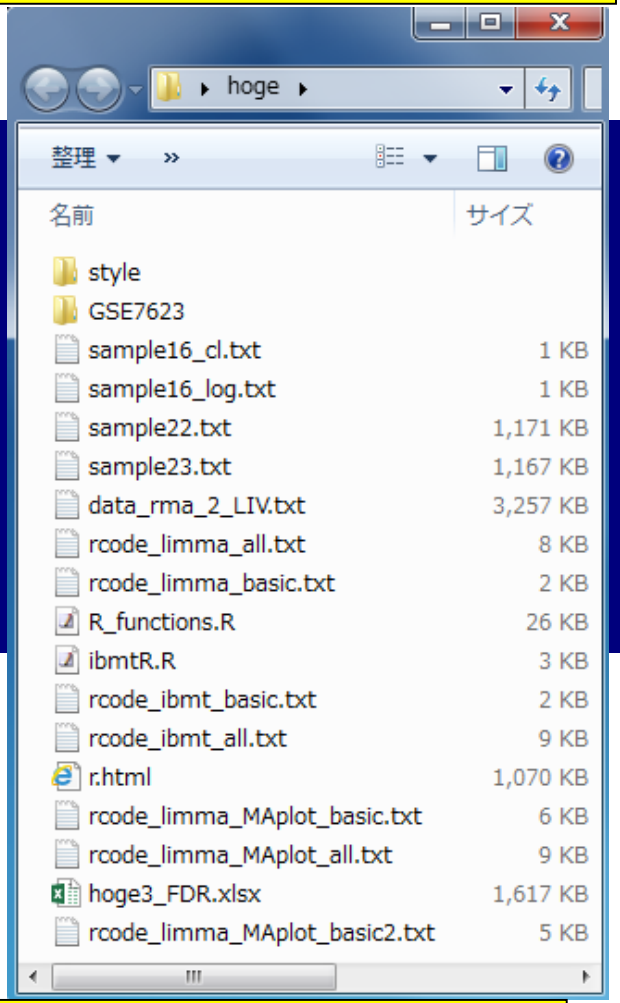
講義室後ろにあるUSBメモリ中のhogeフォルダをデスクトップにコピーしておいてください。



機能ゲノム学 第3回

東京大学大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究ユニット
門田幸二
kadota@iu.a.u-tokyo.ac.jp

前回（5/21）のhogeフォルダがデスクトップに残っているかもしれないのでご注意ください。

講義予定

- 第1回（2014年5月14日）
  - 原理、各種データベース、生データ取得、遺伝子発現行列作成（データ正規化）
  - 教科書の1.2節、2.2節周辺
- 第2回（2014年5月21日）
  - クラスタリング（データ変換や距離の定義など）、実験デザイン、分布
  - 教科書の3.2節周辺
- 第3回（2014年5月28日）
  - 発現変動解析（多重比較問題）、各種プロット（M-A plotや平均-分散プロット）
  - 教科書の3.2節と4.2節周辺
- 第4回（2014年6月4日）
  - 機能解析（Gene Ontology解析やパスウェイ解析）、分類など

授業の目標・概要

細胞中で発現している全転写物（トランスクリプトーム）の解析技術は、マイクロアレイから次世代シーケンサ（RNA-seq）に移行しつつあります。RNA-seqデータ解析の多くは、マイクロアレイの知識を前提としています。また、ニュートリゲノミクス（食品系）分野では、マイクロアレイは現在でも主流派です。**マイクロアレイデータを主な例として**、各種トランスクリプトーム解析手法について解説します。

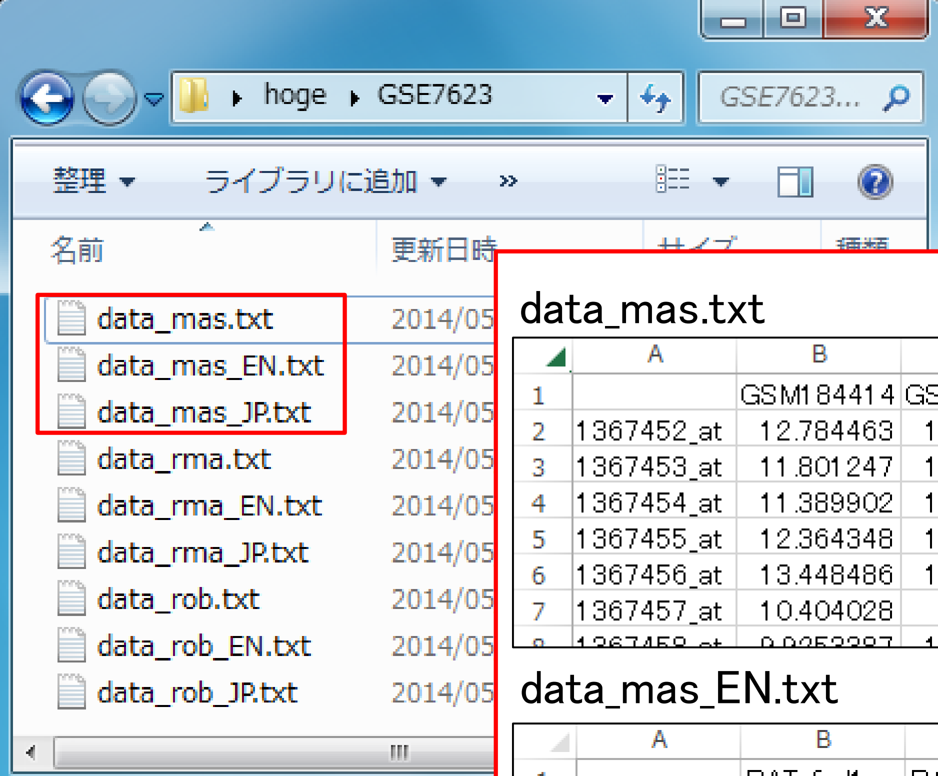
Contents（第3回）

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - 関数の利用法
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - 描画（M-A plot）
    - 作成法
    - 同一群内のばらつき（前処理法間の違い）

遺伝子発現行列データは作成済み

- Affymetrix GeneChip
  - Ge et al., *Genomics*, **86**: 127–141, 2005
    - **GSE2361**、GPL96 (Affymetrix Human Genome U133A Array)、22,283 probesets
    - **ヒト36サンプル**：Heart (心臓)、Thymus (胸腺)、Spleen (脾臓)、Ovary (卵巣)、Kidney (腎臓)、Skeletal Muscle (骨格筋)、Pancreas (膵臓)、Prostate (前立腺)、…
  - Nakai et al., *Biosci Biotechnol Biochem.*, **72**: 139–148, 2008
    - **GSE7623**、GPL1355 (Affymetrix Rat Genome 230 2.0 Array)、31,099 probesets
    - **ラット24サンプル**：Brown adipose tissue (褐色脂肪組織; BAT) 8サンプル、White adipose tissue (白色脂肪組織; WAT) 8サンプル、Liver (肝臓; LIV) 8サンプル
      - BAT 8サンプル：通常 (BAT_fed) 4サンプル　対　24時間絶食 (BAT_fas) 4サンプル
      - WAT 8サンプル：通常 (WAT_fed) 4サンプル　対　24時間絶食 (WAT_fas) 4サンプル
      - LIV 8サンプル：通常 (LIV_fed) 4サンプル　対　24時間絶食 (LIV_fas) 4サンプル
  - Kamei et al., *PLoS One*, **8**: e65732, 2013
    - **GSE30533**、GPL1355 (Affymetrix Rat Genome 230 2.0 Array)、31,099 probesets
    - **ラット10サンプル**：全てLiver (肝臓) サンプル
    - iron-deficient diet (Iron_def) 5サンプル　対　control diet (Control) 5サンプル

hogeフォルダ中に3つの前処理法の実行結果ファイルがあります。
MAS5 (data_mas.txt)、RMA (data_rma.txt)、RMX (data_rob.txt)



GSE7623 (Nakai et al., 2008)の対数変換後のデータ

data_mas.txt

	A	B	C	D	E	F	G	H	I
1		GSM184414	GSM184415	GSM184416	GSM184417	GSM184418	GSM184419	GSM184420	GSM184
2	1367452_at	12.784463	12.447082	12.805908	12.304718	12.589425	12.607532	11.815378	12.43
3	1367453_at	11.801247	12.152935	11.942227	11.968477	11.845375	11.681727	12.078672	12.04
4	1367454_at	11.389902	11.160757	11.145987	11.212088	11.540652	11.308877	11.49885	11.40
5	1367455_at	12.364348	12.529744	12.432574	12.604011	12.441991	12.249935	12.281827	12.19
6	1367456_at	13.448486	13.543046	13.552794	13.629799	13.36913	13.244278	13.424371	13.32
7	1367457_at	10.404028	10.69632	10.475078	10.45579	10.141921	10.290666	10.146529	10.2
8	1367458_at	9.9253387	10.244544	9.9720008	9.9576072	8.702884	9.3578792	9.2134367	9.499

data_mas_EN.txt

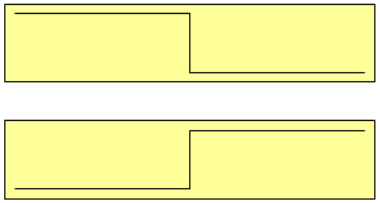
	A	B	C	D	E	F	G	H	I
1		BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fa
2	1367452_at	12.784463	12.447082	12.805908	12.304718	12.589425	12.607532	11.815378	12.43
3	1367453_at	11.801247	12.152935	11.942227	11.968477	11.845375	11.681727	12.078672	12.04
4	1367454_at	11.389902	11.160757	11.145987	11.212088	11.540652	11.308877	11.49885	11.40
5	1367455_at	12.364348	12.529744	12.432574	12.604011	12.441991	12.249935	12.281827	12.19
6	1367456_at	13.448486	13.543046	13.552794	13.629799	13.36913	13.244278	13.424371	13.32
7	1367457_at	10.404028	10.69632	10.475078	10.45579	10.141921	10.290666	10.146529	10.2
8	1367458_at	9.9253387	10.244544	9.9720008	9.9576072	8.702884	9.3578792	9.2134367	9.499

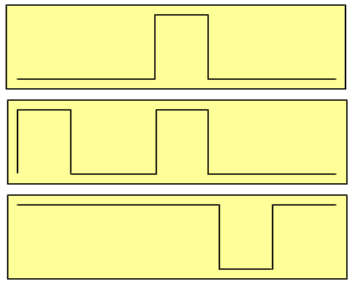
data_mas_JP.txt

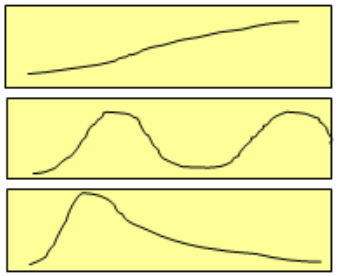
	A	B	C	D	E	F	G
1		褐色脂肪_満腹1	褐色脂肪_満腹2	褐色脂肪_満腹3	褐色脂肪_満腹4	褐色脂肪_空腹1	褐色脂肪_空腹
2	1367452_at	12.7844634	12.44708219	12.80590758	12.30471769	12.58942538	12.6075319
3	1367453_at	11.80124704	12.15293493	11.94222741	11.96847729	11.84537542	11.6817274
4	1367454_at	11.38990178	11.16075717	11.14598707	11.21208786	11.54065185	11.3088766
5	1367455_at	12.36434768	12.52974368	12.43257392	12.60401124	12.44199125	12.2499348
6	1367456_at	13.44848649	13.54304603	13.55279359	13.62979898	13.36912977	13.2442783
7	1367457_at	10.40402803	10.69631952	10.47507777	10.4557902	10.14192076	10.2906657
8	1367458_at	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

データ解析もいろいろ

発現変動遺伝子同定







遺伝子発現行列

二群間比較用

	A群 A1	A2	…	B群 B1	B2	…
gene *1*	$x^A_{1,1}$	$x^A_{1,2}$		$x^B_{1,2}$	$x^B_{1,2}$	
gene *2*	$x^A_{2,1}$	$x^A_{2,2}$		$x^B_{2,2}$	$x^B_{2,2}$	
…	…	…	…	…	…	…
gene *i*	$x^A_{i,1}$	$x^A_{i,2}$		$x^B_{i,2}$	$x^B_{i,2}$	
…	…	…	…	…	…	…
gene *n*	$x^A_{n,1}$	$x^A_{n,2}$		$x^B_{n,2}$	$x^B_{n,2}$	

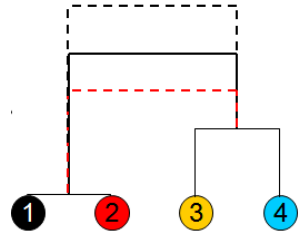
様々な組織（条件）

	S1	S2	S3	S4	…
gene *1*	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	…
gene *2*	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	…
…	…	…	…	…	…
gene *i*	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	
…	…	…	…	…	…
gene *n*	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$x_{n,4}$	…

時系列データ

	T1	T2	T3	T4	…
gene *1*	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	…
gene *2*	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	
…	…	…	…	…	
gene *i*	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	
…	…	…	…	…	…
gene *n*	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$x_{n,4}$	…

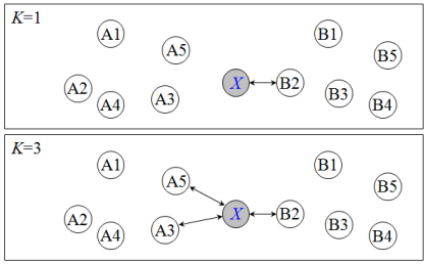
クラスタリング



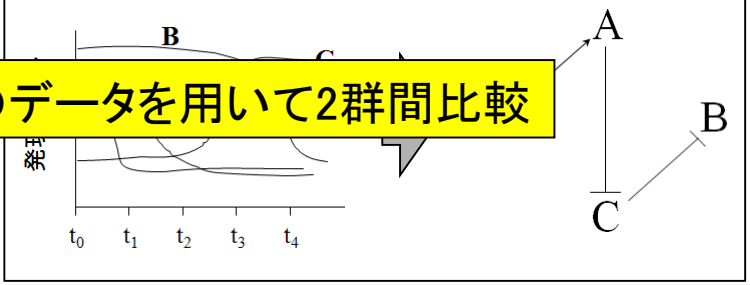
機能解析

- Gene Ontology（GO）
- パスウェイ解析

分類（診断）



遺伝子ネットワーク推定

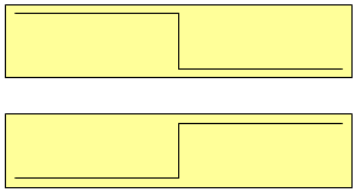


対数変換後のデータを用いて2群間比較

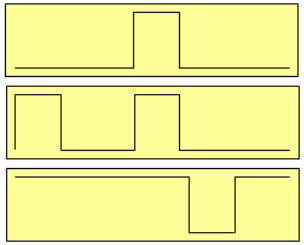
バイオインフォマティクス要素技術

- 相関係数やエントロピーなどの応用例を紹介

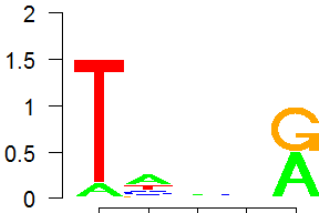
二群間比較



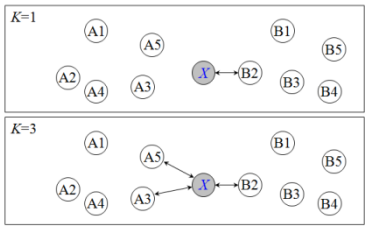
組織特異的遺伝子



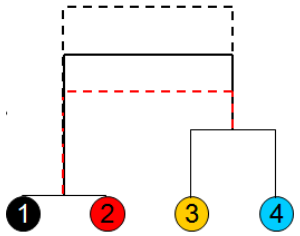
Sequence logo



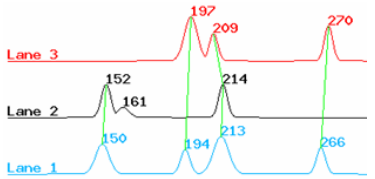
分類（診断）



クラスタリング



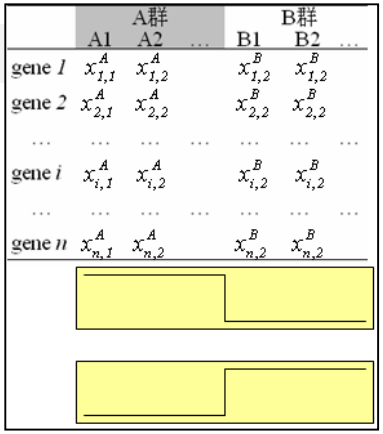
同一ピーク同定

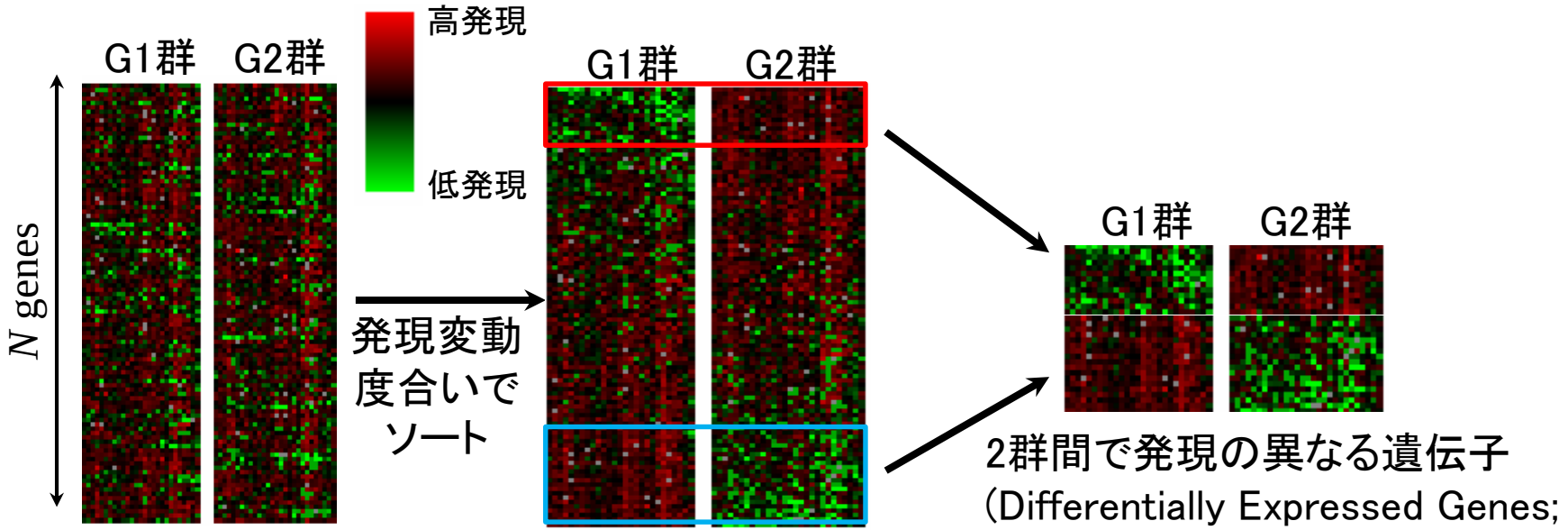


基本スキルのみでいろいろなことができます

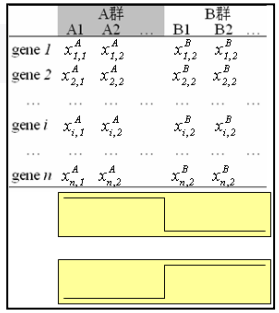
2群間比較

- 農産物の栽培条件の違い（通常 vs. 低温、通常 vs. 乾燥）
- 味の違い（おいしい vs. まずい）
- サンプルの状態の違い（癌 vs. 正常）





比較したいグループ間で発現変動している遺伝子
または転写物を同定することはデータ解析の基本



2群間比較

- パターンマッチング法
  - 理想的なパターンyとの類似度が高い順にランキング

$$\text{相関係数 } r = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y})^2}} \qquad (-1 \le r \le 1)$$

y	1	1	1	1	1	1	0	0	0	0	0

rownan	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5	r
gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09	0.98
gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17	0.81
gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61	−0.71

相関係数rの絶対値が大きいほど発現変動の度合いが大きいと解釈

(Rで)マイクロアレイデータ解析

(last modified 2014/05/23, since 2005)

- 解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法

- 解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Rank products (Breitling_2004) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | samroc (Broberg_2003) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | SAM (Tusher_2001) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Student's t-test(last modified 2014/05/23) NEW
- 解析 | 発現変動 | 2群間 | 対応なし | Welch t-test(last modified 2014/05/23) NEW
- 解析 | 発現変動 | 2群間 | 対応なし | Mann-Whitney U-test(last modified 2013/10/15)
- 解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法(last modified 2014/05/23) NEW
- 解析 | 発現変動 | 2群間 | 対応あり | について(last modified 2009/11/11)
- 解析 | 発現変動 | 2群間 | 対応あり | SAM (Tusher_2001) (last modified 2013/6/2)
- 解析 | 発現変動 | 2群間
- 解析 | 発現変動 | 2群間
- 解析 | 発現変動 | 2群間
- 解析 | 発現変動 | 3群間
- 解析 | 発現変動 | 3群間
- 解析 | 発現変動 | 3群間

テンプレートパターン情報を含むファイルを読み込んでパターンマッチングを行ってみよう

解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法 NEW

パターンマッチング法を用いて、2群間での発現変動遺伝子の同定を行うやり方を紹介します。
「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16のsample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用するやり方です。

```
in_f1 <- "sample16_log.txt"              #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"               #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                     #出力ファイル名を指定してout_fに格納
param <- "pearson"                       #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                      #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテンプレートパターンdata.clと

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合した結果をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

rownan	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5
gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09
gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17
gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61

G1_1	1
G1_2	1
G1_3	1
G1_4	1
G1_5	1
G1_6	1
G2_1	0
G2_2	0
G2_3	0
G2_4	0
G2_5	0

解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法 NEW

パターンマッチング法を用いて、2群間での発現変動遺伝子の同定を行うやり方を紹介します。
「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16の sample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用するやり方です。

```
in_f1 <- "sample16_log.txt"                #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"                 #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                       #出力ファイル名を指定してout_fに格納
param <- "pearson"                         #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                        #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテンプレートパターンdata.clと

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合した結果をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

apply関数は便利です

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	rownan	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5	r
2	gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09	0.984
3	gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17	0.811
4	gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61	-0.708

パターンマッチング法：詳細を解説

解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法 NEW

パターンマッチング法を用いて、2群間での発現変動遺伝子の同定を行うやり方を紹介します。
「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16のsample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用するやり方です。

```
in_f1 <- "sample16_log.txt"                    #入力ファイルを指定して in_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"                     #入力ファイルを指定して in_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                           #出力ファイルを指定して out_fに格納
param <- "pearson"                             #"pearson"または"spearman")

#入力ファイルの読み込みと
data <- read.table(in_f1
hoge <- read.table(in_f2
data.cl <- hoge[,2]

#本番
r <- apply(data, 1, cor,

#ファイルに保存
tmp <- cbind(rownames(da
write.table(tmp, out_f,
```

切り取り(T)
コピー(C)
貼り付け

rownan	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5
gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09
gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17
gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61

入力ファイルの読み込みがうまくできていることがわかる

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> in_f1 <- "sample16_log.txt"                    #入力ファイル名を指$
> in_f2 <- "sample16_cl.txt"                     #入力ファイル名を指$
> out_f <- "hoge1.txt"                           #出力ファイル名を指$
> param <- "pearson"                             #相関係数の種類を指$
>
> #入力ファイルの読み込みとラベル情報の作成
> data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t$
> data
       G1_1 G1_2 G1_3 G1_4 G1_5 G1_6 G2_1 G2_2 G2_3 G2_4 G2_5
gene1 6.44 6.30 6.51 6.36 6.49 6.39 3.58 4.39 4.25 3.70 4.09
gene2 5.81 6.93 6.73 5.55 6.39 6.61 2.81 5.46 1.00 3.46 4.17
gene3 3.91 4.81 5.04 3.17 4.75 5.36 5.58 5.52 5.70 5.64 5.61
>
```

パターンマッチング法：詳細を解説

G1_1	1
G1_2	1
G1_3	1
G1_4	1
G1_5	1
G1_6	1
G2_1	0
G2_2	0
G2_3	0
G2_4	0
G2_5	0

解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法 NEW

パターンマッチング法を用いて、2群間での発現変動遺伝子の同定を行うやり方を紹介します。
「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16のsample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用するやり方です。

```
in_f1 <- "sample16_log.txt"          #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"           #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                 #出力ファイル名を指定してout_fに格納
param <- "pearson"                   #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                  #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテンプレートパター

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合した結果をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

読み込み時にheader=TRUEやrow.names=1の記述がないことに注意!

R Console

```
> hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で$
> data.cl <- hoge[,2]                          #テンプレートパ$
> data.cl
 [1] 1 1 1 1 1 1 0 0 0 0 0
>
```

パターンマッチング法：詳細を解説

解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法 NEW

パターンマッチング法を用いて、2群間での発現変動遺伝子の同定を行うやり方を紹介します。
「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16のsample16_log.txt(対数変換後のデータ)
クラスラベル情報ファイル(sample16_cl.txt)を利用するやり方

```
in_f1 <- "sample16_log.txt"              #入力フ
in_f2 <- "sample16_cl.txt"               #入力フ
out_f <- "hoge1.txt"                     #出力フ
param <- "pearson"                       #相関係

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.nam
hoge <- read.table(in_f2, sep="\t", quote="")#
data.cl <- hoge[,2]                      #テンプ

#本番
r <- apply(data, 1, cor, y=data.cl, method=par

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力デ
write.table(tmp, out_f, sep="\t", append=F, qu
```

R Console

```
> hoge
      V1 V2
1   G1_1  1
2   G1_2  1
3   G1_3  1
4   G1_4  1
5   G1_5  1
6   G1_6  1
7   G2_1  0
8   G2_2  0
9   G2_3  0
10  G2_4  0
11  G2_5  0
> dim(hoge)
[1] 11  2
> hoge[1,]
    V1 V2
1 G1_1  1
> hoge[,2]
 [1] 1 1 1 1 1 1 0 0 0 0 0
>
```

G1_1	1
G1_2	1
G1_3	1
G1_4	1
G1_5	1
G1_6	1
G2_1	0
G2_2	0
G2_3	0
G2_4	0
G2_5	0

hogeの中身は、入力ファイルと同じだが、欲しいのはhogeオブジェクトの2列目部分

パターンマッチング法：詳細を解説

G1_1	1
G1_2	1
G1_3	1
G1_4	1
G1_5	1
G1_6	1
G2_1	0
G2_2	0
G2_3	0
G2_4	0
G2_5	0

解析 | 発現変動 | 2群間 | 対応なし | パターンマッチング法 NEW

パターンマッチング法を用いて、2群間での発現変動遺伝子の同定を行うやり方を紹介します。
「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16のsample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用するやり方です。

```
in_f1 <- "sample16_log.txt"          #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"           #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                 #出力ファイル名を指定してout_fに格納
param <- "pearson"                   #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row
hoge <- read.table(in_f2, sep="\t", quote=
data.cl <- hoge[,2]                  #テ

#本番
r <- apply(data, 1, cor, y=data.cl, method

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入
write.table(tmp, out_f, sep="\t", append=F
```

R Console

```
> hoge <- read.table(in_f2, row.names=1, sep="\t", quote="")
> hoge
      V2
G1_1   1
G1_2   1
G1_3   1
G1_4   1
G1_5   1
G1_6   1
G2_1   0
G2_2   0
G2_3   0
G2_4   0
G2_5   0
> data.cl <- hoge[,1]
> data.cl
 [1] 1 1 1 1 1 1 0 0 0 0 0
>
```

読み込み時にrow.names=1をつけて、こんな風にしてもよい

パターンマッチング法：詳細を解説

```
in_f1 <- "sample16_log.txt"              #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"               #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                     #出力ファイル名を指定してout_fに格納
param <- "pearson"                       #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                      #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテン

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合した結果をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

apply関数は行ごとや列ごとに同じ関数を繰り返して実行させたい場合に便利です

R Console

```
> #本番          ①    ②   ③        ④                ⑤
> r <- apply(data, 1, cor, y=data.cl, method=param)#$
> r
     gene1       gene2       gene3
 0.9839815   0.8107810 -0.7068671
>
```

①dataオブジェクトの、②各行に対して、③cor関数を適用せよ。その際、④テンプレートyはdata.clとし、⑤相関係数の種類はparamで指定したものとする

パターンマッチング法：詳細を解説

```
in_f1 <- "sample16_log.txt"              #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"               #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                     #出力ファイル名を指定してout_fに格納
param <- "pearson"                       #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                      #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテン

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合した結果をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

as.numeric関数は、data.clオブジェクトとデータの型を揃える目的で利用している

R Console

```
> r
     gene1      gene2      gene3
 0.9839815  0.8107810 -0.7068671
> cor(as.numeric(data[1,]), data.cl, method=param)
[1] 0.9839815
> cor(as.numeric(data[2,]), data.cl, method=param)
[1] 0.810781
> cor(as.numeric(data[3,]), data.cl, method=param)
[1] -0.7068671
>
```

パターンマッチング法：詳細を解説

```
in_f1 <- "sample16_log.txt"              #入力ファイル名を指定してin_f1に格納(発現データ)
in_f2 <- "sample16_cl.txt"               #入力ファイル名を指定してin_f2に格納(テンプレート情報)
out_f <- "hoge1.txt"                     #出力ファイル名を指定してout_fに格納
param <- "pearson"                       #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                      #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテン

#ファイルに保存
tmp <- cb
write.tab
```

as.numeric関数は、data.clオブジェクトとデータの型を揃える目的で利用している

R Console

```
> as.numeric(data[1,])
 [1] 6.44 6.30 6.51 6.36 6.49 6.39 3.58 4.39 4.25 3.70 4.09
> data.cl
 [1] 1 1 1 1 1 1 0 0 0 0 0
> cor(data[1,], data.cl, method=param)
 以下にエラー cor(data[1, ], data.cl, method = param) :  互換性のない次元です
> cor(as.vector(data[1,]), data.cl, method=param)
 以下にエラー cor(as.vector(data[1, ]), data.cl, method = param) :
   互換性のない次元です
> data[1,]
      G1_1 G1_2 G1_3 G1_4 G1_5 G1_6 G2_1 G2_2 G2_3 G2_4 G2_5
gene1 6.44  6.3 6.51 6.36 6.49 6.39 3.58 4.39 4.25  3.7 4.09
> as.vector(data[1,])
      G1_1 G1_2 G1_3 G1_4 G1_5 G1_6 G2_1 G2_2 G2_3 G2_4 G2_5
gene1 6.44  6.3 6.51 6.36 6.49 6.39 3.58 4.39 4.25  3.7 4.09
>
```

パターンマッチング法：詳細を解説

R Console

```
> tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合し$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定した$
> tmp
      rownames(data) G1_1 G1_2 G1_3 G1_4 G1_5 G1_6 G2_1 G2_2 G2_3 G2_4 G2_5          r
gene1          gene1 6.44 6.30 6.51 6.36 6.49 6.39 3.58 4.39 4.25 3.70 4.09  0.9839815
gene2          gene2 5.81 6.93 6.73 5.55 6.39 6.61 2.81 5.46 1.00 3.46 4.17  0.8107810
gene3          gene3 3.91 4.81 5.04 3.17 4.75 5.36 5.58 5.52 5.70 5.64 5.61 -0.7068671
>
```

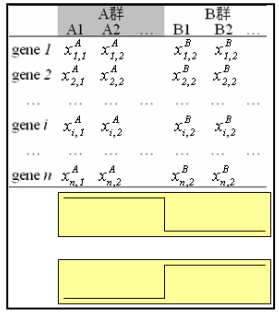
```
in_f1 <- "sa
in_f2 <- "sa                                    #入力ファイル名を指定してin_f2に格納(テンプレートパターン情報)
out_f <- "hoge1.txt"                            #出力ファイル名を指定してout_fに格納
param <- "pearson"                              #相関係数の種類を指定("pearson"または"spearman")

#入力ファイルの読み込みとラベル情報の作成
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイルの読
hoge <- read.table(in_f2, sep="\t", quote="")#in_f2で指定したファイルの読み込み
data.cl <- hoge[,2]                             #テンプレートパターンベクトルdata.clを作成

#本番
r <- apply(data, 1, cor, y=data.cl, method=param)#各（行）遺伝子についてテンプレートパターンdata.clと

#ファイルに保存
tmp <- cbind(rownames(data), data, r)  #入力データの右側に相関係数rのベクトルを結合した結果をtmpに格納
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	rownan	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5	r
2	gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09	0.984
3	gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17	0.811
4	gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61	−0.708

Contents（第3回）

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - 関数の利用法
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - 描画（M-A plot）
    - 作成法
    - 同一群内のばらつき（前処理法間の違い）

2群間比較：Student's t-test



(Rで)マイクロアレイデータ解析

(last modified 2014/05/23, since 2005)

- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | samroc (Broberg_2003) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | SAM (Tusher_2001) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Student's t-test (last modified 2014/05/23) NEW
- 解析 | 発現変動 | 2群間 | 対応なし | Welch t-test (last modified 2014/05/23) NEW
- 解析 | 発現変動 | 2群間 | 対応なし | Mann-Whitney U-test (last modified 2013/10/15)

解析 | 発現変動 | 2群間 | 対応なし | Student's t-test NEW

等分散性を仮定したt検定を用いて、2群間での発現変動遺伝子の同定を行うやり方を示します。

2. サンプルデータ16のsample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用しないやり方です。

```
in_f <- "sample16_log.txt"                #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.txt"                      #出力ファイル名を指定してout_fに格納
param_G1 <- 6                             #G1群のサンプル数を指定
param_G2 <- 5                             #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
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#

#等分散性を仮定（var.equal=T）してt.testを行い、t統計量とp-valueの値を返す関数Students_ttestを作成。
Students_ttest <- function(x, cl){
    x.class1 <- x[(cl == 1)]              #ラベルが1のものをx.class1に格納
    x.class2 <- x[(cl == 2)]              #ラベルが2のものをx.class2に格納
    if((sd(x.class1)+sd(x.class2)) == 0){#両方の群の標準偏差が共に0の場合は計算できないので...
        stat <- 0                         #統計量を0
        pval <- 1                         #p値を1
        return(c(stat, pval))             #として結果を返す
    }
    else{                                 #G1，G2どちらかの群の標準偏差が0(上記条件以外)の場合は
```

このウェブページを用いたDEG検出手順の一般的なグループ指定方法です

2群間比較：Student's t-test

rownan	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5
gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09
gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17
gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61

2. サンプルデータ16の sample16_log.txt(対数変換後のデータ)の場合:

クラスラベル情報ファイル(sample16_cl.txt)を利用しないやり方です。

```
in_f <- "sample16_log.txt"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge2.txt"                          #出力ファイル名を指定してout_fに格納
param_G1 <- 6                                 #G1群のサンプル数を指定
param_G2 <- 5                                 #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
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

#等分散性を仮定（var.equal=T）してt.testを行い、t統計量とp-valueの値を返す関数Students_ttestを作成。
Students_ttest <- function(x, cl){
    x.class1 <- x[(cl == 1)]                  #ラベルが1のものをx.class1に格納
    x.class2 <- x[(cl == 2)]                  #ラベルが2のものをx.class2に格納
    if((sd(x.class1)+sd(x.class2)) == 0){#両方の群の標準偏差が共に0の場合は計算できないので...
        stat <- 0                             #統計量を0
```

rownar	G1_1	G1_2	G1_3	G1_4	G1_5	G1_6	G2_1	G2_2	G2_3	G2_4	G2_5	p.value	q.value	ranking
gene1	6.44	6.30	6.51	6.36	6.49	6.39	3.58	4.39	4.25	3.70	4.09	4.77E-08	1.43E-07	1
gene2	5.81	6.93	6.73	5.55	6.39	6.61	2.81	5.46	1.00	3.46	4.17	0.002465	0.003697	2
gene3	3.91	4.81	5.04	3.17	4.75	5.36	5.58	5.52	5.70	5.64	5.61	0.015006	0.015006	3

gene1のような比較するグループ間（G1群 対 G2群）で明らかに発現の異なる遺伝子（DEG）の*p*値は0に近い値となり、明らかに発現変動遺伝子ではないもの（non-DEG）の*p*値は1に近い値になるという基本的な感覚は重要

2群間比較：Student's t-test

解析 | 発現変動 | 2群間 | 対応なし | Student's t-test NEW

3. サンプルデータ22のsample22.txtの場合:

10000行×6列分の標準正規分布に従う乱数です。G1群3サンプル vs. G2群3サンプルの2群間比較として解析を行っています。乱数を発生させただけのデータなので、発現変動遺伝子(DEG)がない全てがnon-DEGのデータです。

```
in_f <- "sample22.txt"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge3.txt"                      #出力ファイル名を指定してout_fに格納
param_G1 <- 3                             #G1
param_G2 <- 3                             #G2

#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.
data.cl <- c(rep(1, param_G1), rep(2, para

#等分散性を仮定（var.equal=T）してt.testを行
Students_ttest <- function(x, cl){
    x.class1 <- x[(cl == 1)]              #ラ
    x.class2 <- x[(cl == 2)]              #ラ
    if((sd(x.class1)+sd(x.class2)) == 0){#
        stat <- 0                         #統
        pval <- 1                         #p
        return(c(stat, pval))             #とりあえず結果を返す
    }
    else{
        hoge <- t.test(x.class1,
        return(c(hoge$statistic,
    }
```

	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3
gene_1	−0.4458	0.9471	0.3047	−0.6163	−0.2745	0.0957
gene_2	−1.2059	−0.3745	0.2449	1.0468	−1.6776	−0.0616
gene_3	0.0411	−0.2297	−0.6740	1.9160	−1.2744	1.8971
gene_4	0.6394	−0.0078	0.7516	−0.3792	1.7884	−1.2589
gene_5	−0.7866	1.1538	−0.0819	0.5820	−0.5949	2.5911
gene_6	−0.3855	−2.4745	1.1373	−1.6164	0.4605	−0.6176
gene_7	−0.4759	−1.8536	0.0646	1.7821	−0.0340	1.3479
gene_8	0.7198	0.8058	−2.3137	−0.2549	−0.6822	−0.6548
gene_9	−0.0185	−1.5977	−2.0602	1.1553	0.6388	2.0476
gene_10	−1.3731	0.0591	0.2841	−0.0105	0.2229	0.2152
gene_11	−0.9824	−0.5004	−1.3967	−1.3212	0.3319	−2.1954

N = 10,000遺伝子（行）からなる遺伝子発現行列（各群3サンプル）を入力として、遺伝子ごとにt-testを実行し、$p < 0.05$を満たす遺伝子数を眺めることを通じて多重比較問題を実感する

2群間比較：Student's t-test

3. サンプルデータ22のsample22.txtの場合:

10000行×6列分の標準正規分布に従う乱数です。G1群3サンプル vs. G2群3サンプルの2群間比
います。乱数を発生させただけのデータなので、発現変動遺伝子(DEG)がない全てがnon-DEG

```
in_f <- "sample22.txt"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge3.txt"                      #出力ファイル名を指定してout_f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

#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$
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベク$

#等分散性を仮定
Students_ttes
    x.class1
    x.class2
    if((sd(x
        stat
        pval
        retur
    }
    else{
        hoge
        retur
    }
```

切り取り(T)
コピー(C)
貼り付け

	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3
gene_1	−0.4458	0.9471	0.3047	−0.6163	−0.2745	0.0957
gene_2	−1.2059	−0.3745	0.2449	1.0468	−1.6776	−0.0616
gene_3	0.0411	−0.2297	−0.6740	1.9160	−1.2744	1.8971
gene_4	0.6394	−0.0078	0.7516	−0.3792	1.7884	−1.2589
gene_5	−0.7866	1.1538	−0.0819	0.5820	−0.5949	2.5911
gene_6	−0.3855	−2.4745	1.1373	−1.6164	0.4605	−0.6176
gene_7	−0.4759	−1.8536	0.0646	1.7821	−0.0340	1.3479

R Console

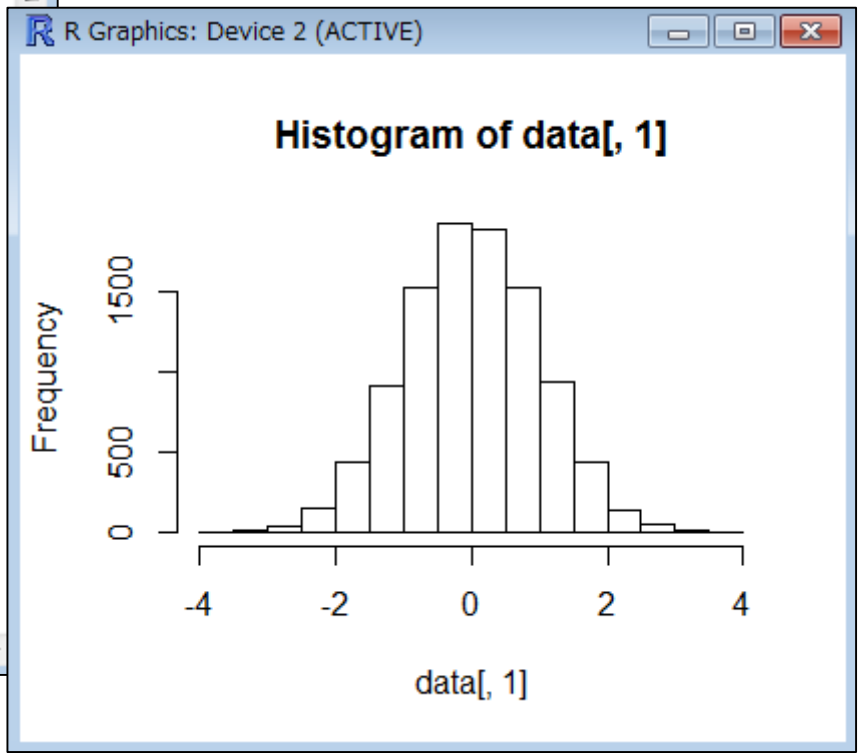
```
> in_f <- "sample22.txt"                    #入力ファイル名を指定してin_fに格納
> out_f <- "hoge3.txt"                      #出力ファイル名を指定してout_fに格納
> param_G1 <- 3                             #G1群のサンプル数を指定
> param_G2 <- 3                             #G2群のサンプル数を指定
>
> #入力ファイルの読み込みとラベル情報の作成
> data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_f$
> data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベク$
> dim(data)
[1] 10000     6
> head(data)
            G1_rep1      G1_rep2     G1_rep3    G2_rep1    G2_rep2     G2_rep3
gene_1 -0.44577826  0.947085174  0.30471680 -0.6162540 -0.2744629  0.09569090
gene_2 -1.20585657 -0.374498928  0.24492090  1.0468297 -1.6776099 -0.06155844
gene_3  0.04112631 -0.229714096 -0.67405000  1.9160200 -1.2743934  1.89711799
gene_4  0.63938841 -0.007755371  0.75159380 -0.3791546  1.7883961 -1.25888353
gene_5 -0.78655436  1.153
gene_6 -0.38548930 -2.474
> data.cl
[1] 1 1 1 2 2 2
>
```

data.clオブジェクトは、テンプレートパターンのようなもの。入力データに相当するdataオブジェクトの1-3列がG1群、4-6列がG2群由来サンプルだということを指し示すクラスラベル情報

2群間比較：Student's t-test

R Console

```
> summary(data)
    G1_rep1            G1_rep2            G1_rep3
 Min.   :-3.561788   Min.   :-4.6296   Min.   :-3.97209
 1st Qu.:-0.673850   1st Qu.:-0.6694   1st Qu.:-0.65026
 Median :-0.001270   Median :-0.0038   Median : 0.01834
 Mean   : 0.001152   Mean   :-0.0036   Mean   : 0.02688
 3rd Qu.: 0.679024   3rd Qu.: 0.6595   3rd Qu.: 0.69573
 Max.   : 3.739140   Max.   : 3.9571   Max.   : 4.31515
    G2_rep1            G2_rep2            G2_rep3
 Min.   :-3.63480   Min.   :-4.096237   Min.   :-3.878653
 1st Qu.:-0.68904   1st Qu.:-0.676368   1st Qu.:-0.655729
 Median :-0.01723   Median :-0.012488   Median :-0.004660
 Mean   :-0.01278   Mean   :-0.003493   Mean   : 0.004555
 3rd Qu.: 0.65439   3rd Qu.: 0.659750   3rd Qu.: 0.674780
 Max.   : 3.83720   Max.   : 3.514435   Max.   : 3.767659
> hist(data[,1])
>
```



確かに正規分布乱数になっている

R Console

```
> head(data)
           G1_rep1      G1_rep2     G1_rep3    G2_rep1    G2_rep2     G2_rep3
gene_1 -0.44577826  0.947085174  0.30471680 -0.6162540 -0.2744629  0.09569090
gene_2 -1.20585657 -0.374498928  0.24492090  1.0468297 -1.6776099 -0.06155844
gene_3  0.04112631 -0.229714096 -0.67405000  1.9160200 -1.2743934  1.89711799
gene_4  0.63938841 -0.007755371  0.75159380 -0.3791546  1.7883961 -1.25888353
gene_5 -0.78655436  1.153805556 -0.08188575  0.5819835 -0.5949475  2.59110024
gene_6 -0.38548930 -2.474468731  1.13727344 -1.6164433  0.4604567 -0.61764893
> data[1,]
          G1_rep1   G1_rep2   G1_rep3   G2_rep1    G2_rep2   G2_rep3
gene_1 -0.4457783 0.9470852 0.3047168 -0.616254 -0.2744629 0.0956909
> data[1, data.cl==1]
          G1_rep1   G1_rep2   G1_rep3
gene_1 -0.4457783 0.9470852 0.3047168
> data[1, data.cl==2]
         G2_rep1    G2_rep2   G2_rep3
gene_1 -0.616254 -0.2744629 0.0956909
> t.test(data[1, data.cl==1], data[1, data.cl==2], var.equal=T)

        Two Sample t-test

data:  data[1, data.cl == 1] and data[1, data.cl == 2]
t = 1.1808, df = 4, p-value = 0.3031
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.7211295  1.7884960
sample estimates:
 mean of x  mean of y
 0.2686746 -0.2650087

>
```

1行目の遺伝子の*p*値は0.05未満ではない
2行目、3行目、…

2群間比較：Student's t-test

3. サンプルデータ22のsample22.txtの場合:

10000行×6列分の標準正規分布に従う乱数です。G1群3サンプル vs. G2群3サンプルの2群間比較として解析を行っています。乱数を発生させただけのデータなので、発現変動遺伝子(DEG)がない全てがnon-DEGのデータです。

```
in_f <- "sample22.txt"
out_f <- "hoge3.txt"
param_G1 <- 3
param_G2 <- 3

#入力ファイルの読み込みと
data <- read.table(in_f,
data.cl <- c(rep(1, para

#等分散性を仮定（var.equa
Students_ttest <- functi
    x.class1 <- x[(cl ==
    x.class2 <- x[(cl ==
    if((sd(x.class1)+sd(
        stat <- 0
        pval <- 1
        return(c(stat, p
    }
    else{
        hoge <- t.test(x.class
        return(c(hoge$statisti
    }
```

切り取り(T)
コピー(C)
貼り付け
すべて選択(A)
印刷(I)...
印刷プレビュー(N)...

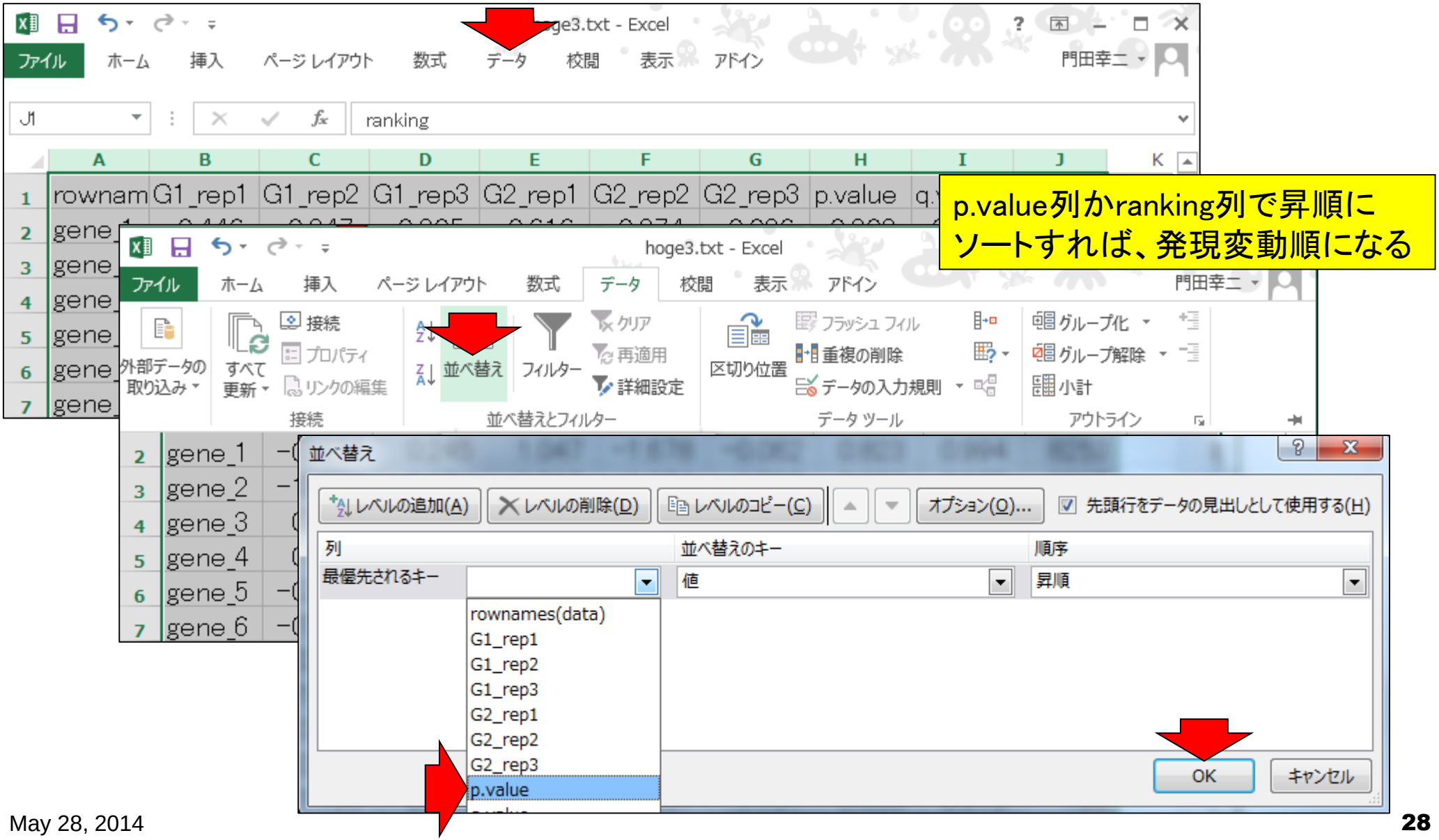
p.value列かranking列で昇順にソートすれば、発現変動順になる

hoge3.txt

rownam	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking
gene_1	−0.446	0.947	0.305	−0.616	−0.274	0.096	0.303	0.975	3074
gene_2	−1.206	−0.374	0.245	1.047	−1.678	−0.062	0.823	0.994	8253
gene_3	0.041	−0.230	−0.674	1.916	−1.274	1.897	0.353	0.977	3615
gene_4	0.639	−0.008	0.752	−0.379	1.788	−1.259	0.683	0.994	6828
gene_5	−0.787	1.154	−0.082	0.582	−0.595	2.591	0.522	0.994	5209
gene_6	−0.385	−2.474	1.137	−1.616	0.460	−0.618	0.989	0.998	9914
ene_7	−0.476	−1.854	0.065	1.782	−0.034	1.348	0.087	0.975	852
	0.720	0.806	−2.314	−0.255	−0.682	−0.655	0.809	0.994	8112
	019	−1.598	−2.060	1.155	0.639	2.048	0.028	0.975	277
	73	0.059	0.284	−0.011	0.223	0.215	0.407	0.989	4111
	82	−0.500	−1.397	−1.321	0.332	−2.195	0.903	0.994	9075
	554	−1.795	−0.657	0.391	−0.387	1.137	0.080	0.975	796

コード内のコピーは
CTRL + ALT + 左クリック

2群間比較：Student's t-test



2群間比較：Student's t-test

rownames(c	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking
gene_2313	0.172	0.422	0.234	−1.211	−1.399	−1.564	0.0002	0.9025	1
gene_7754	1.514	1.426	1.814	−0.324	−0.615	−0.275	0.0002	0.9025	2
gene_1175	−1.437	−1.029	−1.326	1.238	1.336	0.780	0.0003	0.9025	
gene_766	−1.162	−0.936	−1.521	0.538	0.775	0.899	0.0006	0.9025	
gene_9001	0.648	0.386	0.489	−0.737	−0.610	−0.428	0.0007	0.9025	
gene_5866	−1.956	−2.084	−1.932	−0.132	−0.001	0.611	0.0008	0.9025	6
gene_5818	−0.999	−1.526	−1.013	0.860	0.705	1.237	0.0008	0.9025	7
gene_4882	1.911	1.145	0.782	−0.007	−0.983	0.408	0.0498	0.9751	490
gene_8919	0.508	1.239	0.396	−0.149	−0.415	0.135	0.0498	0.9751	491
gene_2545	−1.248	−2.039	−0.106	0.366	0.674	0.332	0.0499	0.9751	492
gene_8229	0.262	0.250	0.715	−0.427	−0.675	0.081	0.0500	0.9751	493
gene_9729	−1.113	−1.714	−0.291	−0.042	0.654	0.123	0.0500	0.9751	494
gene_2484	0.296	0.249	2.137	−0.718	−0.787	−1.059	0.0501	0.9751	495
gene_7406	−0.700	0.393	−1.023	0.136	0.709	1.119	0.0997	0.9751	957
gene_924	0.473	1.117	0.998	0.647	−0.102	−0.597	0.0998	0.9751	958
gene_872	0.236	−0.057	1.111	−1.812	0.031	−1.014	0.0999	0.9751	959
gene_7666	0.112	1.531	1.352	0.491	−0.575	−1.307	0.1003	0.9751	960
gene_4154	1.346	1.255	−0.047	−0.114	0.102	−0.876	0.1003	0.9751	961
gene_8055	−0.590	−0.454	0.163	0.145	0.138	0.543	0.1004	0.9751	962
gene_724	−0.442	1.969	1.071	0.767	1.595	0.236	0.9997	0.9999	9998
gene_287	1.677	0.395	−1.040	−0.120	0.370	0.783	0.9998	0.9999	9999
gene_9776	0.942	−0.389	−0.424	−0.843	−0.763	1.735	1.0000	1.0000	10000

③

DEGの存在しないnon-DEGのみからなるデータなので妥当

①

$p < 0.05$を満たす遺伝子数は492個

②

$p < 0.10$を満たす遺伝子数は959個

ランダムデータの場合

- 有意水準αでN回の検定（多重比較）を行うと、（$N \times \alpha$）個のFalse Positiveが得られる。
- 10000個の遺伝子（N=10000）に対して$p < 0.05$を満たすものを調べる（有意水準αを0.05に設定することと同義）と（$N \times \alpha$）個程度が**本当は発現変動遺伝子 (Differentially Expressed Genes; DEGs)でないにもかかわらず**発現変動遺伝子と判断されてしまう。

Type-I error（false positive）

*p*値だけである程度判断できる…が

- うれしくない結果:「実際に得られた発現変動遺伝子数 ≒（解析遺伝子数*N*× 設定した有意水準 α）個」
  - **このデータ中には「発現変動遺伝子 (DEG)はない」**と判断する。
- うれしい結果:「実際に得られた発現変動遺伝子数 >>（解析遺伝子数*N*× 設定した有意水準 α）個」
  - **このデータ中には「真の発現変動遺伝子が存在する」**ことが期待される。

実際に利用されているRパッケージの多くは、(多重比較を考慮した**補正後の*p*-value**に相当する)***q*-value**の値を出力する
→（*p*値利用時の有意水準 α に相当する) False Discovery Rate (FDR)の閾値を満たす遺伝子数を頼りに発現変動遺伝子の有無を判断する

多重比較問題：FDRって何?

- ***p***–value (false positive rate; FPR)
  - 本当はDEGではないにもかかわらずDEGと判定してしまう確率
  - 全遺伝子に占めるnon-DEGの割合（分母は**遺伝子総数**）
  - 例：10,000個のnon-DEGからなる遺伝子をp–value < 0.05で検定すると、10,000 × 0.05 = 500個程度のnon-DEGを間違ってDEGと判定することに相当
    - 実際のDEG検出結果が900個だった場合：500個は偽物で400個は本物と判断
    - 実際のDEG検出結果が510個だった場合：500個は偽物で10個は本物と判断
    - 実際のDEG検出結果が500個以下の場合：全て偽物と判断
- ***q***–value (false discovery rate: FDR)
  - DEGと判定した中に含まれるnon-DEGの割合
  - DEG中に占めるnon-DEGの割合（分母は**DEGと判定された数**）
  - non-DEGの期待値を計算できれば、p値でも上位x個でもDEGと判定する手段はなんでもよい。以下は10,000遺伝子の検定結果でのFDR計算例
    - p < 0.001を満たすDEG数が100個の場合：FDR = 10,000 × 0.001/100 = 0.1
    - p < 0.01を満たすDEG数が400個の場合：FDR = 10,000 × 0.01/400 = 0.25
    - p < 0.05を満たすDEG数が926個の場合：FDR = 10,000 × 0.05/926 = 0.54

多重比較問題：FDRって何?

- DEGかnon-DEGかを判定する閾値を決める問題
  - 有意水準5%というのがp-value < 0.05に相当
  - False discovery rate (FDR) 5%というのがq-value < 0.05に相当
- 発現変動**ランキング結果は不変**なので上位x個という決め打ちの場合にはこの問題とは無関係

rownames(c	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking
gene_2313	0.172	0.422	0.234	-1.211	-1.399	-1.564	0.0002	0.9025	1
gene_7754	1.514	1.426	1.814	-0.324	-0.615	-0.275	0.0002	0.9025	2
gene_1175	-1.437	-1.029	-1.326	1.238	1.336	0.780	0.0003	0.9025	3
gene_766	-1.162	-0.936	-1.521	0.538	0.775	0.899	0.0006	0.9025	4
gene_9001	0.648	0.386	0.489	-0.737	-0.610	-0.428	0.0007	0.9025	5
gene_5866	-1.956	-2.084	-1.932	-0.132	-0.001	0.611	0.0008	0.9025	6
gene_5818	-0.999	-1.526	-1.013	0.860	0.705	1.237	0.0008	0.9025	7

DEG数に関するよりよい結果を得たい場合には、
p-valueではなくq-value閾値を利用しましょう

2群間比較：Student's t-test

rownames(d	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking
gene_2313	0.172	0.422	0.234	−1.211	−1.399	−1.564	0.0002	0.9025	
gene_7754	1.514	1.426	1.814	−0.324	−0.615	−0.275	0.0002	0.9025	
gene_1175	−1.437	−1.029	−1.326	1.238	1.336	0.780	0.0003	0.9025	
gene_766	−1.162	−0.936	−1.521	0.538	0.775	0.899	0.0006	0.9025	
gene_9001	0.648	0.386	0.489	−0.737	−0.610	−0.428	0.0007	0.9025	
gene_5866	−1.956	−2.084	−1.932	−0.132	−0.001	0.611	0.0008	0.9025	6
gene_5818	−0.999	−1.526	−1.013	0.860	0.705	1.237	0.0008	0.9025	7
gene_4882	1.911	1.145	0.782	−0.007	−0.983	0.408	0.0498	0.9751	490
gene_8919	0.508	1.239	0.396	−0.149	−0.415	0.135	0.0498	0.9751	491
gene_2545	−1.248	−2.039	−0.106	0.366	0.674	0.332	0.0499	0.9751	492
gene_8229	0.262	0.250	0.715	−0.427	−0.675	0.081	0.0500	0.9751	493
gene_9729	−1.113	−1.714	−0.291	−0.042	0.654	0.123	0.0500	0.9751	494
gene_2484	0.296	0.249	2.137	−0.718	−0.787	−1.059	0.0501	0.9751	495
gene_7406	−0.700	0.393	−1.023	0.136	0.709	1.119	0.0997	0.9751	957
gene_924	0.473	1.117	0.998	0.647	−0.102	−0.597	0.0998	0.9751	958
gene_872	0.236	−0.057	1.111	−1.812	0.031	−1.014	0.0999	0.9751	959
gene_7666	0.112	1.531	1.352	0.491	−0.575	−1.307	0.1003	0.9751	960
gene_4154	1.346	1.255	−0.047	−0.114	0.102	−0.876	0.1003	0.9751	961
gene_8055	−0.590	−0.454	0.163	0.145	0.138	0.543	0.1004	0.9751	962
gene_724	−0.442	1.969	1.071	0.767	1.595	0.236	0.9997	0.9999	9998
gene_287	1.677	0.395	−1.040	−0.120	0.370	0.783	0.9998	0.9999	9999
gene_9776	0.942	−0.389	−0.424	−0.843	−0.763	1.735	1.0000	1.0000	10000

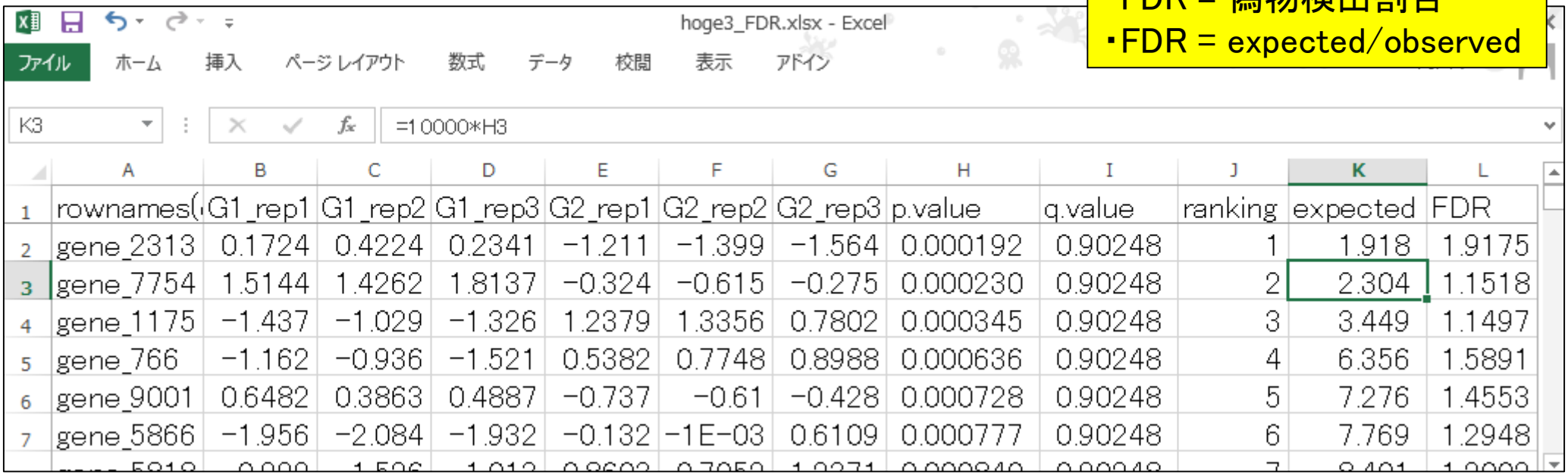
$q < 0.05$を満たす遺伝子数は0個。DEGの存在しないnon-DEGのみからなるデータなので妥当

$p < 0.05$を満たす遺伝子数の実測値は492個。期待値は500個。

$p < 0.10$を満たす遺伝子数の実測値は959個。期待値は1,000個。

2群間比較：Student's t-test

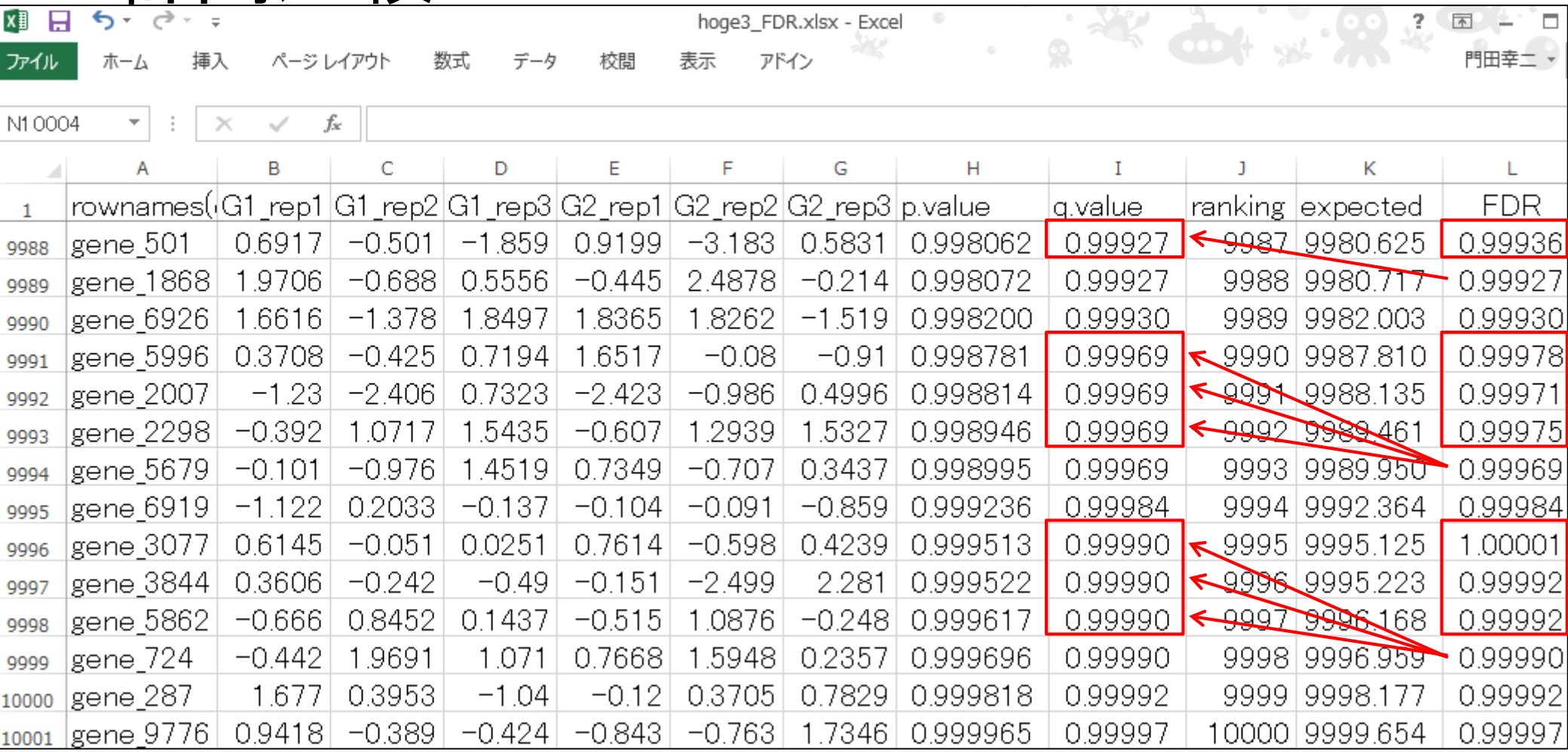
・FDR = 偽物検出割合
・FDR = expected/observed



	A	B	C	D	E	F	G	H	I	J	K	L
1	rownames(	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking	expected	FDR
2	gene_2313	0.1724	0.4224	0.2341	−1.211	−1.399	−1.564	0.000192	0.90248	1	1.918	1.9175
3	gene_7754	1.5144	1.4262	1.8137	−0.324	−0.615	−0.275	0.000230	0.90248	2	2.304	1.1518
4	gene_1175	−1.437	−1.029	−1.326	1.2379	1.3356	0.7802	0.000345	0.90248	3	3.449	1.1497
5	gene_766	−1.162	−0.936	−1.521	0.5382	0.7748	0.8988	0.000636	0.90248	4	6.356	1.5891
6	gene_9001	0.6482	0.3863	0.4887	−0.737	−0.61	−0.428	0.000728	0.90248	5	7.276	1.4553
7	gene_5866	−1.956	−2.084	−1.932	−0.132	−1E−03	0.6109	0.000777	0.90248	6	7.769	1.2948

基本的にこの2つは同じものという理解でよい。より正確には、FDR列の情報をもとに値の分布が滑らかになるように細工しているのがq.value列の数値

2群間比較：Student's t-test



hoge3_FDR.xlsx - Excel

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

N1 0004

	A	B	C	D	E	F	G	H	I	J	K	L
1	rownames(	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking	expected	FDR
9988	gene_501	0.6917	−0.501	−1.859	0.9199	−3.183	0.5831	0.998062	0.99927	9987	9980.625	0.99936
9989	gene_1868	1.9706	−0.688	0.5556	−0.445	2.4878	−0.214	0.998072	0.99927	9988	9980.717	0.99927
9990	gene_6926	1.6616	−1.378	1.8497	1.8365	1.8262	−1.519	0.998200	0.99930	9989	9982.003	0.99930
9991	gene_5996	0.3708	−0.425	0.7194	1.6517	−0.08	−0.91	0.998781	0.99969	9990	9987.810	0.99978
9992	gene_2007	−1.23	−2.406	0.7323	−2.423	−0.986	0.4996	0.998814	0.99969	9991	9988.135	0.99971
9993	gene_2298	−0.392	1.0717	1.5435	−0.607	1.2939	1.5327	0.998946	0.99969	9992	9989.461	0.99975
9994	gene_5679	−0.101	−0.976	1.4519	0.7349	−0.707	0.3437	0.998995	0.99969	9993	9989.950	0.99969
9995	gene_6919	−1.122	0.2033	−0.137	−0.104	−0.091	−0.859	0.999236	0.99984	9994	9992.364	0.99984
9996	gene_3077	0.6145	−0.051	0.0251	0.7614	−0.598	0.4239	0.999513	0.99990	9995	9995.125	1.00001
9997	gene_3844	0.3606	−0.242	−0.49	−0.151	−2.499	2.281	0.999522	0.99990	9996	9995.223	0.99992
9998	gene_5862	−0.666	0.8452	0.1437	−0.515	1.0876	−0.248	0.999617	0.99990	9997	9996.168	0.99992
9999	gene_724	−0.442	1.9691	1.071	0.7668	1.5948	0.2357	0.999696	0.99990	9998	9996.959	0.99990
10000	gene_287	1.677	0.3953	−1.04	−0.12	0.3705	0.7829	0.999818	0.99992	9999	9998.177	0.99992
10001	gene_9776	0.9418	−0.389	−0.424	−0.843	−0.763	1.7346	0.999965	0.99997	10000	9999.654	0.99997

p.valueが高いもののFDR値から順に見ていって、FDR値が低くなるように置換していったものがq.value列の値。

自力でq-value (FDR)計算

1	rownames(	G1_rep1	G1_rep2	G1_rep3	G2_rep1	G2_rep2	G2_rep3	p.value	q.value	ranking	expected	FDR
491	gene_4882	1.911	1.1447	0.7818	−0.007	−0.983	0.4081	0.049820	0.97509	490	498.198	1.01673
492	gene_8919	0.5084	1.2393	0.3958	−0.149	−0.415	0.1354	0.049845	0.97509	491	498.453	1.01518
493	gene_2545	−1.248	−2.039	−0.106	0.3658	0.6739	0.3315	0.049911	0.97509	492	499.114	1.01446
494	gene_8229	0.262	0.2501	0.715	−0.427	−0.675	0.0809	0.050047	0.97509	493	500.468	1.01515
495	gene_9729	−1.113	−1.714	−0.291	−0.042	0.6544	0.1234	0.050049	0.97509	494	500.489	1.01314
496	gene_2484	0.2956	0.2492	2.1367	−0.718	−0.787	−1.059	0.050095	0.97509	495	500.949	1.01202

R Console

```
> head(p.value)
   gene_1    gene_2    gene_3    gene_4    gene_5    gene_6
0.3030816 0.8226226 0.3532968 0.6832627 0.5216252 0.9894427
> observed <- sum(p.value < 0.05)
> observed
[1] 492
> length(p.value)
[1] 10000
> expected <- length(p.value)*0.05
> expected
[1] 500
> FDR <- expected/observed
> FDR
[1] 1.01626
>
```

・FDR = 偽物検出割合
・FDR = expected/observed

自力でq-value (FDR)計算

*p*値計算結果が手元にあれば(つまりp.valueオブジェクトがあれば)このコードを実行することによってFDRの概要がわかります

(Rで)マイクロアレイデータ解析

- 書籍 | トランスクリプトーム解析 | 3.2.1 クラスタリング(データ変換や距離の定義など) (last modified
- 書籍 | トランスクリプトーム解析 | 3.2.2 実験デザイン、データ分布、統計解析との関係 (last modified
- 書籍 | トランスクリプトーム解析 | 3.2.3 多重比較問題 (last modified 2014/04/19)
- 書籍 | トランスクリプトーム解析 | 3.2.4 各種プロット(M-A plotや平均-分散プロットなど) (last modifie
- 書籍 | トランスクリプトーム解析 | 4.2.1 2群間比較 (last modified 2014/04/19)

書籍 | トランスクリプトーム解析 | 3.2.3 多重比較問題

シリーズ Useful R 第7巻トランスクリプトーム解析のp111-121のRコードです。Windowsの場合、コピペは「CTRLキーとALTキーを押しながら枠内で左クリック」でコード内を全選択できます。

p115-116の網掛け部分:

```
threshold <- c(0.001, 0.01, 0.03, 0.05, 0.10)
res <- NULL
for(i in 1:length(threshold)){
  observed <- sum(p.value < threshold[i])
  expected <- nrow(data)*threshold[i]
  FDR <- expected/observed
  res <- rbind(res, c(threshold[i], observed, expected, FDR))
}
colnames(res) <- c("threshold", "observed", "expected", "FDR")
```

自力でq-value (FDR)計算

p115-116の網掛け部分:

```
threshold <- c(0.001, 0.01, 0.03, 0.05, 0.10)
res <- NULL
for(i in 1:length(threshold)){
  obs
  exp
  FDR
  res
}
colna
```

R Console

```
> threshold <- c(0.001, 0.01, 0.03, 0.05, 0.10)
> res <- NULL
> for(i in 1:length(threshold)){
+   observed <- sum(p.value < threshold[i])
+   expected <- nrow(data)*threshold[i]
+   FDR <- expected/observed
+   res <- rbind(res, c(threshold[i], observed, expected, FD$
+ }
> colnames(res) <- c("threshold", "observed", "expected", "F$
> res
     threshold observed expected       FDR
[1,]     0.001        8       10 1.2500000
[2,]     0.010      104      100 0.9615385
[3,]     0.030      295      300 1.0169492
[4,]     0.050      492      500 1.0162602
[5,]     0.100      959     1000 1.0427529
> 
```

ここで指定しているのはp-valueの閾値（つまり有意水準）です

Contents（第3回）

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - 関数の利用法
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - 描画（M-A plot）
    - 作成法
    - 同一群内のばらつき（前処理法間の違い）

2群間比較：Student's t-test

解析 | 発現変動 | 2群間 | 対応なし | Student's t-test NEW

等分散性を仮定したt検定を用いて、2群間での発現変動遺伝子の同定を行うやり方を示します。

4. サンプルデータ23のsample23.txtの場合:

最初の3サンプルがG1群、残りの3サンプルがG2群の標準正規分布に従う乱数からなるシミュレーションデータです。乱数発生後に、さらに最初の10%分についてG1群に相当するところのみ数値を+3している（つまり10%がG1群で高発現というシミュレーションデータを作成している）

```
in_f <- "sample23.txt"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge4.txt"                      #出力ファイル名を指定してout_fに格納
param_G1 <- 3
param_G2 <- 3

#入力ファイルの
data <- read.
data.cl <- c(

#等分散性を仮定
Students_ttes
    x.class1
    x.class2
    if((sd(x.
```

確かにG1群で高発現になっていることがわかります

R Console

```
> in_f <- "sample23.txt"                    #入力ファイル名を指定してin$
> out_f <- "hoge4.txt"                      #出力ファイル名を指定してou$
> 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
>
> #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$
> data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2$
> head(data)
         G1_rep1   G1_rep2  G1_rep3    G2_rep1    G2_rep2     G2_rep3
gene_1 2.554222 3.9470852 3.304717 -0.6162540 -0.2744629  0.09569090
gene_2 1.794143 2.6255011 3.244921  1.0468297 -1.6776099 -0.06155844
gene_3 3.041126 2.7702859 2.325950  1.9160200 -1.2743934  1.89711799
gene_4 3.639388 2.9922446 3.751594 -0.3791546  1.7883961 -1.25888353
gene_5 2.213446 4.1538056 2.918114  0.5819835 -0.5949475  2.59110024
gene_6 2.614511 0.5255313 4.137273 -1.6164433  0.4604567 -0.61764893
>
```

2群間比較：Student's t-test

4. サンプルデータ23のsample23.txtの場合:

最初の3サンプルがG1群、残りの3サンプルがG2群の標準正規分布に従う乱数からなるシミュレーションデータです。乱数発生後に、さらに最初の10%分についてG1群に相当するところのみ数値を+3している(つまり10%がG1群で高発現というシミュレーションデータを作成している)

```
in_f <- "sample23.txt"                    #入力ファイル名を指定してin_fに格納
out_f <- "hoge4                           イル名を指定してout_fに格納
param_G1 <- 3                             ンプル数を指定
param_G2 <- 3                             ンプル数を指定
```

切り取り(T)
コピー(C)

R Console

```
> #以下は(こんなこともできますという)おまけ
> #(G1群 vs. G2群) t-testでp-value < 0.05を満たす遺伝子数を表示さ$
> param4 <- 0.05                                  #閾値を指定
> sum(p.value < param4)                           #p.valueが(param4)未満と$
[1] 1226
> sum(q.value < 0.05)                             #FDR < 0.05を満たす要素数$
[1] 0
> sum(q.value < 0.10)                             #FDR < 0.10を満たす要素数$
[1] 0
> sum(q.value < 0.15)                             #FDR < 0.15を満たす要素数$
[1] 110
> sum(q.value < 0.20)                             #FDR <
[1] 388
> sum(q.value < 0.25)                             #FDR <
[1] 627
>
```

$p < 0.05$を満たす遺伝子数は1,226個。期待値は500個なので、(1,226 − 500)個程度が本物だと判断する

$q < 0.20$を満たす遺伝子数は388個。FDR = 0.20なので、388*0.2 = 77.6個は偽物で残りの80%は本物だと判断する

Contents（第3回）

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - 関数の利用法
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - 描画（M-A plot）
    - 作成法
    - 同一群内のばらつき（前処理法間の違い）

発現変動解析用Rパッケージの利用

- 解析 | 発現変動 | 2群間 | 発現変動遺伝子の割合を調べる (Ploner_2006) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | について (last modified 2011/08/02)
- 解析 | 発現変動 | 2群間 | 対応なし | WAD (Kadota_2008) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | Random forest (Diaz-Uriarte_2007) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | shrinkage t (Opgen-Rhein_2007) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | layer ranking algorithm (Chen_2007) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | fdr2d (Ploner_2006) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Rank products (Breitling_2004) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | samroc (Broberg_2003) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | SAM (Tusher_2001) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Studen
- 解析 | 発現変動 | 2群間 | 対応なし | Welch
- 解析 | 発現変動 | 2群間 | 対応なし | Mann-
- 解析 | 発現変動 | 2群間 | 対応なし | パター
- 解析 | 発現変動 | 2群間 | 対応あり | につい
- 解析 | 発現変動 | 2群間 | 対応あり | SAM (
- 解析 | 発現変動 | 2群間 | 対応あり | SAM (
- 解析 | 発現変動 | 2群間 | 対応あり | 時系列

limmaというパッケージを用いてDEG検出を行います

解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004)

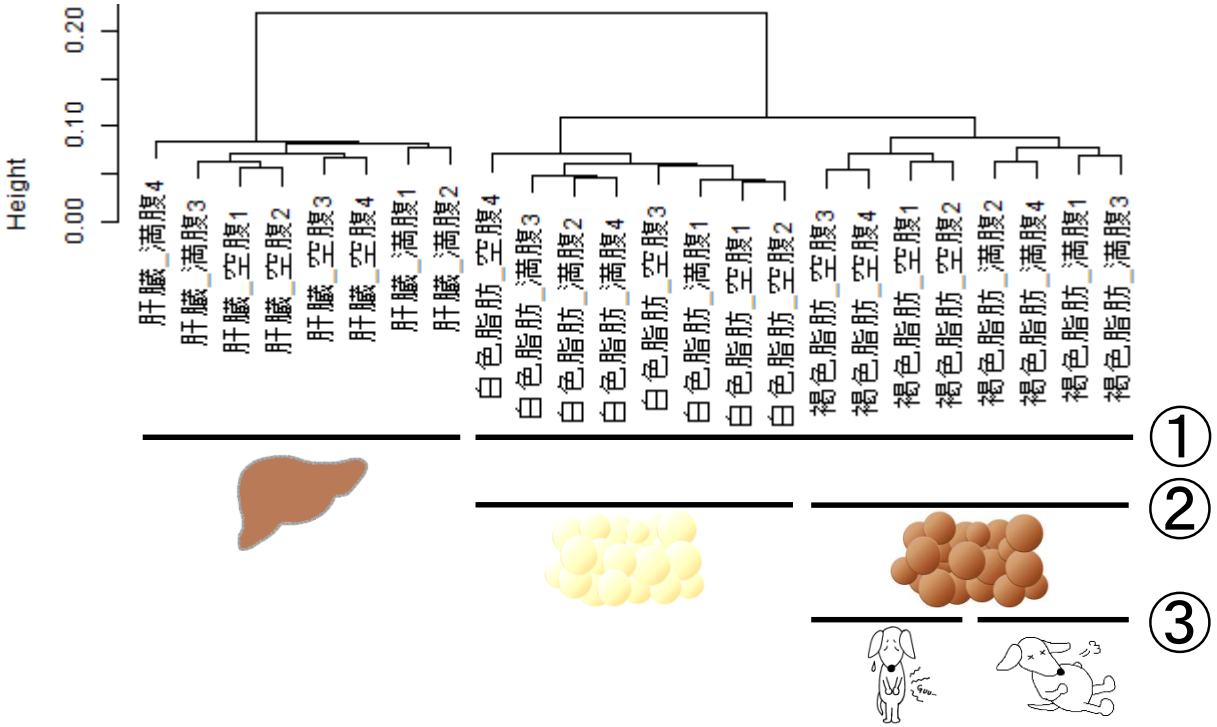
limmaパッケージを用いて2群間比較を行うやり方を示します。
この方法は経験ベイズと表現されたり、moderated t statisticと表現されたりしているようです。
「ファイル」−「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し、以下をコピペ

1. サンプルデータ20の31,099 probesets×8 samplesのdata_rma_2_LIV.txt(G1群4サンプル vs. G2群4サンプル)の場合:

```
in_f <- "data_rma_2_LIV.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
param_G1 <- 4                           #G1群のサンプル数を指定
param_G2 <- 4                           #G2群のサンプル数を指定

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

#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
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

#本番
#design <- model.matrix(~ as.factor(data.cl))#デザイン行列を作成した結果をdesignに格納
design <- model.matrix(~data.cl)        #デザイン行列を作成した結果をdesignに格納
```

発現変動解析用Rパッケージの利用

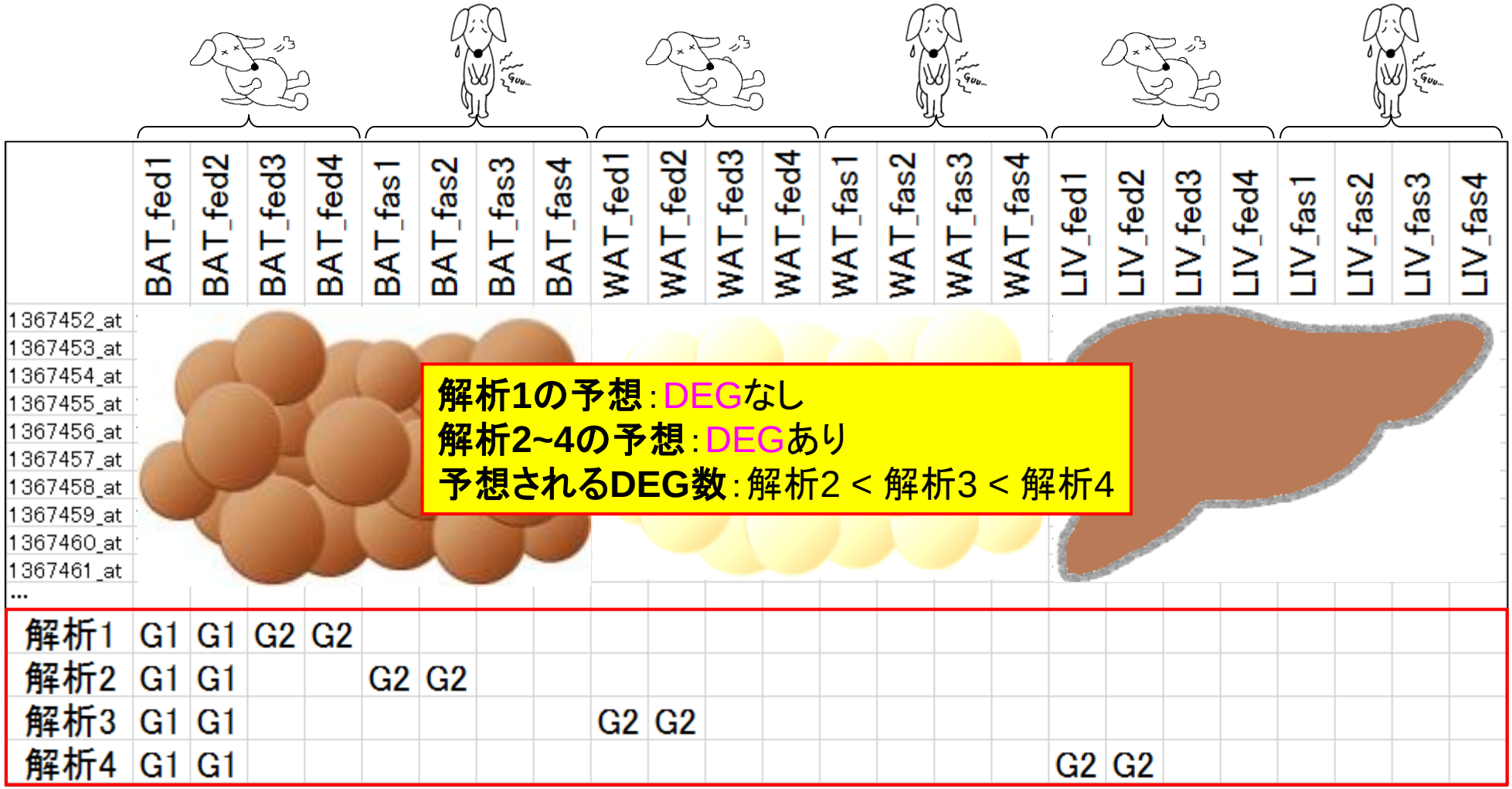
- Nakai et al., *Biosci Biotechnol Biochem.*, **72**: 139-148, 2008
  - **GSE7623**、GPL1355 (Affymetrix Rat Genome 230 2.0 Array)、31,099 probesets
  - **ラット24サンプル**：Brown adipose tissue (褐色脂肪組織; BAT)8サンプル、White adipose tissue (白色脂肪組織; WAT)8サンプル、Liver (肝臓; LIV)8サンプル
    - BAT 8サンプル：通常(BAT_fed) 4サンプル　対　24時間絶食(BAT_fas) 4サンプル
    - WAT 8サンプル：通常(WAT_fed) 4サンプル　対　24時間絶食(WAT_fas) 4サンプル
    - LIV 8サンプル：通常(LIV_fed) 4サンプル　対　24時間絶食(LIV_fas) 4サンプル



GSE7623データを用い、様々な2群間比較を行い、クラスタリング結果とDEG検出結果の関連をみてみよう

rcode_clustering_png.txtの実行結果。
①肝臓と脂肪間で大きく二つのクラスターに分かれている。
②脂肪の中でも白色脂肪と褐色脂肪に分かれている。
③褐色脂肪は空腹(24時間絶食)と満腹(通常)できれいに分かれている。

RパッケージlimmaでDEG検出



解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004)

limmaパッケージを用いて2群間比較を行うや
この方法は経験ベイズと表現されたり、moder
「ファイル」-「ディレクトリの変更」で解析したい

1. サンプルデータ20の31,099 probesets×8 sa
場合:

```
in_f <- "data_rma_2_LIV.txt"
out_f <- "hoge1.txt"
param_G1 <- 4
param_G2 <- 4

#必要なパッケージをロード
library(limma)

#入力ファイルの読み込みとラベル情報(
data <- read.table(in_f, header=T
data.cl <- c(rep(1, param_G1), re

#本番
#design <- model.matrix(~ as.fact
design <- model.matrix(~data.cl)
fit <- lmFit(data, design)
out <- eBayes(fit)
p.value <- out$p.value[,ncol(desi
q.value <- p.adjust(p.value, meth
ranking <- rank(p.value)
sum(q.value < 0.05)
```

rcode_limma_basic.txt（変更点および追加点）

```
############################↓
###  解析1 (Analysis1)  ###↓
############################↓
in_f <- "data_mas_EN.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↓
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↓
↓
#必要なパッケージをロード↓
library(limma)                           #パッケージの読み込み↓
↓
#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
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
↓
#サブセットの作成（解析したいデータのみにする）↓
posi <- c(1,2,3,4)                       #元の発現行列上での列番号を指定↓
data <- data[,posi]                      #サブセットを抽出↓
↓
#本番↓
#design <- model.matrix(~ as.factor(data.cl))#デザイン行列を作成した結果をdesignに格納
design <- model.matrix(~data.cl)         #デザイン行列を作成した結果をdesignに格納↓
fit <- lmFit(data, design)               #モデル構築(ばらつきの程度を見積もっている)↓
out <- eBayes(fit)                       #検定(経験ベイズ)↓
p.value <- out$p.value[,ncol(design)]    #p値をp.valueに格納↓
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納↓
ranking <- rank(p.value)                 #p.valueでランキングした結果をrankingに格納↓
sum(q.value < 0.05)                      #FDR < 0.05を満たす遺伝子数を表示↓
↓
#ファイルに保存↓
tmp <- cbind(rownames(data), data, p.value, q.value, ranking)#入力データの右側にp.valu
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定した
```

rcode_limma_basic.txt

- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth 2004)

入力ファイル（data_mas_EN.txt）読み込み後のdataオブジェクトは24サンプルからなる

```
############################
###  解析1 (Analysis1)  ###
############################
in_f <- "data_mas_EN.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
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

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

#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="
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を

#サブセットの作成（解析したいデータのみにする）
posi <- c(1,2,3,4)                           #元の発現行列上で
data <- data[,posi]                          #サブセットを抽出

#本番
#design <- model.matrix(~ as.factor(data.cl))#デザイン行
design <- model.matrix(~data.cl)             #デザイン行列を作
fit <- lmFit(data, design)                   #モデル構築(ばら
out <- eBayes(fit)                           #検定(経験ベイズ)
p.value <- out$p.value[,ncol(design)]  #p値をp.valueに格
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに
ranking <- rank(p.value)                     #p.valueでランキ
sum(q.value < 0.05)                          #FDR < 0.05を満た

#ファイルに保存
tmp <- cbind(rownames(data), data, p.value, q.value, ran
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row
```

R Console

```
> ############################
> ###  解析1 (Analysis1)  ###
> ############################
> in_f <- "data_mas_EN.txt"                    #入力ファイ$
> out_f <- "hoge1.txt"                         #出力ファイ$
> param_G1 <- 2                                #G1群のサン$
> param_G2 <- 2                                #G2群のサン$
>
> #必要なパッケージをロード
> library(limma)                               #パッケージ$
>
> #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,$
> data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G$
> dim(data)
[1] 31099    24
> colnames(data)
 [1] "BAT_fed1" "BAT_fed2" "BAT_fed3" "BAT_fed4"
 [5] "BAT_fas1" "BAT_fas2" "BAT_fas3" "BAT_fas4"
 [9] "WAT_fed1" "WAT_fed2" "WAT_fed3" "WAT_fed4"
[13] "WAT_fas1" "WAT_fas2" "WAT_fas3" "WAT_fas4"
[17] "LIV_fed1" "LIV_fed2" "LIV_fed3" "LIV_fed4"
[21] "LIV_fas1" "LIV_fas2" "LIV_fas3" "LIV_fas4"
>
```

rcode_limma_basic.txt

- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth 2004)

```
############################
###  解析1 (Analysis1)  ###
############################
in_f <- "data_mas_EN.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
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

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

#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,
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G

#サブセットの作成（解析したいデータのみにする）
posi <- c(1,2,3,4)                     #元の発現行
data <- data[,posi]                    #サブセット

#本番
#design <- model.matrix(~ as.factor(data.cl))#デザ
design <- model.matrix(~data.cl)       #デザイン行
fit <- lmFit(data, design)             #モデル構築
out <- eBayes(fit)                     #検定(経験
p.value <- out$p.value[,ncol(design)]  #p値をp.val
q.value <- p.adjust(p.value, method="BH")#q値をq.v
ranking <- rank(p.value)               #p.valueで
sum(q.value < 0.05)                    #FDR < 0.05

#ファイルに保存
tmp <- cbind(rownames(data), data, p.value, q.valu
write.table(tmp, out_f, sep="¥t", append=F, quote=
```

posiで指定した列番号のみからなるサブセットを抽出できていることがわかる

R Console

```
> #サブセットの作成(解析したいデータのみにする)
> posi <- c(1,2,3,4)                   #元の発現行$
> data <- data[,posi]                  #サブセット$
> dim(data)
[1] 31099     4
> colnames(data)
[1] "BAT_fed1" "BAT_fed2" "BAT_fed3" "BAT_fed4"
> head(data)
           BAT_fed1 BAT_fed2 BAT_fed3 BAT_fed4
1367452_at 12.78446 12.44708 12.80591 12.30472
1367453_at 11.80125 12.15293 11.94223 11.96848
1367454_at 11.38990 11.16076 11.14599 11.21209
1367455_at 12.36435 12.52974 12.43257 12.60401
1367456_at 13.44849 13.54305 13.55279 13.62980
1367457_at 10.40403 10.69632 10.47508 10.45579
> posi
[1] 1 2 3 4
>
```

RパッケージlimmaでDEG検出

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fed1
1367452_at									
1367453_at									
1367454_at									
1367455_at									
1367456_at									
1367457_at									
1367458_at									
1367459_at									
1367460_at									
1367461_at									
…									
解析1	G1	G1	G2	G2					
解析2	G1	G1			G2	G2			
解析3	G1	G1							G2
解析4	G1	G1							

解析1の予想：DEGなし

R Console

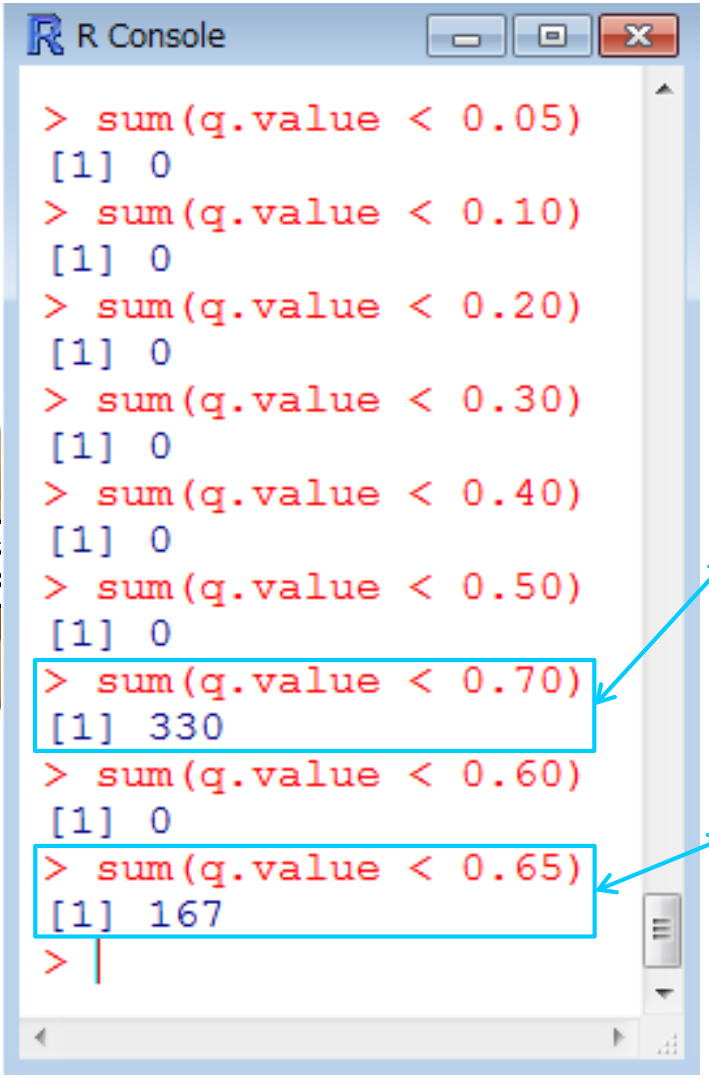
```
> #サブセットの作成(解析したいデータのみにする)
> posi <- c(1,2,3,4)                    #元の発$
> data <- data[,posi]                   #サブセ$
> dim(data)
[1] 31099     4
> colnames(data)
[1] "BAT_fed1" "BAT_fed2" "BAT_fed3" "BAT_fed4"
> head(data)
           BAT_fed1 BAT_fed2 BAT_fed3 BAT_fed4
1367452_at 12.78446 12.44708 12.80591 12.30472
1367453_at 11.80125 12.15293 11.94223 11.96848
1367454_at 11.38990 11.16076 11.14599 11.21209
1367455_at 12.36435 12.52974 12.43257 12.60401
1367456_at 13.44849 13.54305 13.55279 13.62980
1367457_at 10.40403 10.69632 10.47508 10.45579
> posi
[1] 1 2 3 4
> data.cl
[1] 1 1 2 2
>
```

data.clで指定している情報は、グループラベル情報(どの列がどの群由来かということ)

RパッケージlimmaでDEG検出

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4
1367452_at				
1367453_at				
1367454_at				
1367455_at				
1367458_at				
1367459_at				
1367460_at				
1367461_at				
…				
解析1	G1	G1	G2	G2
解析2	G1	G1		
解析3	G1	G1		
解析4	G1	G1		

解析1の予想：DEGな



通常（私）は、0.05～0.30あたりのFDR閾値を調査→DEGがないと判断

$q < 0.70$を満たす遺伝子数は330個。FDR = 0.70なので、330*0.7 = 231個は偽物で残りの30%（つまり330*0.3 = 99個）は本物だと判断することになる…

$q < 0.65$を満たす遺伝子数は167個。FDR = 0.65なので、167*0.65 = 108.55個は偽物で残りの35%（つまり167*0.35 = 58.45個）は本物だと判断する…

```
############################
###  解析1 (Analysis1)  ###
############################
in_f <- "data_mas_EN.txt"
out_f <- "hoge1.txt"
param_G1 <- 2
param_G2 <- 2

#必要なパッケージをロード
library(limma)

#入力ファイルの読み込みとラベル情報
data <- read.table(in_f, header=TRU
data.cl <- c(rep(1, param_G1), rep(

#サブセットの作成 (解析したいデータ
posi <- c(1,2,3,4)
data <- data[,posi]

#本番
#design <- model.matrix(~ as.facto
design <- model.matrix(~data.cl)
fit <- lmFit(data, design)
out <- eBayes(fit)
p.value <- out$p.value[,ncol(desig
q.value <- p.adjust(p.value, metho
ranking <- rank(p.value)
sum(q.value < 0.05)

#ファイルに保存
tmp <- cbind(rownames(data), data,
write.table(tmp, out_f, sep="¥t",
```

rcode_limma_all.txt(の一部)

```
############################
###  解析1 (Analysis1)  ###
############################
in_f <- "data_mas_EN.txt"
out_f <- "hoge1.txt"
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_posi <- c(1,2,3,4)         #元の発現行列上での列番号を指定

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

#入力ファイルの読み込みとラベル情報の作成、そしてサブセットの作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="¥t", quote="")#in_fで指定した
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.c
data <- data[,param_posi]        #サブセットを抽出
colnames(data)                   #サブセット抽出後のサンプル名を表示

#本番
design <- model.matrix(~data.cl)       #デザイン行列を作成した結果をdesignに格納
fit <- lmFit(data, design)             #モデル構築(ばらつきの程度を見積もっている)
out <- eBayes(fit)                     #検定(経験ベイズ)
p.value <- out$p.value[,ncol(design)]  #p値をp.valueに格納
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納
ranking <- rank(p.value)               #p.valueでランキングした結果をrankingに格納
sum(q.value < 0.05)                    #FDR < 0.05を満たす遺伝子数を表示
sum(q.value < 0.10)                    #FDR < 0.10を満たす遺伝子数を表示
sum(q.value < 0.30)                    #FDR < 0.30を満たす遺伝子数を表示
sum(q.value < 0.50)                    #FDR < 0.50を満たす遺伝子数を表示

#ファイルに保存
tmp <- cbind(rownames(data), data, p.value, q.value, ranking)#入力データの右側にp.val
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定した
```

rcode_limma_basic.txtで動作確認をしてから、param_posiのように変更予定箇所を上のほうに移動して、解析2-4用のコードを作成する(のが門田流)

Rパッケージ

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2
1367452_at						
1367453_at						
1367454_at						
1367455_at						
1367458_at						
1367459_at						
1367460_at						
1367461_at						
…						
解析1	G1	G1	G2	G2		
解析2	G1	G1			G2	G2
解析3	G1	G1				
解析4	G1	G1				

解析2の予想：DEGあり

rcode_limma_all.txt（の一部）

```
#############################↓
###  解析2 (Analysis2)  ###↓
#############################↓
in_f <- "data_mas_EN.txt"              #入力ファイル名を指定してin_fに格納↓
out_f <- "hoge2.txt"                   #出力ファイル名を指定してout_fに格納↓
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_posi <- c(1,2,5,6)               #元の発現行列上での列番号を指定↓
↓
#必要なパッケージをロード↓
library(limma)                         #パッケージの読み込み↓
↓
#入力ファイルの読み込みとラベル情報の作成、そしてサブセットの作成↓
data <- read.table(in_f, header=TRUE, row.names=1, sep="¥t", quote="")#in_fで指定し
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.
data <- data[,param_posi]              #サブセットを抽出↓
colnames(data)                         #サブセット抽出後のサンプル名を表示↓
↓
#本番↓
design <- model.matrix(~data.cl)       #デザイン行列を作成した結果をdesignに格納↓
fit <- lmFit(data, design)             #モデル構築(ばらつきの程度を見積もっている)↓
out <- eBayes(fit)                     #検定(経験ベイズ)↓
p.value <- out$p.value[,ncol(design)]  #p値をp.valueに格納↓
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納↓
ranking <- rank(p.value)               #p.valueでランキングした結果をrankingに格納↓
sum(q.value < 0.05)                    #FDR < 0.05を満たす遺伝子数を表示↓
sum(q.value < 0.10)                    #FDR < 0.10を満たす遺伝子数を表示↓
sum(q.value < 0.30)                    #FDR < 0.30を満たす遺伝子数を表示↓
sum(q.value < 0.50)                    #FDR < 0.50を満たす遺伝子数を表示↓
↓
#ファイルに保存↓
tmp <- cbind(rownames(data), data, p.value, q.value, ranking)#入力データの右側にp.va
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定し
```

RパッケージlimmaでDEG検出

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2
1367452_at						
1367453_at						
1367454_at						
1367455_at						
1367458_at						
1367459_at						
1367460_at						
1367461_at						
…						
解析1	G1	G1	G2	G2		
解析2	G1	G1			G2	G2
解析3	G1	G1				
解析4	G1	G1				

解析2の予想：DEGあり

通常（私）は、0.05～0.30あたりのFDR閾値を調査→DEGがあると判断

R Console

```
> colnames(data)                                    #サブ$
[1] "BAT_fed1" "BAT_fed2" "BAT_fas1" "BAT_fas2"
>
> #本番
> design <- model.matrix(~data.cl)                  #デザ$
> fit <- lmFit(data, design)                        #モデ$
> out <- eBayes(fit)                                #検定$
> p.value <- out$p.value[,2]                        #値$
> q.value <- p.adjust(p.value, method="BH")         #補正$
> ranking <- rank(p.value)                          #順位$
> sum(q.value < 0.05)
[1] 0
> sum(q.value < 0.10)
[1] 38
> sum(q.value < 0.30)                               #FDR $
[1] 2375
> sum(q.value < 0.50)                               #FDR $
[1] 8993
>
> #ファイルに保存
> tmp <- cbind(rownames(data), data, p.value, q.value, ranking)  #$
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=T)  #$
>
```

$q < 0.1$を満たす遺伝子数は38個。FDR = 0.1なので、38*0.1 = 3.8個は偽物で残りの90%（つまり38*0.9 = 34.2個）は本物だと判断することになる…

RパッケージlimmaでDEG検出

通常（私）は、0.05～0.30あたりのFDR閾値を調査→DEGがあると判断

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fed1	WAT_fed2	WAT_fas1	WAT_fas2	WAT_fas3	WAT_fas4
1367452_at														
1367453_at														
1367454_at														
1367455_at														
1367														
1367														
1367458_at														
1367459_at														
1367460_at														
1367461_at														
…														
解析1	G1	G1	G2	G2										
解析2	G1	G1			G2	G2								
解析3	G1	G1							G2	G2				
解析4	G1	G1											G2	G2

解析3の予想：DEGあり

R Console

```
> colnames(data)                                          #グループ
[1] "BAT_fed1" "BAT_fed2" "WAT_fed1" "WAT_fed2"
>
> #本番
> design <- model.matrix(~data.cl)                        #デザイン
> fit <- lmFit(data, design)                              #モデル
> out <- eBayes(fit)                                      #検定
> p.value <- out$p.value[,ncol(design)]                   #p値
> q.value <- p.adjust(p.value, method="BH")#q値
> ranking <- rank(p.value)                                #p.value順
> sum(q.value < 0.05)                                     #FDR 5%
[1] 0
> sum(q.value < 0.10)                                     #FDR 10%
[1] 0
> sum(q.value < 0.30)                                     #FDR 30%
[1] 4786
> sum(q.value < 0.50)                                     #FDR 50%
[1] 12733
```

RパッケージlimmaでDEG検出

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fed1
1367452_at									
1367453_at									
1367454_at									
1367455_at									
1367									
1367									
1367458_at									
1367459_at									
1367460_at									
1367461_at									
…									
解析1	G1	G1	G2	G2					
解析2	G1	G1			G2	G2			
解析3	G1	G1							G2
解析4	G1	G1							

解析4の予想：DEGあり

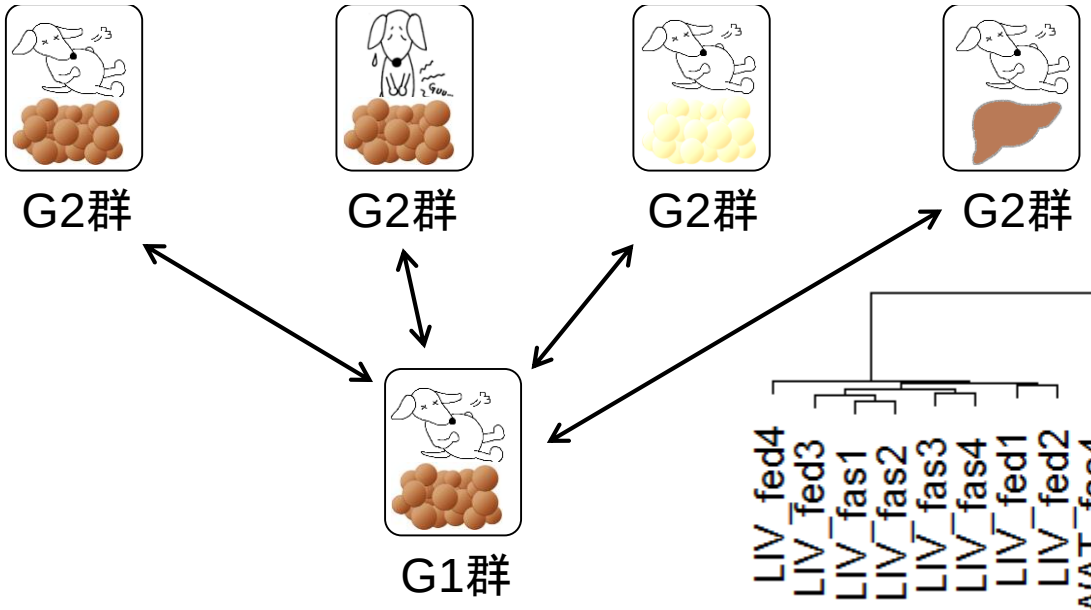
通常（私）は、0.05～0.30あたりのFDR閾値を調査→DEGがあると判断

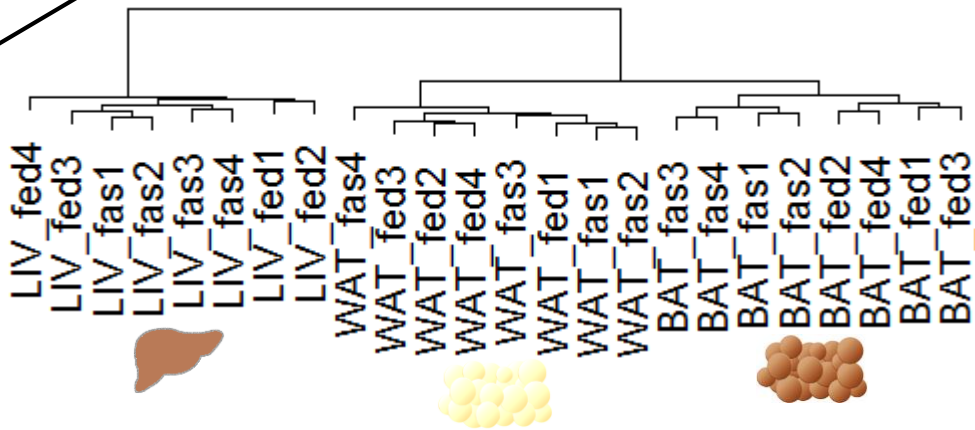
R Console

```
> colnames(data)
[1] "BAT_fed1" "BAT_fed2" "LIV_fed1" "LIV_fed2"
>
> #本番
> design <- model.matrix(~data.cl)          #デザ$
> fit <- lmFit(data, design)                #モデ$
> out <- eBayes(fit)                        #検定$
> p.value <- out$p.value[,ncol(design)]     #p値$
> q.value <- p.adjust(p.value, method="BH")#q$
> ranking <- rank(p.value)                  #p.va$
> sum(q.value < 0.05)                       #FDR $
[1] 2892
> sum(q.value < 0.10)                       #FDR $
[1] 5829
> sum(q.value < 0.30)                       #FDR $
[1] 13771
> sum(q.value < 0.50)                       #FDR $
[1] 19355
```

limmaによるDEG検出結果のまとめ

遺伝子数	解析1 G1群：BAT_fed G2群：BAT_fed	解析2 G1群：BAT_fed G2群：BAT_fas	解析3 G1群：BAT_fed G2群：WAT_fed	解析4 G1群：BAT_fed G2群：LIV_fed
FDR < 0.05	0	0	0	2892
FDR < 0.10	0	38	0	5829
FDR < 0.30	0	2375	4786	13771
FDR < 0.50	0	8993	12733	19355





解析1の予想：DEGなし
解析2~4の予想：DEGあり
予想されるDEG数：解析2 < 解析3 < 解析4

課題

- GSE7623のRMAおよびRobLoxBioCデータについてもlimmaを用いて同様の解析を実行し、以下の問いに答えよ。
  - RMAデータの解析結果

遺伝子数	解析1 G1群：BAT_fed G2群：BAT_fed	解析2 G1群：BAT_fed G2群：BAT_fas	解析3 G1群：BAT_fed G2群：WAT_fed	解析4 G1群：BAT_fed G2群：LIV_fed
FDR＜0.05				
FDR＜0.10				
FDR＜0.30				
FDR＜0.50				

  - RobLoxBioCデータの解析結果

遺伝子数	解析1 G1群：BAT_fed G2群：BAT_fed	解析2 G1群：BAT_fed G2群：BAT_fas	解析3 G1群：BAT_fed G2群：WAT_fed	解析4 G1群：BAT_fed G2群：LIV_fed
FDR＜0.05				
FDR＜0.10				
FDR＜0.30				
FDR＜0.50				

  - MAS5データの結果も含めた考察

Contents（第3回）

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§ 4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - **関数の利用法**
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - 描画（M-A plot）
    - 作成法
    - 同一群内のばらつき（前処理法間の違い）

2群間比較：Student's t-test

4.と5.と6.は実質的に同じです

4. サンプルデータ23のsample23.txtの場合:

最初の3サンプルがG1群、残りの3サンプルがG2群の 標準正規分布に従う乱数からなるシミュレーションデータです。乱数発生後に、さらに最初の10% 分についてG1群に相当するところのみ数値を+3している(つまり10%がG1群で高発現という

```
in_f <- "sample23.txt"
out_f <- "hoge4.txt"
param_G1 <- 3
param_G2 <- 3

#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, r
data.cl <- c(rep(1, param_G1), rep(2, p

#等分散性を仮定（var.equal=T）してt.testを
Students_ttest <- function(x, cl){
    x.class1 <- x[(cl == 1)]
    x.class2 <- x[(cl == 2)]
    if((sd(x.class1)+sd(x.class2)) == 0
        stat <- 0
        pval <- 1
        return(c(stat, pval))
    }
    else{
        hoge <- t.test(x.class1, x.class2, var.equ
        return(c(hoge$statistic, hoge$p.value))#統
    }
```

5. サンプルデータ23のsample23.txtの場合:

4.と同じですが、関数の定義の仕方が異なります。

```
in_f <- "sample23.txt"                  #入力ファイル名を指定してin_fに格納
out_f <- "hoge5.txt"                    #出力ファイル名を指定してout_f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

#必要な関数などをロード
source("http://www.iu.a.u-tokyo.ac.jp/~kadota/R/R_functions.R")#Student'

#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="")#i
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベ

#本番
out <- t
```

6. サンプルデータ23のsample23.txtの場合:

5.とほぼ同じですが、作業ディレクトリ中にStudents_ttest関数を含むR_functions.Rという名前のファイルが存在するという前提です。

```
in_f <- "sample23.txt"                  #入力ファイル名を指定してin_fに格納
out_f <- "hoge6.txt"                    #出力ファイル名を指定してout_f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

#必要な関数などをロード
source("R_functions.R")                 #Student's t-testを行うStudents_tt

#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="")#i
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベ

#本番
out <- t(apply(data, 1, Students_ttest, data.cl))#各（行）遺伝子についてt検
```

5. サンプルデータ23のsample23.txtの場合:

4.と同じですが、関数の定義の仕方が異なります。

```
in_f <- "sample23.txt"                  #入力ファイル名を指定してin_fに格納
out_f <- "hoge5.txt"                    #出力ファイル名を指定してout_f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

#必要な関数などをロード
source("http://www.iu.a.u-tokyo.ac.jp/~kadota/R/R_functions.R")#Student'

#入力ファ
data <-
data.cl

#本番
out <- t
```

• 解析 | 発現変動 | 2群間 | 対応なし | Student's t-test

ネット接続環境であれば、ここで提供している関数を利用可能

http://www.iu.a.u-tokyo.ac.jp/~kadota/R/R_functions.R

iu.a.u-tokyo.ac.jp

```
############################
###  Student's t-test    ###
############################
Students_ttest <- function(x, cl){
    x.class1 <- x[(cl == 1)]                    #ラベルが1のものをx.class1に格納
    x.class2 <- x[(cl == 2)]                    #ラベルが2のものをx.class2に格納
    if((sd(x.class1)+sd(x.class2)) == 0){       #両方の群の標準偏差が共に0の場合は計算できないので...
        stat <- 0                               #統計量を0
        pval <- 1                               #p値を1
        return(c(stat, pval))                   #として結果を結果として返す
    }
    else{                                       #A, Bどちらかの群の標準偏差が0(上記条件以外)の場合は
        hoge <- t.test(x.class1, x.class2, var.equal=T)  #t検定を行って、
        return(c(hoge$statistic, hoge$p.value))  #統計量とp値を結果として返す
    }
}

############################
###  Welth t-test        ###
############################
Welch_ttest <- function(x, cl){
    x.class1 <- x[(cl == 1)]                    #ラベルが1のものをx.class1に格納
    x.class2 <- x[(cl == 2)]                    #ラベルが2のものをx.class2に格納
    if((sd(x.class1)+sd(x.class2)) == 0){       #両方の群の標準偏差が共に0の場合は計算できないので...
        stat <- 0                               #統計量を0
        pval <- 1                               #p値を1
        return(c(stat, pval))                   #として結果を結果として返す
    }
    else{                                       #A, Bどちらかの群の標準偏差が0(上記条件以外)の場合は
        hoge <- t.test(x.class1, x.class2, var.equal=F)  #t検定を行って、
        return(c(hoge$statistic, hoge$p.value))  #統計量とp値を結果として返す
    }
}
```

6. サンプルデータ23のsample23.txtの場合:

5.とほぼ同じですが、作業ディレクトリ中にStudents_ttest関数を含むR_functions.Rという名前のファイルが存在するという前提です。

```
in_f <- "sample23.txt"                  #入力ファイル名を指定してin_fに格納
out_f <- "hoge6.txt"                    #出力ファイル名を指定してout_f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

#必要な関数などをロード
source("R_functions.R")                 #Student's t-testを行うStudents_t

#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="")#i
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベ
```

• 解析 | 発現変動 | 2群間 | 対応なし | Student's t-test

ネット接続環境でなくても、一旦R_functions.Rファイルを作業ディレクトリにダウンロードしておけばStudents_ttest関数を利用可能

C:¥Users¥kadota¥Desktop¥hoge¥R_functions.R - EmEditor

ファイル(F) 編集(E) 検索(S) 表示(V) ツール(T) ウィンドウ(W) ヘルプ(H)

R_functions.R

```
##############################↓
###  Student's t-test     ###↓
##############################↓
Students_ttest <- function(x, cl){↓
    x.class1 <- x[(cl == 1)]                    #ラベルが1のものをx.class1に格納↓
    x.class2 <- x[(cl == 2)]                    #ラベルが2のものをx.class2に格納↓
    if((sd(x.class1)+sd(x.class2)) == 0){       #両方の群の標準偏差が共に0の場合は計算できないので...
        stat <- 0                               #統計量を0↓
        pval <- 1                               #p値を1↓
        return(c(stat, pval))                   #として結果を結果として返す↓
    }↓
    else{                                       #A, Bどちらかの群の標準偏差が0(上記条件以外)の場合は↓
        hoge <- t.test(x.class1, x.class2, var.equal=T)  #t検定を行って、↓
        return(c(hoge$statistic, hoge$p.value))  #統計量とp値を結果として返す↓
    }↓
}↓
```

25.3 KB (25,911 バイト), 681 行。 Text 1行, 1桁 日本語 (シフト JIS)

IBMT法でDEG検出

- 解析 | 発現変動 | 2群間 | 対応なし | Random forest (Diaz-Uriarte_2007) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | shrinkage t (Opgen-Rhein_2007) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | layer ranking algorithm (Chen_2007) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | fdr2d (Ploner_2006) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Rank products (Breitling_2004) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | samroc (Bro
- 解析 | 発現変動 | 2群間 | 対応なし | SAM (Tushe
- 解析 | 発現変動 | 2群間 | 対応なし | Student's t-te
- 解析 | 発現変動 | 2群間 | 対応なし | Welch t-test(
- 解析 | 発現変動 | 2群間 | 対応なし | Mann-Whitn
- 解析 | 発現変動 | 2群間 | 対応なし | パターンマッ
- 解析 | 発現変動 | 2群間 | 対応あり | について(las
- 解析 | 発現変動 | 2群間 | 対応あり | SAM (Tushe
- 解析 | 発現変動 | 2群間 | 対応あり | SAM (Tushe
- 解析 | 発現変動 | 2群間 | 対応あり | 時系列 |につ

IBMT法を用いて
DEG検出を行います。

解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006)

IBMT法 (Sartor et al., 2006)の方法を用いて2群間で発現の異なる遺伝子をランキング。empirical Bayes (Smyth_2004)の改良版という位置づけですね。a novel Bayesian moderated-Tと書いてますし。
「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20の31,099 probesets×8 samplesのdata_rma_2_LIV.txt(G1群4サンプル vs. G2群4サンプル)の場合:

```
in_f <- "data_rma_2_LIV.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
param_G1 <- 4                             #G1群のサンプル数を指定
param_G2 <- 4                             #G2群のサンプル数を指定

#必要なパッケージなどをロード
library(limma)                            #パッケージの読み込み
source("http://eh3.uc.edu/r/ibmtR.R")     #IBMTのRスクリプトの読み込み

#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
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.

#本番
#design <- model.matrix(~ as.factor(data.cl))#デザイン行列を作成した結果をdesignに格
design <- model.matrix(~data.cl)          #デザイン行列を作成した結果をdesignに格納
fit <- lmFit(data, design)                #モデル構築(ばらつきの程度を見積もっている)
fit$Amean<-rowMeans(data)                 #おまじない
fit <- IBMT(fit,2)                        #IBMTプログラムの実行
p.value <- fit$IBMT.p                     #p値をp.valueに格納
```

IBMT法は、limmaパッケージ中の関数を内部的に用いています。limmaを基本としつつ、改良を加えた関数部分のみ提供しているという解釈でもよい。

http://eh3.uc.edu/r/ibmtR.R

eh3.uc.edu

```
IBMT<-function(mdata,testcol)
{ ##############################################################################
# Function for IBMT (Intensity-based Moderated T-statistic) # Written by: Maureen Sartor,
University of Cincinnati, 2006
############################################################################## ##
## This function adjusts the T-statistics and p-values from a linear ## model analysis of
microarrays. The method contains elements similar in ## nature both to Smyth's eBayes
function in limma and to the Cyber-T ## program (Baldi, 2001). It is an empirical
hierarchical Bayesian method. ## Local regression and empirical bayesian theory are used
to ## determine the prior degrees of freedom and the predicted background (prior) ##
variance for each gene dependent on average spot
T-statistic uses a weighted average of prior and li
degrees of freedom are simply the sum of ## prior
Please acknowledge your use of IBMT in publicatio
Tomlinson CR, Wesselkamper SC, Sivaganesan S,
Intensity-based hierarchical Bayes method improv
genes in microarray experiments. BMC Bioinforma
mdata and testcol ## "mdata" should be a list obj
limma, or at least have attributes named sigma, Ar
stdev.unscaled ## "testcol" is an integer or vecto
mdata$coefficients for which the function is to be
augmented form of "mdata" (the input), with the a
t-value for IBMT ## IBMT.p - P-value for IBMT ##
for IBMT ## IBMT.priorvar- prior variance for IBM
Call: ## IBMT.re
ion, contact sar
gVAR<-log(mdat
df[df==0]<-NA e
,iterations=1,spa
ression fit") mean.
priordf<-vector(); testd0<-vector() for (i in 1:(num
abs(mean.myfct-trigamma(testd0[i]/2)) if (i>2) { if
```

IBMT法の実体であるIBMT関数の読み込みを行っている

解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006)

IBMT法 (Sartor et al., 2006)の方法を用いて2群間で発現の異なる遺伝子をランキング。empirical Bayes (Smyth_2004)の改良版という位置づけですね。a novel Bayesian moderated-Tと書いてますし。
「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20の31,099 probesets×8 samplesのdata_rma_2_LIV.txt(G1群4サンプル vs. G2群4サンプル)の場合:

```
in_f <- "data_rma_2_LIV.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
param_G1 <- 4                             #G1群のサンプル数を指定
param_G2 <- 4                             #G2群のサンプル数を指定

#必要なパッケージなどをロード
library(limma)                            #パッケージの読み込み
source("http://eh3.uc.edu/r/ibmtR.R")     #IBMTのRスクリプトの読み込み

#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
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.

#本番
#design <- model.matrix(~ as.factor(data.cl))#デザイン行列を作成した結果をdesignに格
design <- model.matrix(~data.cl)          #デザイン行列を作成した結果をdesignに格納
fit <- lmFit(data, design)                #モデル構築(ばらつきの程度を見積もっている)
fit$Amean<-rowMeans(data)                 #おまじない
fit <- IBMT(fit,2)                        #IBMTプログラムの実行
p.value <- fit$IBMT.p                     #p値をp.valueに格納
```

IBMT法でDEG検出

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fed1	WAT_fed2	WAT_fed3	WAT_fed4	WAT_fas1	WAT_fas2	WAT_fas3	WAT_fas4	LIV_fed1	LIV_fed2	LIV_fed3	LIV_fed4	LIV_fas1	LIV_fas2	LIV_fas3	LIV_fas4
1367452_at																								
1367453_at																								
1367454_at																								
1367455_at																								
1367456_at																								
1367457_at																								
1367458_at																								
1367459_at																								
1367460_at																								
1367461_at																								
...																								
解析1	G1	G1	G2	G2																				
解析2	G1	G1			G2	G2																		
解析3	G1	G1							G2	G2														
解析4	G1	G1															G2	G2						

解析1の予想：DEGなし
解析2~4の予想：DEGあり
予想されるDEG数：解析2 < 解析3 < 解析4

IBMT法で解析1をやってみよう

解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006)

IBMT法 (Sartor et al., 2006)の 方法を用いて2群間で発現の 異なる遺伝子をランキング。empirical Bayes (Smyth_2004)の改良版という位置づけですね。a nov
「ファイル」-「ディレクトリの 変更」で解析したいファイル

1. サンプルデータ20の31,099 probesets×8 samplesの
ル)の場合:

```
in_f <- "data_rma_2_LIV.txt"
out_f <- "hoge1.txt"
param_G1 <- 4
param_G2 <- 4

#必要なパッケージなどをロード
library(limma)
source("http://eh3.uc.edu/r/ibmtR.R")

#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,
data.cl <- c(rep(1, param_G1), rep(2,

#本番
#design <- model.matrix(~ as.factor(da
design <- model.matrix(~data.cl)
fit <- lmFit(data, design)
fit$Amean<-rowMeans(data)
fit <- IBMT(fit,2)
p.value <- fit$IBMT.p
```

rcode_ibmt_basic.txt（変更点および追加点）

```
############################
###  解析1 (Analysis1)  ###
############################
in_f <- "data_mas_EN.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
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

#必要なパッケージをロード
library(limma)                           #パッケージの読み込み
source("ibmtR.R")                        #IBMTのRスクリプトの読み込み

#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
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

#サブセットの作成（解析したいデータのみにする）
posi <- c(1,2,3,4)                       #元の発現行列上での列番号を指定
data <- data[,posi]                      #サブセットを抽出

#本番
design <- model.matrix(~data.cl)         #デザイン行列を作成した結果をdesignに格納
fit <- lmFit(data, design)               #モデル構築(ばらつきの程度を見積もっている)
fit$Amean<-rowMeans(data)                #おまじない
fit <- IBMT(fit,2)                       #IBMTプログラムの実行
p.value <- fit$IBMT.p                    #p値をp.valueに格納
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納
ranking <- rank(p.value)                 #p.valueでランキングした結果をrankingに格納
sum(q.value < 0.05)                      #FDR < 0.05を満たす遺伝子数を表示

#ファイルに保存
tmp <- cbind(rownames(data), data, p.value, q.value, ranking)#入力データの右側にp.v
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定し
```

rcode_ibmt_basic.txt（変更点および追加点）

```
############################↓
###  解析1 (Analysis1)  ###↓
############################↓
in_f <- "data_mas_EN.txt"                #入力ファイル名を指定して
out_f <- "hoge1.txt"                     #出力ファイル名を指定して
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↓
↓
#必要なパッケージをロード↓
library(limma)                           #パッケージの読み込み↓
source("ibmtR.R")                        #IBMTのRスクリプトの読み
↓
#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", qu
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を
↓
#サブセットの作成（解析したいデータのみにする）↓
posi <- c(1,2,3,4)                       #元の発現行列上での列番号
data <- data[,posi]                      #サブセットを抽出↓
↓
#本番↓
design <- model.matrix(~data.cl)         #デザイン行列を作成した
fit <- lmFit(data, design)               #モデル構築(ばらつきの程
fit$Amean<-rowMeans(data)                #おまじない↓
fit <- IBMT(fit,2)                       #IBMTプログラムの実行↓
p.value <- fit$IBMT.p                    #p値をp.valueに格納↓
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納↓
ranking <- rank(p.value)                 #p.valueでランキングした
sum(q.value < 0.05)                      #FDR < 0.05を満たす遺伝子
↓
#ファイルに保存↓
tmp <- cbind(rownames(data), data, p.value, q.value, ranking)#入
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=
```

R Console

```
> #本番
> design <- model.matrix(~data.cl)          #$
> fit <- lmFit(data, design)                #$
> fit$Amean<-rowMeans(data)                 #$
> fit <- IBMT(fit,2)                        #$
[1] "Local regression fit"
[1] "Prior degrees freedom found"
[1] "P-values calculated"
> p.value <- fit$IBMT.p                     #$
> q.value <- p.adjust(p.value, method="BH"$
> ranking <- rank(p.value)                  #$
> sum(q.value < 0.05)                       #$
[1] 0
>
> #ファイルに保存
> tmp <- cbind(rownames(data), data, p.val$
> write.table(tmp, out_f, sep="\t", append$
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/GSE7623"
> sum(q.value < 0.1)
[1] 0
> sum(q.value < 0.3)
[1] 0
> sum(q.value < 0.5)
[1] 0
> sum(q.value < 0.7)
[1] 0
> sum(q.value < 0.8)
[1] 0
>
```

解析1の結果は妥当

IBMT法によるDEG検出結果のまとめ

MAS5

遺伝子数	解析1 G1群：BAT_fed G2群：BAT_fed	解析2 G1群：BAT_fed G2群：BAT_fas	解析3 G1群：BAT_fed G2群：WAT_fed	解析4 G1群：BAT_fed G2群：LIV_fed
FDR＜0.05	0	1927	1999	7256
FDR＜0.10	0	2891	3246	9227
FDR＜0.30	0	6729	8030	14607
FDR＜0.50	0	11491	13602	19125

RMA

遺伝子数	解析1 G1群：BAT_fed G2群：BAT_fed	解析2 G1群：BAT_fed G2群：BAT_fas	解析3 G1群：BAT_fed G2群：WAT_fed	解析4 G1群：BAT_fed G2群：LIV_fed
FDR＜0.05	0	2889	2988	8965
FDR＜0.10	0	4348	5570	10954
FDR＜0.30	0	8973	14364	16059
FDR＜0.50	0	13927	20056	20305

RobLoxBioC

遺伝子数	解析1 G1群：BAT_fed G2群：BAT_fed	解析2 G1群：BAT_fed G2群：BAT_fas	解析3 G1群：BAT_fed G2群：WAT_fed	解析4 G1群：BAT_fed G2群：LIV_fed
FDR＜0.05	0	3066	2169	9196
FDR＜0.10	0	4527	4079	11434
FDR＜0.30	0	10008	11627	17265
FDR＜0.50	0	15485	17558	21451

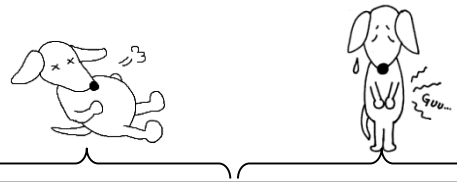
rcode_ibmt_all.txt

同じDEG検出法でも入力データ（前処理法）が違うと結果もずいぶん異なる

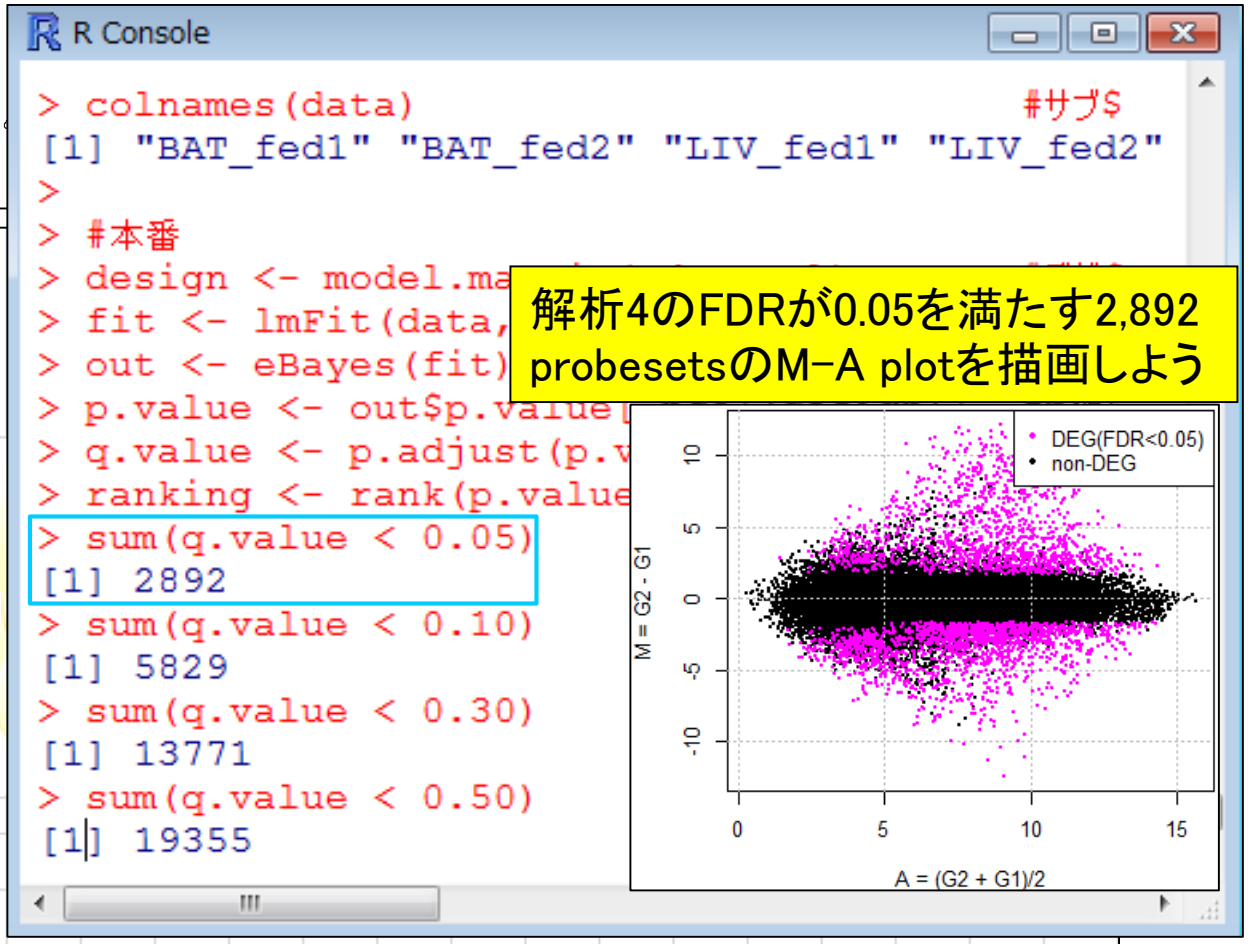
Contents（第3回）

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - 関数の利用法
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - **描画（M-A plot）**
    - **作成法**
    - **同一群内のばらつき（前処理法間の違い）**

limmaでDEG検出した結果のM-A plot



	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fed1
1367452_at									
1367453_at									
1367454_at									
1367455_at									
1367456_at									
1367457_at									
1367458_at									
1367459_at									
1367460_at									
1367461_at									
…									
解析1	G1	G1	G2	G2					
解析2	G1	G1			G2	G2			
解析3	G1	G1							G2
解析4	G1	G1							G2



- 解析 | 発現変動 | 2群間 | 対応なし | IBMT (Sartor_2006) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | Rank products (Breitling_2004) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | samroc (Broberg_2003) (last modified 2014/02/03)
- 解析 | 発現変動 | 2群間 | 対応なし | SAM (Tusher_2001) (last modified 2014/02/03)

- 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004)

解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth_2004)

limmaパッケージを用いて2群間比較を行うやり方を示します。

この方法は経
「ファイル」-「

1. サンプルデ
場合:

```
in_f <-
out_f <-
param_G1
param_G2

#必要なパ
library(

#入力ファ
data <-
data.cl

#本番
#design
design <
```

7. サンプルデータ20の31,099 probesets×8 samplesのdata_rma_2_LIV.txt(G1群4サンプル vs. G2群4サンプル)の場合:

M-A plotのpngファイルを生成しています。limmaパッケージ中のtopTable関数やplotMA関数を使わないやり方です。テキストファイルのほうは、M-A plotのM値とA値も出力させるようにしています。

```
in_f <- "data_rma_2_LIV.txt"              #入力ファイル名を指定してin_fに格納
out_f1 <- "hoge7.txt"                     #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge7.png"                     #出力ファイル名を指定してout_f2に格納
param_G1 <- 4                             #G1群のサンプル数を指定
param_G2 <- 4                             #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
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

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

#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
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.

#本番
#design <- model.matrix(~ as.factor(data.cl))#デザイン行列を作成した結果をdesignに格
design <- model.matrix(~data.cl)          #デザイン行列を作成した結果をdesignに格納
fit <- lmFit(data, design)                #モデル構築(ばらつきの程度を見積もっている)
out <- eBayes(fit)                        #検定(経験ベイズ)
p.value <- out$p.value[,ncol(design)]     #p値をp.valueに格納
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納
```

rcode_limma_MAplot_basic.txtは
これをテンプレートにしています

7. サンプルデータ20の31,099 probesets×8 samplesのdata_rma_2_LIV.txt(G1群4サンプル vs. G2群4サンプル)の場合:

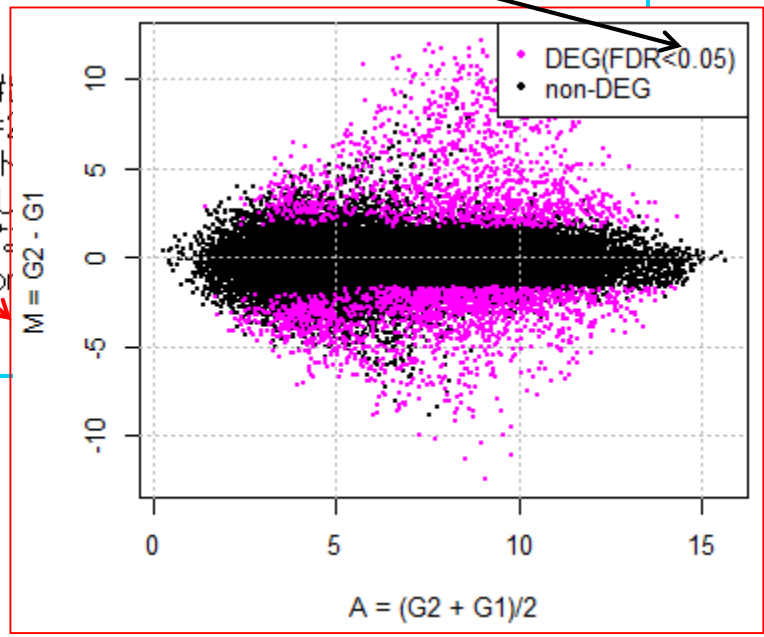
• 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth 2004)

M-A plotのpngファイルを生成しています。limmaパッケージ中のtopTable関数やplotMA関数を使わないやり方です。テキストファイルのほうは、M-A plotのM値とA値も出力させるようにしています。

```
in_f <- "data_mas_EN.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 <- 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_posi <- c(1,2,17,18)         #元の発現行列上での列番号を指定↓
param_FDR <- 0.05                  #DEG検出時のfalse discovery rate (FDR)閾値を指定↓
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
library(li
↓
#入力ファイ
data <- re
data.cl <-
data <- da
colnames(d
↓
#本番(DEG検
design <- m
fit <- lmF
out <- eBay
p.value <-
q.value <-
ranking <-
sum(q.value
```

```
mean_G1 <- apply(as.matrix(data[,data.cl==1]), 1, mean)#遺伝子ごとにG1群の平均を計算した結果をmean_G1に格納
mean_G2 <- apply(as.matrix(data[,data.cl==2]), 1, mean)#遺伝子ごとにG2群の平均を計算した結果をmean_G2に格納
M <- mean_G2 - mean_G1                  #M-A plotのM値(y軸の値)に相当するものをMに格納↓
A <- (mean_G1 + mean_G2)/2              #M-A plotのA値(x軸の値)に相当するものをAに格納↓
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, M, A, p.value, q.value, ranking)#入力データの右側にDEG検出結果を結合した
write.table(tmp, out_f1, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保存↓
↓
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=
grid(col="gray", lty="dotted")          #指定したパラメータでグリッ
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定し
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとな
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "no
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

横軸のAは平均発現レベル、
縦軸のMは$\log_2$(G2/G1)に相当



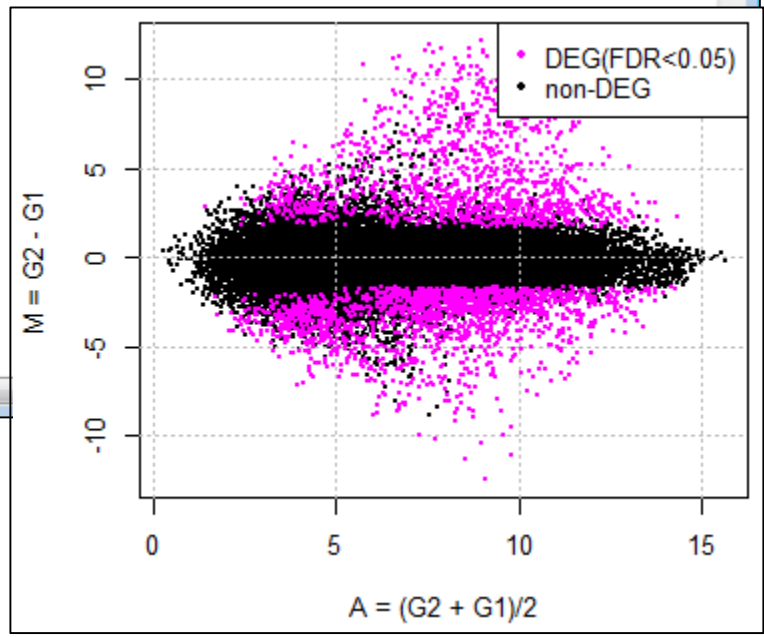
```
mean_G1 <- apply(as.matrix(data[,data.cl==1]), 1, mean)#遺伝子ごとにG1群の平均を計算した結果をmean_G1に格納
mean_G2 <- apply(as.matrix(data[,data.cl==2]), 1, mean)#遺伝子ごとにG2群の平均を計算した結果をmean_G2に格納
M <- mean_G2 - mean_G1                  #M-A plotのM値(y軸の値)に相当するものをMに格納
A <- (mean_G1 + mean_G2)/2              #M-A plotのA値(x軸の値)に相当するものをAに格納
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, M, A, p.value, q.value, ranking)#入力データの右側にDEG検出結果を結合した
write.table(tmp, out_f1, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保存↓
```

横軸のAは平均発現レベル、
縦軸のMは$\log_2$(G2/G1)に相当

R Console

```
> head(tmp)
          rownames(data) BAT_fed1 BAT_fed2 LIV_fed1 LIV_fed2         M        A   p.value   q.value ranking
1367452_at     1367452_at 12.78446 12.44708 12.19593 11.95968 -0.5379634 12.34679 0.1613769 0.3365743   14911
1367453_at     1367453_at 11.80125 12.15293 11.49419 11.49189 -0.4840521 11.73506 0.1846049 0.3647413   15740
1367454_at     1367454_at 11.38990 11.16076 11.66812 12.23333  0.6753955 11.61303 0.1322055 0.2991158   13745
1367455_at     1367455_at 12.36435 12.52974 12.80589 12.96296  0.4373753 12.66573 0.1991063 0.3812952   16239
1367456_at     1367456_at 13.44849 13.54305 13.38086 13.58722 -0.0117270 13.48990 0.9686717 0.9832814   30636
1367457_at     1367457_at 10.40403 10.69632 10.78228 10.61002  0.1459767 10.62316 0.6482376 0.7815199   25794
> head(cbind(mean_G1, mean_G2))
           mean_G1  mean_G2
1367452_at 12.61577 12.07781
1367453_at 11.97709 11.49304
1367454_at 11.27533 11.95072
1367455_at 12.44705 12.88442
1367456_at 13.49577 13.48404
1367457_at 10.55017 10.69615
> (12.78446 + 12.44708)/2
[1] 12.61577
>
```

mean_G1とmean_G2は、
単にグループごとの平
均値を算出しているだけ



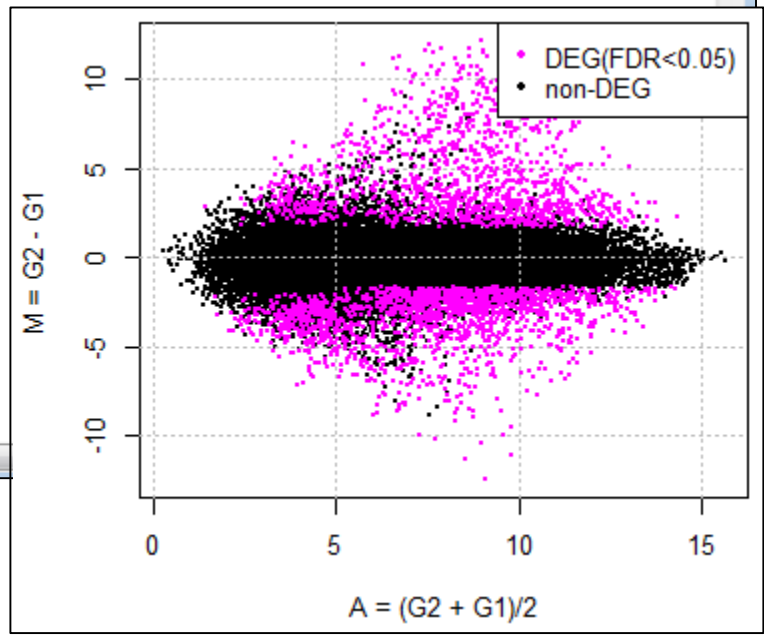
```
mean_G1 <- apply(as.matrix(data[,data.cl==1]), 1, mean)#遺伝子ごとにG1群の平均を計算した結果をmean_G1に格納
mean_G2 <- apply(as.matrix(data[,data.cl==2]), 1, mean)#遺伝子ごとにG2群の平均を計算した結果をmean_G2に格納
M <- mean_G2 - mean_G1                  #M-A plotのM値(y軸の値)に相当するものをMに格納
A <- (mean_G1 + mean_G2)/2              #M-A plotのA値(x軸の値)に相当するものをAに格納
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, M, A, p.value, q.value, ranking)#入力データの右側にDEG検出結果を結合した
write.table(tmp, out_f1, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指定したファイル名で保存↓
```

横軸のAは平均発現レベル、縦軸のMは$\log_2$(G2/G1)に相当

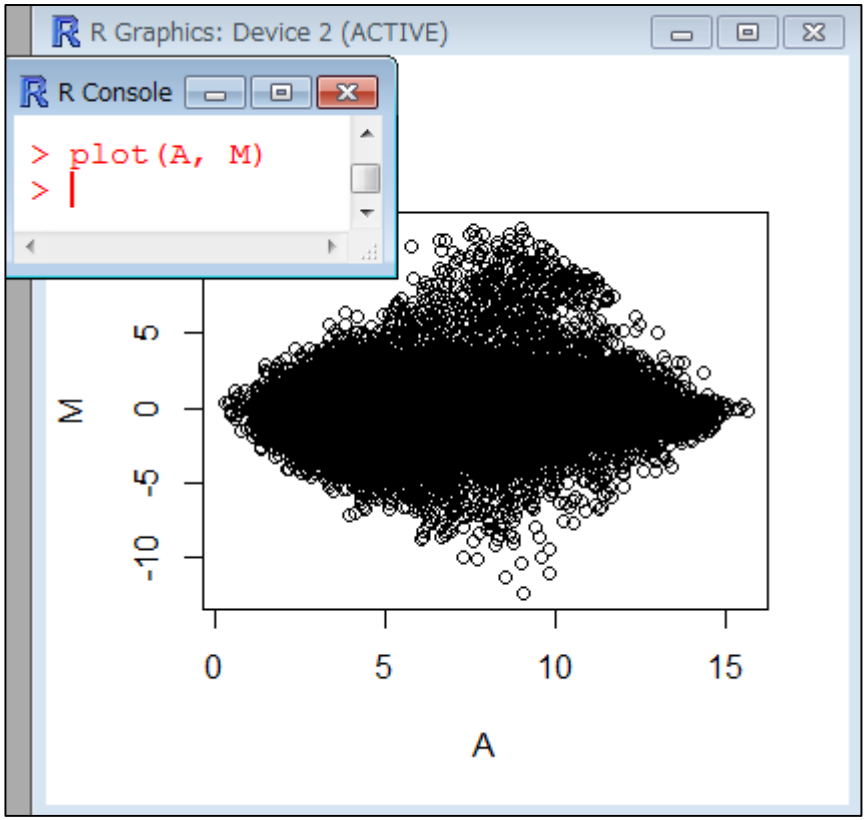
R Console

```
> head(tmp)
          rownames(data) BAT_fed1 BAT_fed2 LIV_fed1 LIV_fed2          M        A   p.value   q.value ranking
1367452_at     1367452_at 12.78446 12.44708 12.19593 11.95968 -0.5379634 12.34679 0.1613769 0.3365743   14911
1367453_at     1367453_at 11.80125 12.15293 11.49419 11.49189 -0.4040521 11.73506 0.1846049 0.3647413   15740
1367454_at     1367454_at 11.38990 11.16076 11.66812 12.23333  0.6753955 11.61303 0.1322055 0.2991158   13745
1367455_at     1367455_at 12.36435 12.52974 12.80589 12.96296  0.4373753 12.66573 0.1991063 0.3812952   16239
1367456_at     1367456_at 13.44849 13.54305 13.38086 13.58722 -0.0117270 13.48990 0.9686717 0.9832814   30636
1367457_at     1367457_at 10.40403 10.69632 10.78228 10.61002  0.1459767 10.62316 0.6482376 0.7815199   25794
> head(cbind(mean_G1, mean_G2))
           mean_G1  mean_G2
1367452_at 12.61577 12.07781
1367453_at 11.97709 11.49304
1367454_at 11.27533 11.95072
1367455_at 12.44705 12.88442
1367456_at 13.49577 13.48404
1367457_at 10.55017 10.69615
> 12.07781 - 12.61577
[1] -0.53796
> (12.07781 + 12.61577)/2
[1] 12.34679
>
```

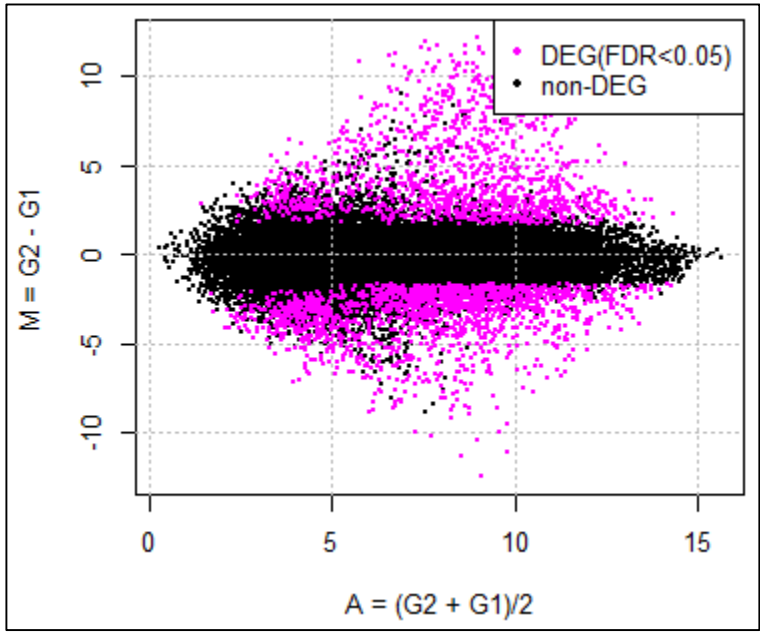
(mean_G2 − mean_G1)の計算結果を縦軸のMとして計算できるのは、発現レベルが対数変換後のデータだから



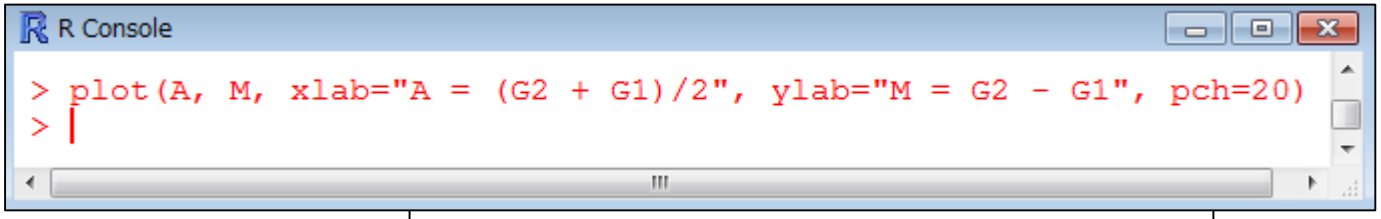
```
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```



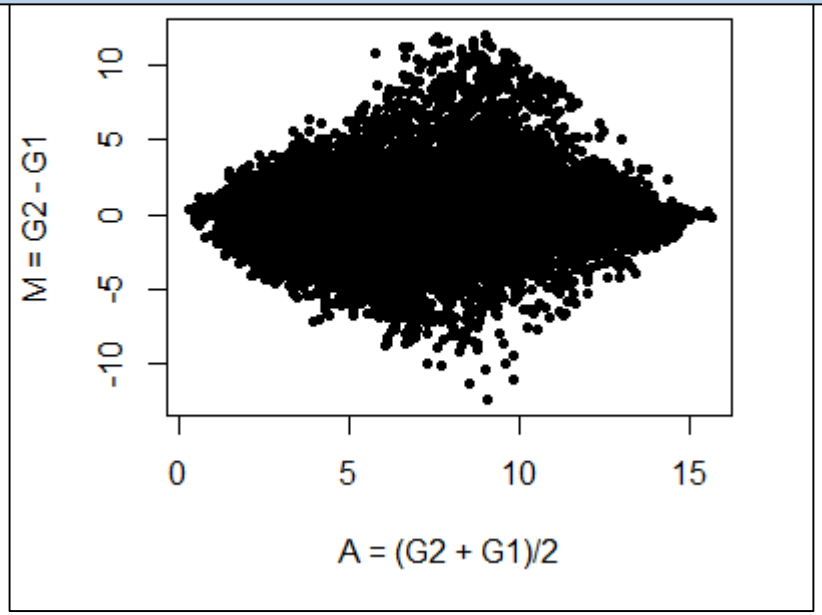
plotの基本形

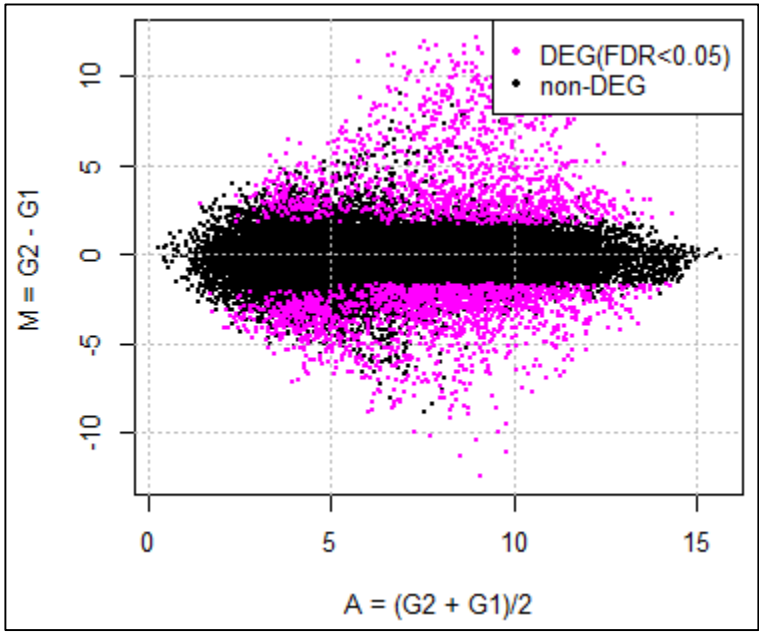


```
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

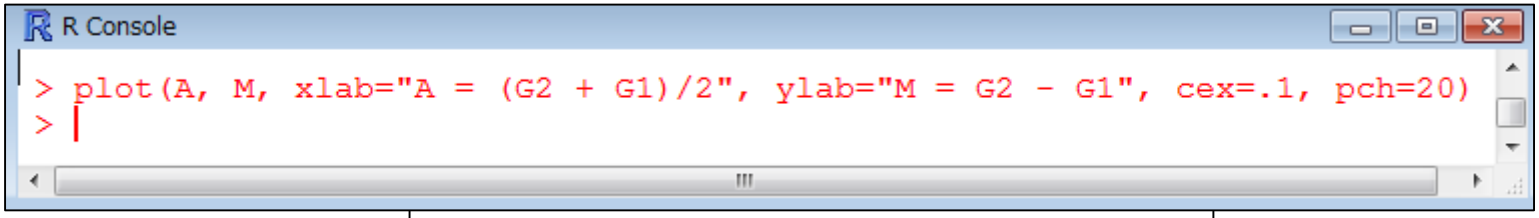


黒丸の塗りつぶしにすべく、pch=20オプションを追加

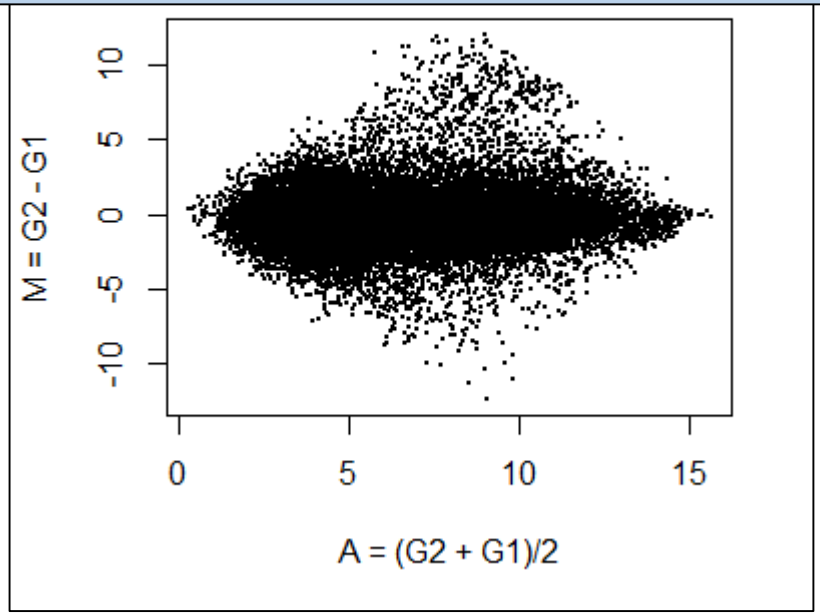


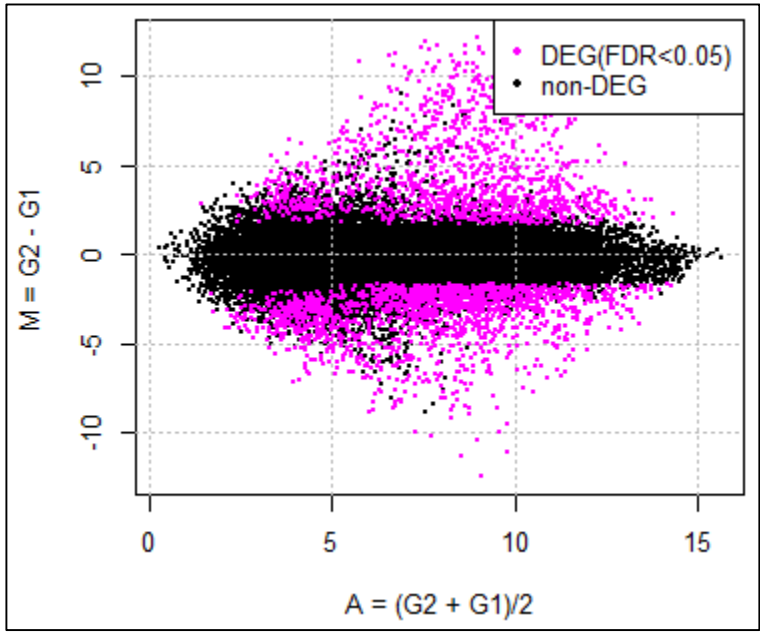


```
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

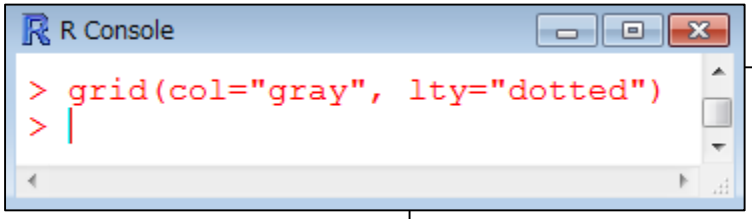


プロットの大きさをデフォルトの10%にすべく、cex=.1オプションを追加

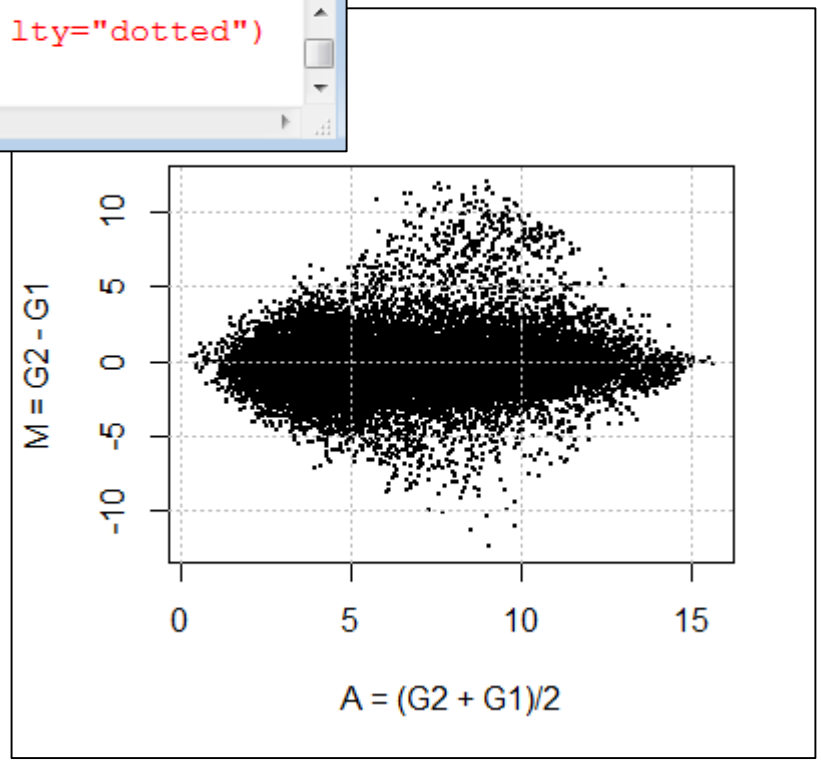


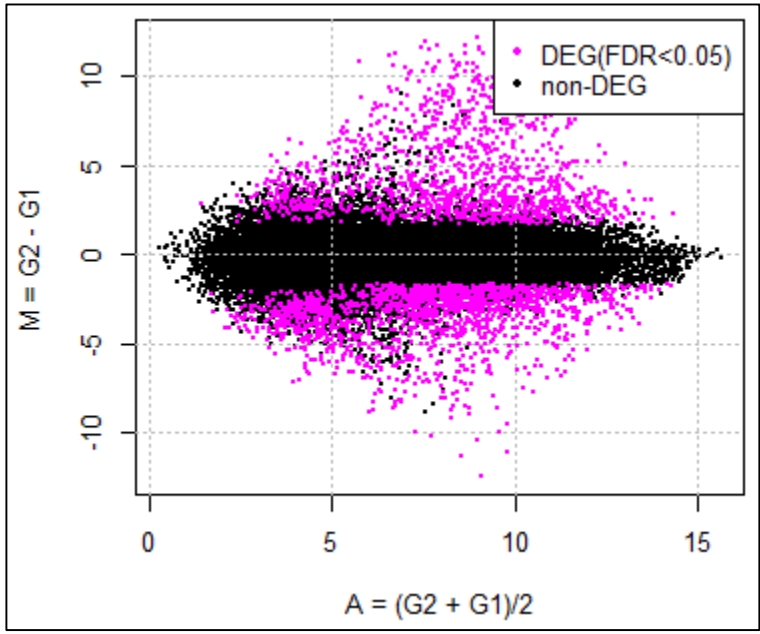


```
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

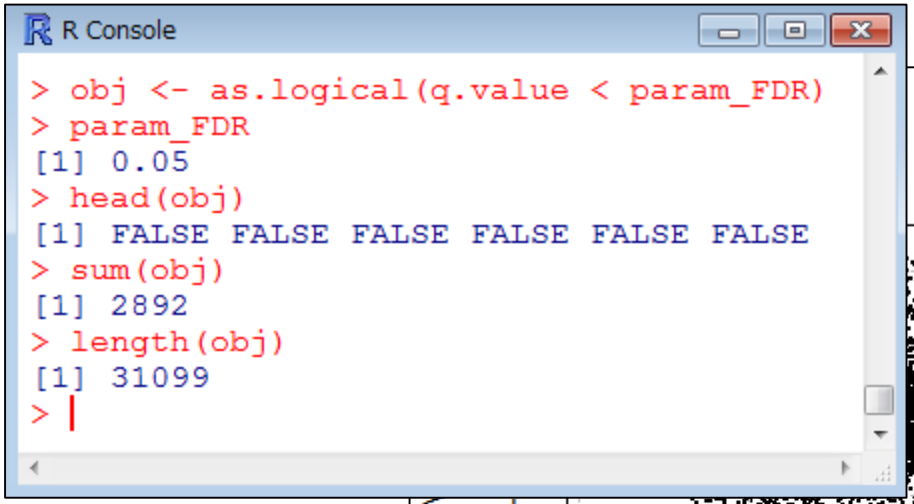


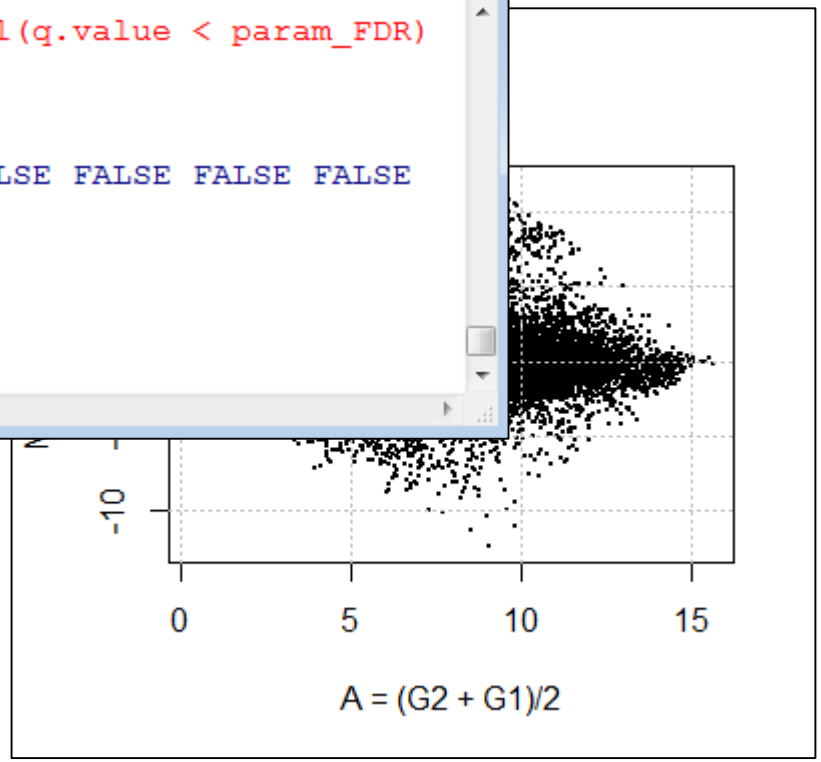
グリッド線を追加



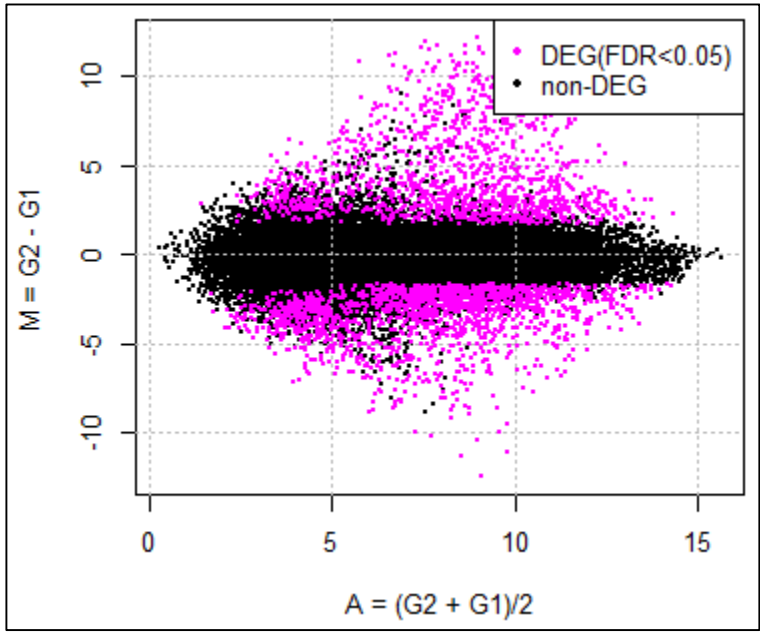


```
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

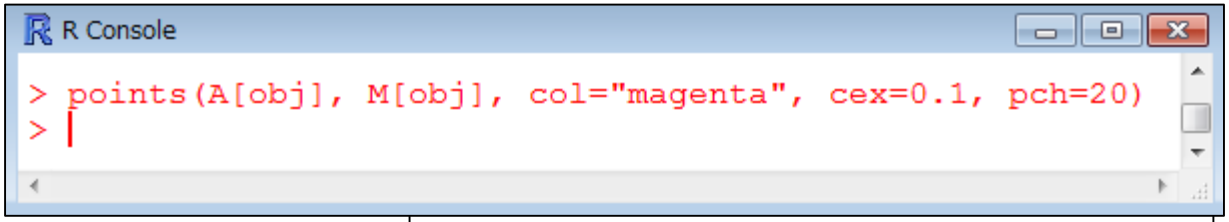


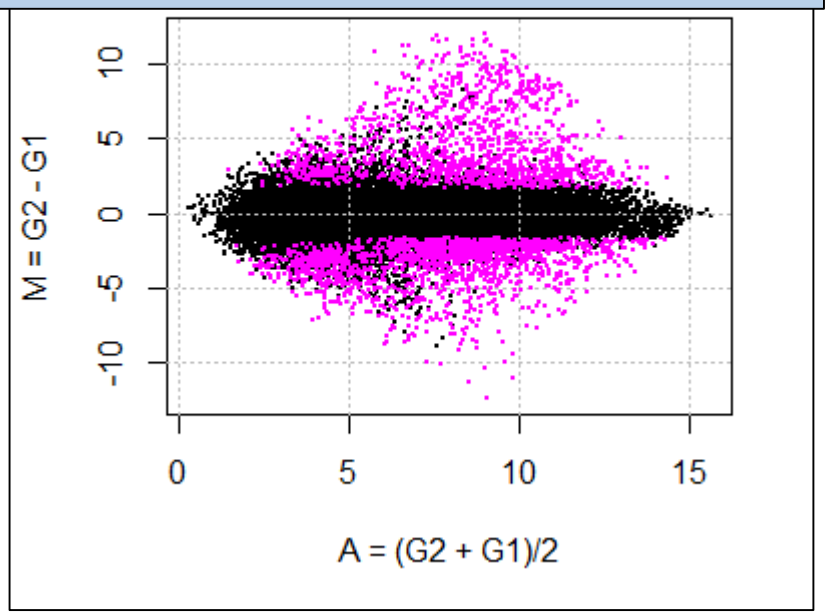


指定したFDR閾値を満たすDEGの位置情報を取得している

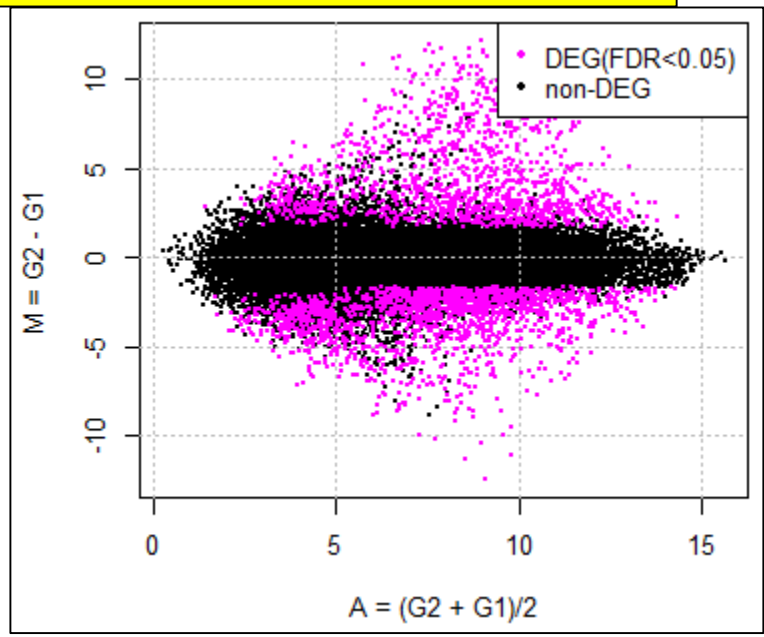


```
#ファイルに保存(M-A plot)↓
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメータを指定↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```





objベクトルがTRUEの場所をmagenta色で描画。cexとpchオプションの値を同じにすることで色だけを変更していることに相当



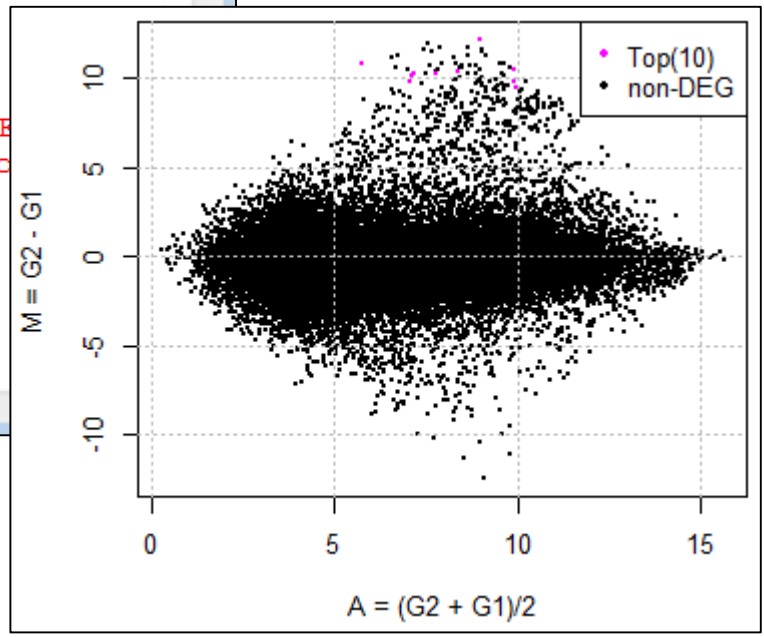
rcode_limma_MAplot_basic.txt
の下のほうのコードです

```
################################↓
#ファイルに保存(M-A plot)1
################################↓
param_TOP <- 10↓
png(out_f2, 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↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(ranking <= param_TOP)#条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("Top(", param_TOP, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

R Console

```
> param_TOP <- 10
> png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出$
> plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1,
> grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを$
> obj <- as.logical(ranking <= param_TOP)#条件を満たすかどうかを判定した$
> points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUE
> legend("topright", c(paste("Top(", param_TOP, ")", sep=""), "no
+        col=c("magenta", "black"), pch=20)#凡例を作成している
> dev.off()                               #おまじない
windows
      2
>
```

上位x個のみ色を変えることも簡単にできます



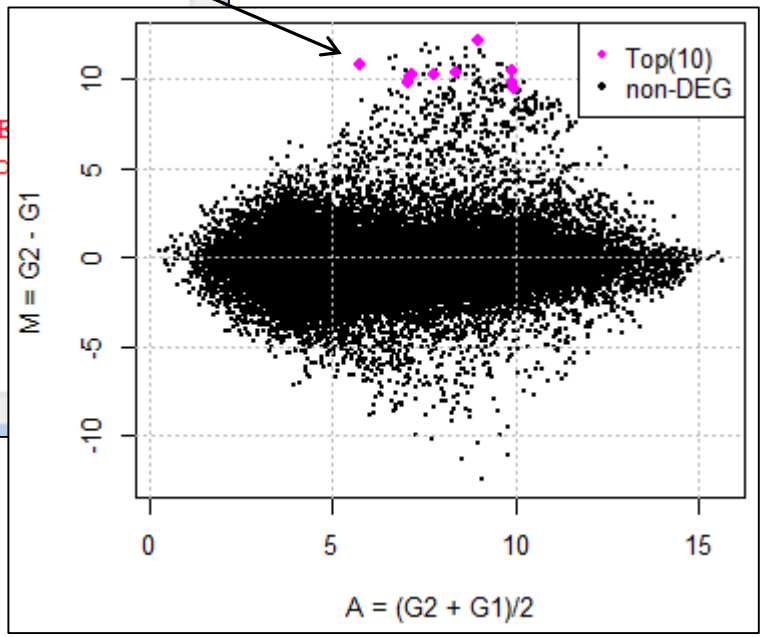
rcode_limma_MAplot_basic.txt
の下のほうのコードです

```
################################↓
#ファイルに保存(M-A plot)2
################################↓
param_TOP <- 10↓
png(out_f2, 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↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(ranking <= param_TOP)#条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)
points(A[obj], M[obj], col="magenta", cex=1.5, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("Top(", param_TOP, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
```

R Console

```
> param_TOP <- 10
> png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出$
> plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1", cex=.1,
> grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを$
> obj <- as.logical(ranking <= param_TOP)#条件を満たすかどうかを判定した$
> points(A[obj], M[obj], col="magenta", cex=1.5, pch=20)#objがTRUE
> legend("topright", c(paste("Top(", param_TOP, ")", sep=""), "no
+        col=c("magenta", "black"), pch=20)#凡例を作成している
> dev.off()                               #おまじない
windows
      2
>
```

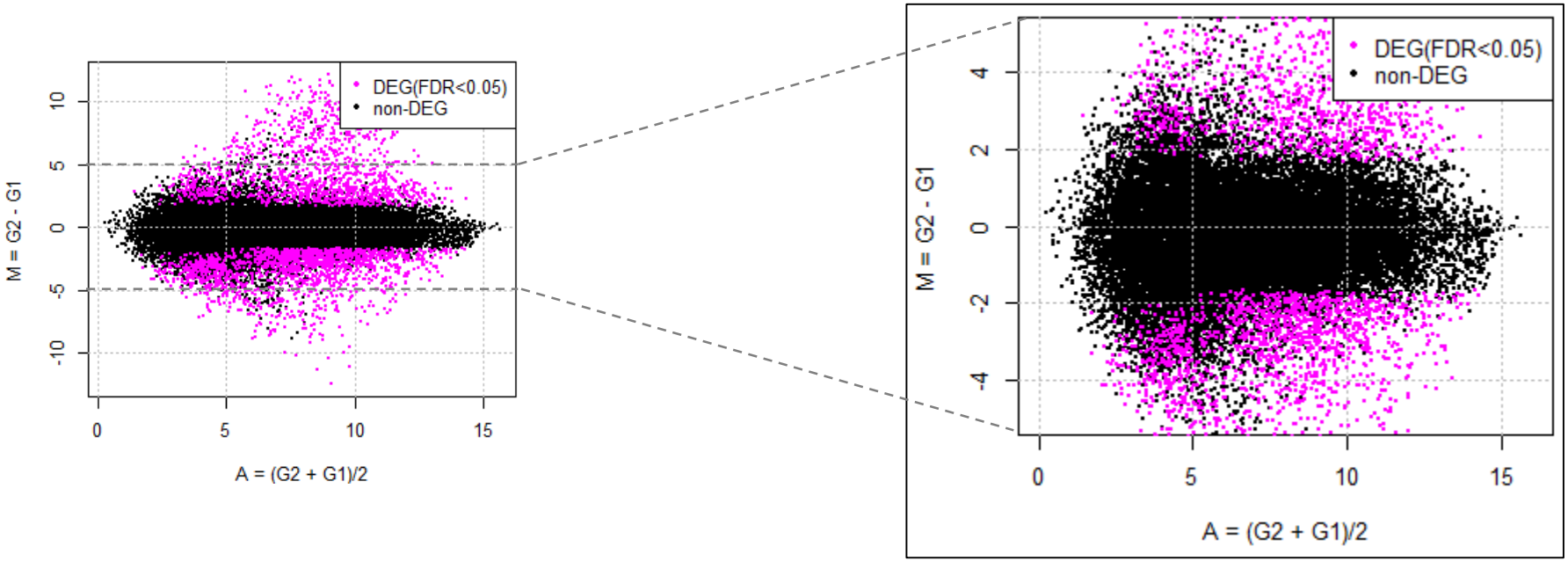
上位x個のみ色を変えて
大きくすることもできます



rcode_limma_MAplot_basic.txt
の下のほうのコードです

```
################################↓
#ファイルに保存(M-A plot)3↓
################################↓
param_xrange <- c(0, 16)                #M-A plotのx軸の範囲を指定↓
param_yrange <- c(-5, 5)                #M-A plotのy軸の範囲を指定↓
png(out_f2, 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↓
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1",    #M-A plotを描画↓
     ylim=param_yrange, xlim=param_xrange, cex=.1, pch=20)#M-A plotを描画↓
grid(col="gray", lty="dotted")          #指定したパラメータでグリッドを表示↓
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRUE、non-DEGがFALSE)↓
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画↓
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している↓
       col=c("magenta", "black"), pch=20)#凡例を作成している↓
dev.off()                               #おまじない↓
↓
↓
```

表示範囲を自在
に変更可能です



```
in_f <- "data_mas_EN.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 <- 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_posi <- c(1,2,3,4)                  #元の発現行列上での列番号を指定
param_FDR <- 0.05                          #DEG検出時のfalse discovery rate (FDR)閾値を指定
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(limma)                             #パッケージの読み込み

#入力ファイルの読み込みとラベル情報の作成、そしてサブセットの作成
data <- read.table(in_f, header=TRUE, row.names=1, sep="¥t", quote=
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
data <- data[,param_posi]                  #サブセットを抽出
colnames(data)                             #サブセット抽出後のサンプル名を表示

#本番(DEG検出)
design <- model.matrix(~data.cl)           #デザイン行列を作成した結果をdesignに格納
fit <- lmFit(data, design)                 #モデル構築(ばらつきの程度を見積もっている)
out <- eBayes(fit)                         #検定(経験ベイズ)
p.value <- out$p.value[,ncol(design)]      #p値をp.valueに格納
q.value <- p.adjust(p.value, method="BH")#q値をq.valueに格納
ranking <- rank(p.value)                   #p.valueでランキングした結果をrankingに格納
sum(q.value < param_FDR)                   #FDR < param_FDRを満たす遺伝子数を表示
mean_G1 <- apply(as.matrix(data[,data.cl==1]), 1, mean)#遺伝子ごとにG1群の平均を計算した結果を
mean_G2 <- apply(as.matrix(data[,data.cl==2]), 1, mean)#遺伝子ごとにG2群の平均を計算した結果を
M <- mean_G2 - mean_G1                     #M-A plotのM値(y軸の値)に相当するものをMに格納
A <- (mean_G1 + mean_G2)/2                 #M-A plotのA値(x軸の値)に相当するものをAに格納

#ファイルに保存(M-A plot)
param_xrange <- c(0, 16)                   #M-A plotのx軸の範囲を指定
param_yrange <- c(-5, 5)                   #M-A plotのy軸の範囲を指定
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの各種パラメー
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1",     #M-A plotを描画
     ylim=param_yrange, xlim=param_xrange, cex=.1, pch=20)#M-A plotを描画
grid(col="gray", lty="dotted")             #指定したパラメータでグリッドを表示
obj <- as.logical(q.value < param_FDR) #条件を満たすかどうかを判定した結果をobjに格納(DEGがTRU
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指定した色で描画
legend("topright", c(paste("DEG(FDR<", param_FDR, ")", sep=""), "non-DEG"),#凡例を作成している
       col=c("magenta", "black"), pch=20)#凡例を作成している
dev.off()                                  #おまじない
```

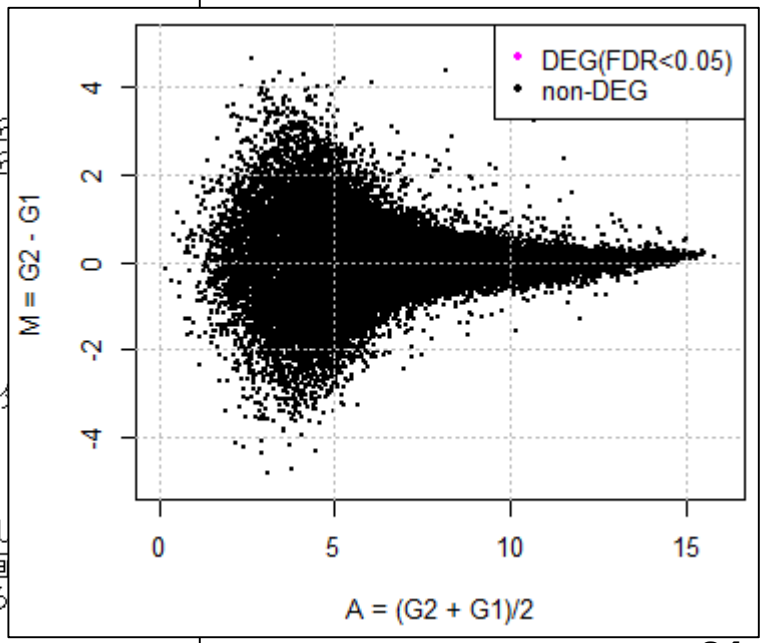
間 | 対応なし | empirical Bayes (Smyth 2004)

rcode_limma_MAplot_basic2.txtです

param_posiをc(5,6,7,8)に変更すればBAT_fas内のばらつきの程度を調べることに相当

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4
1367452_at				
1367453_at				
1367454_at				
1367455_at				
1367456_at				
解析1	G1	G1	G2	G2

同一群内のばらつきの程度を表す



```
#################################################################
###   ファイルに保存(M-A plot) 2倍以上発現変動をマゼンタ色に
#################################################################
param_FC <- 2                          #倍率変化の閾値を指定
param_xrange <- c(0, 16)               #M-A plotのx軸の範囲を指定
param_yrange <- c(-5, 5)               #M-A plotのy軸の範囲を指定
png(out_f2, pointsize=13, width=param_fig[1], height=param_fig[2])#出力ファイルの
plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1",    #M-A plotを描画
     ylim=param_yrange, xlim=param_xrange, cex=.1, pch=20)#M-A plotを描画
grid(col="gray", lty="dotted")         #指定したパラメータでグリッドを表示
obj <- as.logical(abs(M) >= log2(param_FC))#条件を満たすかどうかを判定した結果を
points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#objがTRUEとなる要素のみ指
legend("topright", c(paste("DEG(> ", param_FC, "-fold)", sep=""), "non-DEG"),#凡
       col=c("magenta", "black"), pch=20)#凡例を作成している
abline(h=log2(param_FC), col="red")    #M=log2(param_FC)の直線を表示
abline(h=-log2(param_FC), col="red")   #M=-log2(param_FC)の直線を表示
dev.off()                              #おまじない
```

R Console

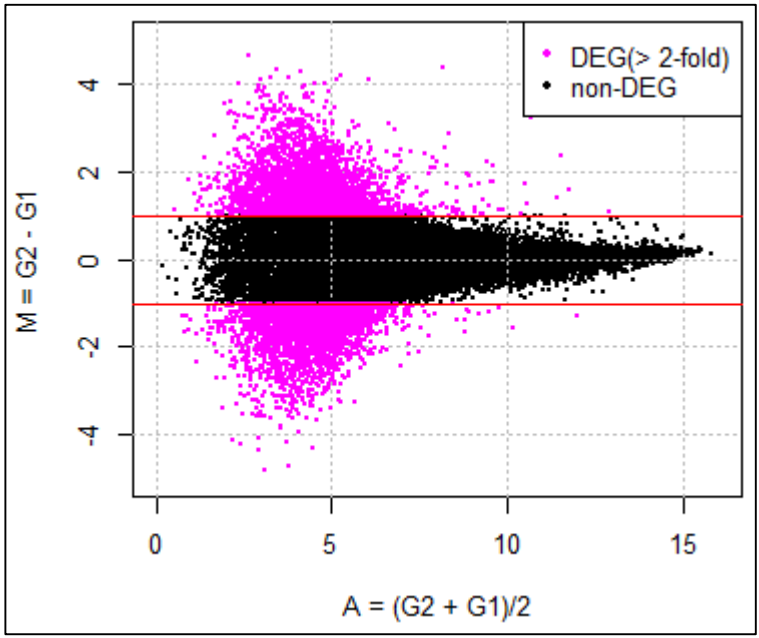
```
> param_FC <- 2                                  #倍率変化の閾値を$
> param_xrange <- c(0, 16)                       #M-A plotのx軸の範$
> param_yrange <- c(-5, 5)                       #M-A plotのy軸の範$
> png(out_f2, pointsize=13, width=param_fig[1], height=para$
> plot(A, M, xlab="A = (G2 + G1)/2", ylab="M = G2 - G1",    $
+      ylim=param_yrange, xlim=param_xrange, cex=.1, pch=20$
> grid(col="gray", lty="dotted")                 #指定したパラメ$
> obj <- as.logical(abs(M) >= log2(param_FC))#条件を満たす$
> points(A[obj], M[obj], col="magenta", cex=0.1, pch=20)#ob$
> legend("topright", c(paste("DEG(> ", param_FC, "-fold)", $
+        col=c("magenta", "black"), pch=20)#凡例を作成して$
> abline(h=log2(param_FC), col="red")            #M=log2(param_FC)$
> abline(h=-log2(param_FC), col="red")           #M=-log2(param_FC)$
> dev.off()                                      #おまじない
null device
          1
> sum(obj)
[1] 5354
>
```

• 解析 | 発現変動 | 2群間 | 対応なし | empirical Bayes (Smyth 2004)

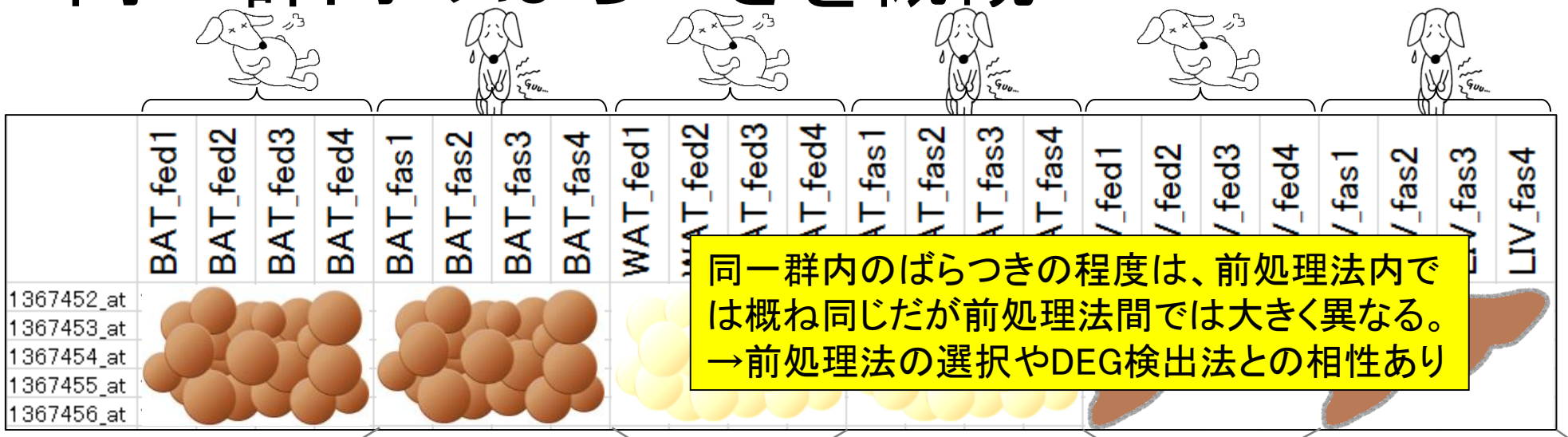
rcode_limma_MAplot_basic2.txtの下の方のコードです

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4
1367456_at				
解析1	G1	G1	G2	G2

Fold-changeによるDEG検出の危険性がよくわかります



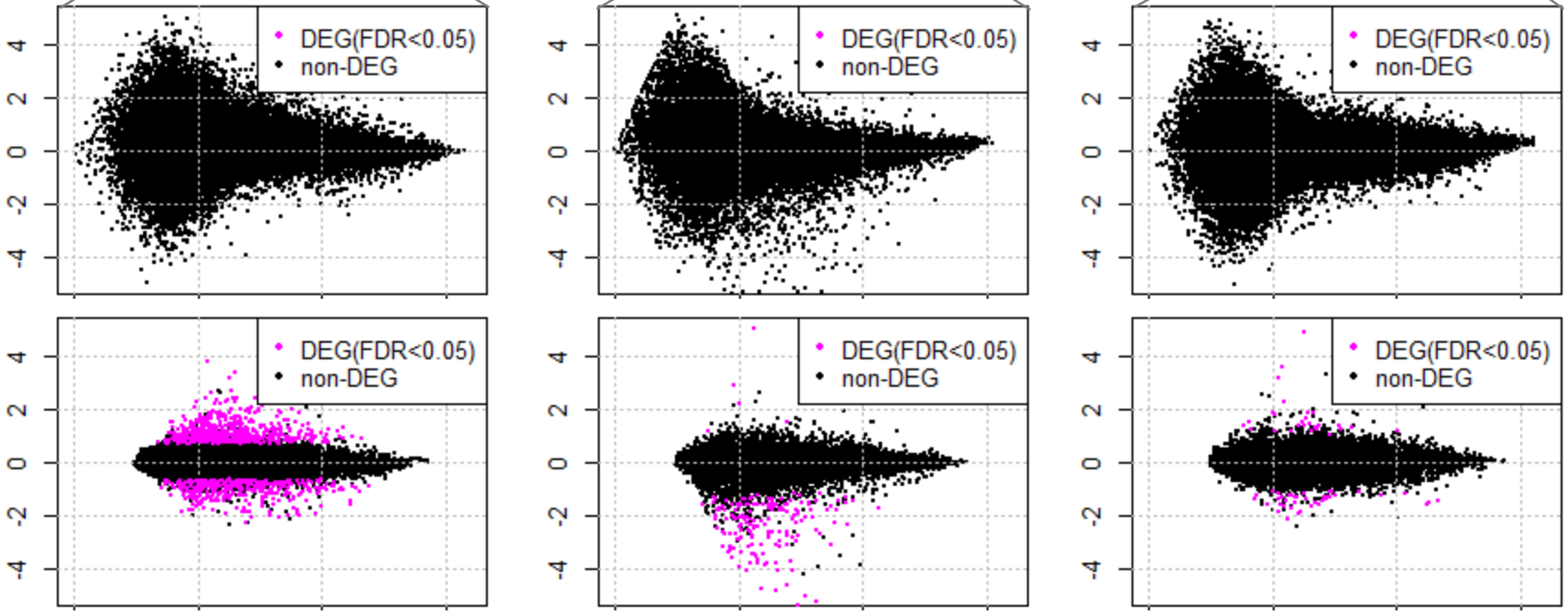
同一群内のばらつきを概観



同一群内のばらつきの程度は、前処理法内では概ね同じだが前処理法間では大きく異なる。
→前処理法の選択やDEG検出法との相性あり

MAS5データ

RMAデータ



まとめ

- 2群間比較：発現変動遺伝子（DEG）検出
  - パターンマッチング法（相関係数の利用）
    - コードの中身をおさらい、apply関数の基本的な利用法など
  - 多重比較問題とFalse Discovery Rate (FDR)
    - 正規分布乱数由来のDEGが存在しないデータでStudent's t-test
    - 10% DEGが存在する正規乱数でデータ（10,000個中1,000個がDEG）でStudent's t-test
  - 発現変動解析用Rパッケージの利用（§4.2.1, p167-）
    - limmaパッケージ (Smyth GK, *SAGMB*, 2004)
    - 関数の利用法
    - IBMT法 (Sartor et al., *BMC Bioinformatics*, 2006)
  - 描画（M-A plot）
    - 作成法
    - 同一群内のばらつき（前処理法間の違い）