを多少変更しています。

```
in_f <- "maqc_count_table.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_G1 <- 7
param_G2 <- 7
param_FDR <- 0.05
param_fig <- c(430, 350)
```

#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, header=TRUE, row.names=1, sep="")
```

#前処理(TCCクラスオブジェクトの作成)

```
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の作成
```

#本番(正規化)

```
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="tmm",
iteration=3, FDR=0.1, floorPDEG=1)
```

```
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)閾値を指定
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

#パッケージの読み込み

```
R Console
> sum(tcc$stat$q.value < 0.05)
[1] 7618
> sum(tcc$stat$q.value < 0.10)
[1] 7843
> sum(tcc$stat$q.value < 0.20)
[1] 8114
> sum(tcc$stat$q.value < 0.30)
[1] 8338
> dim(data)
[1] 52580 14
> |
```

maqcデータでDEG同定

この理由は、①入力ファイルをEXCELなどで眺めてみればわかる。一言で言えばゼロカウントだらけだということ。そのため、②ユニークな発現パターン数は11,177個しかない。また、③全14サンプルのうち、どこかのサンプルで1カウント分だけでも発現している遺伝子数は、11,907個

maqcデータでDEG同定

ReCountのmaqcデータ([maqc_count_table.txt](#))で反復あり2群間比較を行う。「解析 | 発現 | 複製あり | TCC(Sun 2013)」例題1をベースに作成。M-A plot周辺はオプションなど

```
in_f <- "maqc_count_table.txt" #入力ファイル名を指定してin_
out_f1 <- "hoge1.txt" #出力ファイル名を指定してout_
out_f2 <- "hoge1.png" #出力ファイル名を指定してout_
param_G1 <- 7 #G1群のサンプル数を指定
param_G2 <- 7 #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
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)
```

#必要なパッケージをロード

```
library(TCC) #パッケージの読み込
```

#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(TCCクラスオブジェクトの作成)

```
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の作成
```

#本番(正規化)

```
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="tmm",
iteration=3, FDR=0.1, floorPDEG=0)
```

```
R Console
> sum(tcc$stat$q.value < 0.05) # $
[1] 7618
> sum(tcc$stat$q.value < 0.10) # $
[1] 7843
> sum(tcc$stat$q.value < 0.20) # $
[1] 8114
> sum(tcc$stat$q.value < 0.30) # $
[1] 8338
> dim(data)
[1] 52580 14
> dim(unique(data))
[1] 11177 14
> sum(rowSums(data) > 0)
[1] 11907
> |
```

maqcデータでDEG同定

①実質的に遺伝子数の分母は11,907個。それゆえ、②FDR閾値を緩めていっても、遺伝子数があまり増えないように見えたただけだったのである。③このデータはDEGだらけであり、クラスタリング結果と矛盾しない

maqcデータでDEG同定

ReCountのmaqcデータ([maqc_count_table.txt](#))で反復あり2群間比較を行う。「解析し | 複製あり | [TCC\(Sun 2013\)](#)」例題1をベースに作成。M-A plot周辺はオプション

```
in_f <- "maqc_count_table.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_G1 <- 7
param_G2 <- 7
param_FDR <- 0.05
param_fig <- c(430, 350)
```

#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, header=TRUE, row.names=1, sep="")
```

#前処理(TCCクラスオブジェクトの作成)

```
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の作成
```

#本番(正規化)

```
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="tmm",
iteration=3, FDR=0.1, floorPDEG=0)
```

```
#入力ファイル名を指定してin_fに格納
#出力ファイル名を指定してout_f1に格納
#出力ファイル名を指定してout_f2に格納
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)閾値を指定
#ファイル出力時の横幅と縦幅を指定(単位はピクセル)
```

#パッケージの読み込み

```
R Console
> sum(tcc$stat$q.value < 0.05)
[1] 7618
> sum(tcc$stat$q.value < 0.10)
[1] 7843
> sum(tcc$stat$q.value < 0.20)
[1] 8114
> sum(tcc$stat$q.value < 0.30)
[1] 8338
> dim(data)
[1] 52580 14
> dim(unique(data))
[1] 11177 14
> sum(rowSums(data) > 0)
[1] 11907
> |
```

DEG数の見積もり

(Blekhmanデータと異なり)FDR閾値を緩めるとDEG数が減っていることがわかる。私はこんなデータを初めて見たが、technical replicates特有かもしれない。一般的には、FDR閾値を緩めるとDEG数は増えていくという認識でよい

• DEG数の見積もり(maqcデータ)

```
sum(tcc$stat$q.value < 0.05) #FDR = 0.05 (q-value < 0.05)を満たす遺伝子数を
sum(tcc$stat$q.value < 0.10) #FDR = 0.10 (q-value < 0.10)を満たす遺伝子数を
sum(tcc$stat$q.value < 0.20) #FDR = 0.20 (q-value < 0.20)を満たす遺伝子数を
sum(tcc$stat$q.value < 0.30) #FDR = 0.30 (q-value < 0.30)を満たす遺伝子数を
7618*(1 - 0.05)
7843*(1 - 0.10)
8114*(1 - 0.20)
8338*(1 - 0.30)
```

```
R Console
> sum(tcc$stat$q.value < 0.05) # $
[1] 7618
> sum(tcc$stat$q.value < 0.10) # $
[1] 7843
> sum(tcc$stat$q.value < 0.20) # $
[1] 8114
> sum(tcc$stat$q.value < 0.30) # $
[1] 8338
> 7618*(1 - 0.05)
[1] 7237.1
> 7843*(1 - 0.10)
[1] 7058.7
> 8114*(1 - 0.20)
[1] 6491.2
> 8338*(1 - 0.30)
[1] 5836.6
> |
```



Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



maqc (pooled)

①オリジナルのmaqcデータは、7 technical replicates。Blekhmanデータでtechnical replicatesをマージ(merge)したように、通常はbiological replicatesデータを入力として発現変動解析を行う。②ReCountではpooledという表現になっているが、作成作業は同じ(technical replicatesデータを足すだけ)

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	sample.id num.tech.reps tissue
bodymap	22496456	human	19	2,197,622,796	link	link	link	SRX016359.1 1 brain SRX016359.2 1 brain SRX016359.3 1 brain SRX016359.4 1 brain SRX016359.6 1 brain SRX016359.7 1 brain SRX016359.8 1 brain
cheung	20856902	human	2	8,670,342	link	link	link	SRX016367.1 1 UHR SRX016367.2 1 UHR SRX016367.3 1 UHR SRX016367.4 1 UHR SRX016367.6 1 UHR SRX016367.7 1 UHR SRX016367.8 1 UHR
core	19056941	human	2	8,670,342	link	link	link	
gilad	20009012	human	6	41,356,738	link	link	link	
maqc	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

```
sample.id num.tech.reps tissue
SRX016359 7 brain
SRX016367 7 UHR
```

```
original
pooled
```



mergeの基本形

merged (or pooled)データ作成手順の概要。
①基本的には、同一群のデータのみ抽出して、行方向で足しているだけ。②最後にG1群とG2群のデータを列方向で結合して完了

mergeの基本形

technical replicatesのデータを足すだけ。

```
in_f <- "maq_count_table.txt"      #入力ファイル名を指定してin_fに格納
param_G1 <- 7                      #G1群のサンプル数を指定
param_G2 <- 7                      #G2群のサンプル数を指定

#入力ファイルの読み込みとラベル情報の作成
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
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトルdata.clを作成
head(data[, data.cl == 1], n=3)     #最初の3行分を表示(G1群)
head(data[, data.cl == 2], n=3)     #最初の3行分を表示(G2群)

#本番(マージ)
data_G1 <- rowSums(data[, data.cl == 1])#G1群をマージした結果をdata_G1に格納
data_G2 <- rowSums(data[, data.cl == 2])#G2群をマージした結果をdata_G2に格納
data <- cbind(data_G1, data_G2)     #列方向で結合した結果をdataに格納
dim(data)                          #行数と列数を表示
head(data)                          #最初の6行分を表示

dim(unique(data))                  #ユニークな発現パターン数を表示
sum(rowSums(data) > 0)             #どこかのサンプルで1カウント分だけでも発現している遺伝子数を
```



mergeの基本形

黒枠部分を表示。merge前(入力データ)の、G1群(brain)とG2群(UHR)の最初の3行分を表示

mergeの基本形

technical replicatesのデータを足すだけ。

```
in_f <- "maq_count_table.txt" #入力ファイル名を指定してin_fに格納
param_G1 <- 7 #G1群のサンプル数を指定
param_G2 <- 7 #G2群のサンプル数を指定
```

#入力ファイルの読み込みとラベル情報の作成

```
data <- read.table(in_f, head=1)
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
head(data[, data.cl == 1], n=3)
head(data[, data.cl == 2], n=3)
```

#本番(マージ)

```
data_G1 <- rowSums(data[, data.cl == 1])
data_G2 <- rowSums(data[, data.cl == 2])
data <- cbind(data_G1, data_G2)
dim(data)
head(data)

dim(unique(data))
sum(rowSums(data) > 0)
```

```
R Console
> head(data[, data.cl == 1], n=3) #最初の3行分を表示 (G1$)
      SRX016359.1 SRX016359.2 SRX016359.3 SRX016359.4
ENSG00000000003      1          2          3          5
ENSG00000000005      0          0          1          0
ENSG000000000419    9          8          6          7
      SRX016359.6 SRX016359.7 SRX016359.8
ENSG00000000003      3          2          2
ENSG00000000005      0          0          0
ENSG000000000419    7          6          4
> head(data[, data.cl == 2], n=3) #最初の3行分を表示 (G2$)
      SRX016367.1 SRX016367.2 SRX016367.3 SRX016367.4
ENSG00000000003     14         18         27         37
ENSG00000000005      3          0          0          0
ENSG000000000419    28         36         41         37
      SRX016367.6 SRX016367.7 SRX016367.8
ENSG00000000003     20         11         21
ENSG00000000005      2          0          0
ENSG000000000419    43         30         23
> |
```


mergeの基本形

黒枠部分を表示。merge後のデータ(merged or pooled)は、①52,580行×2列。2群間比較用でbrain vs. UHRだから2列になるのは当たり前。②merge後のデータの最初の6行分を表示。technical replicatesデータの場合は、こんな感じで処理しましょう

- mergeの基本形
technical replicatesのデータを足すだけ。

```
in_f <- "maq_count_table.txt"
param_G1 <- 7
param_G2 <- 7
```

```
#入力ファイル名を指定してin_fに格納
#G1群のサンプル数を指定
#G2群のサンプル数を指定
```

```
#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f, head=1)
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
head(data[, data.cl == 1], n=6)
head(data[, data.cl == 2], n=6)
```

```
#本番(マージ)
data_G1 <- rowSums(data[, data.cl == 1])
data_G2 <- rowSums(data[, data.cl == 2])
data <- cbind(data_G1, data_G2)
dim(data)
head(data)
```

```
dim(unique(data))
sum(rowSums(data) > 0)
```

```
R Console
> #本番(マージ)
> data_G1 <- rowSums(data[, data.cl == 1]) #G1群をマージした結果$
> data_G2 <- rowSums(data[, data.cl == 2]) #G2群をマージした結果$
> data <- cbind(data_G1, data_G2) #列方向で結合した結果$
> dim(data) #行数と列数を表示
[1] 52580 2
> head(data) #最初の6行分を表示
```

	data_G1	data_G2
ENSG00000000003	18	148
ENSG00000000005	1	5
ENSG000000000419	47	238
ENSG000000000457	109	126
ENSG000000000460	6	63
ENSG000000000938	8	0

反復なし2群間比較

mergeの基本形

technical replicatesのデータを足すだけ。

```
in_f <- "maq_count_table.txt"      #入力ファイル名を指定してin_fに格納
param_G1 <- 7                      #G1群のサンプル数を指定
param_G2 <- 7                      #G2群のサンプル数を指定
```

#入力ファイルの読み込みとラベル情報の作成

```
data <- read.table(in_f, head=1)
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
head(data[, data.cl == 1], n=1)
head(data[, data.cl == 2], n=1)
```

#本番(マージ)

```
data_G1 <- rowSums(data[, data.cl == 1])
data_G2 <- rowSums(data[, data.cl == 2])
data <- cbind(data_G1, data_G2)
dim(data)
head(data)
```

```
dim(unique(data))
sum(rowSums(data) > 0)
```

R Console

> #本番(マージ)

> data_G1 <- rowSums(data[, data.cl == 1]) #G1群をマージした結果\$

> data_G2 <- rowSums(data[, data.cl == 2]) #G2群をマージした結果\$

> data <- cbind(data_G1, data_G2)

#列方向で結合した結果\$

> dim(data)

#行数と列数を表示

[1] 52580 2

> head(data)

#最初の6行分を表示

	data_G1	data_G2
ENSG000000000003	18	148
ENSG000000000005	1	5
ENSG000000000419	47	238
ENSG000000000457	109	126
ENSG000000000460	6	63
ENSG000000000938	8	0

> dim(unique(data))

#ユニークな発現パターン\$

[1] 9017 2

> sum(rowSums(data) > 0)

#どこかのサンプルで1か\$

[1] 11907

> |

①pooledのファイル(maqc_pooledreps_count_table.txt)の中身と同じです

maqc (pooled)

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human			link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and females
maqc	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

```
sample.id num.tech.reps tissue
SRX016359 7 brain
SRX016367 7 UHR
```

Contents

- カウントデータ、データの正規化(基礎)、RPK、RPM、RPKM
- サンプル間クラスタリング、結果の解釈
 - 20150729の復習(Blekhmanのデータ)、Tips
 - ReCountのbodymapデータ、giladデータ、マージ(bodymap + gilad)後のデータ
- 発現変動解析(反復あり2群間比較)
 - Blekhmanのデータ(DEGが多い場合)、M-A plot
 - モデル、分布、統計的手法、Blekhmanのデータ(DEGがそれほど多くない場合)
 - Blekhmanのデータ(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
 - giladのデータ(オス肝臓3サンプル vs. メス肝臓3サンプル;計6人)
- 実験デザイン、反復なし2群間比較
 - MAQCのtechnical replicatesデータ(7 brain samples vs. 7 UHR samples)
 - MAQCのbiological replicatesデータ(1 brain samples vs. 1 UHR samples)
 - 反復なし2群間比較: maqc (pooled)



maqc (pooled)

① 52,580行 × 2列のmaqc (pooled)のファイル (maqc_pooledreps_count_table.txt)を入力として、反復なし2群間比較を行う。反復なしの場合はほとんどDEGが検出されないことと、その理由などを述べる

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link			Illumina
cheung	20856902	human	41	834,584,950	link			
core	19056941	human	2	8,670,342	link			
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and females
maqc	20167110	human	14 (technical) ** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="maqc_pool")
[1] "maqc_pooledreps_count_table.txt"
> |
```



maqc (pooled)でDEG同定

①「... | 複製なし | TCC」。②若干話がややこしいですが、「TCCとDESeq2とTang et al., 2015」の論文を引用して使えば大丈夫です

- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | [TCC\(Sun_2013\)](#) (last modified 2015/11/13)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | [DESeq2\(Love_2014\)](#) (last modified 2015/11/15)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | [TCC\(Sun_2013\)](#) (last modified 2015/07/07) 推奨
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | [Blekhmanデータ | TCC\(Sun_2013\)](#) (last modified 2015/07/07)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製あり | [SAMseq\(Li_2013\)](#) (last modified 2014/02/07)
- 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 | 対応なし | 複製あり | [WAD\(Kadota_2015\)](#) (last modified 2015/03/30)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | [TCC\(Sun_2013\)](#) (last modified 2016/05/21) 推奨 NEW
- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | [DESeq\(Anders_2014\)](#) (last modified 2014/03/20)
- 解析 | 発現変動 | 2群間 | 対応なし | 複製なし | [edgeR\(Robinson_2010\)](#) (last modified 2014/03/20)

解析 | 発現変動 | 2群間 | 対応なし | 複製なし | TCC (Sun_2013) NEW

TCCを用いたやり方を示します。2016年5月21日に、TCC原著論文([Sun et al., BMC Bioinformatics, 2013](#))発表時の推奨解析パイプラインである、iDEGES/DESeq-DESeqから、iDEGES/DESeq2-DESeq2に切り替えました。これは、その後DESeq2パッケージの論文 ([Love et al., Genome Biol., 2014](#))が発表されたことと、複製なし(反復なし)3群間比較の性能評価論文([Tang et al., BMC Bioinformatics, 2015](#)) 中でTCCの解析パイプライン内部でDESeq2を用いたほうが感度・特異度が高いことが分かったためです。[Tang et al., 2015](#)のpage 10あたりに「We expect the DESeq2-related pipelines (i.e., EDE-S and SSS-S) would be recommended for analyzing two-group data without replicates as an updated guideline.」と書いてありますので大丈夫です。ちなみにTCC中の記述法だと「iDEGES/DESeq2-DESeq2」の解析パイプラインは、[Tang et al., 2015](#) ではSSS-SIに相当します。

「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。

1. サンプルデータ14の10,000 genes×2 samplesのカウントデータ(data_hypodata_lvsl.txt)の場合:

シミュレーションデータ(G1群1サンプル vs. G2群1サンプル)です。gene_1~gene_2000までがDEG (最初の1800個がG1群で高発現、残りの200個がG2群で高発現) gene_2001~gene_10000までがnon-DEGであることが既知です。

```
in_f <- "data_hypodata_lvsl.txt" #入力ファイル名を指定してin_flに格納
out_f <- "hoge1.txt" #出力ファイル名を指定してout_flに格納
param_G1 <- 1 #G1群のサンプル数を指定
param_G2 <- 1 #G2群のサンプル数を指定
```

maqc (pooled)でDEG同定

解析 | 発現変動 | 2群間 | 対応なし | 複製なし | TCC (Sun_2013) **NEW**

TCCを用いたやり方を示します。2016年5月21日に、TCC原著論文([Sun et al., BMC Bioinformatics, 2013](#))発表時の推奨解析パイプラインである、iDEGES/DESeq-DESeqから、iDEGES/DESeq2-DESeq2に切り替えました。これは、その後DESeq2パッケージの論文([Love et al., Genome Biol., 2014](#))が発表されたことと、複製なし(反復なし)3群間比較の性能評価論文([Tang et al., BMC Bioinformatics, 2015](#))中でTCCの解析パイプライン内部でDESeq2を用いたほうが感度・特異度が高いことが分かったためです。[Tang et al., 2015](#)のpage 10あたりでrecommended for analyzing two-group dataにTCC中の記述法だと「iDEGES/DESeq2-D」[ファイル]-「ディレクトリの変更」で解析した

1. サンプルデータ14の10,000 genes×2 sam

シミュレーションデータ(G1群1サンプル vs. 高発現、残りの200個がG2群で高発現) ge

```
in_f <- "data_hypodata_1vs1.txt"
out_f <- "hoge1.txt"
param_G1 <- 1
```

5. ReCountのmaqc (pooled)データ(maqc_pooledreps_count_table.txt)の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

```
in_f <- "maqc_pooledreps_count_table.txt" #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt" #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png" #出力ファイル名を指定してout_f2に格納
param_G1 <- 1 #G1群のサンプル数を指定
param_G2 <- 1 #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
param_fig <- c(400, 380) #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
```

```
#前処理(TCCクラスオブジェクトの作成)
```

```
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2としたベクトルdata
tcc <- new("TCC", data, data.cl) #TCCクラスオブジェクトtccを作成
```

```
#本番(正規化)
```

```
tcc <- calcNormFactors(tcc, norm.method="deseq2", test.method="deseq2", #正規化を身
iteration=3, FDR=0.1, floorPDEG=0.05) #正規化を実行した結果を
normalized <- getNormalizedData(tcc) #正規化後のデータを取り出してnormalizedに格納
```

maqc (pooled)でDEG同定

①コード下部の、②のあたりで警告メッセージが出ますが、入力が「反復なしデータ」であることに起因するものなので、基本的に問題ない

5. ReCountのmaqc (pooled)データ([maqc_pooledreps_count_table.txt](#))の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

#本番(正規化)

```
tcc <- calcNormFactors(tcc, norm.method="deseq2", test.method="deseq2", #正規化を実行した結果を  
                        iteration=3, FDR=0.1, floorPDEG=0.05) #正規化を実行した結果を  
normalized <- getNormalizedData(tcc) #正規化後のデータを取り出してnormalizedに格納
```

#本番(DEG検出)

```
tcc <- estimateDE(tcc, test.method="deseq2", FDR=param_FDR) #DEG検出を実行した結果を  
result <- getResult(tcc, sort=FALSE) #p値などの結果をした結果をresultに格納
```

#ファイルに保存(テキストファイル)

```
tmp <- cbind(rownames(tcc$count), normalized, result) #正規化後のデータの右側にDEG検  
tmp <- tmp[order(tmp$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の中身を指定
```

#ファイルに保存(M-A plot)

```
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2]) #出力ファイルの  
par(mar=c(4, 4, 0, 0)) #余白を指定  
plot(tcc, FDR=param_FDR, 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", #閾値を満たすDEGをマゼンタ色にして描画  
      ylab="M = log2(G2) - log2(G1)", #閾値を満たすDEGをマゼンタ色にして描画  
      legend("topright", c(paste("DEG(FDR =", param_FDR, ")"), "non-DEG"), #凡例  
            col=c("magenta", "black"), pch=20) #凡例を作成
```

②

①

maqc (pooled)でDEG同定

5. ReCountのmaqc (pooled)データ([maqc_pooledreps_count_table.txt](#))の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

#本番(正規化)

```
tcc <- calcNormFactors(tcc, norm.method="deseq2", test.method="deseq2", #正規化を身
iteration=3, FDR=0.1, floorPDEG=0.05)#正規化を実行した結果を
normalized <- getNormalizedData
```

#本番(DEG検出)

```
tcc <- estimateDE(tcc, test.met
result <- getResult(tcc, sort=f
```

#ファイルに保存(テキストファイル)

```
tmp <- cbind(rownames(tcc$count
tmp <- tmp[order(tmp$rank),]
write.table(tmp, out_f1, sep="")
```

#ファイルに保存(M-A plot)

```
png(out_f2, pointsize=13, width
par(mar=c(4, 4, 0, 0))
plot(tcc, FDR=param_FDR, main=
xlab="A = (log2(G2) + log2
ylab="M = log2(G2) - log2
legend("topright", c(paste("DE
col=c("magenta", "black")
```

```
R Console
read the ?DESeq section on 'Experiments without replicates'
3: checkForExperimentalReplicates(object, modelMatrix) で:
same number of samples and coefficients to fit,
estimating dispersion by treating samples as replicates.
read the ?DESeq section on 'Experiments without replicates'
> normalized <- getNormalizedData(tcc) #正規化後のデータを取$
>
> #本番 (DEG検出)
> tcc <- estimateDE(tcc, test.method="deseq2", FDR=param_FDR)#$
TCC::INFO: Identifying DE genes using deseq2 ...
TCC::INFO: Done.
警告メッセージ:
checkForExperimentalReplicates(object, modelMatrix) で:
same number of samples and coefficients to fit,
estimating dispersion by treating samples as replicates.
read the ?DESeq section on 'Experiments without replicates'
> result <- getResult(tcc, sort=FALSE) #p値などの結果をした$
>
```



①

maqc (pooled)でDEG同定

①例題5。コピペ実行結果の最後のほう。FDR 30%までで0個だったので、
②念のためさらに緩い50%と70%のところを眺めてDEGがないことを確認

5. ReCountのmaqc (pooled)データ(maqc_pooledreps_count_table.txt)の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

```
in_f <- "maqc_pooledreps_count_table.txt"#入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt" #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png" #出力ファイル名を指定してout_f2に格納
param_G1 <- 1
param_G2 <- 1
param_FDR <- 0.05
param_fig <- c(400, 380)
```

```
#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, header=TRUE)
```

```
#前処理(TCCクラスオブジェクトの作成)
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
tcc <- new("TCC", data, data.cl)
```

```
#本番(正規化)
tcc <- calcNormFactors(tcc, norm.method="iterative")
normalized <- getNormalizedData(tcc)
```

```
R Console
> legend("topright", c(paste("DEG (FDR =", param_FDR, ")"), sep=" ",
+ col=c("magenta", "black"), pch=20) #凡例を作成
> dev.off() #おまじない
null device
1
> sum(tcc$stat$q.value < 0.05) #FDR = 0.05 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.10) #FDR = 0.10 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.20) #FDR = 0.20 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.30) #FDR = 0.30 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.50) #FDR = 0.30 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.70) #FDR = 0.30 (q-value $
[1] 0
> |
```



maqc (pooled)でDE

①hoge5.txtの中身を確認。結論を先に言えば、「反復なしデータの場合は、②内部的にDESeq2を用いるTCCの解析パイプラインを推奨」

5. ReCountのmaqc (pooled)データ(maqc_pooledreps_count_table.txt)の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

```
in_f <- "maqc_pooledreps_count_table.txt" #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt" #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png" #出力ファイル名を指定してout_f2に格納
param_G1 <- 1 #G1群のサンプル数を指定
param_G2 <- 1 #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
param_fig <- c(400, 380) #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
```

#入力データ	rownames(tcc\$count)	SRX016359	SRX016367	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
	ENSG00000131095	96229.4	17.3	ENSG00000131095	10.33	-12.45	0.00024	0.74489	1	0
#前処理	ENSG00000165023	18460.1	6.6	ENSG00000165023	8.44	-11.46	0.00027	0.74489	2	0
data.c	ENSG00000162728	15655.3	7.4	ENSG00000162728	8.41	-11.05	0.00030	0.74489	3	0
tcc <-	ENSG00000148826	4865.8	0.8	ENSG00000148826	5.98	-12.53	0.00034	0.74489	4	0
#本番()	ENSG00000171532	7523.6	0.8	ENSG00000171532	6.30	-13.16	0.00034	0.74489	5	0
tcc <-	ENSG00000184144	18292.7	13.1	ENSG00000184144	8.94	-10.44	0.00038	0.74489	6	0
normal	ENSG00000149575	4559.1	2.5	ENSG00000149575	6.73	-10.85	0.00039	0.74489	7	0
<	ENSG00000087250	28296.4	22.2	ENSG00000087250	9.63	-10.32	0.00039	0.74489	8	0
	ENSG00000104435	5323.2	3.3	ENSG00000104435	7.05	-10.66	0.00040	0.74489	9	0
	ENSG00000148053	2951.7	1.6	ENSG00000148053	6.12	-10.81	0.00045	0.74489	10	0

① hoge5.txtは、②発現変動順にソートされた結果を返している。赤枠の③brainと④UHRサンプルの数値は、正規化後のデータなので、整数ではなく少数になっている。見た目でも確かに発現変動順になっているので妥当

maqc (pooled) で

5. ReCountのmaqc (pooled)データ (maqc_pooledreps_count_table)

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

```

in_f <- "maqc_pooledreps_count_table.txt" #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt" #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png" #出力ファイル名を指定してout_f2に格納
param_G1 <- 1 #G1群のサンプル数を指定
param_G2 <- 1 #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
param_fig <- c(400, 380) #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)
    
```



#入力データ	rownames(tcc\$count)	SRX016359	SRX016367	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
	ENSG00000131095	96229.4	17.3	ENSG00000131095	10.33	-12.45	0.00024	0.74489	1	0
#前処理	ENSG00000165023	18460.1	6.6	ENSG00000165023	8.44	-11.46	0.00027	0.74489	2	0
data.c	ENSG00000162728	15655.3	7.4	ENSG00000162728	8.41	-11.05	0.00030	0.74489	3	0
tcc <-	ENSG00000148826	4865.8	0.8	ENSG00000148826	5.98	-12.53	0.00034	0.74489	4	0
#本番()	ENSG00000171532	7523.6	0.8	ENSG00000171532	6.30	-13.16	0.00034	0.74489	5	0
tcc <-	ENSG00000184144	18292.7	13.1	ENSG00000184144	8.94	-10.44	0.00038	0.74489	6	0
normal	ENSG00000149575	4559.1	2.5	ENSG00000149575	6.73	-10.85	0.00039	0.74489	7	0
<	ENSG00000087250	28296.4	22.2	ENSG00000087250	9.63	-10.32	0.00039	0.74489	8	0
	ENSG00000104435	5323.2	3.3	ENSG00000104435	7.05	-10.66	0.00040	0.74489	9	0
	ENSG00000148053	2951.7	1.6	ENSG00000148053	6.12	-10.81	0.00045	0.74489	10	0

maqc (pooled)でDEG

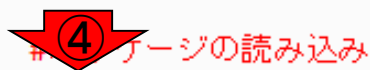
FDR 70%を満たす遺伝子数が0個だった理由は、①1位の②q-valueを見て納得。例えば、FDR 80%で調べると、少なくともこの画面上に見えている上位10個は条件を満たすので10以上の数値になるだろうと予想して確認する

5. ReCountのmaqc (pooled)データ(maqc_pooledreps_count_table.txt)の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。正規化後のデータ、発現変動順にソートして出力しています。M-A plotのとき

```
in_f <- "maqc_pooledreps_count_table.txt" #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt" #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png" #出力ファイル名を指定してout_f2に格納
param_G1 <- 1 #G1群のサンプル数を指定
param_G2 <- 1 #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
param_fig <- c(400, 380) #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)



#入力データ	rownames(tcc\$count)	SRX016359	SRX016367	gene_id	a.value	m.value	p.value	q.value	rank	estimatedDEG
	ENSG00000131095	96229.4	17.3	ENSG00000131095	10.33	-12.45	0.00024	0.74489	1	0
#前処理	ENSG00000165023	18460.1	6.6	ENSG00000165023	8.44	-11.46	0.00027	0.74489	2	0
data.c	ENSG00000162728	15655.3	7.4	ENSG00000162728	8.41	-11.05	0.00030	0.74489	3	0
tcc <-	ENSG00000148826	4865.8	0.8	ENSG00000148826	5.98	-12.53	0.00034	0.74489	4	0
#本番()	ENSG00000171532	7523.6	0.8	ENSG00000171532	6.30	-13.16	0.00034	0.74489	5	0
tcc <-	ENSG00000184144	18292.7	13.1	ENSG00000184144	8.94	-10.44	0.00038	0.74489	6	0
normal	ENSG00000149575	4559.1	2.5	ENSG00000149575	6.73	-10.85	0.00039	0.74489	7	0
<	ENSG00000087250	28296.4	22.2	ENSG00000087250	9.63	-10.32	0.00039	0.74489	8	0
	ENSG00000104435	5323.2	3.3	ENSG00000104435	7.05	-10.66	0.00040	0.74489	9	0
	ENSG00000148053	2951.7	1.6	ENSG00000148053	6.12	-10.81	0.00045	0.74489	10	0



maqc (pooled)でDEG同定



5. ReCountのmaqc (pooled)データ(maqc_pooledreps_count_table.txt)の場合:

52,580 genes×2 samplesのカウントデータ(G1群1サンプル vs. G2群1サンプル)です。例題4と基本的に同じで、正規化後のデータ、発現変動順にソートして出力しています。M-A plotのところも変更しています。

```
in_f <- "maqc_pooledreps_count_table.txt" #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge5.txt" #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge5.png" #出力ファイル名を指定してout_f2に格納
param_G1 <- 1
param_G2 <- 1
param_FDR <- 0.05
param_fig <- c(400, 380)
```

```
#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, header=TRUE)
```

```
#前処理(TCCクラスオブジェクトの作成)
data.cl <- c(rep(1, param_G1), rep(2, param_G2))
tcc <- new("TCC", data, data.cl)
```

```
#本番(正規化)
tcc <- calcNormFactors(tcc, norm.method="TMM", iterative=TRUE)
normalized <- getNormalizedData(tcc)
```

```
R Console
> dev.off() #おまじない
null device
1
> sum(tcc$stat$q.value < 0.05) #FDR = 0.05 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.10) #FDR = 0.10 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.20) #FDR = 0.20 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.30) #FDR = 0.30 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.50) #FDR = 0.30 (q-value $
[1] 0
> sum(tcc$stat$q.value < 0.70) #FDR = 0.30 (q-value $
[1] 0
① > sum(tcc$stat$q.value < 0.80) #FDR = 0.30 (q-value $
[1] 200
> |
```

反復あり or なし(TCC)

コード全体をコピーした結果をまとめたのが右下の表。反復あり(7 technical replicates)では、大量のDEGが得られ、反復なし(1 biological replicates; pooled)ではDEGがほとんど得られなかった。おさらい

- 反復あり or なし(TCC)
TCCパッケージ(Sun et al., BMC Bioinformatics, 2013)のやり方です。

```
#####
### 反復あり(7 technical replicates)
#####
in_f <- "maq_count_table.txt"      #入力ファイル名を指定してin_fに格納
param_G1 <- 7                      #G1群のサンプル数を指定
param_G2 <- 7                      #G2群のサンプル数を指定

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトル

#TCC実行
library(TCC)                       #パッケージの読み込み
tcc <- new("TCC", data, data.cl)   #TCCクラスオブジェクトtccを作成
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="edgeR")#正規化を実行
```

	反復あり2群間比較	反復なし2群間比較
tcc <- estim	TCC(edgeR)	TCC(DESeq2)
p.value <- t	1.2.0	1.2.0
p.value[is.r		
ranking <- r	FDR=5% 7618	0
ranking_TCC	FDR=10% 7843	0
q.value <- t	FDR=20% 8114	0
	FDR=30% 8338	0
	FDR=50% 8621	0
	FDR=70% 8795	0
	FDR=80% 8987	200

反復あり or なし(TCC)

この違いは単純に内部的に用いているパッケージの違い(反復ありの場合は①edgeRで、反復なしの場合は②DESeq2に自動で切り替わる)に起因するのでは?!というヒト用に、edgeRとDESeq2単体で利用した結果も示す

- 反復あり or なし(TCC)

TCCパッケージ(Sun et al., BMC Bioinformatics, 2013)のやり方です。

```
#####
### 反復あり(7 technical replicates)
#####
in_f <- "maqc_count_table.txt"          #入力ファイル名を指定してin_fに格納
param_G1 <- 7                          #G1群のサンプル数を指定
param_G2 <- 7                          #G2群のサンプル数を指定

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトル

#TCC実行
library(TCC)                            #パッケージの読み込み
tcc <- new("TCC", data, data.cl)        #TCCクラスオブジェクトtccを作成
tcc <- calcNormFactors(tcc, norm.method="tmm", test.method="edgeR")#正規化を実
```

反復あり2群間比較

反復なし2群間比較

Package	TCC(edgeR)	TCC(DESeq2)
version	1.2.0	1.2.0
FDR=5%	7618	0
FDR=10%	7843	0
FDR=20%	8114	0
FDR=30%	8338	0
FDR=50%	8621	0
FDR=70%	8795	0
FDR=80%	8987	200

反復あり or なし(edgeR)

- 反復あり or なし(edgeR)

edgeRパッケージ(Robinson et al., *Bioinformatics*, 2010)のやり方です。

```
#####
### 反復あり(7 technical replicates)
#####
in_f <- "maq_count_table.txt"          #入力ファイル名を指定してin_fに格納
param_G1 <- 7                          #G1群のサンプル数を指定
param_G2 <- 7                          #G2群のサンプル数を指定

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで
data <- as.matrix(data)                 #データの型をmatrixにしている
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトル

#edgeR実行
library(edgeR)                          #パッケージの読み込み
d <- DGEList(counts=data, group=data.cl)#DGEListオブジェクトを作成してdに格納
```

	反復あり2群間比較		反復なし2群間比較	
	TCC(edgeR)	edgeR	TCC(DESeq2)	edgeR
Package version	1.2.0	3.14.0	1.2.0	3.14.0
FDR=5%	7618	7585	0	0
FDR=10%	7843	7822	0	0
FDR=20%	8114	8102	0	0
FDR=30%	8338	8309	0	0
FDR=50%	8621	8607	0	0
FDR=70%	8795	8782	0	0
FDR=80%	8987	8962	200	0

反復あり or なし(DESeq2)

- 反復あり or なし(DESeq2)

DESeq2パッケージ(Love et al., *Genome Biol.*, 2014)のやり方です。

```
#####
### 反復あり(7 technical replicates)
#####
in_f <- "maqc_count_table.txt"           #入力ファイル名を指定してin_fに格納
param_G1 <- 7                           #G1群のサンプル数を指定
param_G2 <- 7                           #G2群のサンプル数を指定

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで
data <- as.matrix(data)                  #データの型をmatrixにしている
data.c1 <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトル

#DESeq2実行
library(DESeq2)                          #パッケージの読み込み
colData <- data.frame(condition=as.factor(data.c1))#condition列にクラスラベル情
```

d d t p p r r	反復あり2群間比較			反復なし2群間比較			
	Package version	TCC(edgeR) 1.2.0	edgeR 3.14.0	DESeq2 1.12.0	TCC(DESeq2) 1.2.0	edgeR 3.14.0	DESeq2 1.12.0
FDR=5%		7618	7585	7908	0	0	0
FDR=10%		7843	7822	8269	0	0	204
FDR=20%		8114	8102	8756	0	0	474
FDR=30%		8338	8309	9069	0	0	656
FDR=50%		8621	8607	9631	0	0	945
FDR=70%		8795	8782	10180	0	0	1239
FDR=80%		8987	8962	10518	200	0	1417

細かい話だが...

- 反復あり or なし(DESeq2)
[DESeq2](#)パッケージ(Love et al., *Genome Biol.*, 2014)のやり方です。

#入力ファイルの読み込みとラベル情報の作成

```
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで  
data <- as.matrix(data) #データの型をmatrixにしている  
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトル
```

#DESeq2実行

```
library(DESeq2) #パッケージの読み込み  
colData <- data.frame(condition=as.factor(data.cl))#condition列にクラスラベル情  
d <- DESeqDataSetFromMatrix(countData=data, colData=colData, design=~condition  
d <- DESeq(d) #DESeq2を実行  
tmp <- results(d) #実行結果を抽出  
p.value <- tmp$pvalue ##p-valueをp.valueに格納  
p.value[is.na(p.value)] <- 1 #NAを1に置換している  
ranking <- rank(p.value) #p.valueでランキングした結果をrankingに格納  
ranking_DESeq2_ari <- ranking #このランキング結果をranking_DESeq2_ariに格納  
q.value <- tmp$padj #adjusted p-valueをq.valueに格納  
q.value[is.na(q.value)] <- 1 #NAを1に置換している  
#q.value <- p.adjust(p.value, method="BH") #q-valueをq.valueに格納  
sum(q.value < 0.05) #FDR = 0.05 (q-value < 0.05)を満たす遺伝子  
sum(q.value < 0.10) #FDR = 0.10 (q-value < 0.10)を満たす遺伝子  
sum(q.value < 0.20) #FDR = 0.20 (q-value < 0.20)を満たす遺伝子  
sum(q.value < 0.30) #FDR = 0.30 (q-value < 0.30)を満たす遺伝子
```

①ちょっと下に移動。②DESeq2は、adjusted p-valueが計算結果に含まれているので、それをq.valueとして利用している。それに対し、TCC (edgeRも?!)はp.value情報をもとにp.adjust関数を用いてq.valueを得ている。それゆえ、③のやり方を採用したDESeq2の結果を示す



反復あり or なし(DESeq2)

① p.adjust関数を用いて得たq.valueによるDESeq2の結果。p-valueレベルではどのパッケージも似た結果を返していることがわかる

反復あり or なし(DESeq2)

DESeq2パッケージ (Love et al., Genome Biol., 2014)のやり方です。DESeq2オリジナルのq-valueではなく、p-valueからp.adjust関数を用いてq.valueを計算しています。

```
#####
### 反復あり(7 technical replicates)
#####
in_f <- "maqc_count_table.txt"      #入力ファイル名を指定してin_fに格納
param_G1 <- 7                       #G1群のサンプル数を指定
param_G2 <- 7                       #G2群のサンプル数を指定

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで
data <- as.matrix(data)              #データの型をmatrixにしている
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベクトル

#DESeq2実行
```

Package version	反復あり2群間比較				反復なし2群間比較			
	TCC(edgeR)	edgeR	DESeq2	DESeq2	TCC(DESeq2)	edgeR	DESeq2	DESeq2
FDR=5%	7618	7585	7908	7306	0	0	0	0
FDR=10%	7843	7822	8269	7547	0	0	204	0
FDR=20%	8114	8102	8756	7823	0	0	474	0
FDR=30%	8338	8309	9069	8021	0	0	656	0
FDR=50%	8621	8607	9631	8301	0	0	945	0
FDR=70%	8795	8782	10180	8529	0	0	1239	0
FDR=80%	8987	8962	10518	8628	200	0	1417	204

まとめ

①反復あり(7 technical replicates)では、大量のDEGが得られ、②反復なし(1 biological replicates; pooled)ではDEGがほとんど得られなかった。この傾向は、反復なしに対応しているどの統計的手法においてもおそらく不変。なぜこういう結果になるのかは統計的手法の計算手順をおさらいすればよい



反復あり2群間比較

Package version	TCC(edgeR)	edgeR	DESeq2	DESeq2
	1.2.0	3.14.0	1.12.0	1.12.0
FDR=5%	7618	7585	7908	7306
FDR=10%	7843	7822	8269	7547
FDR=20%	8114	8102	8756	7823
FDR=30%	8338	8309	9069	8021
FDR=50%	8621	8607	9631	8301
FDR=70%	8795	8782	10180	8529
FDR=80%	8987	8962	10518	8628



反復なし2群間比較

Package version	TCC(DESeq2)	edgeR	DESeq2	DESeq2
	1.2.0	3.14.0	1.12.0	1.12.0
FDR=5%	0	0	0	0
FDR=10%	0	0	204	0
FDR=20%	0	0	474	0
FDR=30%	0	0	656	0
FDR=50%	0	0	945	0
FDR=70%	0	0	1239	0
FDR=80%	200	0	1417	204

赤字部分は、①G1(brain)群内のバラツキと②G2(UHR)群内のバラツキを独立に調べることに相当。この場合は縦軸の距離がせいぜい0.03程度のバラツキだと評価

統計的手法とは

おさらい

■ 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)

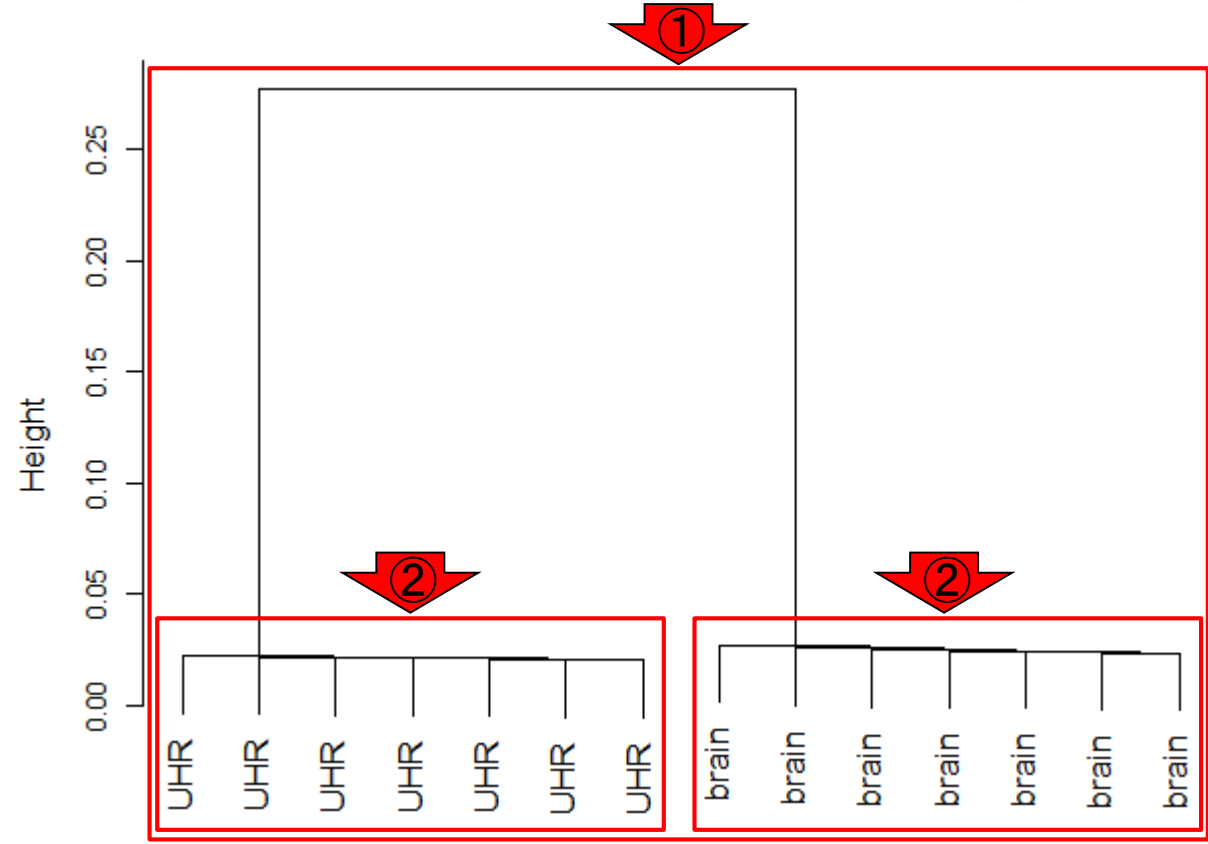


検定は、①比較したいG1群とG2群が②同じ群由来だとした場合(これが帰無仮説)、②同じ群の分布のどのあたりのバラツキの程度のところに位置するかをp値で評価していると思えばよい。②の分布のど真ん中だとp=1、外れるほどp=0に近い値

統計的手法とは

- 同一群内の遺伝子のばらつきの種類に従う分布の全体像を把握して
 - non-DEGのばらつき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が non-DEG分布のどのあたりに位置するかを評価(検定)

おさらい

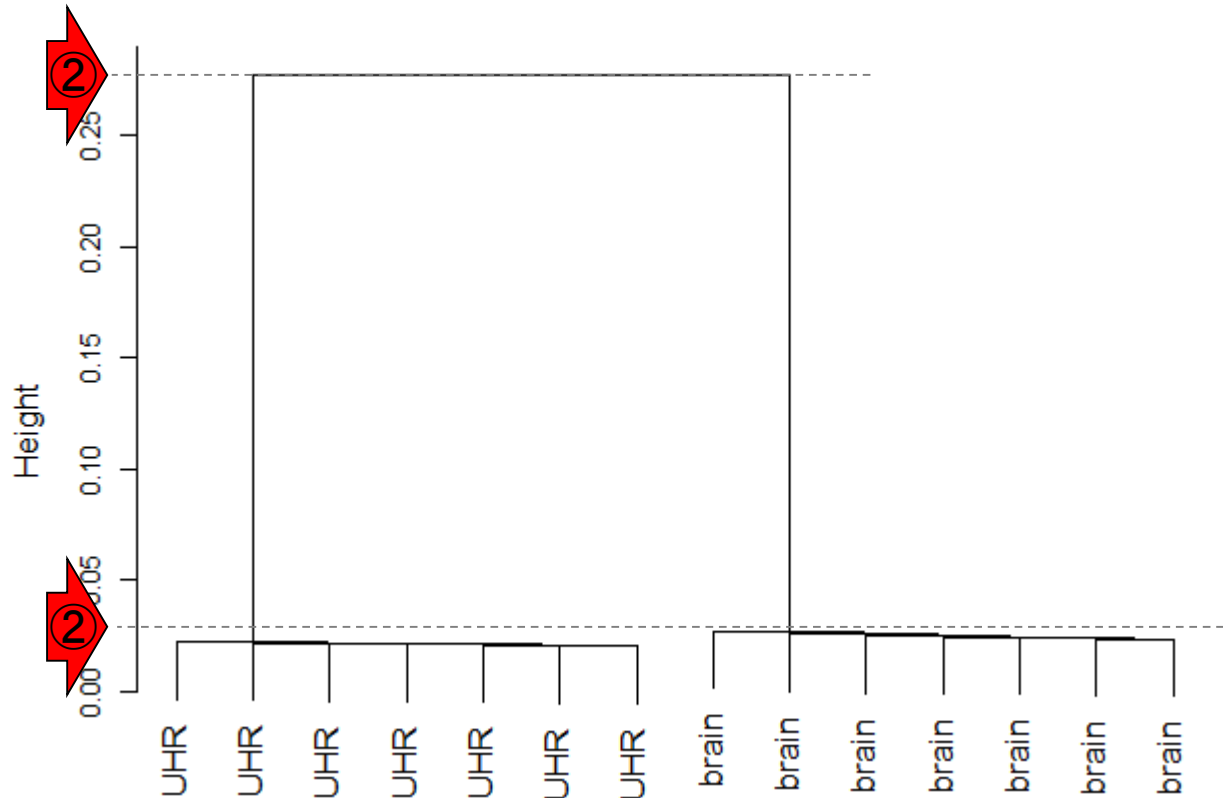


統計的手法とは

このデータの場合、①検定中の2群をマージした場合のバラツキは大きく(類似度が低いので距離が遠い)、②同一群内のバラツキ(類似度が高いので距離が近い)の範囲にはどう見ても収まっていない。
→ p値が1から遠く離れた0に近い値、多数のDEG

おさらい

- 同一群内の遺伝子のばらつきの種類と分布の全体像を把握しておくことと同義
 - non-DEGのばらつき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がnon-DEG分布のどのあたりに位置するかを評価(検定)



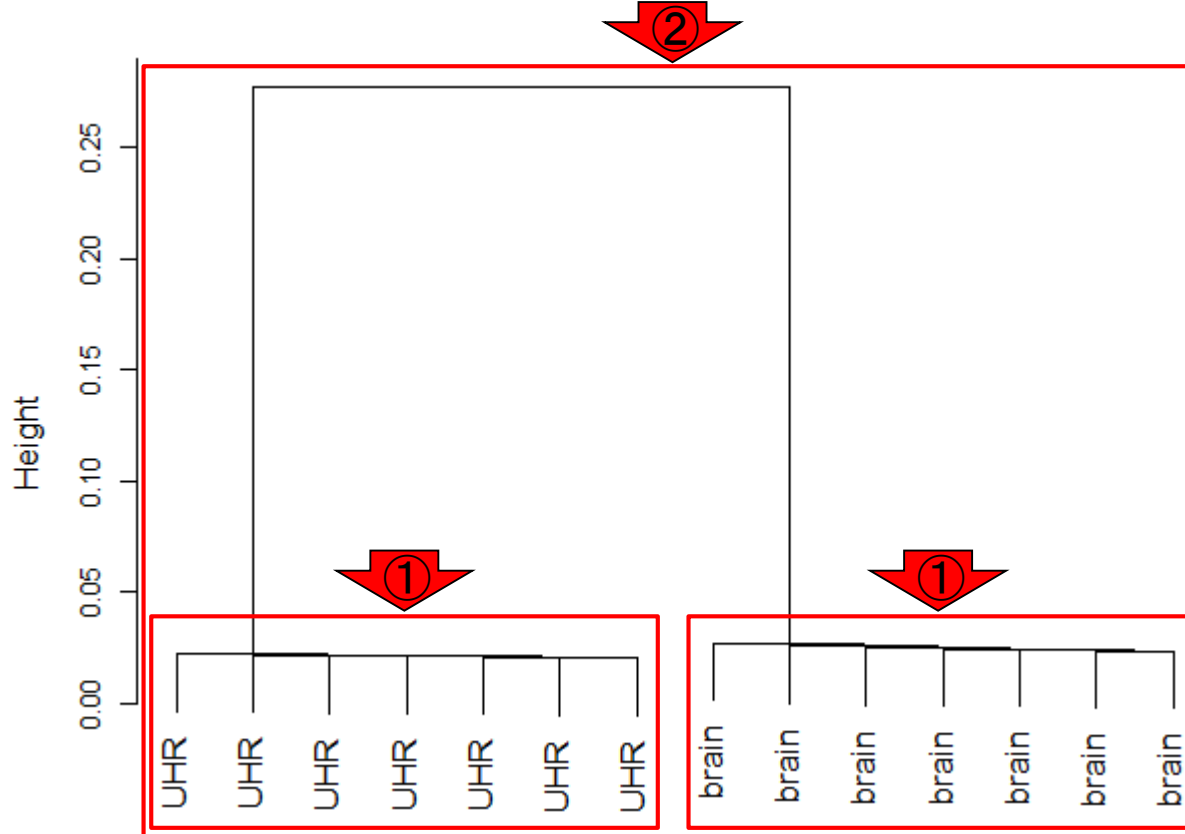
反復なしの場合は

①同一群がマージされて反復なしの場合は、そもそもバラツキを見積もりようがない。それゆえ反復なし非対応の統計的手法を適用するとエラーが出る。反復なしに対応済みのTCC, edgeR, DESeq2は、おそらく②の検定に用いるG1群とG2群を合わせたものを、仮想「反復ありデータの同一群」として取り扱っている

■ 同一群内の遺伝子のばらつきの説に従う分布の全体像を把握して

□ non-DEGのばらつきを把握しておくこと同義

■ 実際に比較したい2群の遺伝子のばらつきがnon-DEG分布のどのあたりに位置するかを評価(検定)



反復なしの場合は

その場合、たとえ②DEGを多く含むデータであったとしても、それをモデル構築に用いるのだから、結果として仮想同一群のバラツキは大きくなる傾向にある。したがって、それを用いて検定してもDEGはほとんど得られないのは至極妥当

- 同一群内の遺伝子のばらつきの程度をモデル構築に用いるのだから、結果として仮想同一群のバラツキは大きくなる傾向にある。したがって、それを用いて検定してもDEGはほとんど得られないのは至極妥当
 - 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)

