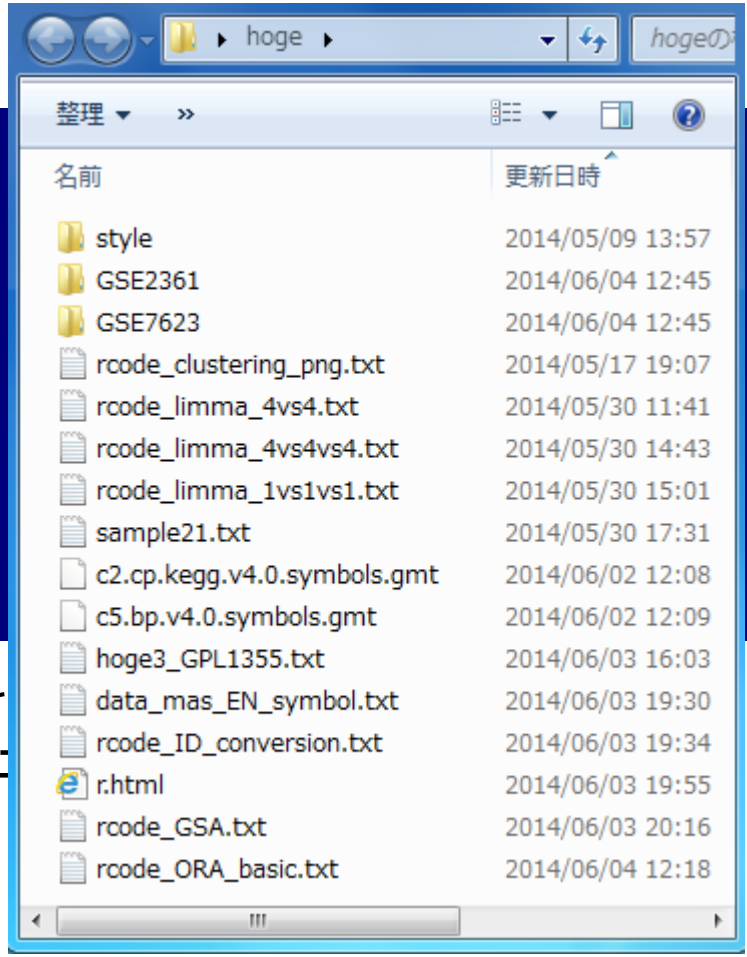
講義室後ろにあるUSBメモリ中のhogeフォルダをデスクトップにコピーしておいてください。

コード内のコピーは
CTRL + ALT + 左クリック

機能ゲノム学 第4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
門田幸二
kadota@iu.a.u-tokyo.ac.jp



前回(5/28)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。

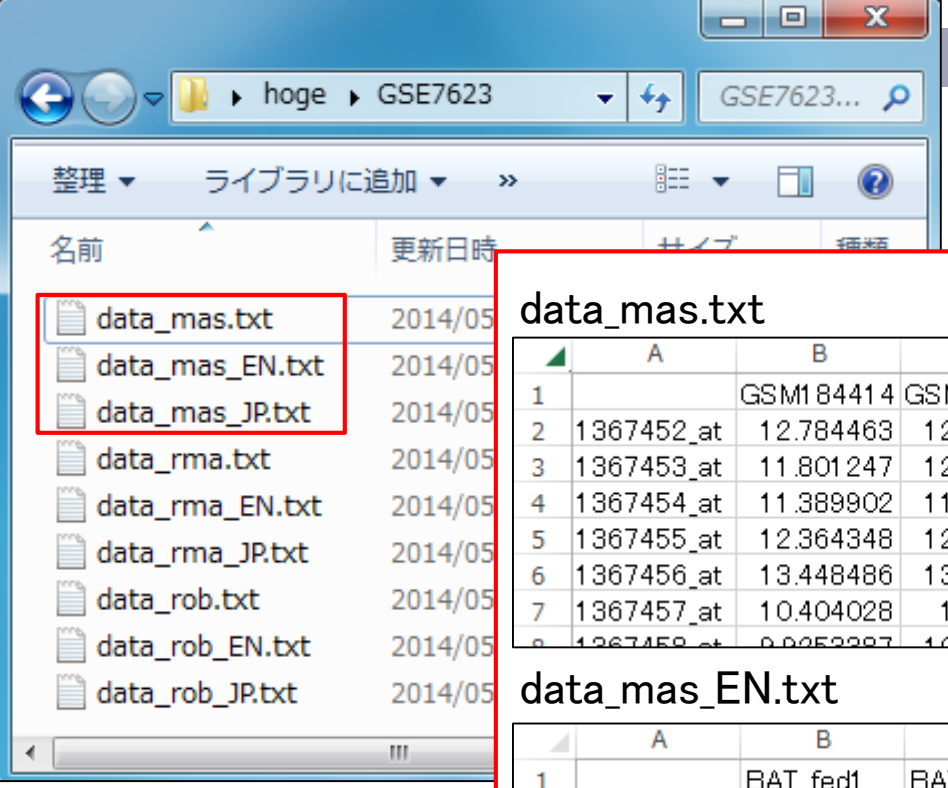
Contents（第4回）

- **デザイン行列の意味を理解（教科書p173-182）**
  - **limmaパッケージを用いた2群間比較のおさらい**
  - limmaパッケージを用いた3群間比較（複製あり）
- 複製なし多群間比較（教科書p182-188）
  - limmaパッケージを用いた3群間比較（複製なし）
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- 機能解析（遺伝子セット解析）
  - 基本的な考え方
  - 前処理
    - MSigDBからの遺伝子セット情報（GMT形式ファイル）取得
    - ID変換（probe ID → gene symbol）
  - GSAパッケージを用いたパスウェイ解析
  - その他
- 分類

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	12.439
3	1367453_at	11.801247	12.152935	11.942227	11.968477	11.845375	11.681727	12.078672	12.048
4	1367454_at	11.389902	11.160757	11.145987	11.212088	11.540652	11.308877	11.49885	11.402
5	1367455_at	12.364348	12.529744	12.432574	12.604011	12.441991	12.249935	12.281827	12.190
6	1367456_at	13.448486	13.543046	13.552794	13.629799	13.36913	13.244278	13.424371	13.329
7	1367457_at	10.404028	10.69632	10.475078	10.45579	10.141921	10.290666	10.146529	10.26
8	1367458_at	9.9253387	10.244544	9.9720008	9.9576072	8.702884	9.3578792	9.2134367	9.4998

data_mas_EN.txt

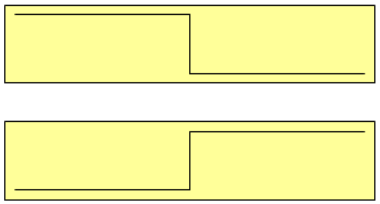
	A	B	C	D	E	F	G	H	I
1		BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas
2	1367452_at	12.784463	12.447082	12.805908	12.304718	12.589425	12.607532	11.815378	12.439
3	1367453_at	11.801247	12.152935	11.942227	11.968477	11.845375	11.681727	12.078672	12.048
4	1367454_at	11.389902	11.160757	11.145987	11.212088	11.540652	11.308877	11.49885	11.402
5	1367455_at	12.364348	12.529744	12.432574	12.604011	12.441991	12.249935	12.281827	12.190
6	1367456_at	13.448486	13.543046	13.552794	13.629799	13.36913	13.244278	13.424371	13.329
7	1367457_at	10.404028	10.69632	10.475078	10.45579	10.141921	10.290666	10.146529	10.26
8	1367458_at	9.9253387	10.244544	9.9720008	9.9576072	8.702884	9.3578792	9.2134367	9.4998

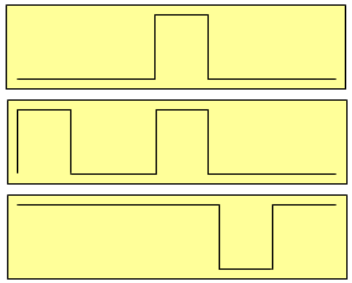
data_mas_JP.txt

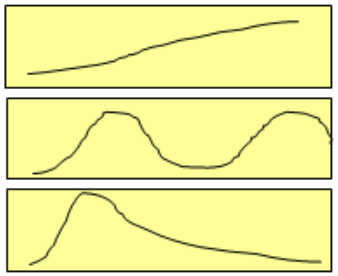
	A	B	C	D	E	F	G
1		褐色脂肪_満腹1	褐色脂肪_満腹2	褐色脂肪_満腹3	褐色脂肪_満腹4	褐色脂肪_空腹1	褐色脂肪_空腹
2	1367452_at	12.7844634	12.44708219	12.80590758	12.30471769	12.58942538	12.6075319
3	1367453_at	11.80124704	12.15293493	11.94222741	11.96847729	11.84537542	11.6817274
4	1367454_at	11.38990178	11.16075717	11.14598707	11.21208786	11.54065185	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	9.9053387	10.244454	9.9720008	9.9576071	8.702884	9.3578791

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）

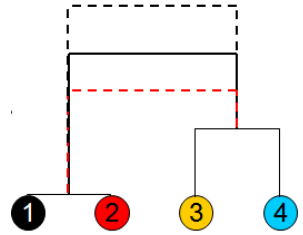
	S1	S2	S3	S4	…
gene 1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	…
gene 2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	…
…	…	…	…	…	…
gene i	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	
…	…	…	…	…	…
gene n	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$x_{n,4}$	…

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 i	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	
…	…	…	…	…	…
gene n	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$	$x_{n,4}$	…

対数変換後のデータを用いて2群、3群、多群間比較

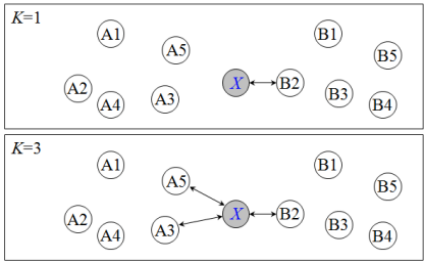
クラスタリング



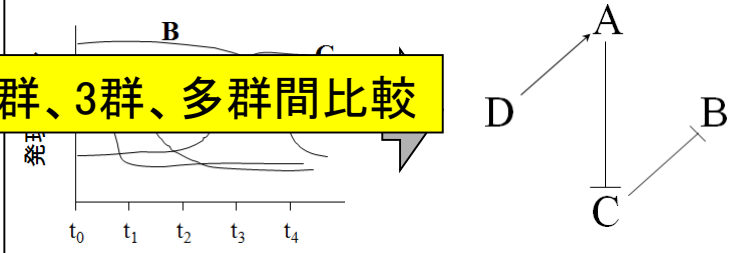
機能解析

・Gene Ontology（GO）
・パスウェイ解析

分類（診断）

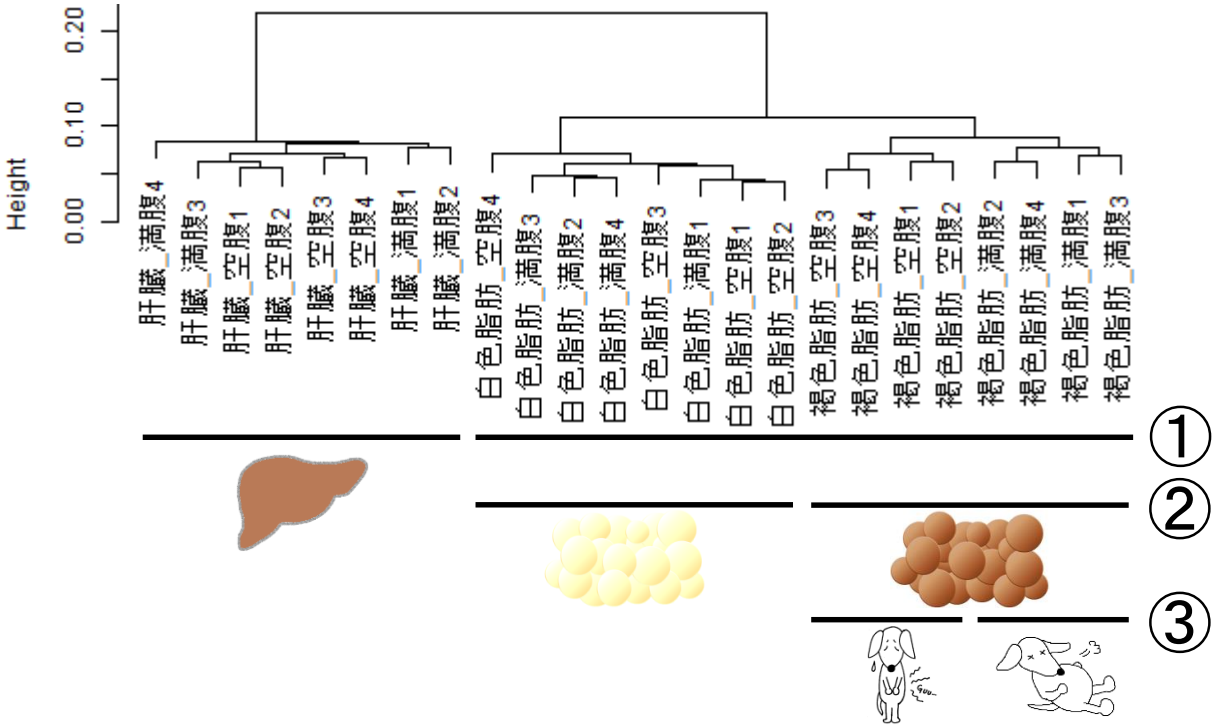


遺伝子ネットワーク推定



発現変動解析用Rパッケージの利用

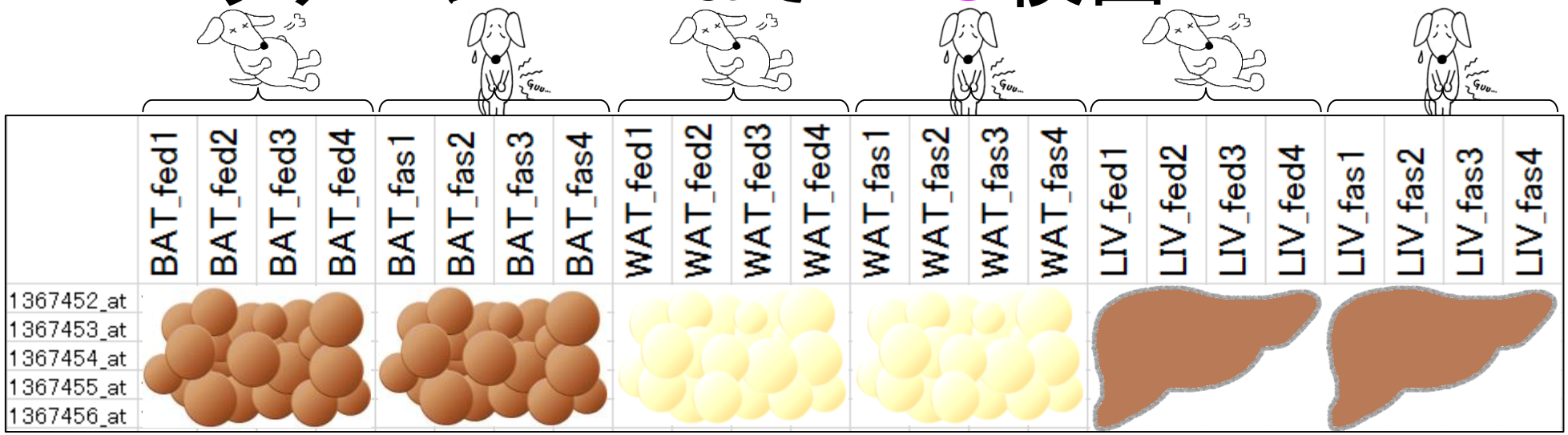
- Nakai et al., *Biosci Biotechnol Biochem.*, **72**: 139-148, 2008
  - **GSE7623**、GPL1355 (Affymetrix Rat Genome 230 2.0 Array)、31,099 probesets
  - **ラット24サンプル**：Brown adipose tissue (褐色脂肪組織; BAT)8サンプル、White adipose tissue (白色脂肪組織; WAT)8サンプル、Liver (肝臓; LIV)8サンプル
    - BAT 8サンプル：通常(BAT_fed) 4サンプル　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

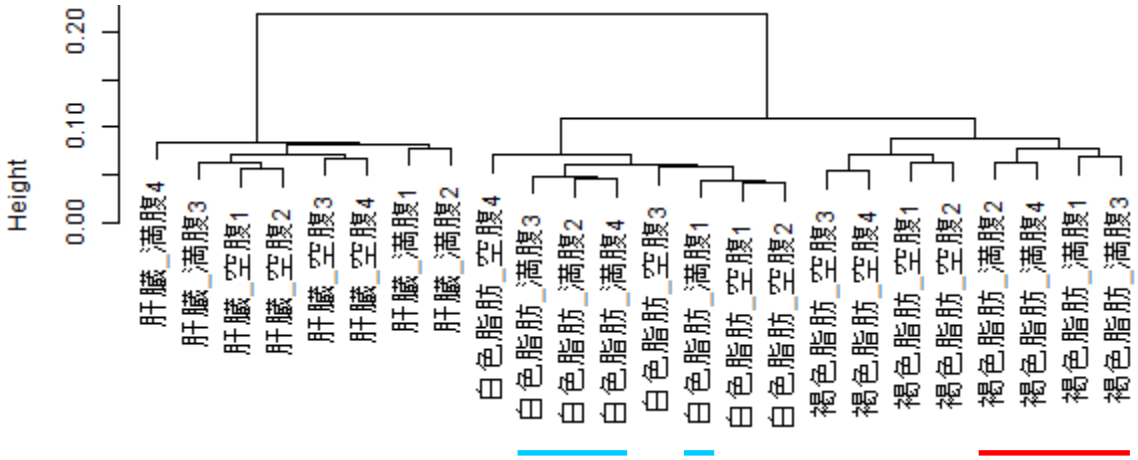


GSE7623データを用い、様々な2群間比較を行い、クラスタリング結果とDEG検出結果の関連をみてみよう

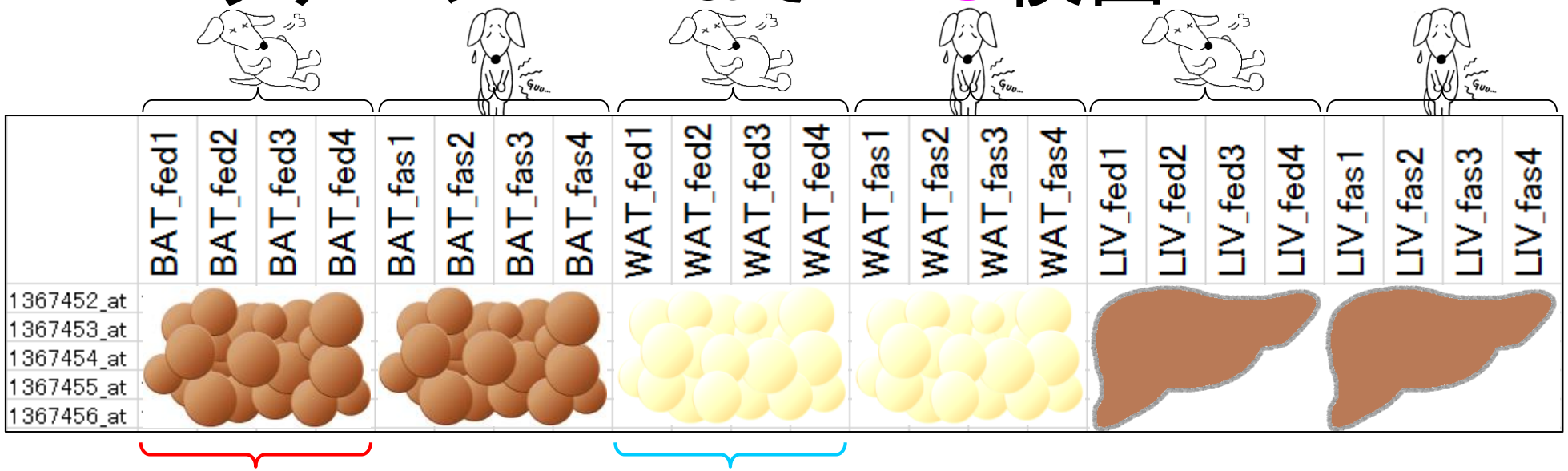
rcode_clustering_png.txtの実行結果。
①肝臓と脂肪間で大きく二つのクラスターに分かれている。
②脂肪の中でも白色脂肪と褐色脂肪に分かれている。
③褐色脂肪は空腹(24時間絶食)と満腹(通常)できれいに分かれている。

RパッケージlimmaでDEG検出





RパッケージlimmaでDEG検出



G1群　　G2群

rcode_limma_4vs4.txt

```
##################################################↓
###  4 BAT_fed samples vs. 4 WAT_fed samples  ###↓
##################################################↓
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 <- 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_posi <- c(1:4, 9:12)                 #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↓
```

rcode_limma_4vs4.txt

```
#############################################################
###  4 BAT_fed samples vs. 4 WAT_fed samples  ###
#############################################################
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"
param_G1 <- 4
param_G2 <- 4
param_posi <- c(1:4, 9:12)
param_FDR <- 0.05
param_fig <- c(400, 380)

#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の
data <- read.table(in_f, header=TRUE,
data.cl <- c(rep(1, param_G1), rep(2,
data <- data[,param_posi]
colnames(data)

#本番
#design <- model.matrix(~ as.factor(d
design <- model.matrix(~data.cl)
fit <- lmFit(data, design)
out <- eBayes(fit)
p.value <- out$p.value[,ncol(design)]
q.value <- p.adjust(p.value, method="
ranking <- rank(p.value)
```

R Console

```
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/GSE7623"
> in_f <- "data_mas_EN.txt"                    #入力ファイ$
> out_f1 <- "hoge1.txt"                        #出力ファイ$
> out_f2 <- "hoge1.png"                        #出力ファイ$
> param_G1 <- 4                                #G1群のサン$
> param_G2 <- 4                                #G2群のサン$
> param_posi <- c(1:4, 9:12)                   #元の発現行$
> param_FDR <- 0.05                            #DEG検出時$
> param_fig <- c(400, 380)                     #ファイル出$
>
> #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、そして$
> data <- read.table(in_f, header=TRUE, row.names=1,$
> data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G$
> data <- data[,param_posi]                    #サブセット$
> colnames(data)                               #サブセット$
[1] "BAT_fed1" "BAT_fed2" "BAT_fed3" "BAT_fed4"
[5] "WAT_fed1" "WAT_fed2" "WAT_fed3" "WAT_fed4"
> param_posi
[1]  1  2  3  4  9 10 11 12
>
```

解析したいサブセットに正しくできていることがわかります

rcode_limma_4vs4.txt

```
↓
#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 < 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)#行ごとにG1群の平均を計算した結
mean_G2 <- apply(as.matrix(data[,data.
M <- mean_G2 - mean_G1
A <- (mean_G1 + mean_G2)/2
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, M,
write.table(tmp, out_f1, sep="¥t", app
```

R Console

```
> 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$p.value[,ncol(design)]     #p値をp.val$
> design
  (Intercept) data.cl
1           1       1
2           1       1
3           1       1
4           1       1
5           1       2
6           1       2
7           1       2
8           1       2
attr(,"assign")
[1] 0 1
>
```

designオブジェクトが(実験)デザイン行列です。この行列の2列目がG1群とG2群がどれに相当するかを表すクラスラベル情報であることもわかります。

```
#本番↓
#design <- model.matrix(~ as.factor(data.cl))
design <- model.matrix(~data.cl)
fit <- lmFit(data, design)
out <- eBayes(fit)
p.value <- out$p.value[,ncol(design)]
```

R Console

```
> design
  (Intercept) data.cl
1           1       1
2           1       1
3           1       1
4           1       1
5           1       2
6           1       2
7           1       2
8           1       2
attr(,"assign")
[1] 0 1
> dim(design)
[1] 8 2
> nrow(design)
[1] 8
> ncol(design)
[1] 2
> design[1,]
(Intercept)     data.cl
          1           1
> design[,2]
1 2 3 4 5 6 7 8
1 1 1 1 2 2 2 2
> design[,ncol(design)]
1 2 3 4 5 6 7 8
1 1 1 1 2 2 2 2
>
```

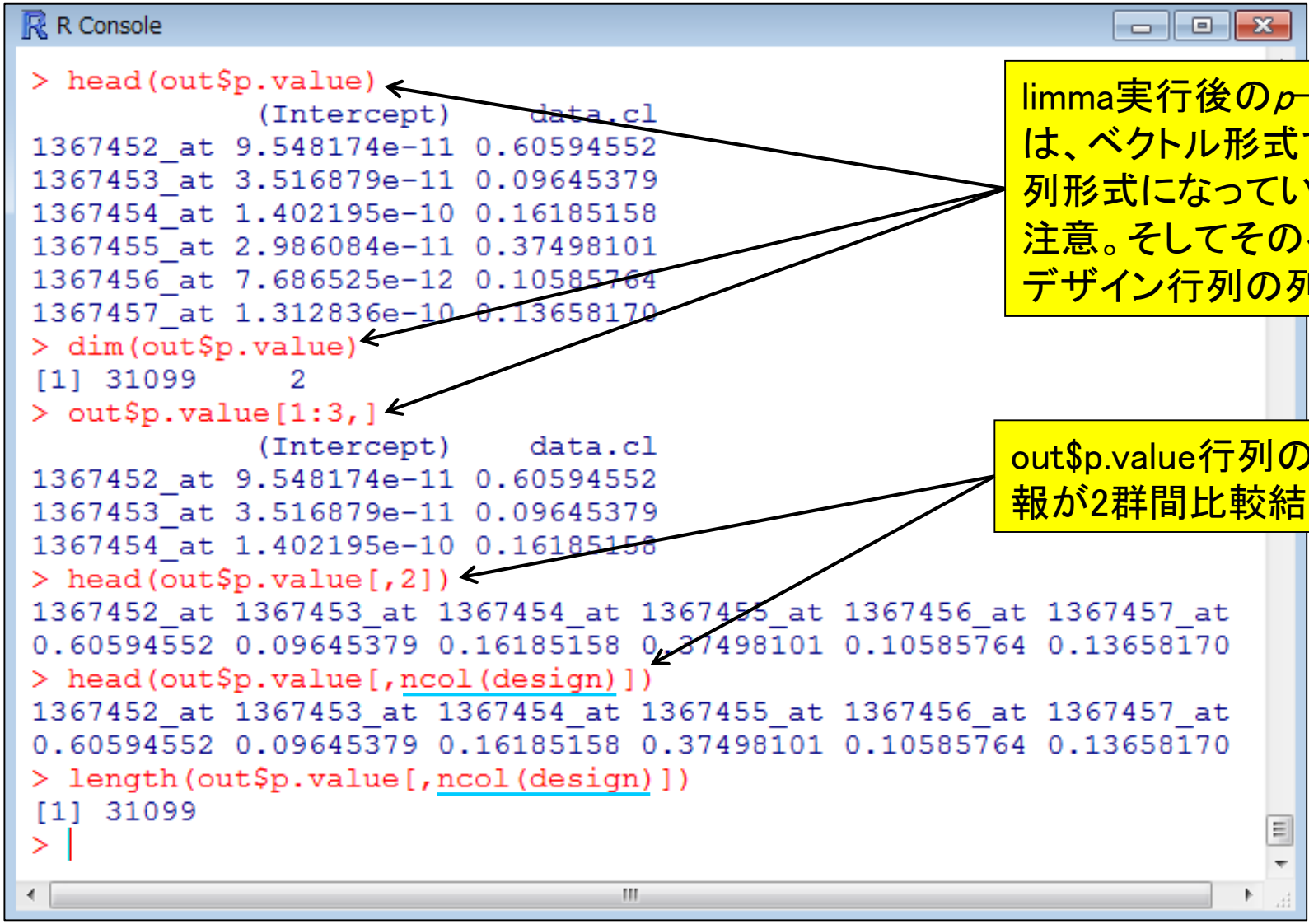
dim関数で行数と列数を表示

nrow関数で行数を表示

ncol関数で列数を表示

行列の要素抽出の基本は[行, 列]

```
#本番↓
#design <- model.matrix(~ as.factor(data.cl))
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値を
```



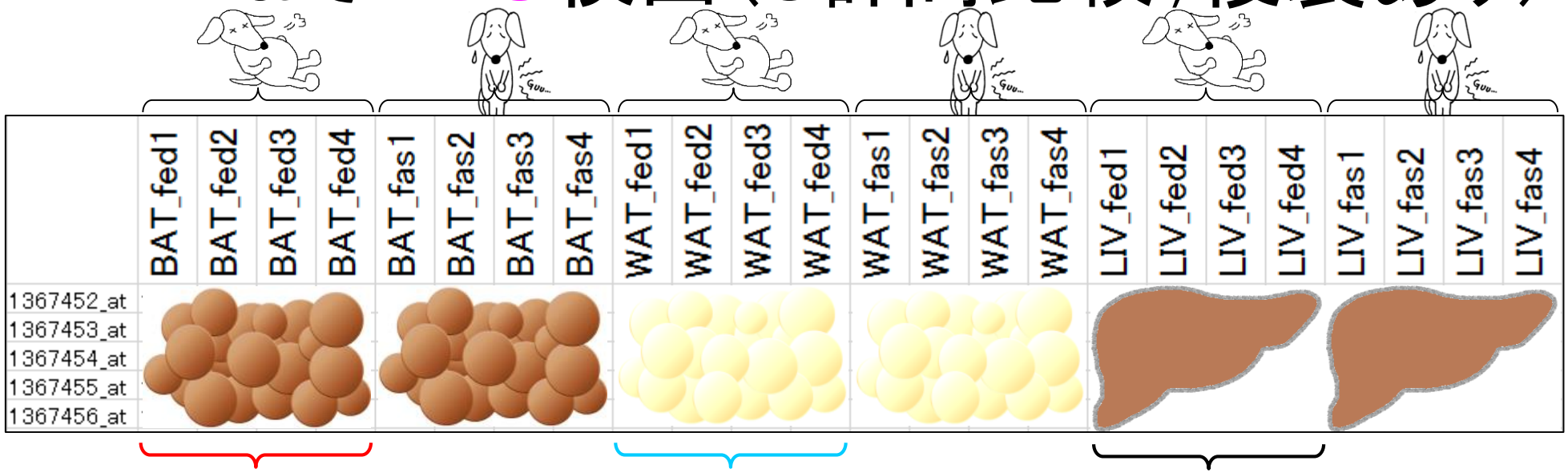
limma実行後の*p*-value情報は、ベクトル形式ではなく行列形式になっていることに注意。そしてその列数は、デザイン行列の列数と同じ。

out$p.value行列の2列目の情報が2群間比較結果に相当

Contents（第4回）

- デザイン行列の意味を理解（教科書p173-182）
  - limmaパッケージを用いた2群間比較のおさらい
  - **limmaパッケージを用いた3群間比較（複製あり）**
- 複製なし多群間比較（教科書p182-188）
  - limmaパッケージを用いた3群間比較（複製なし）
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- 機能解析（遺伝子セット解析）
  - 基本的な考え方
  - 前処理
    - MSigDBからの遺伝子セット情報（GMT形式ファイル）取得
    - ID変換（probe ID → gene symbol）
  - GSAパッケージを用いたパスウェイ解析
  - その他
- 分類

limmaでDEG検出(3群間比較;複製あり)



rcode_limma_4vs4vs4.txt

```
##################################################################################↓
###  4 BAT_fed samples vs. 4 WAT_fed samples vs. 4 LIV_fed samples ###↓
##################################################################################↓
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 <- 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_G3 <- 4                              #G2群のサンプル数を指定↓
param_posi <- c(1:4, 9:12, 17:20)          #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↓
↓
#必要なパッケージをロード↓
library(limma)                             #パッケージの読み込み↓
```

```
##########################################################################
###  4 BAT_fed samples vs. 4 WAT_fed samples vs. 4 LIV_fed samples ###
##########################################################################
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 <- 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_G3 <- 4                            #G2群のサンプル数を指定
param_posi <- c(1:4, 9:12, 17:20)        #元の発
param_FDR <- 0.05                        #DEG検出

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

#入力ファイルの読み込みとラベル情報の作成、そ
data <- read.table(in_f, header=TRUE, row.name
data.cl <- c(rep("G1", param_G1), rep("G2", pa
data <- data[,param_posi]                #サブセ
colnames(data)                           #サブセ
```

R Console

```
> in_f <- "data_mas_EN.txt"                #入力ファイル$
> out_f1 <- "hoge1.txt"                    #出力ファイル$
> param_G1 <- 4                            #G1群のサンプ$
> param_G2 <- 4                            #G2群のサンプ$
> param_G3 <- 4                            #G2群のサンプ$
> param_posi <- c(1:4, 9:12, 17:20)        #元の発現行列$
> param_FDR <- 0.05                        #DEG検出時のfa$
>
> #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の作成、そしてサブ$
> data <- read.table(in_f, header=TRUE, row.names=1, se$
> data.cl <- c(rep("G1", param_G1), rep("G2", param_G2)$
> data <- data[,param_posi]                #サブセットを$
> colnames(data)                           #サブセット抽$
 [1] "BAT_fed1" "BAT_fed2" "BAT_fed3" "BAT_fed4"
 [5] "WAT_fed1" "WAT_fed2" "WAT_fed3" "WAT_fed4"
 [9] "LIV_fed1" "LIV_fed2" "LIV_fed3" "LIV_fed4"
> data.cl
 [1] "G1" "G1" "G1" "G1" "G2" "G2" "G2" "G2" "G3" "G3"
[11] "G3" "G3"
>
```

解析したいサブセットに正しくできています

```
#本番
#design <- model.matrix(~ 0 + 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(~ 0 + data.cl)  #デザイン行列を作成した結果をdesignに格納
colnames(design) <- levels(as.factor(data.cl))#デザイン行列の列名を付与
fit <- lmFit(data, design)                #モデル構築(ばらつきの程度を見積もっている)
contrast <- makeContrasts(                #比較したい2群の情報を作成
    G1vsG2 = G1 - G2,                     #比較したい2群の情
    G1vsG3 = G1 - G3,                     #比較したい2群の情
    G2vsG3 = G2 - G3,                     #比較したい2群の情
    levels = design)                      #比較したい2群の情
fit2 <- contrasts.fit(fit, contrast)      #モデル構築
out <- eBayes(fit2)                       #検定(経験ベイズ)
p.value <- out$p.value                    #p値をp.valueに格納
q.value <- apply(p.value, MARGIN=2, p.adjust, method="BH"
ranking <- apply(p.value, MARGIN=2, rank)#p.valueでランキ
```

R Console

```
> design <- model.matrix(~ 0 + data.cl)
> design
   data.clG1 data.clG2 data.clG3
1          1         0         0
2          1         0         0
3          1         0         0
4          1         0         0
5          0         1         0
6          0         1         0
7          0         1         0
8          0         1         0
9          0         0         1
10         0         0         1
11         0         0         1
12         0         0         1
attr(,"assign")
[1] 1 1 1
attr(,"contrasts")
attr(,"contrasts")$data.cl
[1] "contr.treatment"

>
```

```
#本番
#design <- model.matrix(~ 0 + 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(~ 0 + data.cl)  #デザイン行列を作成した結果をdesignに格納
colnames(design) <- levels(as.factor(data.cl))#デザイン行列の列名を付与
fit <- lmFit(data, design)                    #モデル構築(ばらつきの程度を見積もっている)
contrast <- makeContrasts(                    #比較
    G1vsG2 = G1 - G2,                         #比較
    G1vsG3 = G1 - G3,                         #比較
    G2vsG3 = G2 - G3,                         #比較
    levels = design)                          #比較
fit2 <- contrasts.fit(fit, contrast)          #モデ
out <- eBayes(fit2)                           #検定
p.value <- out$p.value                        #p値
q.value <- apply(p.value, MARGIN=2, p.adjust
ranking <- apply(p.value, MARGIN=2, rank)#p.
```

R Console

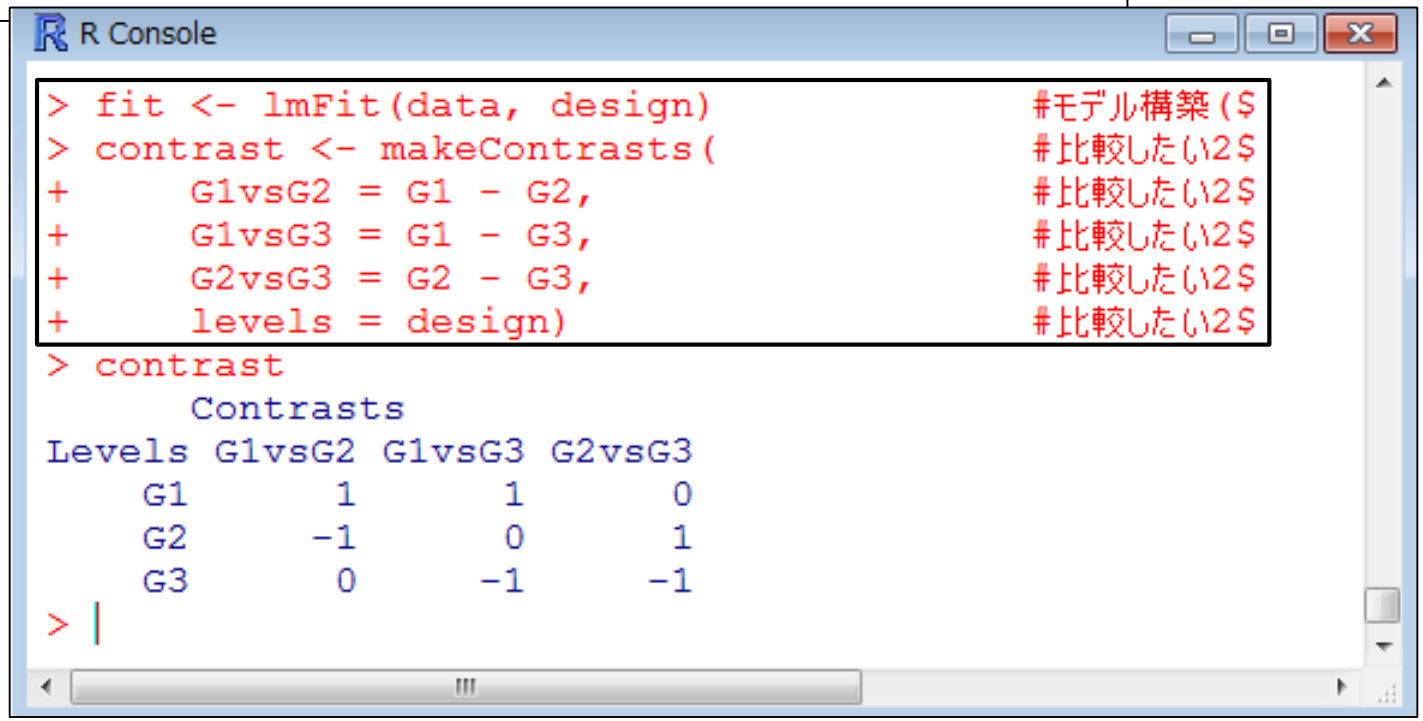
```
> as.factor(data.cl)
 [1] G1 G1 G1 G1 G2 G2 G2 G2 G3 G3 G3 G3
Levels: G1 G2 G3
> levels(as.factor(data.cl))
[1] "G1" "G2" "G3"
> colnames(design) <- levels(as.factor(data.cl))
> design
   G1 G2 G3
1   1  0  0
2   1  0  0
3   1  0  0
4   1  0  0
5   0  1  0
6   0  1  0
7   0  1  0
8   0  1  0
9   0  0  1
10  0  0  1
11  0  0  1
12  0  0  1
attr(,"assign")
[1] 1 1 1
attr(,"contrasts")
attr(,"contrasts")$data.cl
[1] "contr.treatment"

>
```

デザイン行列designの列名を
変更して取扱いやすくしている

```
#本番
#design <- model.matrix(~ 0 + 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(~ 0 + data.cl)  #デザイン行列を作成した結果をdesignに格納
colnames(design) <- levels(as.factor(data.cl))#デザイン行列の列名を付与
fit <- lmFit(data, design)                    #モデル構築(ばらつきの程度を見積もっている)
contrast <- makeContrasts(                    #比較したい2群の情報を作成
    G1vsG2 = G1 - G2,                         #比較したい2群の情報を作成
    G1vsG3 = G1 - G3,                         #比較したい2群の情報を作成
    G2vsG3 = G2 - G3,                         #比較したい2群の情報を作成
    levels = design)                          #比較したい2群の情報を作成
fit2 <- contrasts.fit(fit, contrast)          #モデル構築
out <- eBayes(fit2)                           #検定(経験ベイズ)
p.value <- out$p.value                        #p値をp.valueに格納
q.value <- apply(p.value, MARGIN=2, p.adjust, method="BH")#q値をq.valueに格納
ranking <- apply(p.value, MARGIN=2, rank)#p.valueでランキングした結果をrankingに格納
```

デザイン行列の列名を変更して取扱いやすくしておかないと、この部分での指定時にややこしいことになる。ここでは3種類の2群間比較を行うようにしている。



```
#本番
#design <- model.matrix(~ 0 + 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(~ 0 + data.cl)  #デザイン行列を作成した結果をdesignに格納
colnames(design) <- levels(as.factor(data.cl))#デザイン行列の列名を付与
fit <- lmFit(data, design)             #モデル構築(ばらつきの程度を見積もっている)
contrast <- makeContrasts(             #比較したい2群の情報を作成
    G1vsG2 = G1 - G2,                  #比較したい2群の情報を作成
    G1vsG3 = G1 - G3,                  #比較したい2群の情報を作成
    G2vsG3 = G2 - G3,                  #比較したい2群の情報を作成
    levels = design)                   #比較したい2群の情報を作成
fit2 <- contrasts.fit(fit, contrast)   #モデル構築
out <- eBayes(fit2)                    #検定(経験ベイズ)
p.value <- out$p.value                 #p値をp.valueに格納
q.value <- apply(p.value, MARGIN=2, p.adjust, method="BH")#q値をq.valueに格納
ranking <- apply(p.value, MARGIN=2, rank)#p.valueでランキングした結果をrankingに格納
```

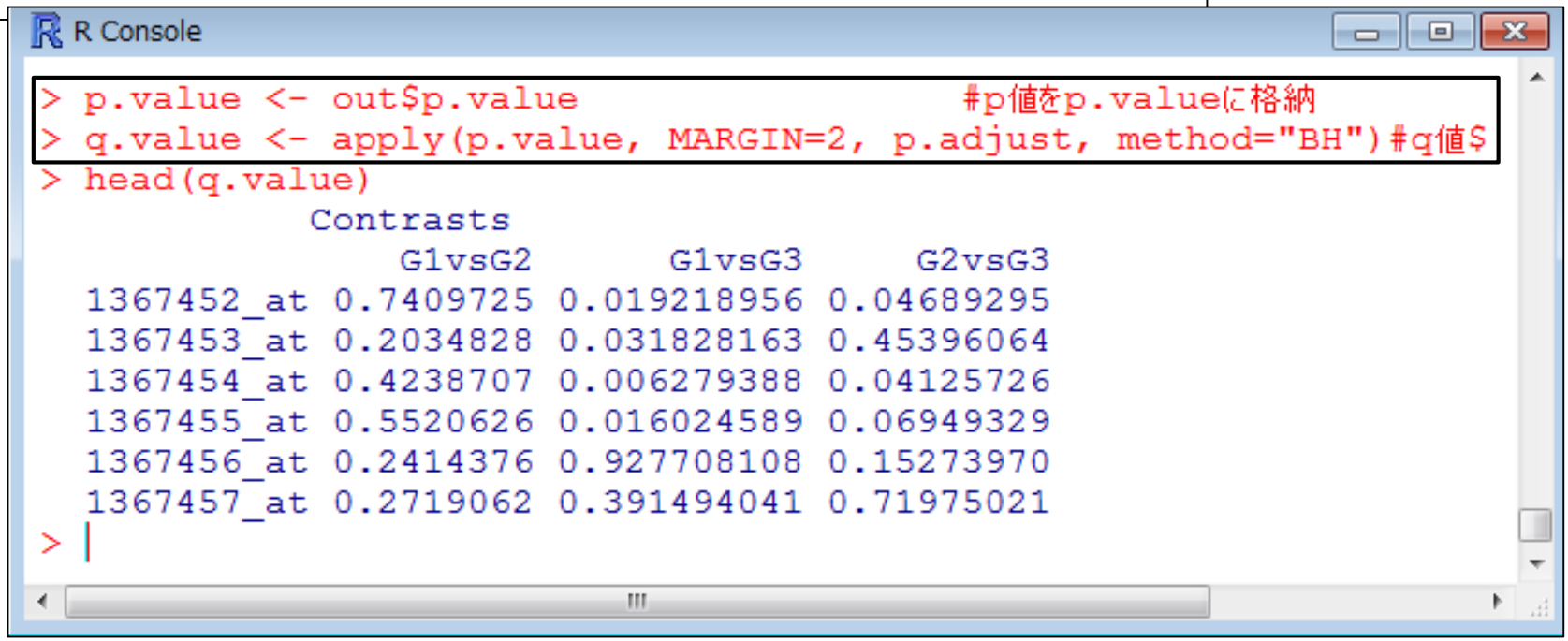
3種類の2群間比較を行うようにしたコントラスト行列contrastを入力としているので、DEG検出結果として31,099行×3列からなるp-value行列が得られることになる。

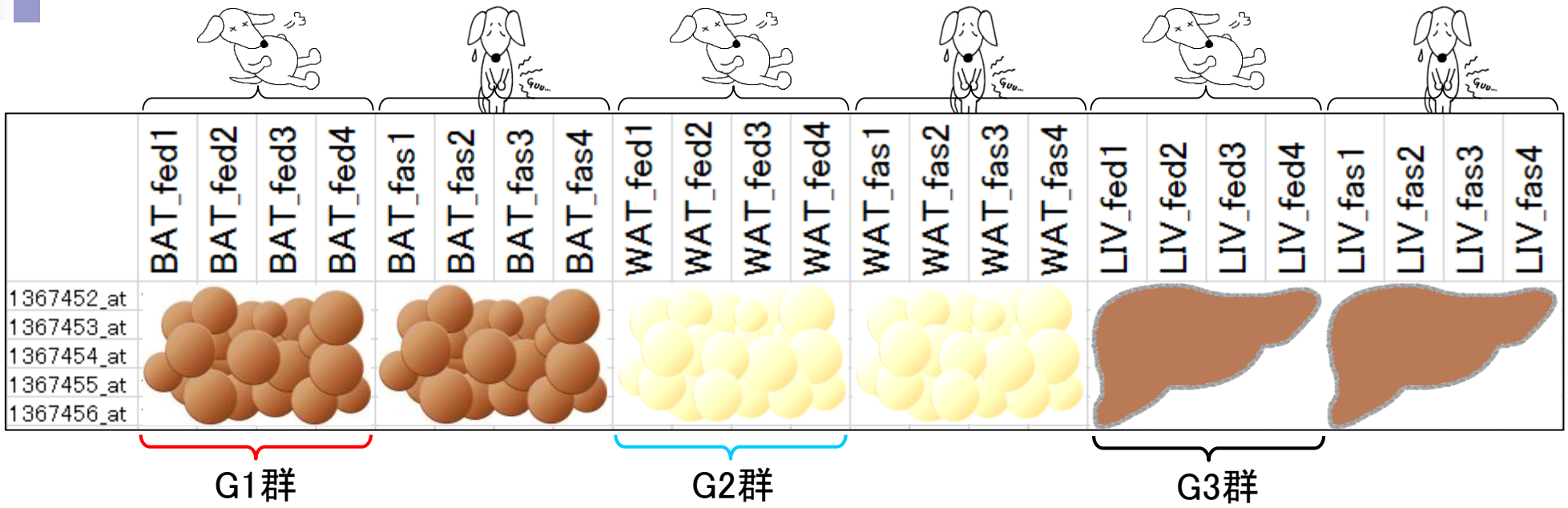
R Console

```
> fit2 <- contrasts.fit(fit, contrast)    #モデル構築
> out <- eBayes(fit2)                     #検定(経験ベ$
> head(out$p.value)
             Contrasts
                  G1vsG2       G1vsG3     G2vsG3
  1367452_at 0.57066249 0.005780086 0.01612904
  1367453_at 0.07035758 0.010713438 0.30569612
  1367454_at 0.22337752 0.001491958 0.01373208
  1367455_at 0.34340818 0.004630791 0.02635020
  1367456_at 0.09121355 0.887708741 0.07149528
  1367457_at 0.11033976 0.258495085 0.59513570
> dim(out$p.value)
[1] 31099     3
>
```

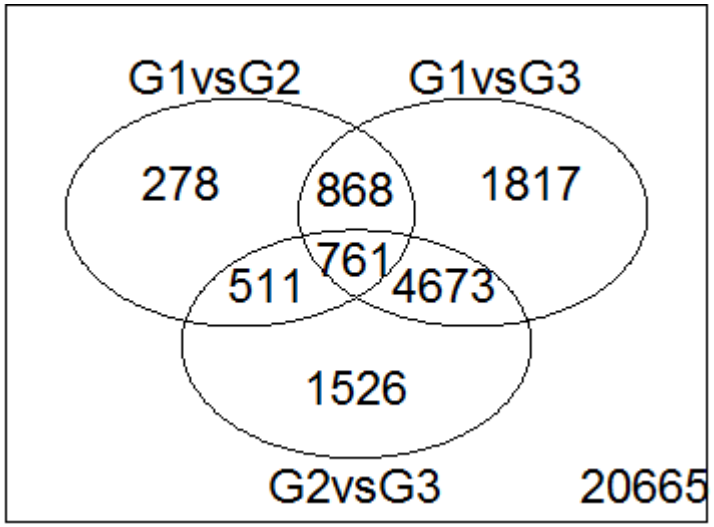
```
#本番
#design <- model.matrix(~ 0 + as.factor(data.cl))#デザイン行列を作成した結果をdesignに格納
design <- model.matrix(~ 0 + data.cl)  #デザイン行列を作成した結果をdesignに格納
colnames(design) <- levels(as.factor(data.cl))#デザイン行列の列名を付与
fit <- lmFit(data, design)             #モデル構築(ばらつきの程度を見積もっている)
contrast <- makeContrasts(             #比較したい2群の情報を作成
    G1vsG2 = G1 - G2,                  #比較したい2群の情報を作成
    G1vsG3 = G1 - G3,                  #比較したい2群の情報を作成
    G2vsG3 = G2 - G3,                  #比較したい2群の情報を作成
    levels = design)                   #比較したい2群の情報を作成
fit2 <- contrasts.fit(fit, contrast)   #モデル構築
out <- eBayes(fit2)                    #検定(経験ベイズ)
p.value <- out$p.value                 #p値をp.valueに格納
q.value <- apply(p.value, MARGIN=2, p.adjust, method="BH")#q値をq.valueに格納
ranking <- apply(p.value, MARGIN=2, rank)#p.valueでランキングした結果をrankingに格納
```

apply関数を用いて列ごと(MARGIN=2)に*q*-valueを計算している





G1vsG2のDEG数が他に
比べて少ないので妥当



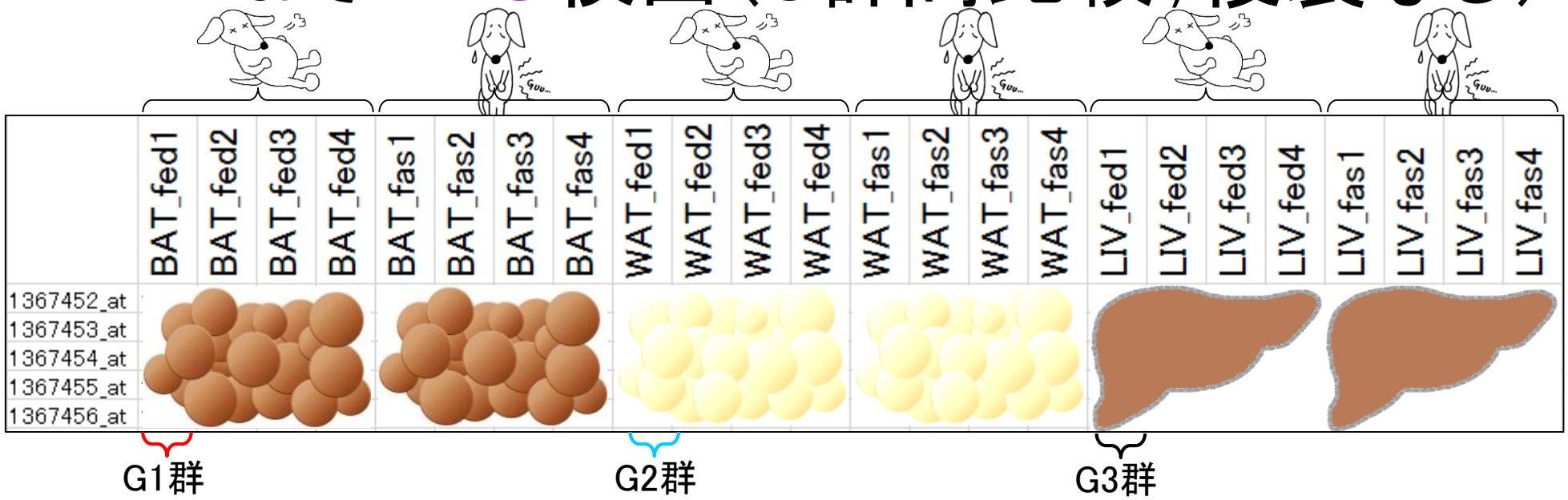
R Console

```
> sum(q.value[,1] < 0.01)
[1] 2418
> sum(q.value[,2] < 0.01)
[1] 8119
> sum(q.value[,3] < 0.01)
[1] 7471
> vennDiagram(decideTests(out, adjust.method="BH", p.value=0.01))
>
```

Contents（第4回）

- デザイン行列の意味を理解（教科書p173-182）
  - limmaパッケージを用いた2群間比較のおさらい
  - limmaパッケージを用いた3群間比較（複製あり）
- **複製なし多群間比較（教科書p182-188）**
  - **limmaパッケージを用いた3群間比較（複製なし）**
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- 機能解析（遺伝子セット解析）
  - 基本的な考え方
  - 前処理
    - MSigDBからの遺伝子セット情報（GMT形式ファイル）取得
    - ID変換（probe ID → gene symbol）
  - GSAパッケージを用いたパスウェイ解析
  - その他
- 分類

limmaでDEG検出(3群間比較;複製なし)



rcode_limma_1vs1vs1.txt

```
##########################################################################↓
###  1 BAT_fed sample vs. 1 WAT_fed sample vs. 1 LIV_fed sample ###↓
##########################################################################↓
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↓
param_G1 <- 1                      #G1群のサンプル数を指定↓
param_G2 <- 1                      #
param_G3 <- 1                      #
param_posi <- c(1, 9, 17)          #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↓
```

(biological) replicatesがないデータの場合

rcode_limma_1vs1vs1.txt

```
##########################################################################
###  1 BAT_fed sample vs. 1 WAT_fed sample vs. 1 LIV_fed sample ###
##########################################################################
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
param_G1 <- 1                              #G1群のサンプル数を指定
param_G2 <- 1                              #G2群のサンプル数を指定
param_G3 <- 1                              #
param_posi <- c(1, 9, 17)                  #
param_FDR <- 0.05                          #

#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の作成
data <- read.table(in_f, header=TRUE, ro
data.cl <- c(rep("G1", param_G1), rep("G
data <- data[,param_posi]                  #
colnames(data)                             #

#本番
#design <- model.matrix(~ 0 + as.factor(
design <- model.matrix(~ 0 + data.cl)      #
colnames(design) <- levels(as.factor(dat
fit <- lmFit(data, design)                 #
contrast <- makeContrasts(                 #
    G1vsG2 = G1 - G2,                      #
    G1vsG3 = G1 - G3,                      #
    G2vsG3 = G2 - G3,                      #
    levels = design)                       #
fit2 <- contrasts.fit(fit, contrast)       #
out <- eBayes(fit2)                        #
p.value <- out$p.value                     #
q.value <- apply(p.value, MARGIN=2, p.ad
ranking <- apply(p.value, MARGIN=2, rank
```

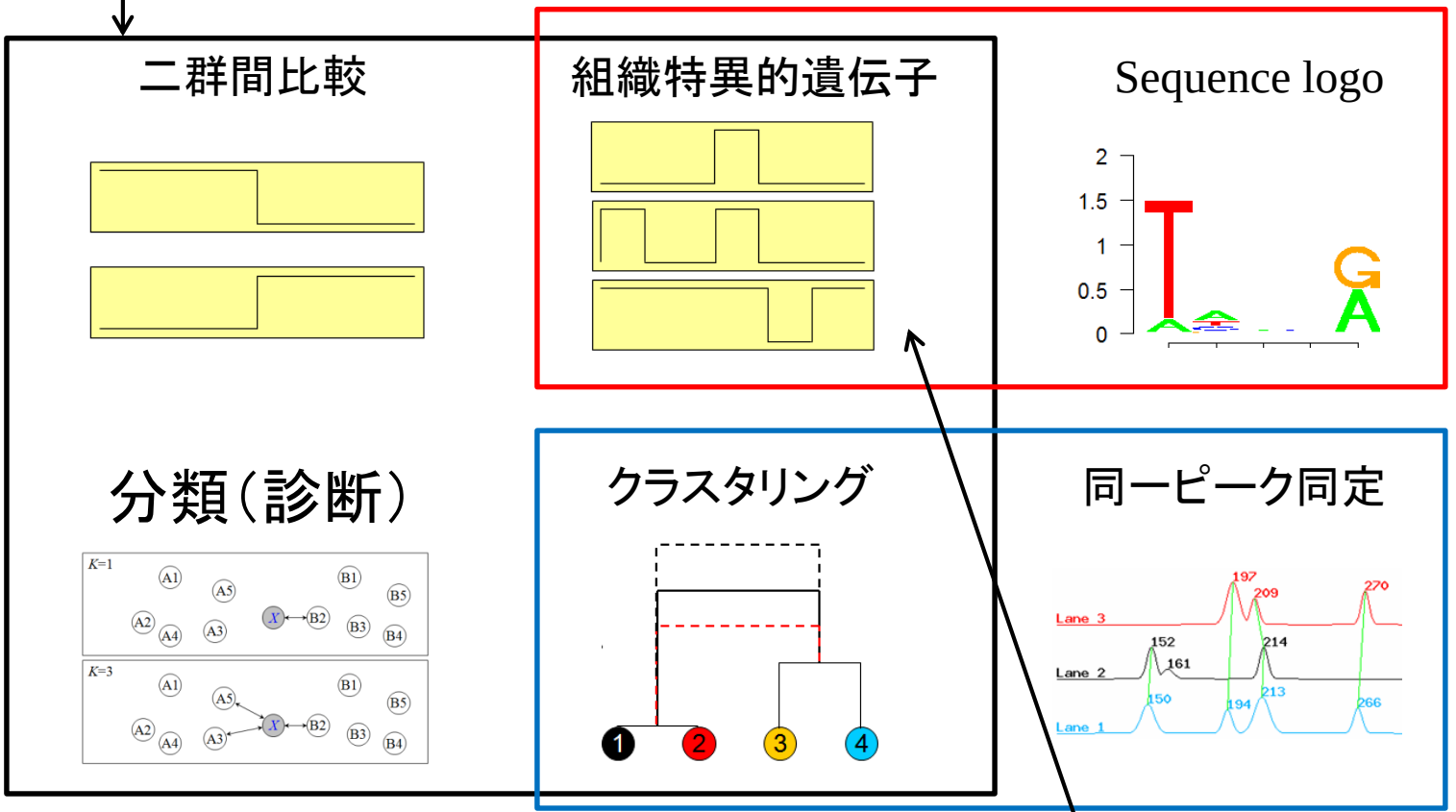
R Console

```
> #本番
> #design <- model.matrix(~ 0 + as.factor(data.cl))#デ$
> design <- model.matrix(~ 0 + data.cl)   #デザイン行列$
> colnames(design) <- levels(as.factor(data.cl))#デザ$
> fit <- lmFit(data, design)                        #モデル構築($
> contrast <- makeContrasts(                        #比較したい2$
+     G1vsG2 = G1 - G2,                             #比較したい2$
+     G1vsG3 = G1 - G3,                             #比較したい2$
+     G2vsG3 = G2 - G3,                             #比較したい2$
+     levels = design)                              #比較したい2$
> fit2 <- contrasts.fit(fit, contrast)    #モデル構築
> out <- eBayes(fit2)                               #検定(経験ベ$
 以下にエラー ebayes(fit = fit, proportion = proportio$
  No residual degrees of freedom in linear model fits
> p.value <- out$p.value                            #p値をp.value$
 エラー:  オブジェクト 'out' がありません
> q.value <- apply(p.value, MARGIN=2, p.adjust, method$
 以下にエラー apply(p.value, MARGIN = 2, p.adjust, met$
   オブジェクト 'p.value' がありません
> ranking <- apply(p.value, M
 以下にエラー apply(p.value, MARGI
   オブジェクト 'p.value' がありません
>
```

(biological) replicatesがないデータの場合は、通常モデル構築ができないのでエラーが出ます

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



エントロピーで組織特異的遺伝子を
ランキングするやり方を紹介します

Sequence logos：計算手順

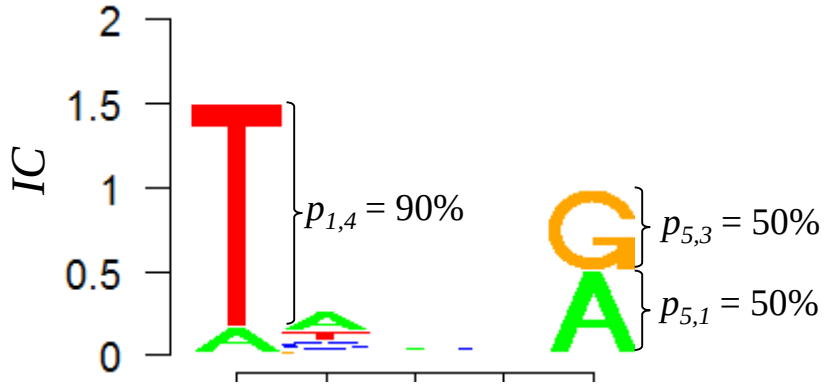
- position iの情報量 $IC_i = \log_2(N) - H(x_i)$

$N = 2$

Sequence logosは、あるポジションに特定の塩基が濃縮されている状態をうまく表すために、エントロピーを内部的に計算している

position i

	1	2	3	4	5	…
配列 1	T	A	C	G	G	…
配列 2	T	A	A	C	G	…
配列 3	T	G	T	A	G	…
配列 4	A	C	T	T	A	…
配列 5	T	T	G	G	A	…
配列 6	T	C	A	A	G	…
配列 7	T	A	C	T	A	…
配列 8	T	T	G	C	A	…
配列 9	T	A	A	C	A	…
配列10	T	A	C	T	G	…



IC	1.53	0.24	0.03	0.03	1.00	…

x_{ij}	1	2	3	4	5	…
Aの数 (j=1)	1	5	3	2	5	…
Cの数 (j=2)	0	2	3	3	0	…
Gの数 (j=3)	0	1	2	2	5	…
Tの数 (j=4)	9	2	2	3	0	…
$\Sigma_j\ x_{ij}$	10	10	10	10	10	

j

p_{ij}	1	2	3	4	5	…
1	0.1	0.5	0.3	0.2	0.5	…
2	0.0	0.2	0.3	0.3	0.0	…
3	0.0	0.1	0.2	0.2	0.5	…
4	0.9	0.2	0.2	0.3	0.0	…
Σ_j	1.0	1.0	1.0	1.0	1.0	

$-p_{ij}\log_2(p_{ij})$	1	2	3	4	5	…
1	0.33	0.50	0.52	0.46	0.50	…
2	0.00	0.46	0.52	0.52	0.00	…
3	0.00	0.33	0.46	0.46	0.50	…
4	0.14	0.46	0.46	0.52	0.00	…
$H = \Sigma_j$	0.47	1.76	1.97	1.97	1.00	

エントロピー(組織特異的遺伝子検出)

- 遺伝子iのエントロピー $H(x_i) = -\sum_{j=1}^{N} p_{ij} \log_2(p_{ij}),\ \text{where}\ p_{ij} = x_{ij} / \sum_{j=1}^{N} x_{ij}$

i

x_{ij}	遺伝子1	遺伝子2	遺伝子3	遺伝子4	遺伝子5	...
組織1	1	5	6	4	10	...
組織2	0	2	6	4	10	...
組織3	0	1	6	4	10	...
組織4	9	2	6	4	10	...
組織5	0	4	6	10	4	
組織6	0	6	6	4	10	
組織7	0	3	6	4	10	
組織8	0	5	6	4	10	
$\Sigma_j\ x_{ij}$	10	28	48	38	74	

j

i

p_{ij}	遺伝子1	遺伝子2	遺伝子3	遺伝子4	遺伝子5	...
1	0.10	0.18	0.13	0.11	0.14	...
2	0.00	0.07	0.13	0.11	0.14	...
3	0.00	0.04	0.13	0.11	0.14	...
4	0.90	0.07	0.13	0.11	0.14	...
5	0.00	0.14	0.13	0.26	0.05	
6	0.00	0.21	0.13	0.11	0.14	
7	0.00	0.11	0.13	0.11	0.14	
8	0.00	0.18	0.13	0.11	0.14	
Σ_j	1.00	1.00	1.00	1.00	1.00	

i

$-p_{ij}\log_2(p_{ij})$	遺伝子1	遺伝子2	遺伝子3	遺伝子4	遺伝子5	...
1	0.33	0.44	0.38	0.34	0.39	...
2	0.00	0.27	0.38	0.34	0.39	...
3	0.00	0.17	0.38	0.34	0.39	...
4	0.14	0.27	0.38	0.34	0.39	...
5	0.00	0.40	0.38	0.51	0.23	
6	0.00	0.48	0.38	0.34	0.39	
7	0.00	0.35	0.38	0.34	0.39	
8	0.00	0.44	0.38	0.34	0.39	
Σ_j	0.47	2.83	3.00	2.90	2.96	

組織特異的遺伝子は低いエントロピー

そうでないものは高い値

N:組織数(jの数) = 8

Hの取りうる範囲:$0 \leqq H \leqq \log_2 N \rightarrow 0 \leqq H \leqq 3$

- 解析 | 発現変動 | 3群間 | 対応なし | について(last modified 2014/02/02)
- 解析 | 発現変動 | 3群間 | 対応なし | Mulcom (Isella_2011)(last modified 2013/12/06)
- 解析 | 発現変動 | 3群間 | 対応なし | limma (Smyth_2004)(last modified 2014/02/03)
- 解析 | 発現変動 | 3群間 | 対応なし | 一元配置分散分析(One-way ANOVA)(last modified 2013/11/12)
- 解析 | 発現変動 | 3群間 | 対応なし | Kruskal-Wallis(クラスカル-ウォリス)検定 (last modified 2013/6/2)
- 解析 | 発現変動 | 多群間 | について(last modified 2013/6/2)
- 解析 | 発現変動 | 多群間 | SpeCond(Cavalli_2011) (last modified 2013/6/10)
- 解析 | 発現変動 | 多群間 | ROKU(Kadota_2006) (last modified 2014/05/15) NEW
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現
- 解析 | 発現

- 解析 | 発現変動 | 多群間 | ROKU(Kadota_2006)

解析 | 発現変動 | 多群間 | ROKU (Kadota_2006) NEW

ROKU法(Kadota et al., 2006)を用いて、遺伝子発現行列中の遺伝子を全体的な組織特異性の度合いでランキングします。
出力ファイル中の"modH"列の数値は、「ROKU論文中のAdditional file 1(Suppl.xls)の"H(**x'**)"列の数値」と対応しています。つまり、データ変換後のエントロピー値です。
"ranking"列は、modHの値でランキングした結果です。"ranking"列で昇順にソートすることで、全体的な組織特異性の度合いでランキングしていることになります。つまり、上位が「(どの組織で特異的かはこのスコアだけでは分からないが)組織特異性が高い遺伝子」ということになります。
残りの結果は「1:特異的高発現、-1:特異的低発現、0:その他」からなる「外れ値行列」です。例えば、組織A and Bで1, それ以外の組織で0を示す遺伝子(群)は「AとB特異的高発現遺伝子」と判断します。
「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21のsample21.txtの場合:

入力と出力の関係を簡単に説明します

log2変換後のデータであるという前提です。

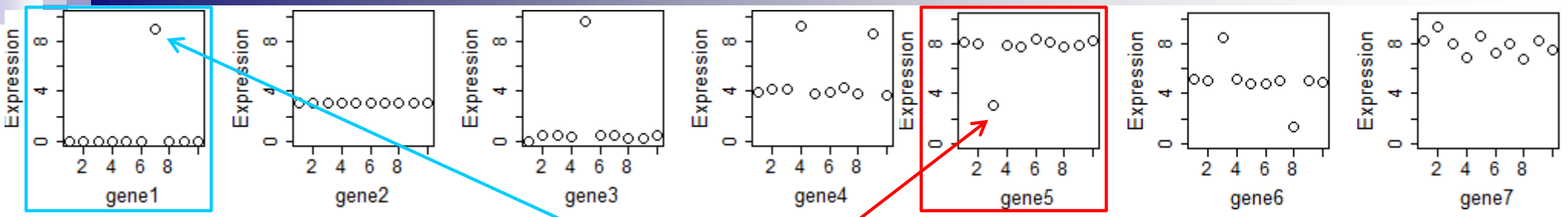
```
in_f <- "sample21.txt"                  #入力ファイル名を指定してin_fに格納
out_f <- "hoge1.txt"                    #出力ファイル名を指定してout_fに格納

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

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

#本番
hoge <- ROKU(data)                      #ROKUを実行した結果をhogeに格納
outlier <- hoge$outliers                #外れ値行列をoutlierに格納
modH <- hoge$modH                       #データ変換後のエントロピー値をmodHに格納(原著論文のH(x')の値に相当)
ranking <- hoge$rank                    #modHでランキングした結果をrankingに格納

#ファイルに保存
tmp <- cbind(rownames(data), outlier, modH, ranking)#左端の列が遺伝子ID, 次にサンプル数分だけの列からなる「外れ値行列
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存
```



入力：sample21.txt

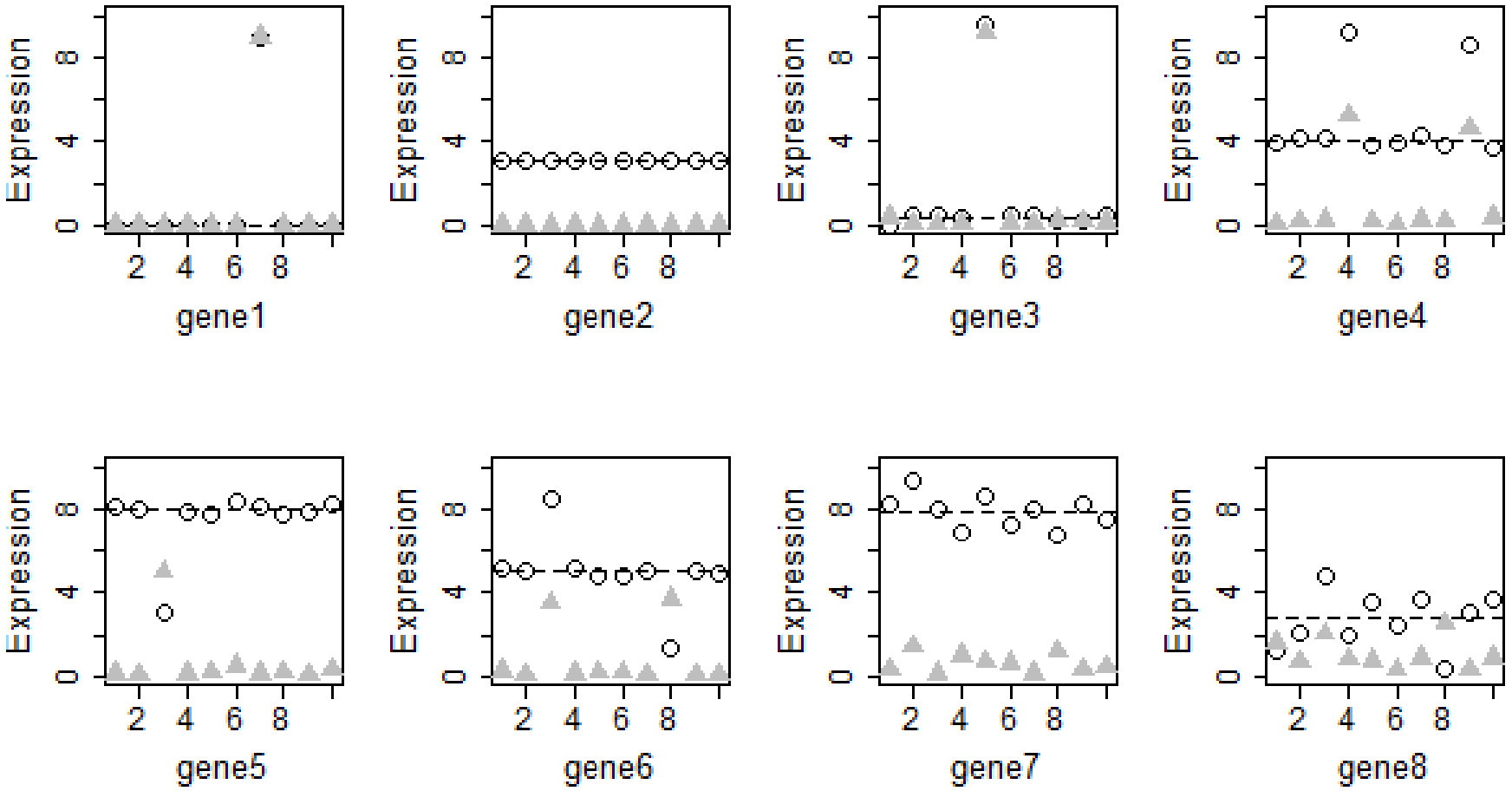
	tissue1	tissue2	tissue3	tissue4	tissue5	tissue6	tissue7	tissue8	tissue9	tissue10
gene1	0.00	0.00	0.00	0.00	0.00	0.00	9.00	0.00	0.00	0.00
gene2	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
gene3	0.02	0.41	0.41	0.38	9.60	0.49	0.44	0.16	0.21	0.52
gene4	3.95	4.12	4.20	9.20	3.84	3.97	4.23	3.80	8.60	3.64
gene5	8.06	7.93	3.00	7.82	7.75	8.42	8.06	7.75	7.88	8.26
gene6	5.20	5.00	8.50	5.10	4.84	4.78	5.00	1.30	5.00	4.89
gene7	8.20	9.30	8.00	6.90	8.60	7.30	8.00	6.70	8.20	7.50
gene8	1.20	2.10	4.80	2.00	3.50	2.50	3.65	0.30	3.10	3.63

これがデータ変換後のエントロピーとその順位

出力：hoge1.txt

	tissue1	tissue2	tissue3	tissue4	tissue5	tissue6	tissue7	tissue8	tissue9	tissue10	modH	ranking
gene1	0	0	0	0	0	0	1	0	0	0	0.000	1
gene2	0	0	0	0	0	0	0	0	0	0	3.322	8
gene3	0	0	0	0	1	0	0	0	0	0	0.768	2
gene4	0	0	0	1	0	0	0	0	1	0	1.718	5
gene5	0	0	−1	0	0	0	0	0	0	0	1.492	3
gene6	0	0	1	0	0	0	0	−1	0	0	1.645	4
gene7	0	0	0	0	0	0	0	0	0	0	2.952	6
gene8	0	0	0	0	0	0	0	0	0	0	3.032	7

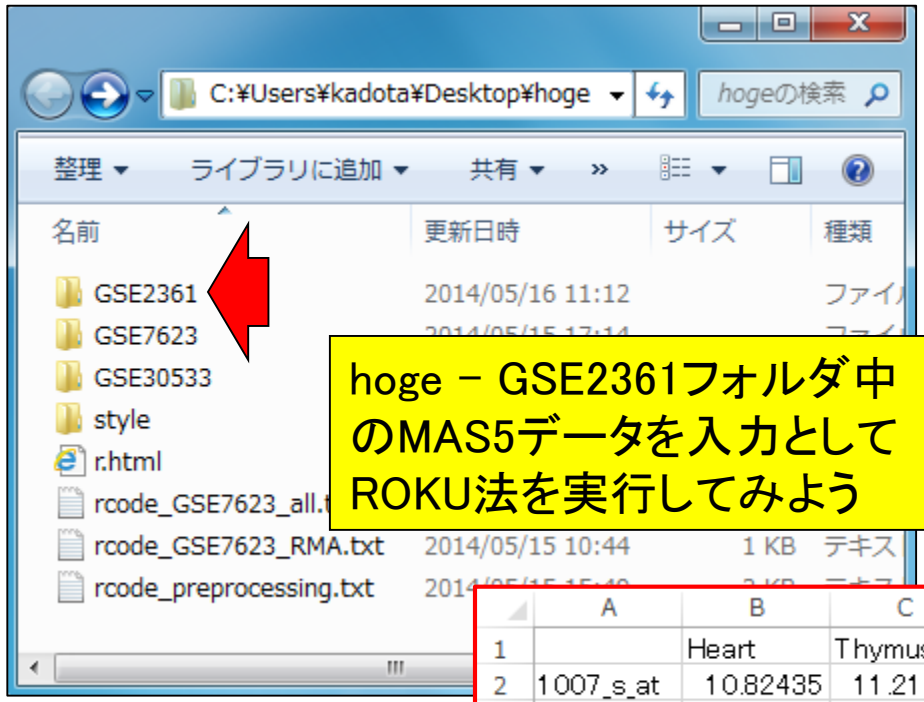
エントロピー(組織特異的遺伝子検出)



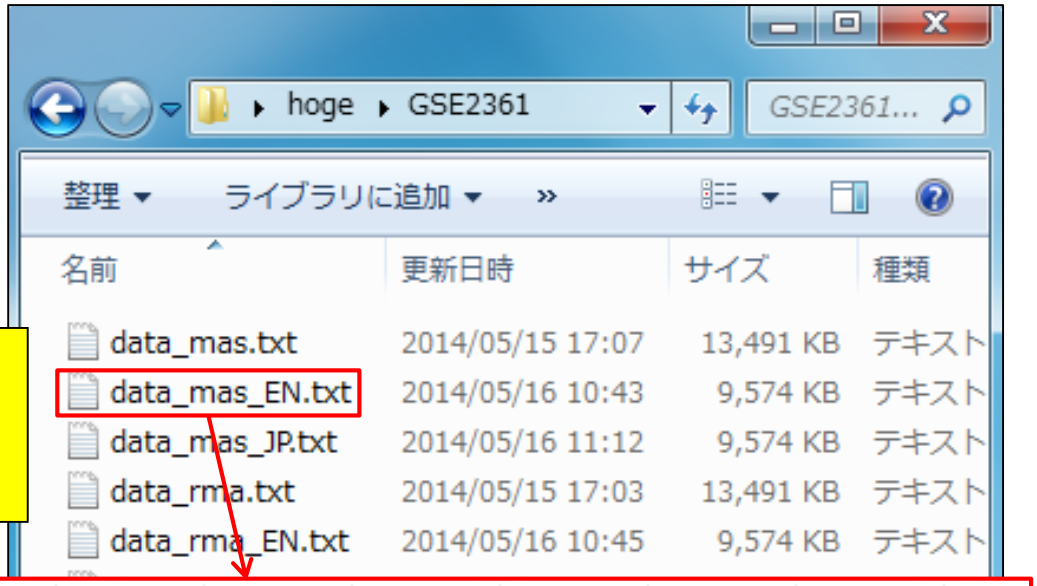
ROKU法はデータの変換を行うことでよりよいエントロピーでのランキング結果を得ている(変換前:○、変換後:▲)

GSE2361データを用いてROKUを実行

- 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)、…



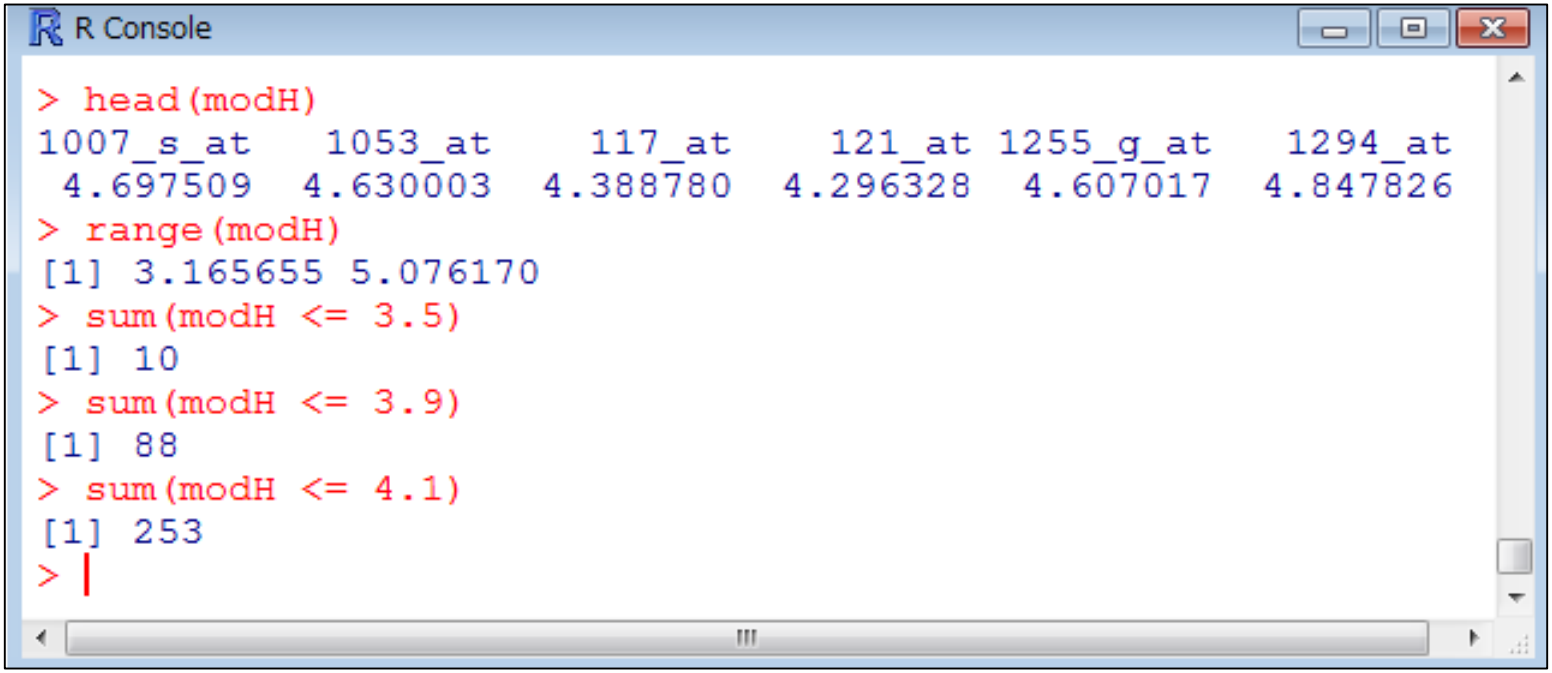
hoge – GSE2361フォルダ中のMAS5データを入力としてROKU法を実行してみよう



	A	B	C	D	E	F	G	H	I	J
1		Heart	Thymus	Spleen	Ovary	Kidney	Skeletal_Mu	Pancreas	Prostate	Small_I
2	1007_s_at	10.82435	11.21261	9.898072	10.8948	11.75531	10.92032	11.61243	11.22101	11.48
3	1053_at	6.621657	6.47488	7.371465	4.168622	7.350244	7.209918	8.362777	7.176571	7.75
4	117_at	8.259603	8.647364	8.689724	7.875649	5.083001	8.165044	7.96707	8.418082	5.719
5	121_at	11.26699	11.04186	11.39999	11.02855	13.13267	11.39138	12.32899	11.0559	11.28
6	1255_g_at	7.1757	6.477278	6.781766	7.048799	7.30767	6.600267	6.42359	6.694412	7.30
7	1294_at	9.137586	9.718507	9.083742	9.014997	8.813377	8.402562	9.146946	8.421351	10.12

課題(ROKU実行結果の解釈)

1. MAS5データ変換後のエントロピー値(modH列の値)の最小値と最大値を示せ。
2. MAS5データ変換後のエントロピー値(modH列の値)が4.0以下のprobeset数を示せ。
3. ROKU実行結果全体について簡単に考察せよ。(例:特異的高発現と特異的低発現の組織数分布、特異的組織数とエントロピー値との関係など)



1. サンプルデータ21のsample21.txtの場合:

log2変換後のデータであるという前提です。

```
in_f <- "sample21.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

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

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

#本番
hoge <- ROKU(data)                      #ROKUを実行した結果をhogeに格納
outlier <- h
modH <- hoge
ranking <- h

#ファイルに保
tmp <- cbind
write.table(
```

R Console

```
> in_f <- "sample21.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
>
> #必要なパッケージをロード
> library(TCC)                            #パッケージの読み込み
>
> #入力ファイルの読み込み
> data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定した$
> data
      tissue1 tissue2 tissue3 tissue4 tissue5 tissue6 tissue7 tissue8 tissue9 tissue10
gene1    0.00    0.00    0.00    0.00    0.00    0.00    9.00    0.00    0.00     0.00
gene2    3.00    3.00    3.00    3.00    3.00    3.00    3.00    3.00    3.00     3.00
gene3    0.02    0.41    0.41    0.38    9.60    0.49    0.44    0.16    0.21     0.52
gene4    3.95    4.12    4.20    9.20    3.84    3.97    4.23    3.80    8.60     3.64
gene5    8.06    7.93    3.00    7.82    7.75    8.42    8.06    7.75    7.88     8.26
gene6    5.20    5.00    8.50    5.10    4.84    4.78    5.00    1.30    5.00     4.89
gene7    8.20    9.30    8.00    6.90    8.60    7.30    8.00    6.70    8.20     7.50
gene8    1.20    2.10    4.80    2.00    3.50    2.50    3.65    0.30    3.10     3.63
> |
```

これが一般的な手元の入力ファイル読み込みです。他の手段として、Rパッケージが提供しているデータの読み込み法についても説明します

実行例が意味不明?!...ではなくて、hypoData_tsというサンプルデータがTCCパッケージ中で提供されているということです

R Console

```
> ?ROKU
>
```

http://127.0.0.1:29076/library/T(R: detect tissue-sp...

ROKU {TCC} R Documentation

detect tissue-specific (or tissue-selective) patterns from microarray data with many kinds of samples

Description

ROKU is a method for detecting tissue-specific (or tissue-selective) patterns from gene expression data for many tissues (or samples). ROKU (i) ranks genes according to their overall tissue-specificity
specific to each gene if a

Usage

```
ROKU(data, upper.lim
```

Arguments

data	numeric m where eac tissue, and gene in th expressior
upper.limit	numeric v tissues (or

http://127.0.0.1:29076/library/T(R: detect tissue-sp...

Kadota K, Nishimura SI, Bono H, Nakamura S, Hayashizaki Y, Okazaki Y, Takahashi K: Detection of genes with tissue-specific expression patterns using Akaike's Information Criterion (AIC) procedure. Physiol Genomics 2003, 12: 251-259.

Ueda T. Simple method for the detection of outliers. Japanese J Appl Stat 1996, 25: 17-26.

Examples

```
data(hypoData_ts)

result <- ROKU(hypoData_ts)
```

[Package *TCC* version 1.4.0 Index]

R Console

```
> in_f <- "sample21.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
>
> #必要なパッケージをロード
> library(TCC)                              #パッケージの読み込み
>
> #入力ファイルの読み込み
> data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="")#in_fで指定した$
> data
      tissue1 tissue2 tissue3 tissue4 tissue5 tissue6 tissue7 tissue8 tissue9 tissue10
gene1    0.00    0.00    0.00    0.00    0.00    0.00    9.00    0.00    0.00     0.00
gene2    3.00    3.00    3.00    3.00    3.00    3.00    3.00    3.00    3.00     3.00
gene3    0.02    0.41    0.41    0.38    9.60    0.49    0.44    0.16    0.21     0.52
gene4    3.95    4.12    4.20    9.20    3.84    3.97    4.23    3.80    8.60     3.64
gene5    8.06    7.93    3.00    7.82    7.75    8.42    8.06    7.75    7.88     8.26
gene6    5.20    5.00    8.50    5.10    4.84    4.78    5.00    1.30    5.00     4.89
gene7    8.20    9.30    8.00    6.90    8.60    7.30    8.00    6.70    8.20     7.50
gene8    1.20    2.10    4.80    2.00    3.50    2.50    3.65    0.30    3.10     3.63
> |
```

R Console

```
> data(hypoData_ts)
>
> result <- ROKU(hypoData_ts)
> hypoData_ts
      tissue1 tissue2 tissue3 tissue4 tissue5 tissue6 tissue7 tissue8 tissue9 tissue10
gene1    0.00    0.00    0.00    0.00    0.00    0.00    9.00    0.00    0.00     0.00
gene2    3.00    3.00    3.00    3.00    3.00    3.00    3.00    3.00    3.00     3.00
gene3    0.02    0.41    0.41    0.38    9.60    0.49    0.44    0.16    0.21     0.52
gene4    3.95    4.12    4.20    9.20    3.84    3.97    4.23    3.80    8.60     3.64
gene5    8.06    7.93    3.00    7.82    7.75    8.42    8.06    7.75    7.88     8.26
gene6    5.20    5.00    8.50    5.10    4.84    4.78    5.00    1.30    5.00     4.89
gene7    8.20    9.30    8.00    6.90    8.60    7.30    8.00    6.70    8.20     7.50
gene8    1.20    2.10    4.80    2.00    3.50    2.50    3.65    0.30    3.10     3.63
> |
```

上のdataオブジェクトと下のhypoData_tsオブジェクトの中身は同じです

Contents（第4回）

- デザイン行列の意味を理解（教科書p173-182）
  - limmaパッケージを用いた2群間比較のおさらい
  - limmaパッケージを用いた3群間比較（複製あり）
- 複製なし多群間比較（教科書p182-188）
  - limmaパッケージを用いた3群間比較（複製なし）
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- **機能解析（遺伝子セット解析）**
  - **基本的な考え方**
  - 前処理
    - MSigDBからの遺伝子セット情報（GMT形式ファイル）取得
    - ID変換（probe ID → gene symbol）
  - GSAパッケージを用いたパスウェイ解析
  - その他
- 分類

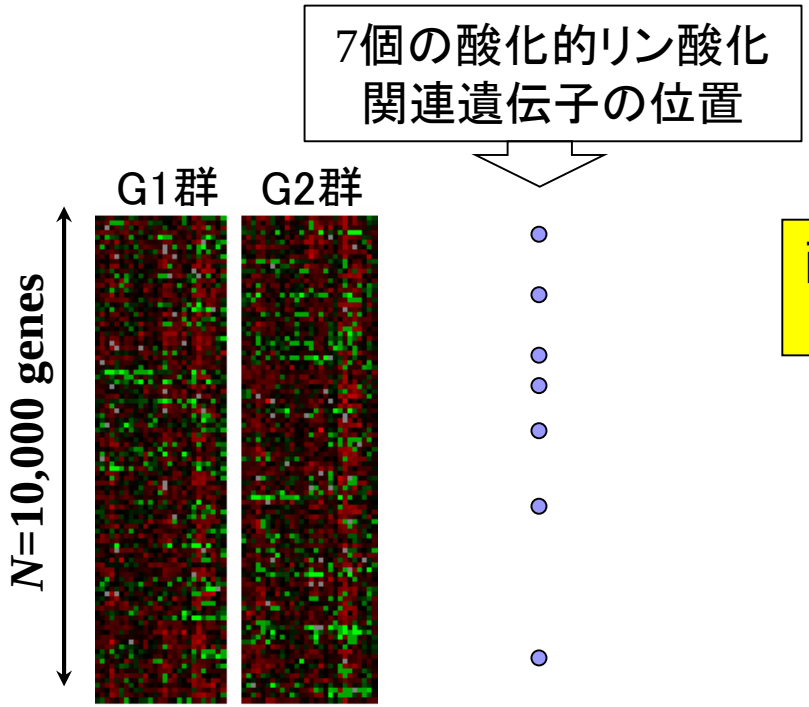
機能解析

発現に差のある遺伝子**セット**を探したい

- Gene Ontology (GO)解析(発現に差のあるGO termを探索)
  - 基本3カテゴリ(Cellular component (CC), Molecular Function (MF), Biological Process (BP))のどれでも可能
    - 例:肝臓の空腹状態 vs. 満腹状態のGO(BP)解析の結果、「脂肪酸β酸化」関連GO term (GO:0006635)が動いていることが分かった
- パスウェイ解析(発現に差のあるパスウェイを探索)
  - KEGG, BioCarta, Reactome pathway databaseのどれでも可能
    - 例:酸化的リン酸化パスウェイ関連遺伝子セットが糖尿病患者で動いていた
- モチーフ解析(発現に差のあるモチーフを探索)
  - 同じ3'-UTR microRNA結合モチーフをもつ遺伝子セット
  - 同じ転写因子結合領域(TATA-boxなど)をもつ遺伝子セット
    - 例:TATA-boxをもつ遺伝子セットがG1群 対 G2群比較で動いていた
- …

機能解析

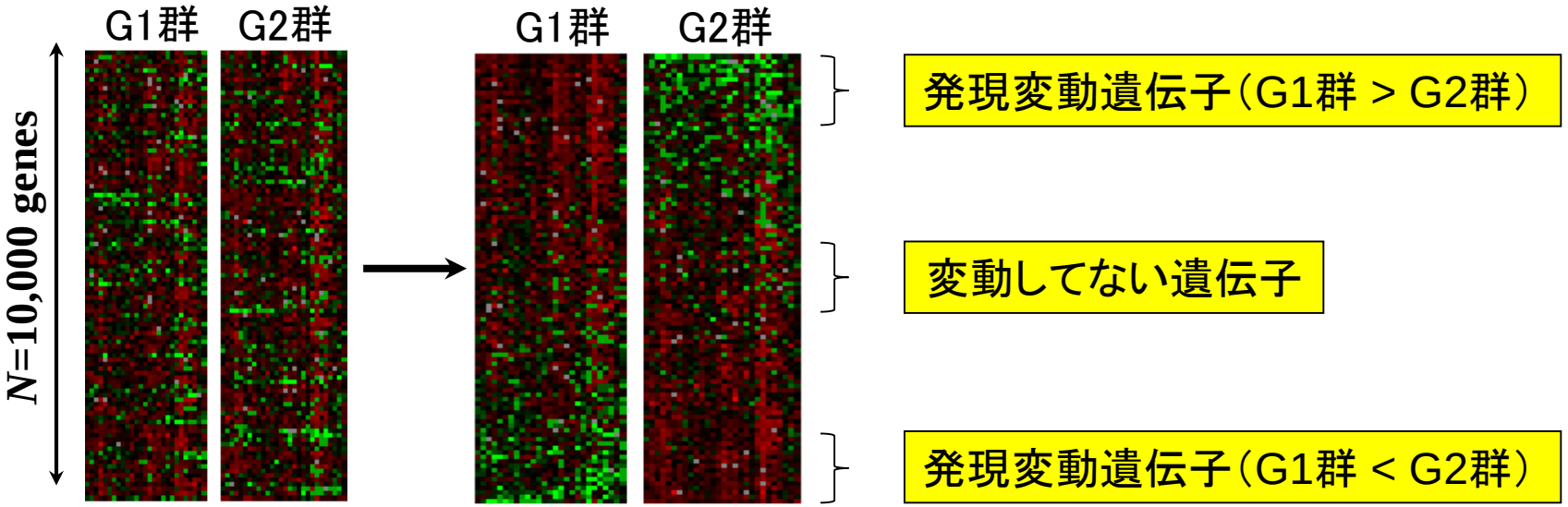
- 発現変動遺伝子セット解析手法（2群間比較用がほとんど）
  - *N*=10,000個の遺伝子からなる2群間比較用データ
  - この中に、XXX関連遺伝子が*n*個含まれている
    - 例：酸化的リン酸化（=XXX）関連遺伝子が7（=*n*）個含まれている



酸化的リン酸化関連遺伝子**セット**が
変動しているかどうかを調べたい

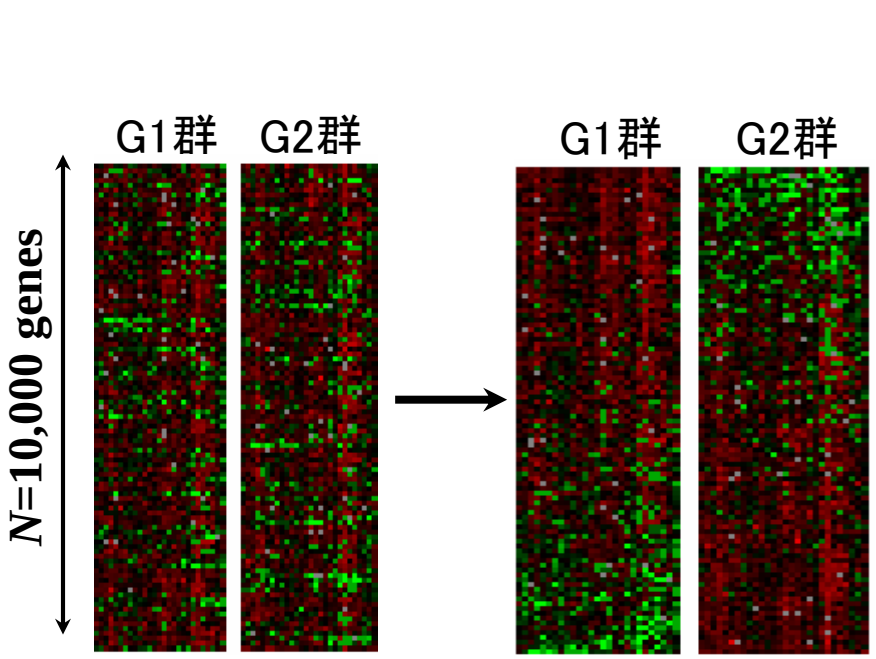
機能解析（遺伝子セット解析）

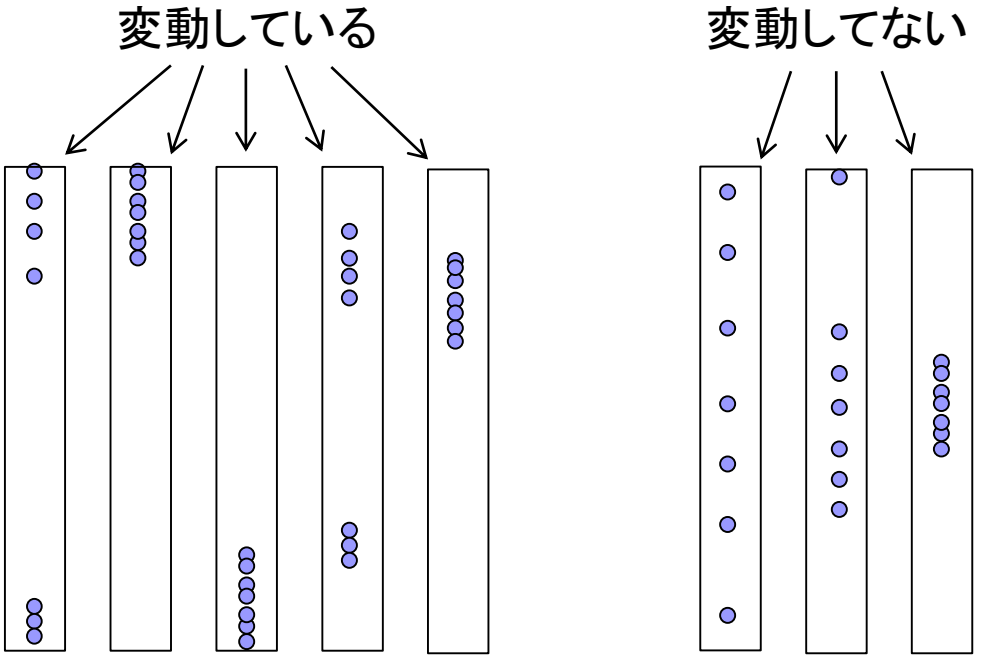
- 遺伝子ごとの統計量を算出（発現変動の度合いを数値化）
  - 例：*t*–統計量、$\log_2$（G2/G1）、相関係数、SAM、WAD



機能解析（遺伝子セット解析）

- 発現変動順にソート後の酸化的リン酸化関連遺伝子セットのステレオタイプな分布

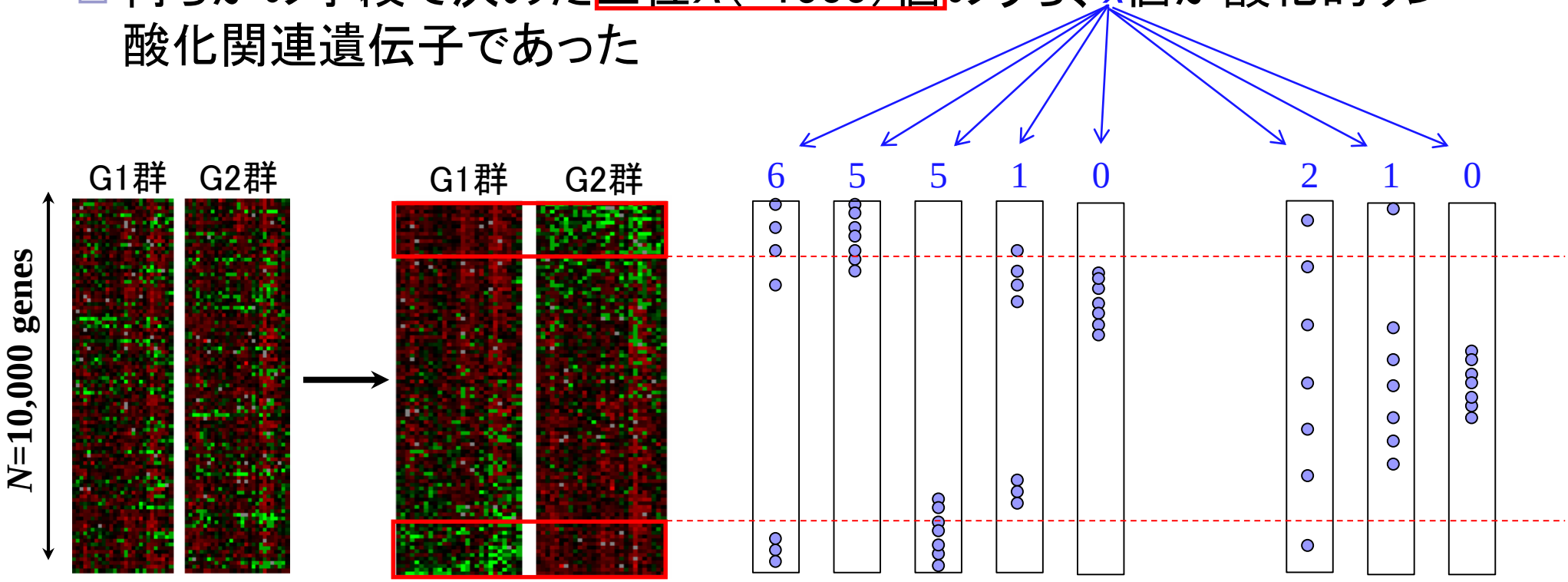




どうやって偏りを評価するのか?

遺伝子セット解析法（第一世代）

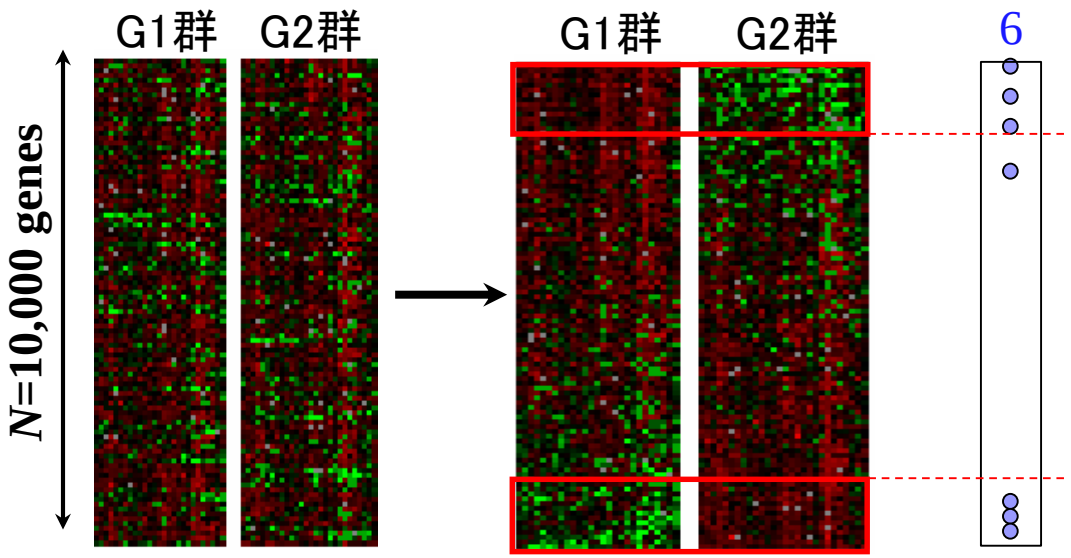
- Over-Representation Analysis (ORA)
  - 何らかの手段で決めた上位X(=1500)個のうち、x個が酸化的リン酸化関連遺伝子であった



酸化的リン酸化関連遺伝子セット（n=7）が変動していない場合：$x/n \fallingdotseq X/N$ (= 1500/10000)
酸化的リン酸化関連遺伝子セット（n=7）が変動している場合：$x/n \gg X/N$ (= 15%)

遺伝子セット解析法（第一世代）

- Over-Representation Analysis (ORA)
  - 何らかの手段で決めた上位X(=1500)個のうち、x個が酸化的リン酸化関連遺伝子であった



XXX=酸化的リン酸化関連遺伝子セット

	XXX	XXX以外	計
non-DEG数	1	8500-1	N-X
DEG数	6	1500-6	X
計	n	N-n	

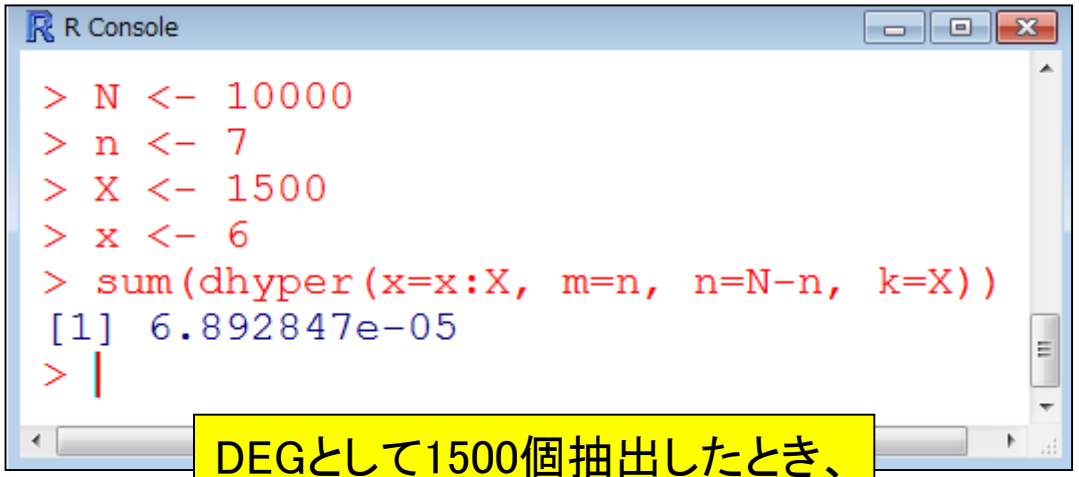
2×2分割表に基づく方法
- 超幾何検定
- カイ二乗検定
- …

遺伝子セット解析法(超幾何検定)

- N=10000個の遺伝子発現データ中にXXX=酸化的リン酸化関連遺伝子はn=7個含まれていた。上位X=1500個の発現変動遺伝子(DEG)の中にx=6個の酸化的リン酸化関連遺伝子が含まれていた
  - 帰無仮説:酸化的リン酸化関連遺伝子の割合はDEGとnon-DEG間で差がない

	XXX	XXX以外	計
non-DEG数	1	8500-1	8500
DEG数	6	1500-6	1500
計	7	9993	10000

	XXX	XXX以外	計
non-DEG数	n-x	$(N$-$n)$-$(X$-$x)$	N-X
DEG数	x	X-x	X
計	n	N-n	N



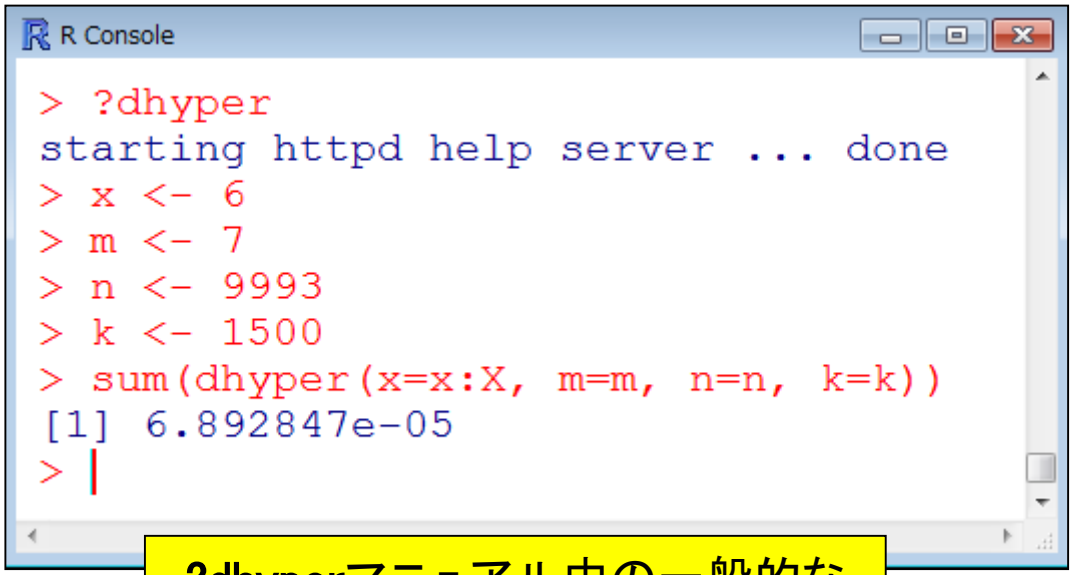
DEGとして1500個抽出したとき、酸化的リン酸化関連遺伝子が6個以上含まれる確率として算出

遺伝子セット解析法（超幾何検定）

- m=7個の白いボールとn=9993個の黒いボールが入った箱があります（トータルで$N=m+n$=10,000個）。この中からk=1500個ランダムに取り出したときにx=6個以上白いボールが含まれる確率を計算しなさい。

	白	黒	計
箱の中	1	9993−(1500−6)	8500
箱の外	6	1500−6	1500
計	7	9993	10000

	白	黒	計
箱の中	m-x	n-(k-x)	m+n-k
箱の外	x	k-x	k
計	m	n	N



?dhyperマニュアル中の一般的な説明に置き換えるとこんな感じです

遺伝子セット解析法(カイ二乗検定)

R Console

```
> N <- 10000
> n <- 7
> X <- 1500
> x <- 6
> data <- matrix(c((n-x), (N-n)-(X-x), x, (X-x)), ncol=2, byrow=T)
> data
     [,1] [,2]
[1,]    1 8499
[2,]    6 1494
> chisq.test(data)

        Pearson's Chi-squared test with Yates' continuity
        correction

data:  data
X-squared = 22.2032, df = 1, p-value = 2.453e-06

 警告メッセージ:
In chisq.test(data) :  カイ自乗近似は不正確かもしれません
>
```

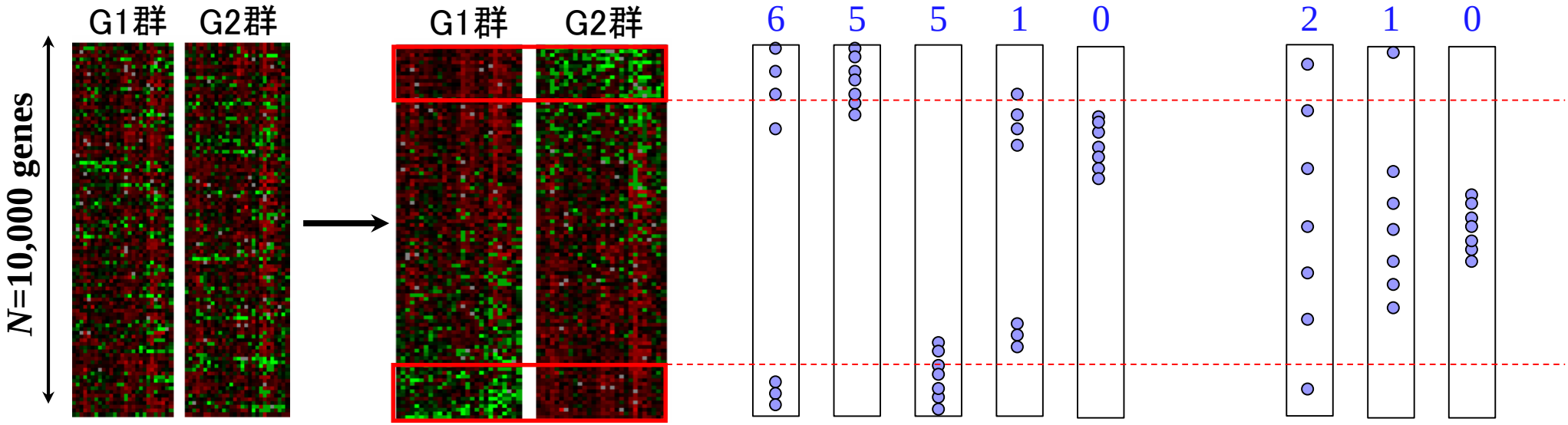
	XXX	XXX以外	計
non-DEG数	1	8500-1	8500
DEG数	6	1500-6	1500
計	7	9993	10000

	XXX	XXX以外	計
non-DEG数	n-x	$(N$-$n)$-$(X$-$x)$	N-X
DEG数	x	X-x	X
計	n	N-n	N

DEGとして1500個抽出したとき、酸化的リン酸化関連遺伝子が6個以上含まれる確率として算出

遺伝子セット解析法（第一世代）

- Over-Representation Analysis (ORA)
  - 何らかの手段で決めた上位X(=1500)個のうち、x個が酸化的リン酸化関連遺伝子であった



p-value	x=6	x=5	x=4	x=3	x=2	x=1	x=0
超幾何検定	6.89E-05	0.0012	0.0121	0.0737	0.2834	0.6795	1.0000
カイ二乗検定	2.45E-06	0.0003	0.0095	0.1247	0.6337	0.6337	0.5603
Fisher test	6.89E-05	0.0012	0.0121	0.0737	0.2834	1.0000	0.6039

$p < 0.05$を灰色で示した

遺伝子セット解析法(第一世代)

- Over-Representation Analysis (ORA)
  - GenMAPP (Dahlquist et al., *Nature Genet.*, **31**: 19-20, 2002)
  - FatiGO (Al-Shahrour et al., *Bioinformatics*, **20**: 578-580, 2004)
  - GOstat (Beissbarth et al., *Bioinformatics*, **20**: 1464-1465, 2004)
  - GOFFA (Sun et al., *BMC Bioinformatics*, **7 Suppl 2**: S23, 2006)
  - agriGO (Du et al., *Nucleic Acids Res.*, **38**: W64-W70, 2010)
  - …

第一世代(ORA)の短所

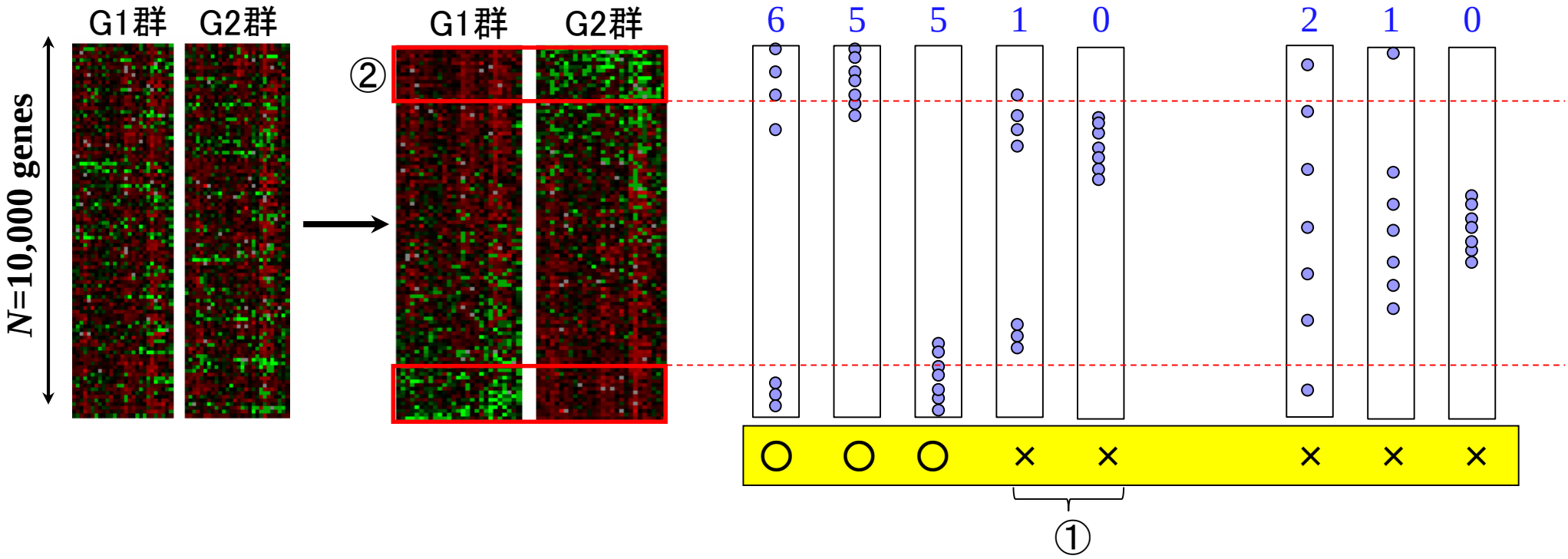
	XXX	XXX以外	計
non-DEG数	n-x	$(N$-$n)$-$(X$-$x)$	N-X
DEG数	x	X-x	X
計	n	N-n	N

③

① 全体的には動いているものの、個々の発現変動の度合いが弱い場合に検出困難

② 上位X個のX次第で結果が変わる

③ 情報量が落ちている(発現変動の度合い → カウント情報)



遺伝子セット解析法（第二世代）

- Functional Class Scoring (FCS)
  1. 遺伝子ごとの統計量を算出（発現変動の度合いを数値化）
     例：*t*-統計量、log（B/A）、相関係数、SAM、WAD
  2. 目的の遺伝子セットXXX（=酸化的リン酸化関連遺伝子）の偏りを何らかの方法で評価
     - *t*検定（XXX中の遺伝子群の統計量 vs. それ以外の遺伝子群の統計量）
     - Wilcoxon rank sum test（XXX中の遺伝子群の発現変動の順位 vs. それ以外）
     - XXX中の*n*個の遺伝子群の何らかの要約統計量S_{XXX}を計算しておき、*N*個の全遺伝子の中からランダムに*n*個を抽出して同じ統計量を計算する（例えば10万回）。10万回のうちS_{XXX}「以上」（大きければ大きいほど発現変動していることを意味する場合；その逆のときは「以下」）だった回数（例えばj回）に基づいてp値（= j / 100,000）を算出（いわゆるgene set permutationというアプローチ）
     - 本来のG1群 vs. G2群のラベル情報を用いて得られたXXX中の*n*個の遺伝子群の何らかの要約統計量S_{XXX}を計算しておく。ランダムにラベル情報を入れ替えて、同じ統計量を計算することを何回も繰り返して*p*値を算出（いわゆるPhenotype permutationというアプローチ）

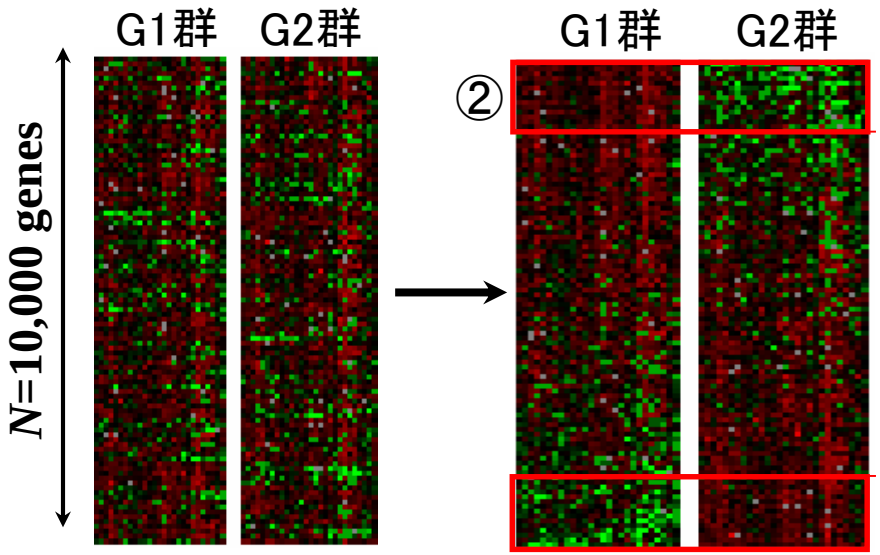
第一世代(ORA)→ 第二世代(FCS)

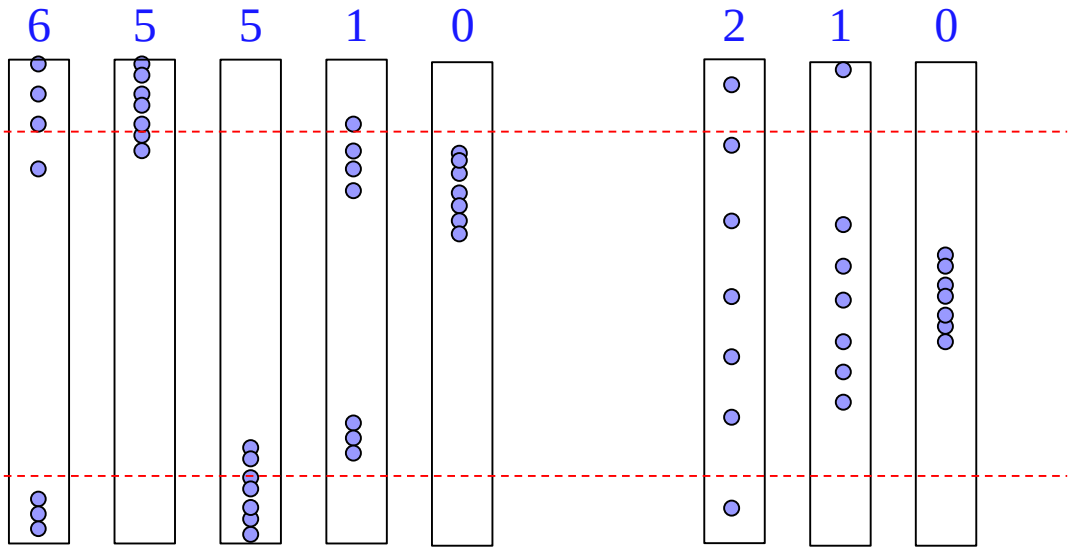
第一世代の欠点が改善

① 全体的には動いているものの、個々の発現変動の度合いが弱い場合に検出困難

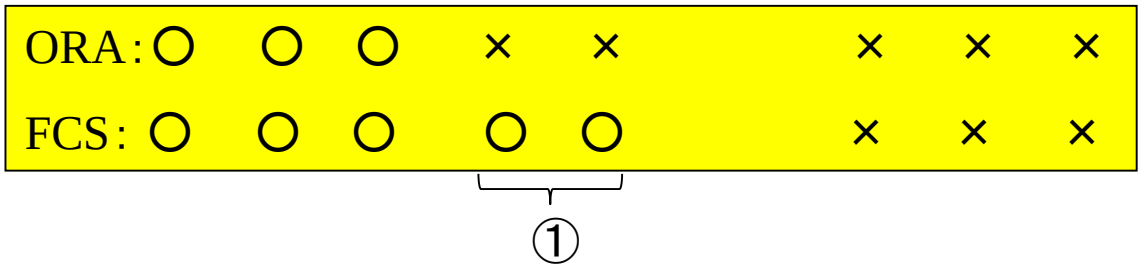
② 上位X個のX次第で結果が変わる

③ 情報量が落ちている(発現変動の度合い → カウント情報)





③	XXX	XXX以外	計
non-DEG数	n-x	$(N$-$n)$-$(X$-$x)$	N-X
DEG数	x	X-x	X
計	n	N-n	N



遺伝子セット解析法(第二世代)

- Functional Class Scoring (FCS)
  - **GSEA** (Subramanian et al., *PNAS*, **102**: 15545–15550, 2005)
  - PAGE (Kim and Volsky, *BMC Bioinformatics*, **6**: 144, 2005)
  - sigPathway (Tian et al., *PNAS*, **102**: 13544–13549, 2005)
  - GSA (Efron and Tibshirani, *Ann. Appl. Stat.*, **1**: 107–129, 2007)
  - GeneTrail (Backes et al., *Nucleic Acids Res.*, **35**: W186–W192, 2007)
  - SAM-GS (Dinu et al., *BMC Bioinformatics*, **8**: 242, 2007)
  - …

最も有名なのはGSEAです

遺伝子セット解析法（共通の問題）

- （知識ベースの解析法なので）解析対象がアノテーションの情報の豊富な生物種に限定
  - それ以外の生物種は、まずは地道にアノテーション情報を増やしていくことが先決（ではないだろうか）
- アノテーション情報の信頼度が高いとはいえない
  - なんらかのGO termがついていたとしても、その大部分のevidence codeが自動でつけられたもの（IEA, inferrred from electronic annotations）である…
- 遺伝子セット間の独立性の問題
  - 「数百個程度の遺伝子セットの中から、比較するサンプル間で動いている遺伝子セットはどれか？」という解析を遺伝子セット間の独立性を仮定して調べるが、そもそも独立ではない（GO term間の親子関係などから明らか）
    → いくつくらいの遺伝子セットが動いているのか？という問いに答えるすべがない
- 評価に用いられる「よく研究されているデータセット」は答えが完全に分かっているものではない（the actual biology is never fully known!）
  - “感度が高い”と謳っているだけの方法は…（全部の遺伝子セットが動いている → 感度100%）

最も有名なGSEAソフトウェアの使い方は**統合TV**で独学

TOGO TV CURATED 生命科学系DB・ツール使い倒し系チャンネル

はじめての方へ　番組ランキング　ほかの便利な方法　よくある質問　スタッフ　お問い合わせ

使い倒し系チャンネル 統合TV
旧 統合TVはこちらから!

DBCLS Database Center for Life Science

目的別に検索！

- ゲノム・核酸 配列解析
- タンパク質 配列・構造解析
- 発現制御解析
- 文献検索・辞書 情報収集 PC環境構築
- DBCLSサービス 講演・講習動画
- データベース別 分類

全番組のリストから調べたいDBやウェブツールに関するキーワードで検索！

検索窓にキーワードを入れると、入力のたびごとに即座に候補の番組が絞り込まれます。
先頭のタイトル行をクリックすると、昇順・降順で並び替えができます。

10 エントリを表示

検索: gsea

番組（YOUTUBEへのリンク）	番組の概要（画像をクリックすると番組の再生ページへ移動します。）
GSEA softwareの使い方 発展編	 Gene Set Enrichment Analysis (GSEA)は、予め用意した遺伝子セットが異なる条件下でどう振舞うかを調べる手法です。これを利用し発現プロファイルを解釈することができます。詳しいアルゴリズムは、Gene Set Enrichment Analysis PNAS paper (pdf)を参照してください。GSEA softwareは Broad Institutelによって実装されたGSEAを行うソフトウェアです。前回は、GSEA softwareのGUI版の導入と簡単な解析方法・結果の閲覧方法を解説しました。今回は、NCBI GEOより取得した公共の遺伝子発現データ(GSE1657:Adipocyte Differentiation [Homo sapiens]のSeries Matrix Files)を表計算ソフトを使い加工し、GSEA softwareに読み込ませ、解析を行う手順を解説します。
GSEA softwareの使い方 基本編	 Gene Set Enrichment Analysis (GSEA)は、予め用意した遺伝子セットが異なる条件

Contents（第4回）

- デザイン行列の意味を理解（教科書p173-182）
  - limmaパッケージを用いた2群間比較のおさらい
  - limmaパッケージを用いた3群間比較（複製あり）
- 複製なし多群間比較（教科書p182-188）
  - limmaパッケージを用いた3群間比較（複製なし）
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- 機能解析（遺伝子セット解析）
  - 基本的な考え方
  - **前処理**
    - **MSigDBからの遺伝子セット情報（GMT形式ファイル）取得**
    - **ID変換（probe ID → gene symbol）**
  - GSAパッケージを用いたパスウェイ解析
  - その他
- 分類

発現変動遺伝子セット解析おさらい

- Gene Ontology (GO)解析(発現に差のあるGO termを探索)
  - 基本3カテゴリ(Cellular component (CC), Molecular Function (MF), Biological Process (BP))のどれでも可能
    - 例:肝臓の空腹状態 vs. 満腹状態のGO(BP)解析の結果、「脂肪酸β酸化」関連GO term (GO:0006635)が動いていることが分かった
- パスウェイ解析(発現に差のあるパスウェイを探索)
  - KEGG, BioCarta, Reactome pathway databaseのどれでも可能
    - 例:酸化的リン酸化パスウェイ関連遺伝子セットが糖尿病患者で動いていた
- モチーフ解析(発現に差のあるモチーフを探索)
  - 同じ3'-UTR microRNA結合モチーフをもつ遺伝子セット
  - 同じ転写因子結合領域(TATA-boxなど)をもつ遺伝子セット
    - 例:TATA-boxをもつ遺伝子セットがG1群 対 G2群比較で動いていた
- …

どの遺伝子セットにどの遺伝子が所属しているかというgmt形式ファイルの取得が第一歩

Molecular Signature Database (MSigDB, ver. 4.0)

- c1: positional gene sets (326 gene sets)
  - ヒト染色体の位置ごとの遺伝子セットリストファイル (326 gene sets)
- c2: curated gene sets (4,722 gene sets)
  - **CGP**: chemical and genetic perturbations (3,402 gene sets)
  - **CP**: canonical pathways (1,320 gene sets)
  - **CP:BIOCARTA**: BioCarta gene sets (217 gene sets)
  - **CP:KEGG**: KEGG gene sets (186 gene sets)
  - **CP:REACTOME**: Reactome gene sets (674 gene sets)
- c3: motif gene sets (836 gene sets)
  - **MIR**: microRNA targets (221 gene sets)
  - **TFT**: transcription factor targets (615 gene sets)
- c4: computational gene sets (858 gene sets)
  - **CGM**: cancer gene neighborhoods (427 gene sets)
  - **CM**: cancer modules (431 gene sets)
- c5: gene ontology (GO) gene sets (1,454 gene sets)
  - **BP**: biological process (825 gene sets)
  - **CC**: cellular component (233 gene sets)
  - **MF**: molecular function (396 gene sets)
- c6: oncogenic signatures gene sets (189 gene sets)
- c7: immunologic signatures gene sets (1,910 gene sets)

発現変動と関連するKEGG パスウェイを調べたいとき

様々な遺伝子セット解析を行うためのgmt形式ファイルをダウンロード可能です

発現変動と関連するBP中のGO termsを調べたいとき

- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | pcot2 (Kong 2006)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | SAFE (Barry 2005)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | global[illegible] (Goeman
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | について [illegible]modified
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | Pathview ([illegible] 2013)([illegible]
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GAGE (Luo 2009)(las
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | SPIA (Tarca 2009)(las
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | sigPathway (Tian 200
- ...
- 解析 | 機能解析(GSEA周辺)について(以下は再編予定)(last modif
- 解析 | 機能解析 | PAGE法(Kim 2005:統計量の変換なし)の考え方
- 解析 | 機能解析 | PAGE法(Kim 2005:統計量の変換なし)を用いて
- 解析 | 機能解析 | PAGE法(Kim 2005;統計量の変換あり)を用いて

- 解析 | 機能解析 | パスウェイ(Pathway)解析 | について

解析 | 機能解析 | パスウェイ(Pathway)解析 | について NEW

機能解析の実体は遺伝子セット解析です。パスウェイ(Pathway)解析は遺伝子セット解析の一つであり、パスウェイ解析を行うためのパッケージもいくつか出ています。Reviewや手法評価系論文も下のほうにリストアップしています。
2014年6月に調査した結果をリストアップします(R)。

- GSA: Efron and Tibshirani, Ann. Appl. Stat., 2007
- GSAのウェブページ
- SPIA: Tarca et al., Bioinformatics, 2009
- dCoxS: Cho et al., BMC Bioinformatics, 2009
- GAGE: Luo et al., BMC Bioinformatics, 2009
- PADOG: Tarca et al., BMC Bioinformatics, 2012
- Pathview: Luo et al., Bioinformatics, 2013

2014年6月に調査した結果をリストアップします(R以外)。

- PLAGE(リンク切れ): Tomfohr et al., BMC Bioinformatics, 2005
- GSEA: Subramanian et al., PNAS, 2005
- GSEAのユーザーガイド
- ToppGene Suite(loginも要求なし; webtool): Chen et al., Nucleic Acids Res., 2009
- PINTA(loginも要求なし; webtool): Nitsch et al., Nucleic Acids Res., 2011
- FIDEA(loginも要求なし; webtool): D'Andrea et al., Nucleic Acids Res., 2013

2014年6月に調査した結果をリストアップします(Reviewや手法評価系論文)。

- Abatangelo et al., BMC Bioinformatics, 2009
- Khatri H, PLoS Comput Biol., 2012
- Maciejewski H, Brief Bioinform., 2013
- Tarca et al., PLoS One, 2013

2014年6月に調査した結果をリストアップします(遺伝子セットDB系)。

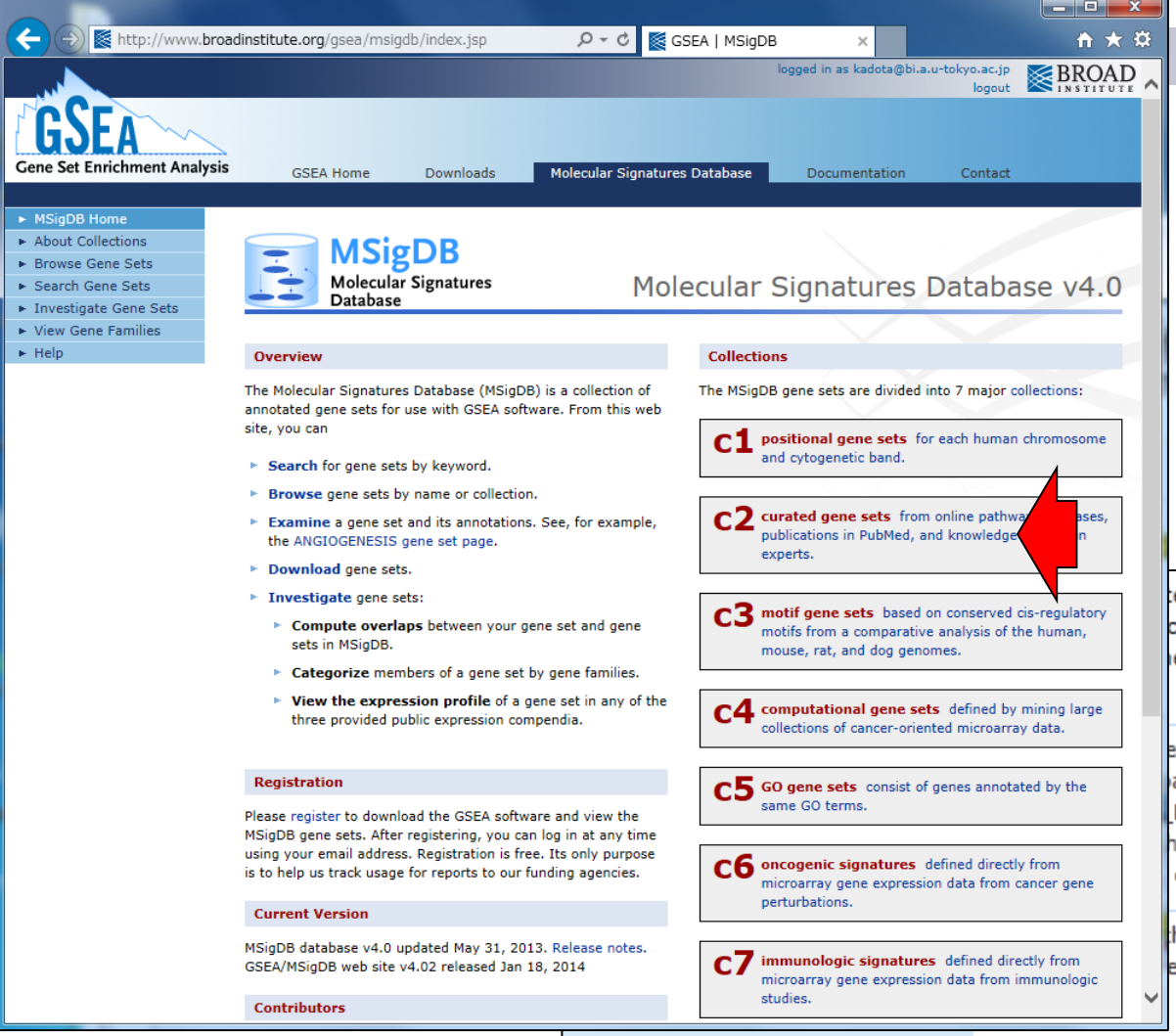
- MSigDB: Subramanian et al., PNAS, 2005
- hiPathDB: Yu et al., Nucleic Acids Res., 2012
- IPAVS: Sreenivasaiah et al., Nucleic Acids Res., 2012
- PAGED: Huang et al., BMC Bioinformatics, 2012
- IPAD: Zhang et al., BMC Bioinformatics, 2012

GSEAに代表される発現変動遺伝子セット解析は、基本的にGSEAの開発者らが作成した様々な遺伝子セット情報を収めたMolecular Signatures Database (MSigDB)からダウンロードした.gmt形式ファイルを読み込んで解析を行います。それゆえ、自分がどの遺伝子セットについて機能解析を行いたいのかを予め決めておく必要がありますが、パスウェイ解析の場合はc2のBioCarta, KEGG, Reactomeあたりを解析するのでしょう。gmt形式ファイルの基本的なダウンロード方法は以下の通りです:

1. Molecular Signatures Database (MSigDB)
   「register」のページで登録し、遺伝子セット[illegible]ンロード可能な状態にする。
2. Molecular Signatures Database (MSigDB)の
   「Download gene sets」の"Download"のところをクリックし、Loginページで登録したe-mail addressを入力。
3. これでMSigDBのダウンロードページに行けるので、目的に応じたgmtファイルをダウンロード(2014/06/02現在のバージョンは4.0)。
   「c2: curated gene sets」の「all canonical pathways, gene symbols」を解析したい場合:c2.cp.v4.0.symbols.gmt
   「c2: curated gene sets」の「BioCarta gene sets, gene symbols」を解析したい場合:c2.cp.biocarta.v4.0.symbols.gmt
   「c2: curated gene sets」の「KEGG gene sets, gene symbols」を解析したい場合:c2.cp.kegg.v4.0.symbols.gmt
   「c2: curated gene sets」の「Reactome gene sets, gene symbols」を解析したい場合:c2.cp.reactome.v4.0.symbols.gmt

トップページへ

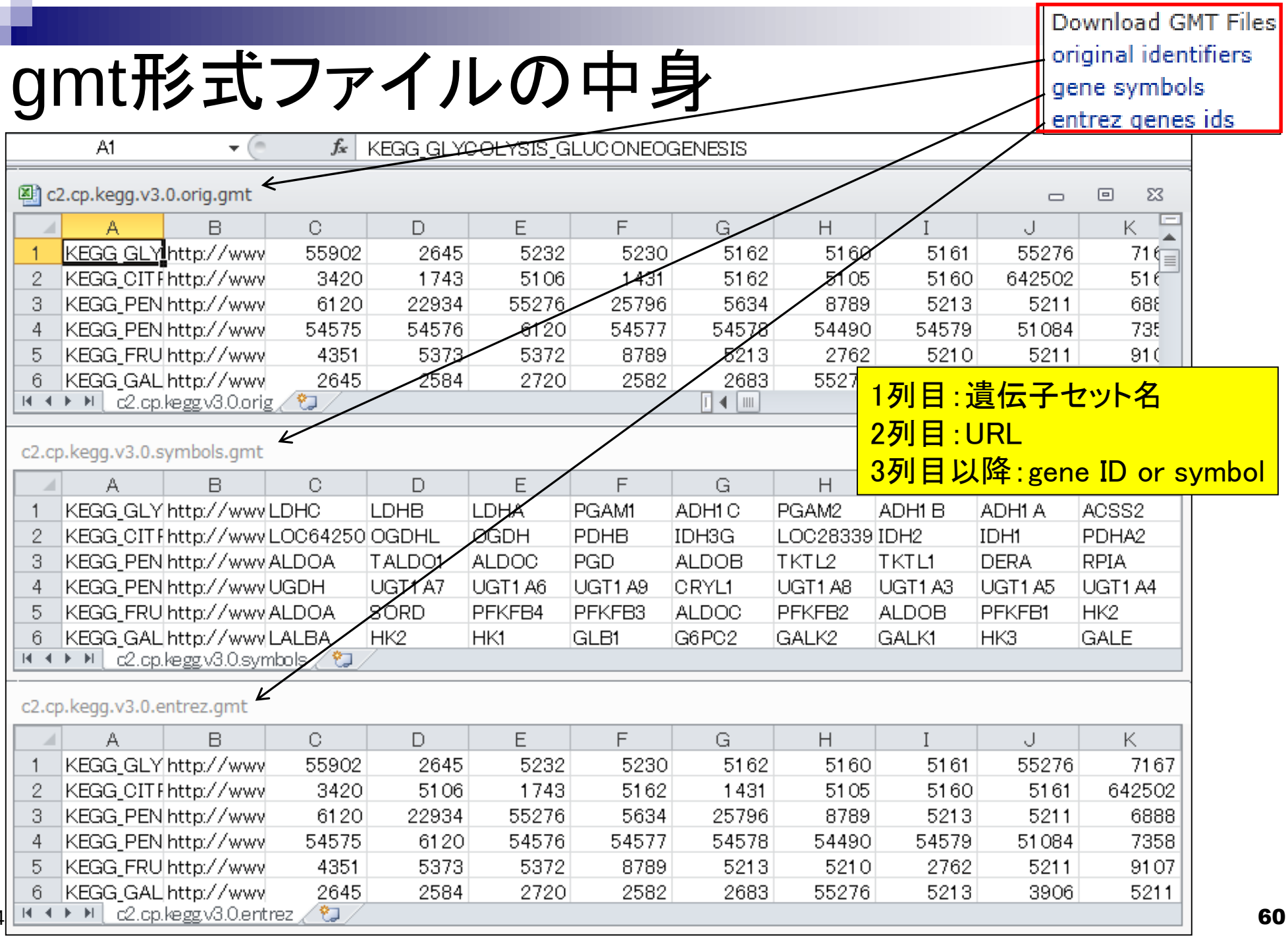
遺伝子セット解析(パスウェイ解析)を行うためのgmt形式ファイルのダウンロード方法はこちら



KEGG Pathway解析を行いたい場合は、ここからgmtファイルを取得

	ted from various sources such as online pathway cations in PubMed, and knowledge of domain e set page for each gene set lists its source. details	Download GMT Files original identifiers gene symbols entrez genes ids
	epresent gene expression signatures of genetic and ations. A number of these gene sets come in pairs: DN) gene set representing genes induced he perturbation. The gene set page for each gene set citation on which it is based.	Download GMT Files original identifiers gene symbols entrez genes ids
	he pathway databases. Usually, these gene sets are entations of a biological process compiled by domain	Download GMT Files original identifiers gene symbols entrez genes ids
CP:BIOCARTA: BioCarta gene sets (browse 217 gene sets)	Gene sets derived from the BioCarta pathway database (http://www.biocarta.com/genes/index.asp).	Download GMT Files original identifiers gene symbols entrez genes ids
CP:KEGG: KEGG gene sets (browse 186 gene sets)	Gene sets derived from the KEGG pathway database (http://www.genome.jp/kegg/pathway.html).	Download GMT Files original identifiers gene symbols entrez genes ids

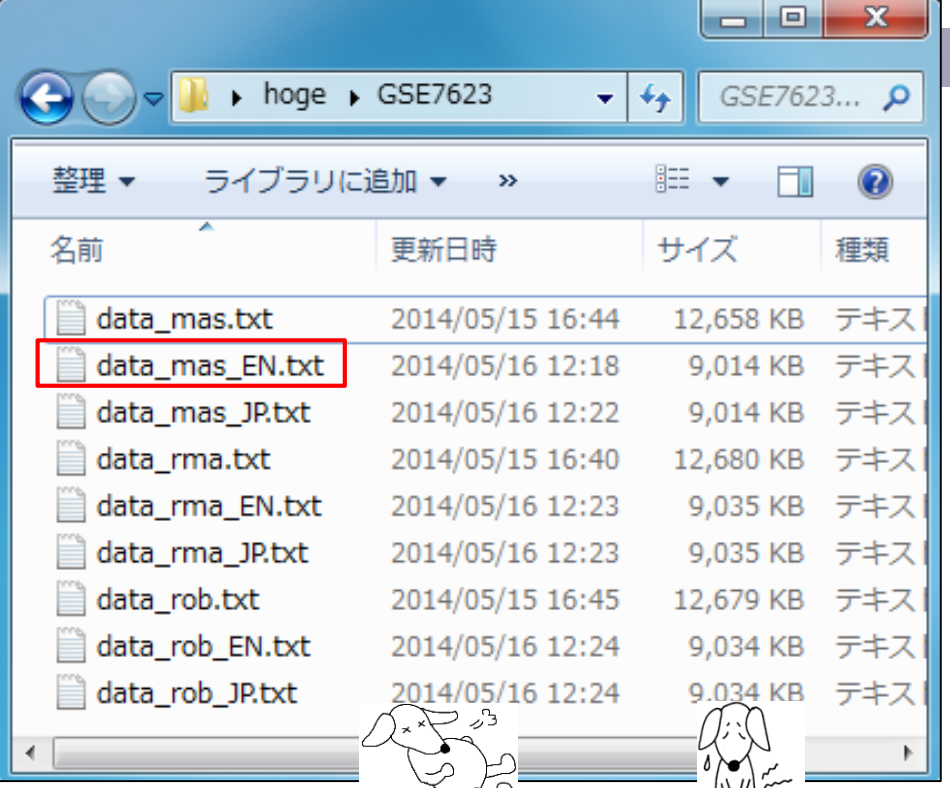
gmt形式ファイルの中身



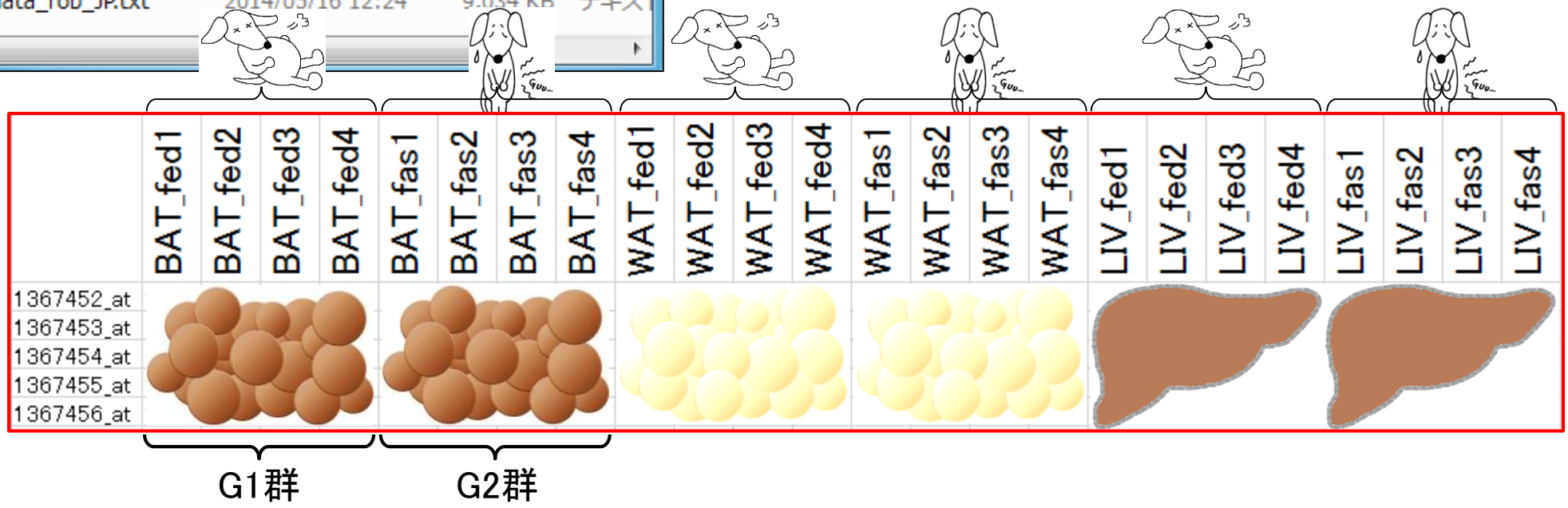
	A	B	C	D	E	F	G	H	I	J	K
1	KEGG_GLY	http://www	55902	2645	5232	5230	5162	5160	5161	55276	716
2	KEGG_CITR	http://www	3420	1743	5106	1431	5162	5105	5160	642502	516
3	KEGG_PEN	http://www	6120	22934	55276	25796	5634	8789	5213	5211	688
4	KEGG_PEN	http://www	54575	54576	6120	54577	54578	54490	54579	51084	735
5	KEGG_FRU	http://www	4351	5373	5372	8789	5213	2762	5210	5211	910
6	KEGG_GAL	http://www	2645	2584	2720	2582	2683	5527			

	A	B	C	D	E	F	G	H	I	J	K
1	KEGG_GLY	http://www	LDHC	LDHB	LDHA	PGAM1	ADH1C	PGAM2	ADH1B	ADH1A	ACSS2
2	KEGG_CITR	http://www	LOC64250	OGDHL	OGDH	PDHB	IDH3G	LOC28339	IDH2	IDH1	PDHA2
3	KEGG_PEN	http://www	ALDOA	TALDO1	ALDOC	PGD	ALDOB	TKTL2	TKTL1	DERA	RPIA
4	KEGG_PEN	http://www	UGDH	UGT1A7	UGT1A6	UGT1A9	CRYL1	UGT1A8	UGT1A3	UGT1A5	UGT1A4
5	KEGG_FRU	http://www	ALDOA	SORD	PFKFB4	PFKFB3	ALDOC	PFKFB2	ALDOB	PFKFB1	HK2
6	KEGG_GAL	http://www	LALBA	HK2	HK1	GLB1	G6PC2	GALK2	GALK1	HK3	GALE

	A	B	C	D	E	F	G	H	I	J	K
1	KEGG_GLY	http://www	55902	2645	5232	5230	5162	5160	5161	55276	7167
2	KEGG_CITR	http://www	3420	5106	1743	5162	1431	5105	5160	5161	642502
3	KEGG_PEN	http://www	6120	22934	55276	5634	25796	8789	5213	5211	6888
4	KEGG_PEN	http://www	54575	6120	54576	54577	54578	54490	54579	51084	7358
5	KEGG_FRU	http://www	4351	5373	5372	8789	5213	5210	2762	5211	9107
6	KEGG_GAL	http://www	2645	2584	2720	2582	2683	55276	5213	3906	5211



GSE7623 (Nakai et al., 2008)の**対数変換後**のデータを入力として、BAT_fed vs. BAT_fasの遺伝子セット解析をやってみよう

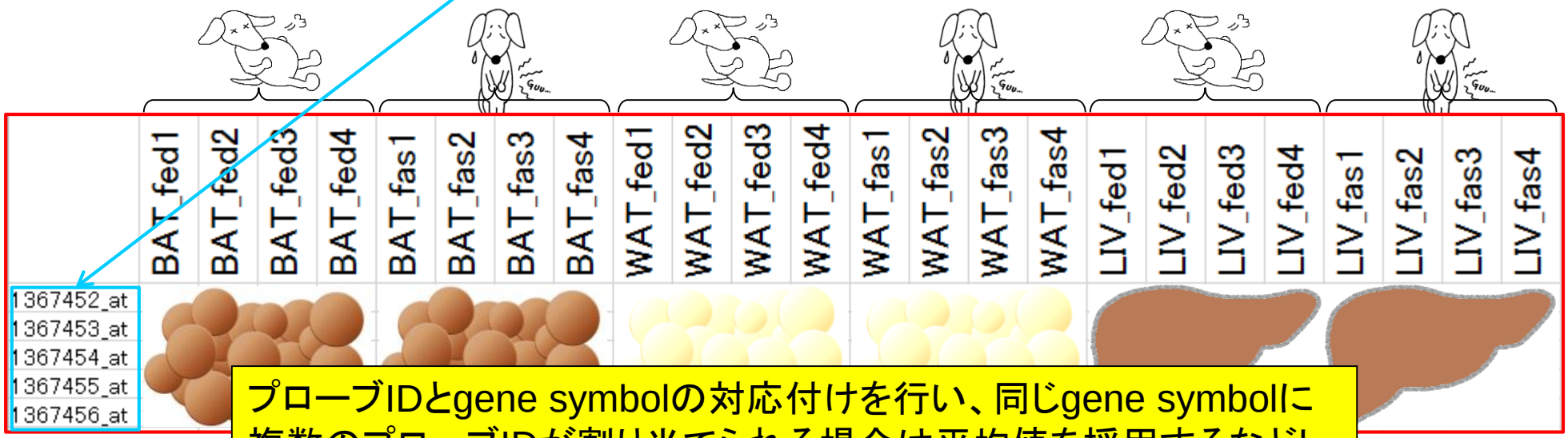


解析前に対応付けを行う必要がある

c2.cp.kegg.v3.0.symbols.gmt

	A	B	C	D	E	F	G	H	I
1	KEGG_GLYCOLYSIS_GLUCONEOGENESIS	http://www	LDHC	LDHB	LDHA	PGAM1	ADH1C	PGAM2	ADH1B
2	KEGG_CITRATE_CYCLE_TCA_CYCLE	http://www	LOC64250	OGDHL	OGDH	PDHB	IDH3G	LOC28339	IDH2
3	KEGG_PENTOSE_PHOSPHATE_PATHWAY	http://www	ALDOA	TALDO1	ALDOC	PGD	ALDOB	TKTL2	TKTL1
4	KEGG_PENTOSE_AND_GLUCURONATE_INTERCONVERSIONS	http://www	UGDH	UGT1A7	UGT1A6	UGT1A9	CRYL1	UGT1A8	UGT1A3
5	KEGG_FRUCTOSE_AND_MANNOSE_METABOLISM	http://www	ALDOA	SORD	PFKFB4	PFKFB3	ALDOC	PFKFB2	ALDOB
6	KEGG_GALACTOSE_METABOLISM	http://www	LALBA	HK2	HK1	GLB1	G6PC2	GALK2	GALK1

c2.cp.kegg.v3.0.symbols



プローブIDとgene symbolの対応付けを行い、同じgene symbolに複数のプローブIDが割り当てられる場合は平均値を採用するなどしてnon-redundantにする（折り畳む；つぶす；collapse）作業が必要

- イントロ | 発現データ取得 | 公共DBから (last modified 2014/05/11) NEW
- イントロ | 発現データ取得 | inSilicoDb(Taminau_2011) (last modified 2013/08/20)
- イントロ | 発現データ取得 | ArrayExpress(Kauffmann_2009) (last modified 2014/05/15)推奨 NEW
- イントロ | 発現データ取得 | GEOquery(Davis_2007) (last modified 2013/08/20)
- イントロ | アノテーション情報取得 | 公共DB(GEO)から (last modified 2013/08/18)
- イントロ | アノテーション情報取得 | GEOquery(Davis_2007) (last modified 2014/06/03)推奨 NEW
- イントロ | アノテーション情報取得 | Rのパッケージ*.dbから (last modified 2014/06/02) NEW
- イントロ | プローブ配列情報取得 | Rのパッケージから (last modified 2013/08/16)
- イントロ | トラ
- イントロ | トラ
- イントロ | Aff

- イントロ | アノテーション情報取得 | GEOquery(Davis_2007)

教科書p70-71

イントロ | アノテーション情報取得 | GEOquery (Davis_2007) NEW

公共DBGene Expression Omnibus (GEO)に登録されているアレイ(Platform)のアノテーション情報をGEOqueryというRパッケージを用いてゲットするやり方を示します。

「ファイル」

1. Affymetrix R

```
out_f <- "
param <- "

#必要なパッ
library(GE

#前処理
data <- ge

#本番
out <- dat
write.table
```

3. "GSE"から始まるIDをたよりにアレイのID情報を内部的に入手してアノテーション情報を取得したい場合:

GSE7623 (Nakai et al., BBB, 2008)は1種類のアレイ(GPL1355)しか使っていないので、1つのファイルのみ生成されます。出力ファイルの情報に相当するoutオブジェクト中の、自分がほしいprobe ID列とgene symbol列が1列目と11列目に存在することがあらかじめ分かっているという前提です。colnames(out)でわかります。

```
param1 <- "GSE7623"                    #入手したいGEO IDを指定
param2 <- "hoge3_"                     #出力ファイル名の最初の部分を指定
param_posi <- c(1, 11)

#必要なパッケージをロード
library(GEOquery)

#前処理
data <- getGEO(param1)                 #指定したGEO IDのデータを取得
sapply(data, annotation)               #用いられたアレイ情報(GPL ID)を表示

#本番
hoge <- sapply(data, annotation)       #用いられたアレイ情報(GPL ID)をhogeに格納
for(i in 1:length(hoge)){              #hogeの要素数(用いられたアレイ数)分だけループを
    out_f <- paste(param2, hoge[i], ".txt", sep="")#出力ファイル名を作成した結果をout_
    out <- data[[i]]@featureData@data  #アノテーション情報抽出結果をoutに格納
    out <- out[,param_posi]            #サブセットを抽出
    write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を指
}
```

遺伝子発現データは、公共DBのGEOからGSE7623というIDで取得したものだった。ここから、プローブIDとgene symbolの対応付けを行うためのアノテーションファイルを取得可能

教科書p70-71

3. "GSE"から始まるIDをたよりにアレイのID情報を内部的に入手してアノテーション情報を取得したい場合:

GSE7623 (Nakai et al., BBB, 2008)は1種類のアレイ(GPL1355)しか使っていないので、1つのファイルのみ生成されます。出力ファイルの情報に相当するoutオブジェクト中の、自分がほしいprobe ID列とgene symbol列が1列目と11列目に存在することがあらかじめ分かっているという前提です。colnames(out)でわかります。

```
param1 <- "GSE7623"                    #入手したいGEO IDを指定
param2 <- "hoge3_"                     #出力ファイル名の最初の部分を指定
param_posi <- c(1, 11)                 #outオブジェクト中のID列とgene symbol列の位置情

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

#前処理
data <- getGEO(param1)                 #指定したGEO IDのデータを取得
sapply(data, annotation)               #用いられたアレイ情報(GPL ID)を表示

#本番
hoge <- sapply(data, annotation)       #用いられたアレイ情報(GPL ID)をhogeに格納
for(i in 1:length(hoge)){              #hogeの要素数(用いられたアレイ数)分だけループを
    out_f <- paste(param2, hoge[i], ".txt", sep="")#出力ファイル名を作成した結果をout_
    out <- data[[i]]@featureData@data  #アノテーション情報抽出結果をoutに格納
    out <- out[,param_posi]            #サブセットを抽出
    write.table(out, out_f, sep="\t", append=F, quote=F, row.names=F)#outの中身を指
}
```

プローブIDとgene symbolからなるアノテーションファイルを取得できています

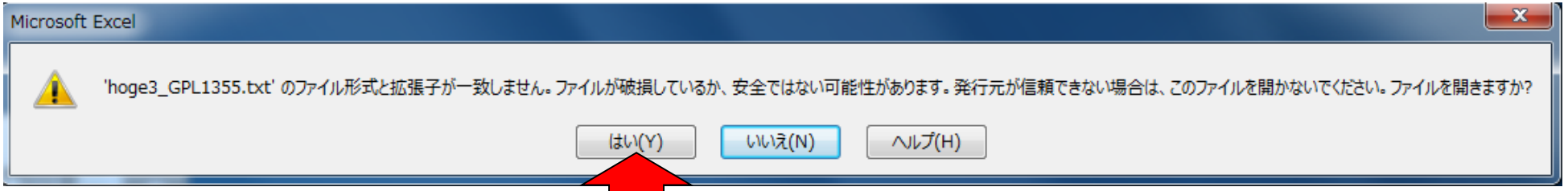
R Console

```
> sapply(data, annotation)                    #用いられたアレイ情報$
GSE7623_series_matrix.txt.gz
                  "GPL1355"
>
> #本番
> hoge <- sapply(data, annotation)            #用いられたアレイ情報$
> for(i in 1:length(hoge)){                   #hogeの要素数(用いら$
+     out_f <- paste(param2, hoge[i], ".txt", sep="")#出力ファ$
+     out <- data[[i]]@featureData@data        #アノテーション情報抽$
+     out <- out[,param_posi]                  #サブセットを抽出
+     write.table(out, out_f, sep="\t", append=F, quote=F, row$
+ }
>
```

hoge3_GPL1355.txt

ID	Gene Symbol
1367452_at	Sumo2
1367453_at	Cdc37
1367454_at	Copb2
1367455_at	Vcp
1367456_at	Ube2d3
1367457_at	Becn1
1367458_at	Lypla2
1367459_at	Arf1
1367460_at	Gdi2
1367461_at	Copb1
1367462_at	Capns1
1367463_at	Phb2
1367464_at	Puf60
1367465_at	Dad1
1367466_at	Prpf8
1367467_at	Iscu
1367468_at	Scand1
1367469_at	Eif4g2
1367470_at	Sar1a
1367471_at	Polr2e
1367472_at	Uba1
1367473_at	Tomm22

エクセルで開くときには注意が必要！



① 1行1列目のところが”ID”から始まる文字列の場合にこのような現象が起こるようですが、基本無視で構いません

BMC Bioinformatics. 2004 Jun 23;5:80.

Mistaken identifiers: gene name errors can be introduced inadvertently when using Excel in bioinformatics.

Zeeberg BR[1], Riss J, Kane DW, Bussey KJ, Uchio E, Linehan WM, Barrett JC, Weinstein JN.

Author information

Abstract

BACKGROUND: When processing microarray data sets, we recently noticed that some gene names were being changed inadvertently to non-gene names.

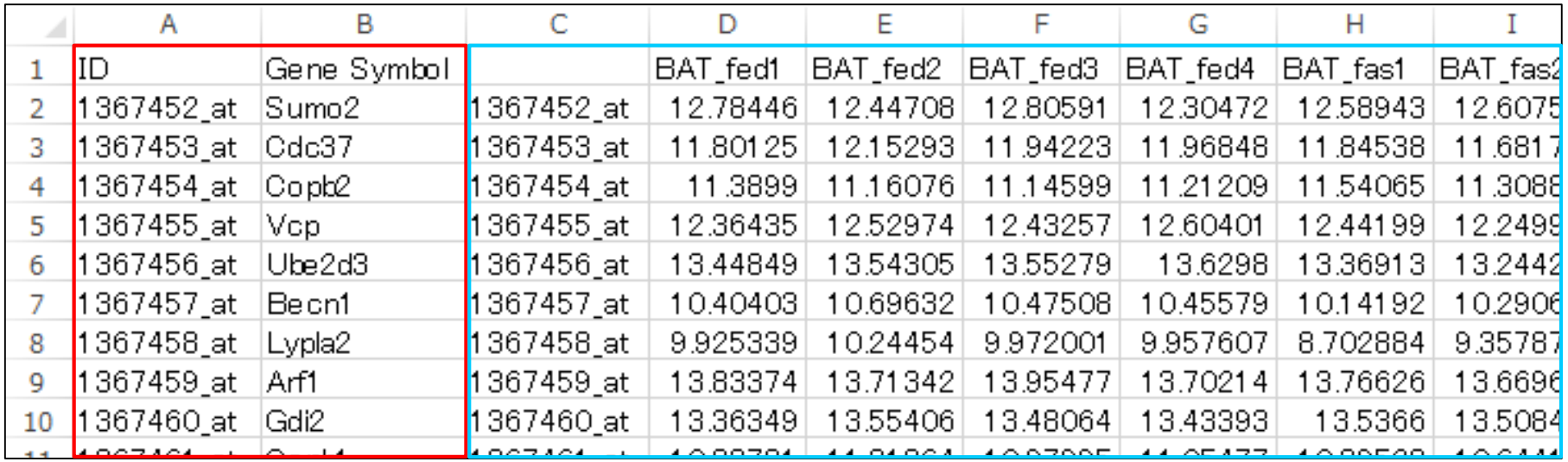
RESULTS: A little detective work traced the problem to default date format conversions and floating-point format conversions in the very useful Excel program package. The date conversions affect at least 30 gene names; the floating-point conversions affect at least 2,000 if Riken identifiers are included. These conversions are irreversible; the original gene names cannot be recovered.

CONCLUSIONS: Users of Excel for analyses involving gene names should be aware of this problem, which can cause genes, including medically important ones, to be lost from view and which has contaminated even carefully curated public databases. We provide work-arounds and scripts for circumventing the problem.

② エクセルを開いたあと、ドラッグ&ドロップで開いてはだめ!
編集して保存したい場合には、「ファイル」–「開く」でファイルを指定して開くべし!
そのまま開くと例えば**March2**というgene symbolが日付と認識されてしまうため、これを防ぐ必要があります！

対応付けの基礎情報はあるが...

Gene Symbol列でソートしてみると



	A	B	C	D	E	F	G	H	I
1	ID	Gene Symbol		BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2
2	1367452_at	Sumo2	1367452_at	12.78446	12.44708	12.80591	12.30472	12.58943	12.6075
3	1367453_at	Cdc37	1367453_at	11.80125	12.15293	11.94223	11.96848	11.84538	11.6817
4	1367454_at	Copb2	1367454_at	11.3899	11.16076	11.14599	11.21209	11.54065	11.3088
5	1367455_at	Vcp	1367455_at	12.36435	12.52974	12.43257	12.60401	12.44199	12.2499
6	1367456_at	Ube2d3	1367456_at	13.44849	13.54305	13.55279	13.6298	13.36913	13.2442
7	1367457_at	Becn1	1367457_at	10.40403	10.69632	10.47508	10.45579	10.14192	10.2906
8	1367458_at	Lypla2	1367458_at	9.925339	10.24454	9.972001	9.957607	8.702884	9.35787
9	1367459_at	Arf1	1367459_at	13.83374	13.71342	13.95477	13.70214	13.76626	13.6696
10	1367460_at	Gdi2	1367460_at	13.36349	13.55406	13.48064	13.43393	13.5366	13.5084

hoge3_GPL1355.txt

data_mas_EN.txt

対応付けの基礎情報はあるが...

	A	B	C	D	E	F	G	H	I
1	ID	Gene Symbol		BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2
2	1367452_at	Sumo2	1367452_at	12.78446	12.44708	12.80591	12.30472	12.58943	12.6075
3	1367453_at	Cdc37	1367453_at	11.80125	12.15293	11.94223	11.96848	11.84538	11.6817
4	1367454_at	Copb2	1367454_at	11.3899	11.16076	11.14599	11.21209	11.54065	11.3088
5	1367455_at	Vcp	1367455_at	12.36435	12.52974	12.43257	12.60401	12.44199	12.2499
6	1367456_at	Ube2d3	1367456_at	13.44849	13.54305	13.55279	13.6298	13.36913	13.2442
7	1367457_at	Becn1	1367457_at	10.40403	10.69632	10.47508	10.45579	10.14192	10.2906
8	1367458_at	Lypla2	1367458_at	9.925339	10.24454	9.972001	9.957607	8.702884	9.35787
9	1367459_at	Arf1	1367459_at	13.83374	13.71342	13.95477	13.70214	13.76626	13.6696
10	1367460_at	Gdi2	1367460_at	13.36349	13.55406	13.48064	13.43393	13.5366	13.5084

ID	Gene Symbol		BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fe
1369509_a_at	A1bg	1369509_a_at	2.283274	2.663464	3.393108	5.581337	4.444774	2.070498	2.366629	1.242299	3.202
1368784_at	A1cf	1368784_at	5.844694	5.913832	5.755455	5.509977	4.184778	6.572447	5.657811	5.381227	5.246
1370027_a_at	A1i3 /// Mug1	1370027_a_at	7.186455	6.369903	3.668318	3.498324	7.211794	6.427345	2.480638	3.215938	5.959
1378371_at	A2bp1	1378371_at	3.395028	5.712191	5.063096	6.231098	6.612678	6.064106	6.833413	6.143807	4.328
1380516_at	A2bp1	1380516_at	4.359652	4.69428	3.010153	4.315149	4.516653	2.632701	3.892409	5.18021	1.917
1383130_at	A2bp1	1383130_at	6.081817	4.593572	4.535909	8.811696	9.702333	9.75088	9.105133	8.825036	5.409
1383257_at	A2bp1	1383257_at	1.501018	2.966239	2.232853	7.283267	8.426518	8.651317	7.625359	7.31545	1.329
1385235_at	A2bp1	1385235_at	3.847942	4.026388	3.911792	8.330072	9.3733	9.137933	8.885068	8.734736	4.634
1390594_at	A2bp1	1390594_at	4.641244	6.627334	5.234572	5.89079	2.245885	1.417343	6.138234	6.252581	5.294
1382945_at	A2ld1	1382945_at	9.974381	10.2494	9.811327	9.831842	9.093722	8.947993	8.82087	9.13077	9.386
1392725_at	A2ld1	1392725_at	6.696741	5.425293	6.610669	6.104502	6.084417	6.684742	5.49265	2.381073	6.770

同じgene symbolを持つプローブIDが複数存在することがわかる

同じgene symbolをもつものをまとめる

入力1：hoge3_GPL1355.txt

ID	Gene Symbol
1369509_a_at	A1 bg
1368784_at	A1 cf
1370027_a_at	A1 i3 /// Mug
1378371_at	A2 bp1
1380516_at	A2 bp1
1383130_at	A2 bp1
1383257_at	A2 bp1
1385235_at	A2 bp1
1390594_at	A2 bp1
1382945_at	A2 ld1
1392725_at	A2 ld1

入力2：data_mas_EN.txt

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fe
1369509_a_at	2.283274	2.663464	3.393108	5.581337	4.444774	2.070498	2.366629	1.242299	3.202
1368784_at	5.844694	5.913832	5.755455	5.509977	4.184778	6.572447	5.657811	5.381227	5.246
1370027_a_at	7.186455	6.369903	3.668318	3.498324	7.211794	6.427345	2.480638	3.215938	5.959
1378371_at	3.395028	5.712191	5.063096	6.231098	6.612678	6.064106	6.833413	6.143807	4.328
1380516_at	4.359652	4.69428	3.010153	4.315149	4.516653	2.632701	3.892409	5.18021	1.917
1383130_at	6.081817	4.593572	4.535909	8.811696	9.702333	9.75088	9.105133	8.825036	5.409
1383257_at	1.501018	2.966239	2.232853	7.283267	8.426518	8.651317	7.625359	7.31545	1.329
1385235_at	3.847942	4.026388	3.911792	8.330072	9.3733	9.137933	8.885068	8.734736	4.634
1390594_at	4.641244	6.627334	5.234572	5.89079	2.245885	1.417343	6.138234	6.252581	5.294
1382945_at	9.974381	10.2494	9.811327	9.831842	9.093722	8.947993	8.82087	9.13077	9.386
1392725_at	6.696741	5.425293	6.610669	6.104502	6.084417	6.684742	5.49265	2.381073	6.770

出力：data_mas_EN_symbol.txt

	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fe
A1 bg	2.283274	2.663464	3.393108	5.581337	4.444774	2.070498	2.366629	1.242299	3.202
A1 cf	5.844694	5.913832	5.755455	5.509977	4.184778	6.572447	5.657811	5.381227	5.246
A1 i3 /// Mug1	7.186455	6.369903	3.668318	3.498324	7.211794	6.427345	2.480638	3.215938	5.959
A2 bp1	3.971117	4.770001	3.998062	6.810345	6.812894	6.275713	7.079936	7.075304	3.81
A2 ld1	8.335561	7.837348	8.210998	7.968172	7.58907	7.816367	7.15676	5.755922	8.07
A2 m	4.201252	5.612429	4.637						
A3 galt2	5.848709	7.467876	6.4374						
A4 galt	4.71439	1.895368	3.0720						
AA926063 ///	4.5776	2.525739	2.21708	4.756383	4.458302	6.10321	6.484667	5.121584	5.833
Aaas									9
Aacs									2

マイクロアレイごとに搭載されている遺伝子の種類や重複度が異なるため、この作業は重要

R Console

```
> (3.395028+4.359652+6.081817+1.501018+3.847942+4.641244)/6
[1] 3.971117
>
```

- 前処理 | フィルタリング | 分散が小さいものを除去 (last modified 2013/11/15)
- 前処理 | ID変換 | について(last modified 2014/06/03) NEW
- 前処理 | ID変換 | probe ID --> gene symbol(last modified 2014/06/03) NEW
- 前処理 | ID変換 | probe ID --> Entrez ID(last modified 2014/06/03) NEW
- 前処理 | ID変換 | probe ID --> その他(last modified 2014/06/03) NEW
- 前処理 |
- 解析 | 基
- 解析 | 基
- 解析 | 基

- 前処理 | ID変換 | probe ID --> gene symbol
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GAGE (Luo 2009)

前処理 | ID変換 | probe ID --> gene symbol NEW

probe IDの遺伝子発現行列を入力として、gene symbolの遺伝子発現行列を出力するやり方を示します。probe IDとgene symbol
の対応関係情報が必要ですので、様々なやり方を示しています。同じgene symbolをもつ複数のprobe IDsが存
paramで指定した方法で要約統計量を計算します。代表値(要約統計量)は、平均値(mean)、中央値(median)、
好きなものを指定できます。
「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

プログラムの組み方で速度が結構違います(データフレーム形式より行列形式のほうが早いらしい)

1. サンプルデータ20の31,099 probesets×24

Affymetrix Rat Genome 230 2.0 Arrayを用い
イントロ | アノテーション情報取得 | GEOque
data_rma_2.txtの1列目のID情報とhoge3_G
前提です。1分程度で終わります。

```
in_f1 <- "data_rma_2.txt"
in_f2 <- "hoge3_GPL1355.txt"
out_f <- "hoge1.txt"
param <- mean

#前処理(IDとGene symbolとの対応関
sym <- read.table(in_f2, header=
IDs <- as.vector(sym[,1])
names(IDs) <- rownames(sym)
uniqID <- unique(IDs)
uniqID <- uniqID[uniqID != ""]
uniqID <- uniqID[!is.na(uniqID)]
uniqID <- uniqID[!is.nan(uniqID)

#本番
data <- read.table(in_f1, header
hoge <- t(apply(as.matrix(uniqID
                apply(d[which(s
        }, data, IDs, param))
rownames(hoge) <- uniqID

#ファイルに保存
```

rcode_ID_conversion.txt

```
in_f1 <- "data_mas_EN.txt"          #入力ファイル名を指定してin_f1に格納(発現
in_f2 <- "hoge3_GPL1355.txt"        #入力ファイル名を指定してin_f2に格納(Gene
out_f <- "data_mas_EN_symbol.txt"   #出力ファイル名を指定してout_fに格納
param <- mean                       #代表値を指定

#前処理(IDとGene symbolとの対応関係を含む情報を入手)
sym <- read.table(in_f2, header=TRUE, row.names=1, sep="¥t", quote="")#in_f2で指
IDs <- as.vector(sym[,1])           #Gene symbol情報をベクトルに変換し、IDsに
names(IDs) <- rownames(sym)
uniqID <- unique(IDs)
uniqID <- uniqID[uniqID != ""]
uniqID <- uniqID[!is.na(uniqID)]
uniqID <- uniqID[!is.nan(uniqID)]   #uniqIDの中から指定したIDが"NaN"のものを

#本番
data <- read.table(in_f1, header=TRUE, row.names=1, sep="¥t", quote="")#in_f1で
hoge <- t(apply(as.matrix(uniqID), 1, function(i, d = data, s = IDs, p = param)
                apply(d[which(s == i), ], 2, p, na.rm = TRUE)#dataの中から現在の
        }, data, IDs, param))       #apply関数でdata,IDs,paramを使えるように
rownames(hoge) <- uniqID            #non-redundant IDをhogeの行の名前として

#ファイルに保存
tmp <- cbind(rownames(hoge), hoge)  #指定したIDの列を行列hogeの左端に挿入し、
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を指
```

data_mas_EN_symbol.txtは、このコードのコピペで作成しています

Contents（第4回）

- デザイン行列の意味を理解（教科書p173-182）
  - limmaパッケージを用いた2群間比較のおさらい
  - limmaパッケージを用いた3群間比較（複製あり）
- 複製なし多群間比較（教科書p182-188）
  - limmaパッケージを用いた3群間比較（複製なし）
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- 機能解析（遺伝子セット解析）
  - 基本的な考え方
  - 前処理
    - MSigDBからの遺伝子セット情報（GMT形式ファイル）取得
    - ID変換（probe ID → gene symbol）
  - **GSAパッケージを用いたパスウェイ解析**
  - **その他**
- 分類

- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | SAFE (Barry 2005)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | globaltest (Goeman 2004)(last modified 2014/06/01)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | について(last modified 2014/06/02) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | Pathview (Luo 2013)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GAGE (Luo 2009)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | SPIA (Tarca 2009)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GSA (Efron 2007)(last modified 2014/06/03)推奨 NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | sigPathway (Tian 2009)(last modified 2014/06/01) NEW

- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GSA (Efron 2007)

解析 | 機能解析 | パスウェイ(Pathway)解析 | GSA (Efron_2007) NEW

GSAパッケージを用いた解析のやり方を示します。

「ファイル」

1.サンプルデー

肝臓のみから
からgene symb

```
in_f1 <- "
in_f2 <- "
out_f1 <-
out_f2 <-
param_G1 <
param_G2 <
param_FDR

#必要なパッ
library(GS

#入力ファイ
gmt <- GSA
data <- re
rownames(d
data.cl <-

#GSA本番
out <- GSA
        ge
        ge
        re
```

2.サンプルデータ20のdata_rma_2_nr.txtの場合:

14,132 genes×24 samplesのデータです。オリジナルのprobeset IDからgene symbolにID変換がなされています。LIV_fed vs. LIV_fastedの2群間比較を行うべく、それらの位置情報を指定しています。

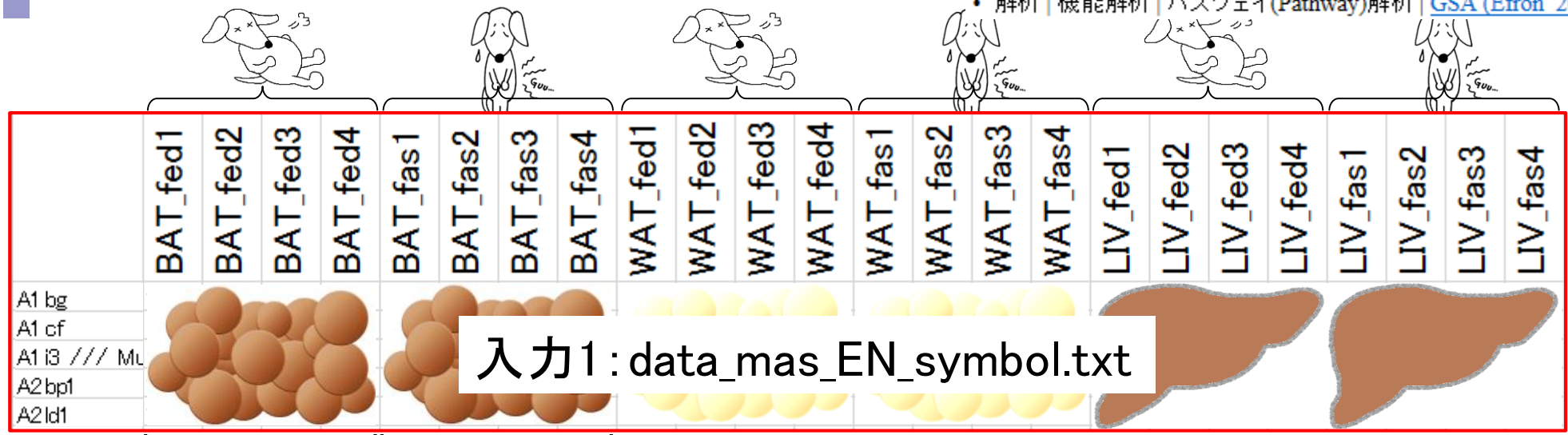
data_mas_EN_symbol.txtを入力としてBAT_fed vs. BAT_fasの遺伝子セット解析をやってみよう

```
in_f1 <- "data_rma_2_nr.txt"              #入力ファイル名を指
in_f2 <- "c2.cp.kegg.v4.0.symbols.gmt"    #入力ファイル名を指
out_f1 <- "hoge2_G1.txt"                  #出力ファイル名を指
out_f2 <- "hoge2_G2.txt"                  #出力ファイル名を指
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 <- 0.1                          #DEG検出時のfalse discovery rate (FDR)閾値を指定
param_posi <- c(17:20, 21:24)             #G1群およびG2群の位置情報を指定

#必要なパッケージをロード
library(GSA)                              #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
gmt <- GSA.read.gmt(in_f2)                #in_f2で指定したファイルの読み込み（不完全な最終行が見つかり
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定したファイル(
rownames(data) <- toupper(rownames(data))#IDを大文字に変換している(gmtファイルに合わせるため)
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)                            #確認してるだけです

#GSA本番
out <- GSA(data, data.cl,                 #GSAを実行し、結果をoutに格納
```



褐色脂肪「満腹 対 空腹」の発現変動に関連したKEGG Pathway遺伝子セットを**GSA法**で解析するための前処理が完了

KEGG_GLYCOLYSIS_GLUCONEOGENESIS	http://www	ACSS2	GCK	PGK2	PGK1	PDHB	PDHA1	PDHA2	PGM2
KEGG_CITRATE_CYCLE_TCA_CYCLE	http://www	IDH3B	DLST	PCK2	CS	PDHB	PCK1	PDHA1	LOC64
KEGG_PENTOSE_PHOSPHATE_PATHWAY	http://www	RPE	RPIA	PGM2	PGLS	PRPS2	FBP2	PFKM	PFKL
KEGG_PENTOSE_AND_GLUCURONATE_INT	http://www	UGT1A10	UGT1A8	RPE	UGT1A7	UGT1A6	UGT2B28	UGT1A5	CRYL1
KEGG_FRUCTOSE_AND_MANNOSE_METAE	http://www	MPI	PMM2	PMM1	FBP2	PFKM	GMDS	PFKFB4	PFKL
KEGG_GALACTOSE_METABOLISM	http://www	GCK	GALK1	GLB1	GALE	PMGALT1	PGM2	LALBA	PFKM
KEGG_ASCORBATE_AND_							UGT2B28	ALDH2	UGT1A
KEGG_FATTY_ACID_META							ACADL	ALDH2	ACADM
KEGG_STEROID_BIOSYNTHESIS	http://www	SOAT1	LSS	SQLE	EBP	CYP51A1	DHCR7	CYP27B1	DHCR2
KEGG_PRIMARY_BILE_ACID_BIOSYNTHESI	http://www	CYP46A1	SLC27A5	BAAT	CYP7B1	AKR1C4	HSD17B4	SCP2	AKR1D
KEGG_STEROID_HORMONE_BIOSYNTHESI	http://www	SRD5A3	AKR1C4	CYP3A5	HSD3B2	UGT2B28	HSD3B1	COMT	SULT2
KEGG_OXIDATIVE_PHOSPHORYLATION	http://www	ATP6V1G1	UQCR10	NDUFA5	NDUFA4	COX6CP3	PPA2	ATP5J2	NDUFS
KEGG_PURINE_METABOLISM	http://www	POLR2G	NT5C2	POLR2H	ENPP3	POLR2E	POLR2E	ENPP1	XDH

入力2：c2.cp.kegg.v4.0.symbols.gmt

```
in_f1 <- "data_mas_EN_symbol.txt"      #入力ファイル名を指定してin_f1に格納
in_f2 <- "c2.cp.kegg.v4.0.symbols.gmt" #入力ファイル名を指定してin_f2に格納
out_f1 <- "hoge2_G1.txt"               #出力ファイル名を指定してout_f1に格納
out_f2 <- "hoge2_G2.txt"               #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 <- 0.1                       #DEG検出時のfalse discovery rate (FD
param_posi <- c(1:4, 5:8)              #G1群およびG2群の位置情報を指定↓
↓
#必要なパッケージをロード↓
library(GSA)
↓
#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↓
gmt <- GSA.read.gmt(in_f2)             #in_f2で指定したファイルの読み込み (
data <- read.table(in_f1, header=TRUE, row.names=1, sep="¥t", quote="")#in_
rownames(data) <- toupper(rownames(data))#IDを大文字に変換している(gmtファイ
data.cl <- c(rep(1, param_G1), rep(2, param_G2))#G1群を1、G2群を2としたベク
data <- data[,param_posi]              #サブセットを抽出↓
colnames(data)                         #確認してるだけです↓
↓
#GSA本番↓
out <- GSA(data, data.cl,
        genesets=gmt$genesets,
        genenames=rownames(data),
        resp.type="Two class unpair
tmp <- GSA.listsets(out,
        geneset.names=gmt$geneset.n
        maxchar=max(nchar(gmt$genes
        FDRcut=param_FDR)
↓
#ファイルに保存↓
write.table(tmp$negative, out_f1, s
write.table(tmp$positive, out_f2, s
```

rcode_GSA.txt

G1群（満腹）で発現が上がった遺伝子セット（FDR < 0.1）

Gene_set	Gene_set_name	Score	p-value	FDR
9	KEGG_STEROID_BIOSYNTHESIS	-1.7542	0	0
39	KEGG_GLYCEROLIPID_METABOLISM	-0.6628	0	0
61	KEGG_PORPHYRIN_AND_CHLOROPHYLL_METABOLISM	-0.6396	0	0
62	KEGG_TERPENOID_BACKBONE_BIOSYNTHESIS	-2.3067	0	0
70	KEGG_BIOSYNTHESIS_OF_UNSATURATED_FATTY_ACIDS	-1.0358	0	0
169	KEGG_THYROID_CANCER	-0.5225	0	0

G2群（空腹）で発現が上がった遺伝子セット（FDR < 0.1）

Gene_set	Gene_set_name	Score	p-value	FDR
45	KEGG_LINOLEIC_ACID_METABOLISM	0.9554	0	0
67	KEGG_METABOLISM_OF_XENOBIOTICS_BY_CYTOCHROME_P450	0.62	0	0
68	KEGG_DRUG_METABOLISM_CYTOCHROME_P450	0.7861	0	0
110	KEGG_TGF_BETA_SIGNALING_PATHWAY	0.3628	0	0

その他情報

- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | について(last mod
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | GAGE (Luo 2009)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | GSA (Efron 2007)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | Category (Jiang 20
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | pcot2 (Kong 2006
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | SAFE (Barry 2005
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | globalt (Goemar
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | について(modified
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | Pathview (Luo 2013)(l
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GAGE (Luo 2009)(last
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | SPIA (Tarca 2009)(last
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GSA (Efron 2007)(last
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | sigPathway (Tian 2005

解析 | 機能解析 | パスウェイ(Pathway)解析 | について NEW

機能解析の実体は遺伝子セット解析です。パスウェイ(Pathway)解析は遺伝子セット解析の一つであり、パスウェイ解析を行うためのパッケージもいくつか出ています。Reviewや手法評価系論文も下のほうにリストアップしています。
2014年6月に調査した結果をリストアップします(R)。

- GSA: Efron and Tibshirani, Ann. Appl. Stat., 2007
- GSAのウェブページ
- SPIA: Tarca et al., Bioinformatics, 2009
- dCoxS: Cho et al., BMC Bioinformatics, 2009
- GAGE: Luo et al., BMC Bioinformatics, 2009
- PADOG: Tarca et al., BMC Bioinformatics, 2012
- Pathview: Luo et al., Bioinformatics, 2013

Pathviewはパスウェイマップまで色づけできるようです

2014年6月に調査した結果をリストアップします(R以外)。

- PLAGE(リンク切れ): Tomfohr et al., BMC Bioinformatics, 2005
- GSEA: Subramanian et al., PNAS, 2005
- GSEAのユーザーガイド
- ToppGene Suite(loginも要求なし; webtool): Chen et al., Nucleic Acids Res., 2009
- PINTA(loginも要求なし; webtool): Nitsch et al., Nucleic Acids Res., 2011
- FIDEA(loginも要求なし; webtool): D'Andrea et al., Nucleic Acids Res., 2013

2014年6月に調査した結果をリストアップします(Reviewや手法評価系論文)。

- Abatangelo et al., BMC Bioinformatics, 2009
- Khatri H, PLoS Comput Biol., 2012
- Maciejewski H, Brief Bioinform., 2013
- Tarca et al., PLoS One, 2013

Review系

2014年6月に調査した結果をリストアップします(遺伝子セットDB系)。

- MSigDB: Subramanian et al., PNAS, 2005
- hiPathDB: Yu et al., Nucleic Acids Res., 2012
- IPAVS: Sreenivasaiah et al., Nucleic Acids Res., 2012
- PAGED: Huang et al., BMC Bioinformatics, 2012
- IPAD: Zhang et al., BMC Bioinformatics, 2012

遺伝子セットDB系(MSigDB以外にも多数あり)

GSEAに代表される発現変動遺伝子セット解析は、基本的にGSEAの開発者らが作成した様々な遺伝子セット情報を収めたMolecular Signatures Database (MSigDB)からダウンロードした.gmt形式ファイルを読み込んで解析を行います。それゆえ、自分がどの遺伝子セットについて機能解析を行いたいのかを予め決めておく必要がありますが、パスウェイ解析の場合はc2のBioCarta, KEGG, Reactomeあたりを解析するのでしょう。gmt形式ファイルの基本的なダウンロード方法は以下の通りです:

1. Molecular Signatures Database (MSigDB)の「register」のページで登録し、遺伝子セットをダウンロード可能な状態にする。
2. Molecular Signatures Database (MSigDB)の「Download gene sets」の"Download"のところをクリックし、Loginページで登録したe-mail addressを入力。
3. これでMSigDBのダウンロードページに行けるので、目的に応じたgmtファイルをダウンロード(2014/06/02現在のバージョンは4.0)。
   「c2: curated gene sets」の「all canonical pathways, gene symbols」を解析したい場合:c2.cp.v4.0.symbols.gmt
   「c2: curated gene sets」の「BioCarta gene sets, gene symbols」を解析したい場合:c2.cp.biocarta.v4.0.symbols.gmt
   「c2: curated gene sets」の「KEGG gene sets, gene symbols」を解析したい場合:c2.cp.kegg.v4.0.symbols.gmt
   「c2: curated gene sets」の「Reactome gene sets, gene symbols」を解析したい場合:c2.cp.reactome.v4.0.symbols.gmt

トップページへ

- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | globaltest (Goeman_2004)(last modified 2014/06/01)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | について(last modified 2014/06/02) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | Pathview (Luo_2013)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | GAGE (Luo_2009)(last modified 2014/06/01) NEW
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | SPIA (Tarca_2009)(last modified 2014/06/01) NEW
- 解析
- 解析
- ...

解析 | 機能解析 | パスウェイ(Pathway)解析 | Pathview (Luo_2013

NEW

Pathviewパッケージを用い
「ファイル」−「ディレクトリ(

1. ...の場合:

```
library(Pathview)
```

- Pathview: Luo et al.

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home | Install | Hel

Home » Bioconductor 2.14 » Software Packages » pathview

pathview

a tool set for pathway based data integration

Bioconductor version: Release (2.14)

Pathview is a tool set for pathway based data integration and v
variety of biological data on relevant pathway graphs. All users
the target pathway. Pathview automatically downloads the pat
maps user data to the pathway, and render pathway graph wit
also seamlessly integrates with pathway and gene set (enrichm
automated analysis.

Author: Weijun Luo

Maintainer: Weijun Luo <luo_weijun at yahoo.com>

To install this package, start R and enter:

```
source("http://bioconductor.org/biocLite.R")
biocLite("pathview")
```

Citation (from within R, enter `citation("pathview")`):

Luo, Weijun, Brouwer and Cory (2013). "Pathview: an R/Bio
integration and visualization." *Bioinformatics*, **29**(14), pp. 18
http://dx.doi.org/10.1093/bioinformatics/btt285.

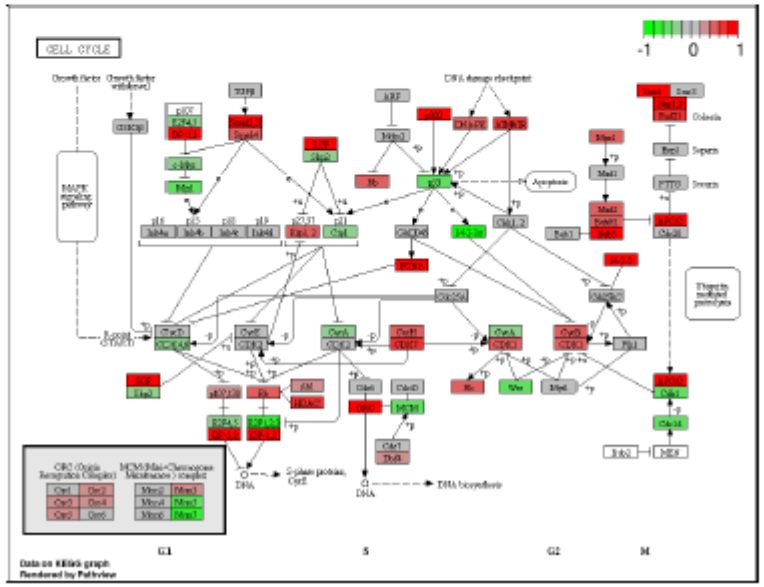
Documentation

PDF	R Script	Pathview: pathway based data integratio
PDF		Reference Manual
Text		NEWS

get more controls over the nodes and edge attributes and look. Importantly, the graph is a vector image in PDF format in your working directory.

```
> pv.out <- pathview(gene.data = gse16873.d[, 1], pathway.id = demo.paths$sel.paths[1],
+                    species = "hsa", out.suffix = "gse16873", kegg.native = F,
+                    sign.pos = demo.paths$spos[1])
> #pv.out remains the same
> dim(pv.out$plot.data.gene)
```

4



(a)

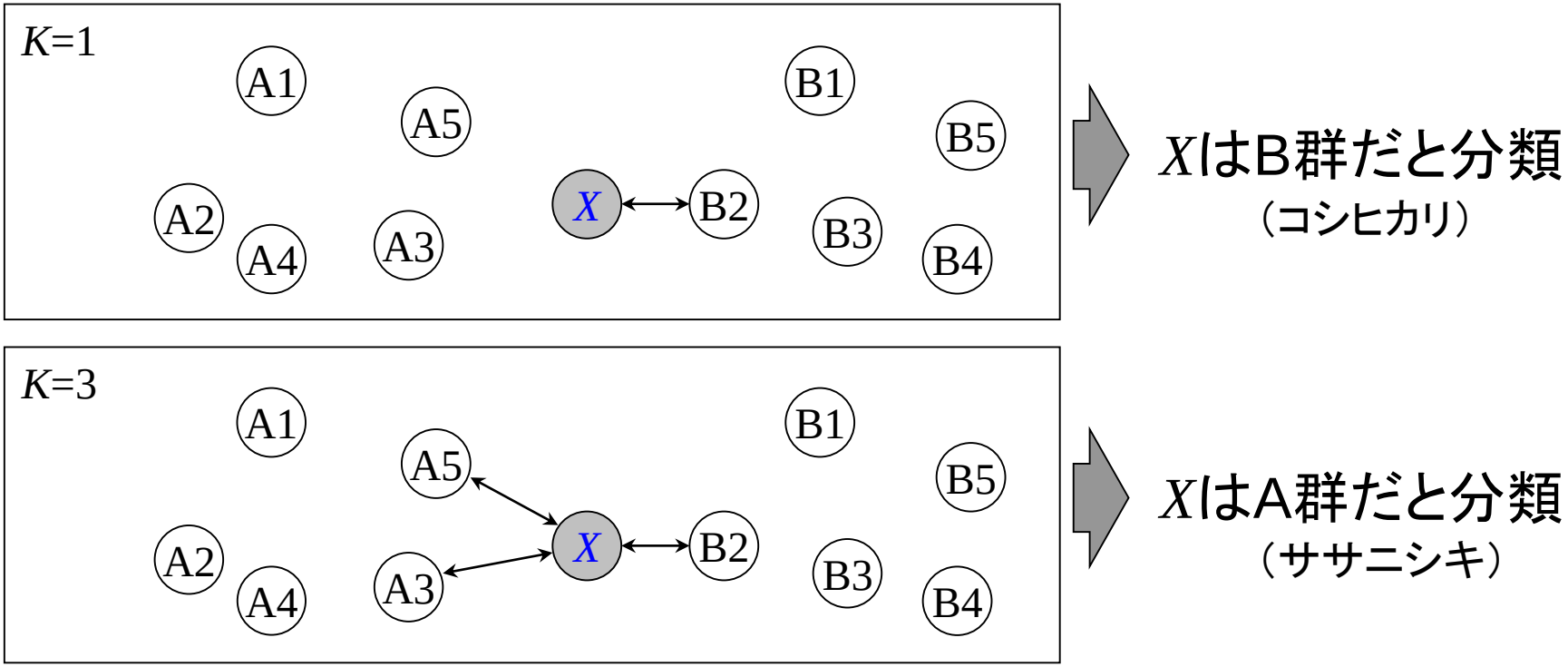
Pathviewはパスウェイマップまで色づけできるようです

Contents（第4回）

- デザイン行列の意味を理解（教科書p173-182）
  - limmaパッケージを用いた2群間比較のおさらい
  - limmaパッケージを用いた3群間比較（複製あり）
- 複製なし多群間比較（教科書p182-188）
  - limmaパッケージを用いた3群間比較（複製なし）
  - TCCパッケージ中のROKU法を用いた特異的発現遺伝子検出
- 機能解析（遺伝子セット解析）
  - 基本的な考え方
  - 前処理
    - MSigDBからの遺伝子セット情報（GMT形式ファイル）取得
    - ID変換（probe ID → gene symbol）
  - GSAパッケージを用いたパスウェイ解析
  - その他
- **分類**

K-Nearest Neighbor (K-NN) 法

- 未知サンプルXからの距離がもっとも近いK個のサンプルのうち、所属するクラスが最も多いクラスに分類



細胞内局在予測プログラムPSORTでも利用されている

距離の定義

- 目的：*x*と*y*の発現パターンの距離*D*を定義したい
  - 似ていれば*D*が0になるようにしたい

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

xとyの発現パターンが酷似	$\rightarrow r \approx 1$	$\rightarrow D = 1\text{-}r = 0$
xとyの発現パターンがばらばら	$\rightarrow r \approx 0$	$\rightarrow D = 1\text{-}r = 1$
xとyの発現パターンがほぼ正反対	$\rightarrow r \approx -1$	$\rightarrow D = 1\text{-}r = 2$

	X	B2
i	***x***	***y***
1	x_1	y_1
2	x_2	y_2
3	x_3	y_3
4	x_4	y_4
5	x_5	y_5
…	…	…
n	x_n	y_n

全遺伝子のデータではなく、二群間で発現の異なる遺伝子セット（～数百個程度）のみを用いて（Feature Selection）、未知サンプル*X*と既知サンプルの距離*D*を計算する