

①持ち込みPCのヒトは後ろにある
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



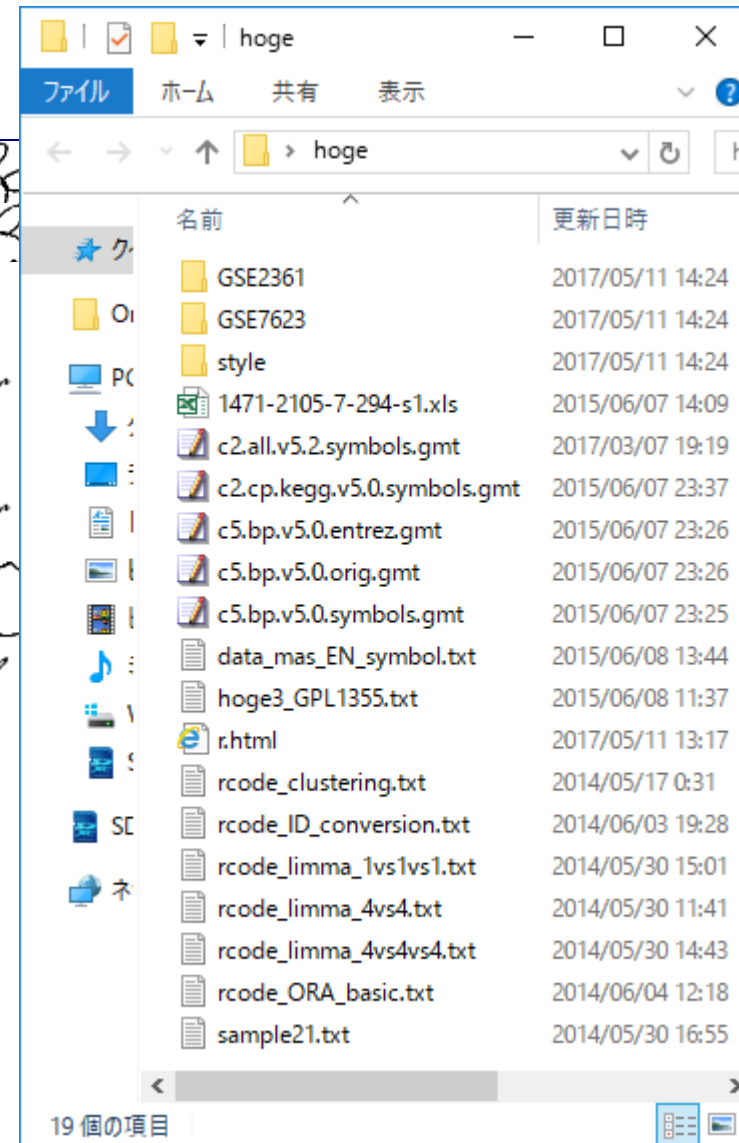
機能ゲノム学 第4回

大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究プログラム

門田幸二(かどた こうじ)

kadota@iu.a.u-tokyo.ac.jp

<http://www.iu.a.u-tokyo.ac.jp/~kadota/>



①は教科書通りなので飛ばす。実験デザイン行列、3群間比較、反復なしの場合などは、平成28年度NGSハンズオン講習会(2016.07.22の資料)にも情報があります

Contents



- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

アレイデータ

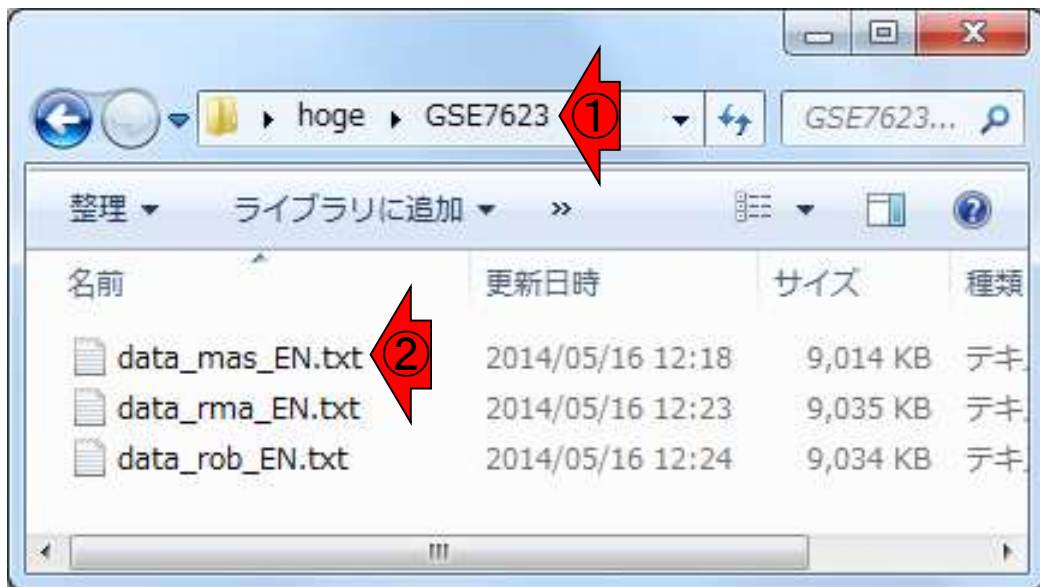
hogeフォルダ中に3つの前処理法の実行結果ファイルがあります。MAS5 (data_mas_EN.txt)、RMA (data_rma_EN.txt)、RMX (data_rob_EN.txt)

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 vs. 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 vs. 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 vs. 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 vs. control diet (Control) 5サンプル

アレイデータ

①「hoge - GSE7623」。GSE7623 (Nakai et al., 2008)の対数変換後のデータ

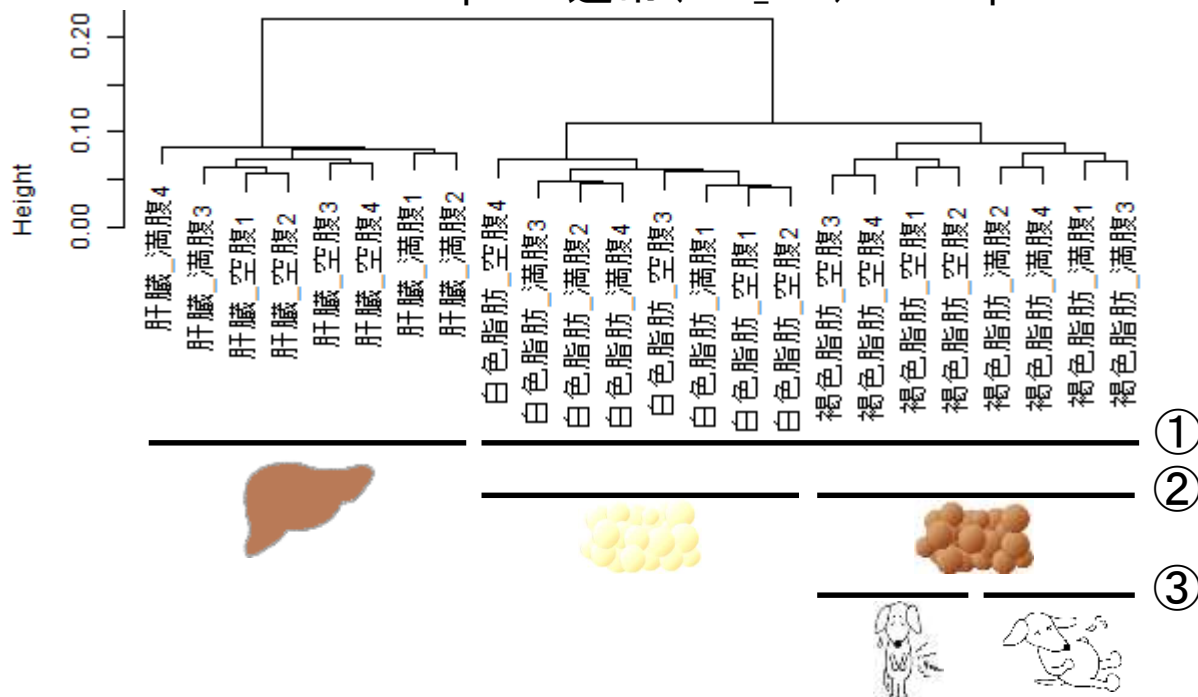


③ data_mas_EN.txt

	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_fas4
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	12.049
4	1367454_at	11.389902	11.160757	11.145987	11.212088	11.540652	11.308877	11.49885	11.409
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	10.269
8	1367458_at	9.9253387	10.244544	9.9720008	9.9576072	8.702884	9.9578792	9.2134367	9.499

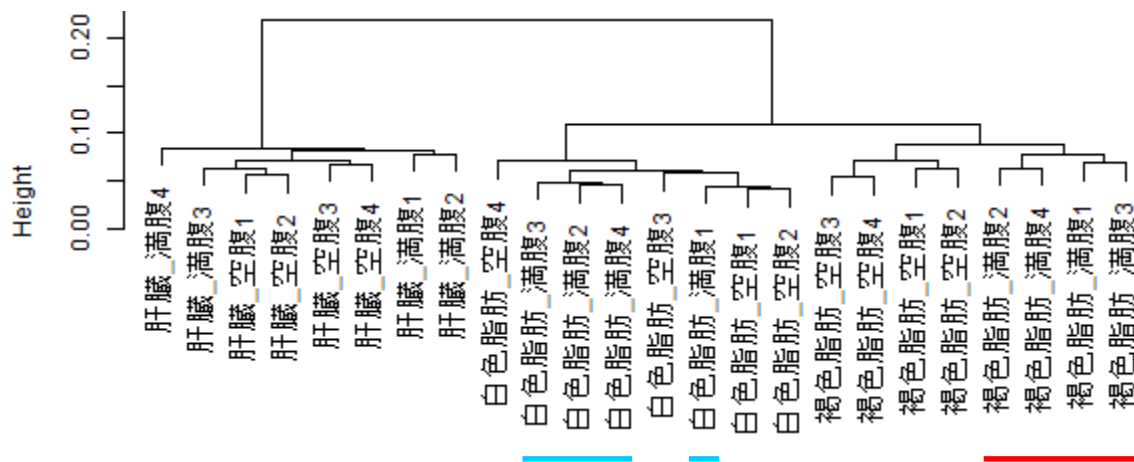
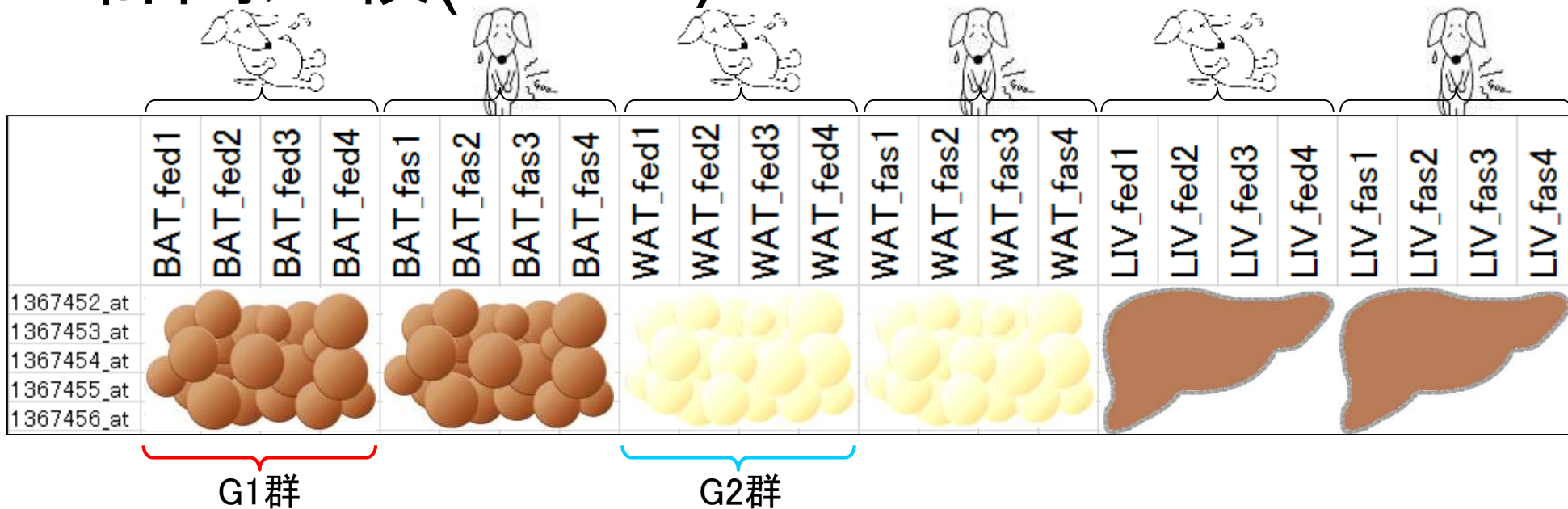
2群間比較(limma)

- Nakai et al., Biosci Biotechnol Biochem., 72: 139–148, 2008
 - GSE7623、GPL1355 (Affymetrix Rat Genome 230 2.0 Array)、31,099 probesets
 - Rat 24 samples: 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 samples、Liver (肝臓; LIV) 8 samples
 - BAT 8 samples: 通常 (BAT_fed) 4 samples vs. 24時間絶食 (BAT_fas) 4 samples
 - WAT 8 samples: 通常 (WAT_fed) 4 samples vs. 24時間絶食 (WAT_fas) 4 samples
 - LIV 8 samples: 通常 (LIV_fed) 4 samples vs. 24時間絶食 (LIV_fas) 4 samples



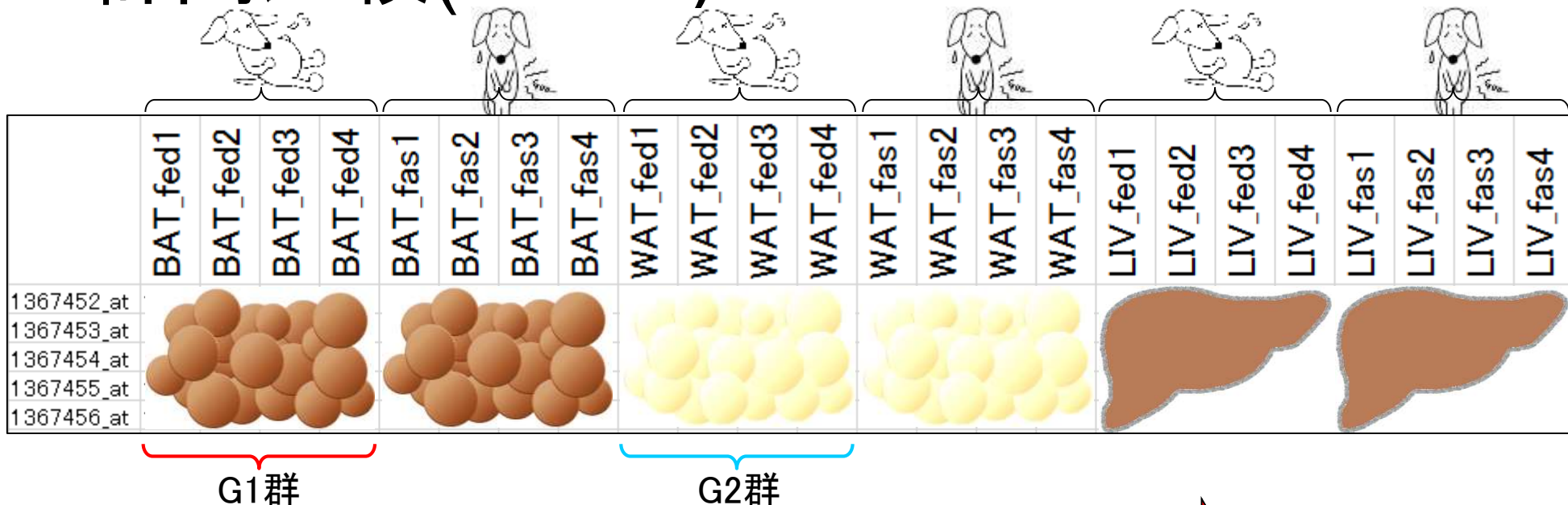
rancode_clustering_png.txtの実行結果。
 ①肝臓と脂肪間で大きく2つのクラスターに分かれている。
 ②脂肪の中でも白色脂肪と褐色脂肪に分かれている。
 ③褐色脂肪は空腹(24時間絶食)と満腹(通常)できれいに分かれている。

2群間比較(limma)



① hogeフォルダ中のrcode_limma_4vs4.txt。サブセット抽出のための情報は、②で与えている

2群間比較(limma)



```
#####↓
### 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↓
```



① rcode_limma_4vs4.txt

②

2群間比較(limma)

rcode_limma_4vs4.txt

```
#####↓
### 4 BAT_fed samples vs. 4 WAT_fed samples ###↓
#####↓
```

```
in_f <- "data_mas_EN.txt"
out_f1 <- "hogel.txt"
out_f2 <- "hogel.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作
data <- read.table(in_f, header=TRUE, r
data.cl <- c(rep(1, param_G1), rep(2, p
data <- data[,param_posi]
colnames(data)
```

```
↓
#本番↓
#design <- model.matrix(~ as.factor(dat
design <- model.matrix(~data.cl)
fit <- lmFit(data, design)
```

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/GSE7623"
> in_f <- "data_mas_EN.txt" #入力ファイル$
> out_f1 <- "hogel.txt" #出力ファイル$
> out_f2 <- "hogel.png" #出力ファイル$
> 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検出時の$
> 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、そしてサ$
> data <- read.table(in_f, header=TRUE, row.names=1, $
> data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #G1$
> 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
> |
```


2群間比較(limma)

designオブジェクトが(実験)デザイン行列です。この行列の①2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

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
q.value <- p.adjust(p.value, method="BH") #p
ranking <- rank(p.value) #p
sum(q.value < param_FDR) #F
mean_G1 <- apply(as.matrix(data[,data.cl=
mean_G2 <- apply(as.matrix(data[,data.cl=
M <- mean_G2 - mean_G1 #M
A <- (mean_G1 + mean_G2)/2 #M
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, M, A,
write.table(tmp, out_f1, sep="¥t", append

```

```

R Console
> design <- model.matrix(~data.cl)
> 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
> data.cl
[1] 1 1 1 1 2 2 2 2
> as.factor(data.cl)
[1] 1 1 1 1 2 2 2 2
Levels: 1 2
> |

```



2群間比較(limma)

①赤下線のout\$p.valueで表される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②列数は、デザイン行列designの列数と同じ。③out\$p.valueの2列目の情報が解析結果に相当

```

#本番↓
#design <- model.matrix(~ as.factor(data.cl))#デザイン行列
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
mean_G1 <- apply(as.matrix(data[,data.cl=
mean_G2 <- apply(as.matrix(data[,data.cl=
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, A,
write.table(tmp, out_f1, sep="¥t", append

```

```

rcode_limma_4vs4.t
#デザイン行列を作成した結果をdesignに格納↓
#モデル構築(ばらつきの程度を見積もっている)↓
#検定(経験ベイズ)↓
#p値をp.valueに格納↓
#q値をq.valueに格納↓
#p.valueでランキングした結果をrankingに格納↓
#FDR

```

```

R Console
> fit <- lmFit(data, design) #モデル構築($
> out <- eBayes(fit) #検定(経験ベ$
> p.value <- out$p.value[,ncol(design)] #p値をp.valu$
> head(out$p.value, n=5)
              (Intercept)      data.cl
1367452_at 9.548174e-11 0.60594552
1367453_at 3.516879e-11 0.09645379
1367454_at 1.402195e-10 0.16185158
1367455_at 2.986084e-11 0.37498101
1367456_at 7.686525e-12 0.10585764
> dim(out$p.value)
[1] 31099      2
> head(out$p.value[, 2], n=4)
1367452_at 1367453_at 1367454_at 1367455_at
0.60594552 0.09645379 0.16185158 0.37498101
> |

```



2群間比較(limma)

① ncol(design)と一般式で表すことで、常にデザイン行列の一番右側の列情報を抽出するようにしている。ここまでの話は、様々な記載方法への慣れと次の多群間比較のイントロ

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) #
p.value <- out$p.value[,ncol(design)] #p
q.value <- p.adjust(p.value, method="BH") #p
ranking <- rank(p.value) #p
sum(q.value < param_FDR) #F
mean_G1 <- apply(as.matrix(data[,data.cl=
mean_G2 <- apply(as.matrix(data[,data.cl=
M <- mean_G2 - mean_G1 #M
A <- (mean_G1 + mean_G2)/2 #M
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, M, A,
write.table(tmp, out_f1, sep="¥t", append
```

```
R Console
> head(out$p.value[, 2], n=4)
1367452_at 1367453_at 1367454_at 1367455_at
0.60594552 0.09645379 0.16185158 0.37498101
> 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
> ncol(design)
[1] 2
> |
```

Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

3群間比較(limma)

(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

①が教科書に沿った内容。実験デザイン行列だとかコントラスト行列だとか様々な記述形式があります。②「応用1」の例題を一通り行くと感覚がつかめると思います。赤枠内では、分散分析(ANOVA)的な「比較するどこかの群間で発現変動している遺伝子」の同定について記述しています

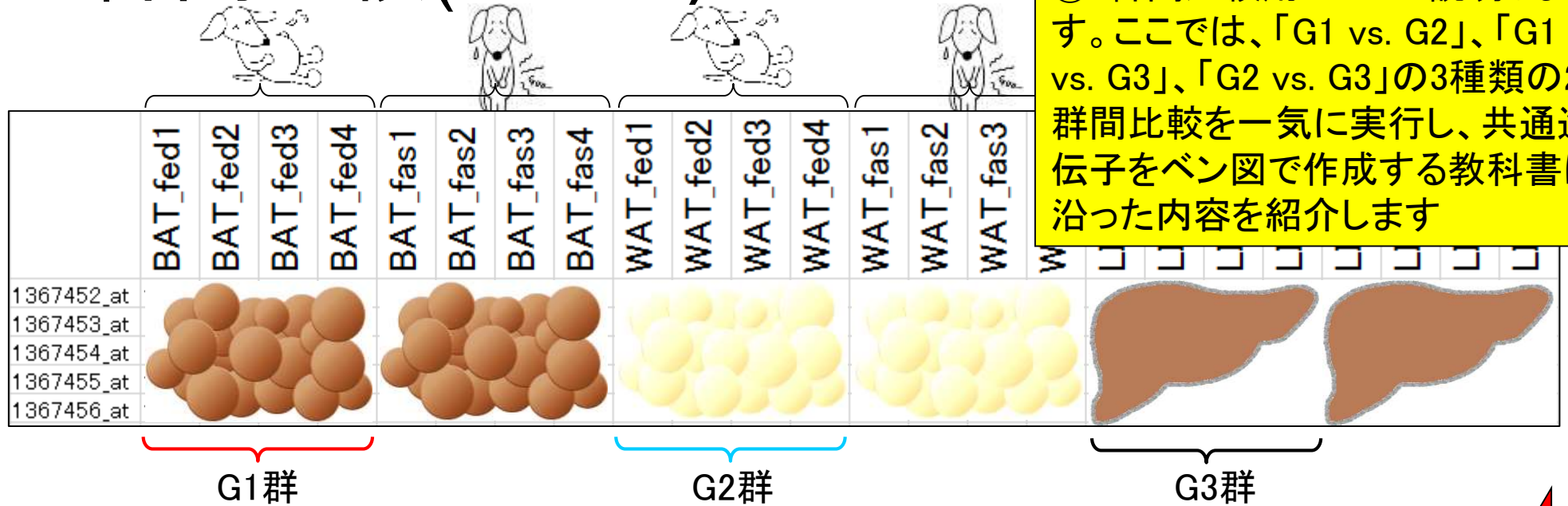
What's new?

- 「解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah 2017\)](#)」を追加しました。比較するだけでなく、ネットワーク図やグループごとの最有力遺伝子(most influential genes or hub genes)も
- 「解析 | 発現変動 | 2群間 | 対応なし | [Rank products \(Breitling 2004\)](#)」の例題3の下のほうで書いてい算術平均値)に関する注釈を追加しました。(2016/08/30)
- 門田幸二 著 [シリーズ Useful R 第7巻](#) (al., 2006)、WAD法 (Kadota et al., 2006)、「書籍 | トランスクリプトーム解析 | ...」
- お知らせは主に [\(Rで\)塩基配列解析](#) での [参考資料\(講義、講習会、本など\)](#) から

- 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)
- 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 2014/06/02)
- 解析 | 発現変動 | 2群間 | 対応あり | [SAM \(Tusher 2001\)+WADの重みづけ](#) (last modified 2013/06/02)
- 解析 | 発現変動 | 2群間 | 対応あり | [時系列 | について](#) (last modified 2013/6/8)
- 解析 | 発現変動 | 2群間 | 対応あり | 時系列 | [maSigPro \(Conesa 2006\)](#) (last modified 2013/6/2)
- 解析 | 発現変動 | 3群間 | 対応なし | [について](#) (last modified 2015/01/16)
- 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 | 基礎 | [limma \(Ritchie 2015\)](#) ② (last modified 2015/06/08)
- 解析 | 発現変動 | 3群間 | 対応なし | 応用1 | [limma \(Ritchie 2015\)](#) (last modified 2015/06/08)
- 解析 | 発現変動 | 3群間 | 対応なし | 応用2 | [limma \(Smyth 2004\)](#) (last modified 2015/06/08)
- 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 2014/05/30)

3群間比較(limma)

デザイン行列を理解しておかないと、多群間比較時に苦労します。
 ①3群間比較用コードで説明します。ここでは、「G1 vs. G2」、「G1 vs. G3」、「G2 vs. G3」の3種類の2群間比較を一気に実行し、共通遺伝子をベン図で作成する教科書に沿った内容を紹介します



```
#####↓
### 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↓
```

rcode_limma_4vs4vs4.txt ①

3群間比較(limma)

```
#####↓
### 4 BAT_fed samples vs. 4 WAT_fed samples vs. 4 LIV_fed samples ###↓
#####↓
```

```
in_f <- "data_mas_EN.txt"
out_f1 <- "hoge1.txt"
out_f2 <- "hoge1.png"
param_G1 <- 4
param_G2 <- 4
param_G3 <- 4
param_posi <- c(1:4, 9:12, 17:20)
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, row.names=1, as.is=TRUE)
data.cl <- c(rep("G1", param_G1), rep("G2", param_G2), rep("G3", param_G3))
data <- data[,param_posi]
colnames(data)
```

```
↓
#本番↓
#design <- model.matrix(~ 0 + as.factor(data.cl))
design <- model.matrix(~ 0 + data.cl) #デザインマトリクス
colnames(design) <- levels(as.factor(data.cl))
```

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/GSE7623"
> list.files(pattern="mas_EN")
[1] "data_mas_EN.txt"
> in_f <- "data_mas_EN.txt"
> out_f1 <- "hoge1.txt"
> param_G1 <- 4
> param_G2 <- 4
> param_G3 <- 4
> param_posi <- c(1:4, 9:12, 17:20)
> 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、そしてサブセットの作成
> data <- read.table(in_f, header=TRUE, row.names=1, as.is=TRUE)
> data.cl <- c(rep("G1", param_G1), rep("G2", param_G2), rep("G3", param_G3))
> data <- data[,param_posi]
> colnames(data)
[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"
> |
```

3群間比較(limma)

実験デザイン行列には様々が記述の仕方があって難解ですが、慣れです。例えば①model.matrix実行直後のdesignは、②全部で12列の解析データのどこがどの群かを1で指し示している

```
#本番↓
#design <- model.matrix(~ 0 + as.factor(data.c1))
design <- model.matrix(~ 0 + data.c1) #デザイン行列
colnames(design) <- levels(as.factor(data.c1))
fit <- lmFit(data, design) #モデル
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) #モデル
out <- eBayes(fit2) #検定
p.value <- out$p.value #p値
q.value <- apply(p.value, MARGIN=2, p.adjust="fdr")
ranking <- apply(p.value, MARGIN=2, rank="na")
```

```
R Console
> design <- model.matrix(~ 0 + data.c1)
> design
      data.c1G1 data.c1G2 data.c1G3
 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.c1
[1] "contr.treatment"

> data.c1
[1] "G1" "G1" "G1" "G1" "G2" "G2" "G2" "G2" "G3" "G3"
[11] "G3" "G3"
> |
```



3群間比較(limma)

```
#本番↓
#design <- model.matrix(~ 0 + as.factor(data.c1))
design <- model.matrix(~ 0 + data.cl) #データ
colnames(design) <- levels(as.factor(data.c1))
fit <- lmFit(data, design) #モデル
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) #モデル
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
> 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"
① > data.cl
[1] "G1" "G1" "G1" "G1" "G2" "G2" "G2" "G2" "G3" "G3"
[11] "G3" "G3"
> |
```

3群間比較(limma)

①levels関数を用いてグループラベル情報を利用し、②デザイン行列designの列名情報を変更して取扱いやすくしています。③かなりすっきりしました

```
#本番↓
#design <- model.matrix(~ 0 + as.factor(data.cl))
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))
fit <- lmFit(data, design) #モデルをfitに格納
contrast <- makeContrasts( #比較行列を作成
  G1vsG2 = G1 - G2, #比較1
  G1vsG3 = G1 - G3, #比較2
  G2vsG3 = G2 - G3, #比較3
  levels = design) #比較3
fit2 <- contrasts.fit(fit, contrast) #モデルをfit2に格納
out <- eBayes(fit2) #検定
p.value <- out$p.value #p値
q.value <- apply(p.value, MARGIN=2, p.adjust="fdr") #q値
ranking <- apply(p.value, MARGIN=2, rank="na") #p値
```

rcode_limma_4vs4vs4.t

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"
```



3群間比較(limma)

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、この部分の指定時にややこしいことになる。①ここでは3種類の2群間比較を行うようにしている。ここで作成しているのは、コントラスト行列というもの。「比較したいものを作成した行列」という位置づけ

```

#本番↓
#design <- model.matrix(~ 0 + as.factor(data.cl))#デザイン行列を作成した
design <- model.matrix(~ 0 + data.cl) #デザイン行列を作成した結果をdes
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
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
q.value <- apply(p.value, MARGIN=2, p.adjust)
ranking <- apply(p.value, MARGIN=2, rank)

```



```

R Console
> fit <- lmFit(data, design)
> contrast <- makeContrasts(
+   G1vsG2 = G1 - G2,
+   G1vsG3 = G1 - G3,
+   G2vsG3 = G2 - G3,
+   levels = design)
#モデル構築($
#比較したい2$
#比較したい2$
#比較したい2$
#比較したい2$
#比較したい2$
#比較したい2$
> contrast
  Contrasts
Levels G1vsG2 G1vsG3 G2vsG3
  G1      1      1      0
  G2     -1      0      1
  G3      0     -1     -1
> |

```

3群間比較(limma)

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行列である①out\$p.valueが得られる

```
#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
p.value <- out$p.value #p値
q.value <- apply(p.value, MARGIN=2, p.adjust)
ranking <- apply(p.value, MARGIN=2, rank)#
```

```
R Console
> fit2 <- contrasts.fit(fit, contrast) #モデル構築
> out <- eBayes(fit2) #検定(経験バ$
① > head(out$p.value, n=5)
      Contrasts
      G1vsG2    G1vsG3    G2vsG3
1367452_at 0.57066249 0.005780086 0.01612904
1367453_at 0.07035758 0.010713438 0.30569612
1367454_at 0.22337752 0.001491958 0.01373208
1367455_at 0.34340818 0.004630791 0.02635020
1367456_at 0.09121355 0.887708741 0.07149528
① > dim(out$p.value)
[1] 31099    3
> |
```


3群間比較(limma)

```

#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↓

```

```

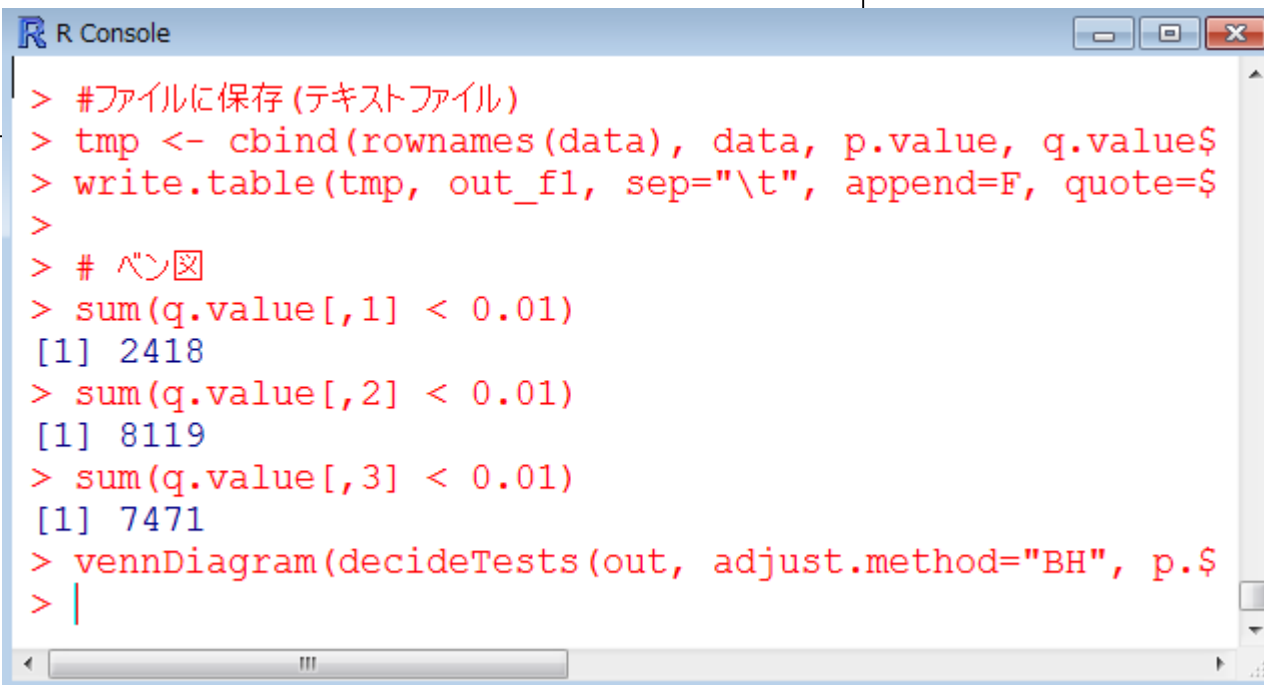
R Console
> p.value <- out$p.value #p値をp.valu$
> q.value <- apply(p.value, MARGIN=2, p.adjust, metho$
> head(q.value, n=4)
      Contrasts
      G1vsG2      G1vsG3      G2vsG3
1367452_at 0.7409725 0.019218956 0.04689295
1367453_at 0.2034828 0.031828163 0.45396064
1367454_at 0.4238707 0.006279388 0.04125726
1367455_at 0.5520626 0.016024589 0.06949329
> |

```

3群間比較(limma)

FDR 1%という閾値を満たす遺伝子数は、G1 vs. G2が2,418個、G1 vs. G3が8,119個、そしてG2 vs. G3が7,471個という結果

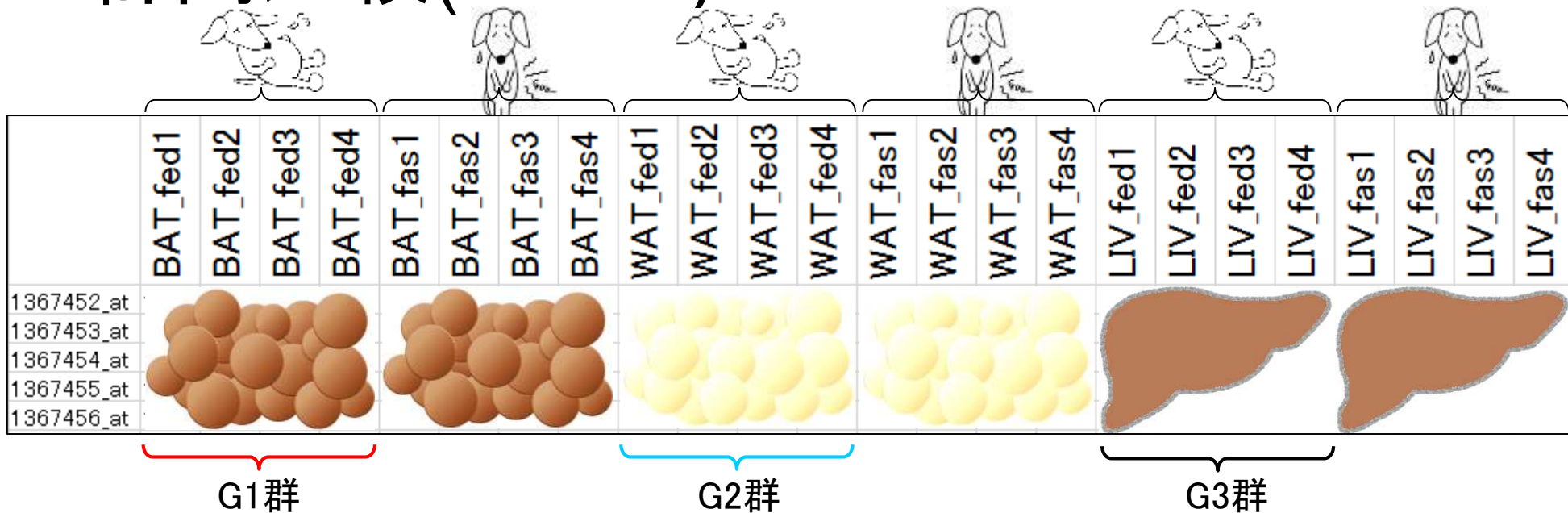
```
p.value <- out$p.value #p値をp.valueに格納↓
q.value <- apply(p.value, MARGIN=2, p.adjust, method="BH")#q値をq.valueに格納↓
ranking <- apply(p.value, MARGIN=2, rank)#p.valueでランキングした結果をrankingに格納↓
↓
#ファイルに保存(テキストファイル)↓
tmp <- cbind(rownames(data), data, p.value, q.value, ranking)#入力データの右側にDEG検出結果
を結合したものをtmpに格納↓
write.table(tmp, out_f1, sep="\t", append=F, quote=F, row.names=F)#tmpの中身を指定したファ
イル名で保存↓
↓
# ベン図↓
sum(q.value[,1] < 0.01)↓
sum(q.value[,2] < 0.01)↓
sum(q.value[,3] < 0.01)↓
vennDiagram(chooseTests(out, adjust.method="BH", p.value=q.value))
```



```
R Console
> #ファイルに保存(テキストファイル)
> tmp <- cbind(rownames(data), data, p.value, q.value$
> write.table(tmp, out_f1, sep="\t", append=F, quote=$
>
> # ベン図
> 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(chooseTests(out, adjust.method="BH", p.$
> |
```

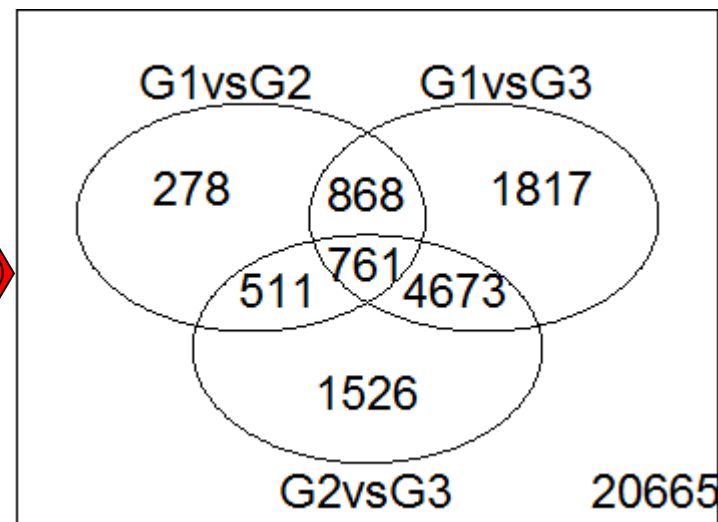
G1vsG2のDEG数が他に比べて少ないので妥当。①ベン図

3群間比較(limma)



```

R Console
> # ベン図
> 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(chooseTests(out, adj$
> |
    
```



Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

反復なし3群間比較(limma)

```
#####↓
### 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 #G3群のサンプル数を指定↓
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
↓
#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
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
↓
#本番↓
#design <- model.matrix(~ 0 + as.factor(data.c
design <- model.matrix(~ 0 + data.cl) #デザイ
colnames(design) <- levels(as.factor(data.cl))
fit <- lmFit(data, design) #モデル
contrast <- makeContrasts( #比較し
+ G1vsG2 = G1 - G2, #比較し
+ G1vsG3 = G1 - G3, #比較し
+ G2vsG3 = G2 - G3, #比較し
+ levels = design) #比較し
fit2 <- contrasts.fit(fit, contrast) #モデル
out <- eBayes(fit2) #検定(経
p.value <- out$p.value #p値を
q.value <- apply(p.value, MARGIN=2, p.adjust,
ranking <- apply(p.value, MARGIN=2, rank)#p.v
↓
```

```
R Console
> #design <- model.matrix(~ 0 + as.factor(data.cl)) #デザイ$
> design <- model.matrix(~ 0 + data.cl) #デザイン行列を作$
> colnames(design) <- levels(as.factor(data.cl)) #デザイン$
> fit <- lmFit(data, design) #モデル構築 (ばら$
> contrast <- makeContrasts( #比較したい2群の$
+ G1vsG2 = G1 - G2, #比較したい2群の$
+ G1vsG3 = G1 - G3, #比較したい2群の$
+ G2vsG3 = G2 - G3, #比較したい2群の$
+ levels = design) #比較したい2群の$
> fit2 <- contrasts.fit(fit, contrast) #モデル構築
> out <- eBayes(fit2) #検定 (経験ベイズ)
Error in ebayes(fit = fit, proportion = proportion, stdev.$
No residual degrees of freedom in linear model fits
> |
```


Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

反復なし多群間比較(TCC)

ROKU入出力のイメージ。出力は入力の対応する位置に「特異的組織でない部分には0、特異的高発現が1、特異的低発現が-1」と返す。①黒枠内の遺伝子(gene2, 7, 8)は組織特異的遺伝子ではない。②ROKU出力結果では、対応するどの組織も0になっていて妥当

入力

	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

出力

	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

①出力の右側は全体的な組織特異度の高さでランキングした結果を返す。modH列の値が0に近いほど、組織特異度が高いことを表す

反復なし多群間比較(TCC)

入力

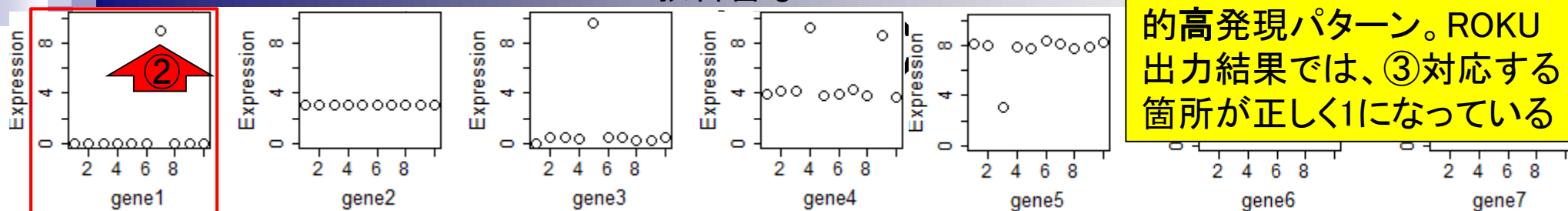
	tissue1	tissue2	tissue3	tissue4	tissue5	tissue6	tissue7	tissue8	tissue9	tissue10
gene1	0.00	0.00	0.00	0.00	0.00	0.00	9.00	0.00	0.00	0.00
gene2	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
gene3	0.02	0.41	0.41	0.38	9.60	0.49	0.44	0.16	0.21	0.52
gene4	3.95	4.12	4.20	9.20	3.84	3.97	4.23	3.80	8.60	3.64
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



出力

	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

①gene1は、②tissue7特異的高発現パターン。ROKU出力結果では、③対応する箇所が正しく1になっている



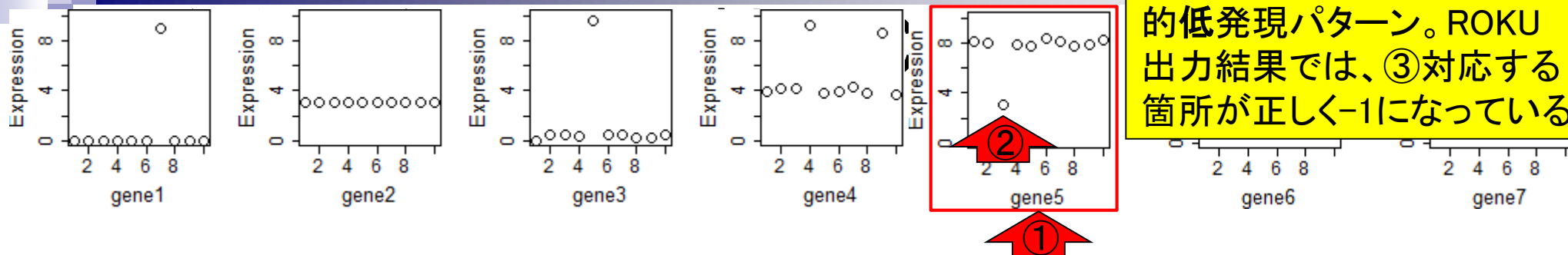
①
入力

	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

出力

	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

①gene5は、②tissue3特異的低発現パターン。ROKU出力結果では、③対応する箇所が正しく-1になっている



入力

	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	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	2.00	3.50	2.50	3.65	0.30	3.10	3.63	

出力

	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	0	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

入出力のイメージは、②例題1実行結果。modH列の値は、「データ変換後のエントロピー値」に相当

反復なし多群間比較(TCC)

(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

What's

- 「解析する」
- 「解析算術」
- 「門田 al., 2011」
- 「書籍」
- 「お知」
- 「の参考資料」

- ・解析 | 発現変動 | 2群間 | 対応あり | 時系列 | [maSigPro \(Conesa 2006\)](#) (last modified 2013/6/2)
- ・解析 | 発現変動 | 3群間 | 対応なし | [について](#) (last modified 2015/01/16)
- ・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 2014/05/30)
- ・解析 | 発現変動 | 多群間 | [Sprent's non-parametric method \(Ge 2005\)](#) (last modified 2009/07/31)
- ・解析 | 発現変動 | 多群間 | [Schug's H](#)
- ・解析 | 発現変動 | 多群間 | [Schug's H](#)

解析 | 発現変動 | 多群間 | ROKU (Kadota_2006)

TCCパッケージで提供している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となります。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」ということとなります。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と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、以下をコピー

② 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

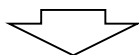
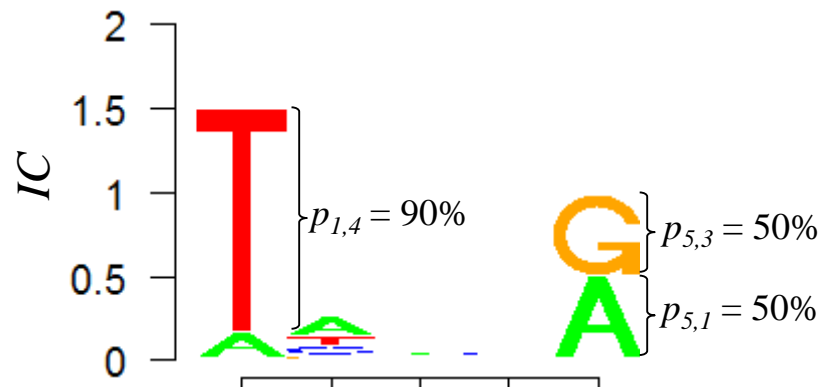
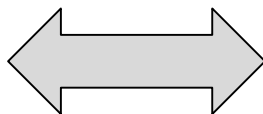
#入力ファイルの読み込み
data <- read.table(in_f, header=TRUE, row.names=1, sep="\t", quote="") #in_fで指定したフ
```


① 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だけです。②赤枠内がエントロピー

Sequence logos

■ position i の情報量 $IC_i = \frac{\log_2(N) - H(x_i)}{2}$

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

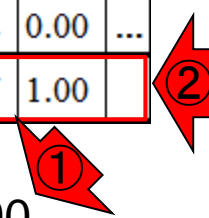


x_{ij}	1	2	3	4	5	...
Aの数 ($j=1$)	1	5	3	2	5	...
Cの数 ($j=2$)	0	2	3	3	0	...
Gの数 ($j=3$)	0	1	2	2	5	...
Tの数 ($j=4$)	9	2	2	3	0	...
$\sum_j x_{ij}$	10	10	10	10	10	

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	...
\sum_j	1.0	1.0	1.0	1.0	1.0	

IC	1.53	0.24	0.03	0.03	1.00	...
----	------	------	------	------	------	-----

$-p_{ij} \log_2(p_{ij})$	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	...
$H = \sum_j$	0.47	1.76	1.97	1.97	1.00	



組織特異的発現遺伝子検出にエントロピーを最初に応用したのは①Schug et al. 2005。ROKU法はその改良版という位置づけ

エントロピー

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

<i>j</i>	<i>x_{ij}</i>	<i>i</i>				
		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	
$\sum_j x_{ij}$	10	28	48	38	74	

<i>p_{ij}</i>	<i>i</i>					
	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	
\sum_j	1.00	1.00	1.00	1.00	1.00	

<i>-p_{ij} log₂(p_{ij})</i>	<i>i</i>					
	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	
\sum_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: $0 \leq H \leq \log_2 N \rightarrow 0 \leq H \leq 3$



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)、...

[BMC Bioinformatics](#). 2006 Jun 12;7:294.

ROKU: a novel method for identification of tissue-specific genes.

[Kadota K](#)¹, [Ye J](#), [Nakai Y](#), [Terada T](#), [Shimizu K](#).

⊕ Author information

Abstract

BACKGROUND: One of the important goals of microarray analysis is to identify genes whose expression is considerably higher or lower in some tissues. Identifying such tissue-specific genes is a challenging task.

RESULTS: We describe a method, ROKU, which identifies tissue-specific genes from many tissues and thousands of genes. ROKU ranks genes by their Shannon entropy and detects tissues specific to each gene. We evaluated the capacity for the detection of various tissues by ROKU. We observed that ROKU was superior to a conventional method according to overall tissue specificity and to detect objective tissues.

CONCLUSION: ROKU is useful for the detection of tissue-specific genes. It is also directly applicable to the selection of diagnostic markers.

Analysis of real data

To further investigate the validity of our method (ROKU), we applied the method to a public gene expression matrix consisting of 36 normal human tissues and 22,283 probesets [5]. Briefly, ROKU (1) processes each probeset expression vector and makes a processed vector x' , (2) calculates the entropy $H(x')$, and (3) assigns specific tissues to each probeset whose observations are detected to be 'outliers' (see Methods). We compared the performance of ROKU to that of Schug's method, which directly uses the original/non-processed vector x for measuring the entropy $H(x)$ [4]. The two entropy scores ($H(x')$ and $H(x)$) for all probesets are available in the additional file [see 1].

Additional file 1. Full information analyzed by ROKU for dataset of Ge et al. (2005). For the original gene expression matrix, an outlier matrix (consisting of 1 for over-expressed outliers, -1 for under-expressed outliers, and 0 for non-outliers) is provided. It also contains two entropy scores measured by ROKU and Schug's method and their ranks.

Format: XLS Size: 8.1MB [Download file](#)

This file can be viewed with: [Microsoft Excel Viewer](#)

OPEN DATA

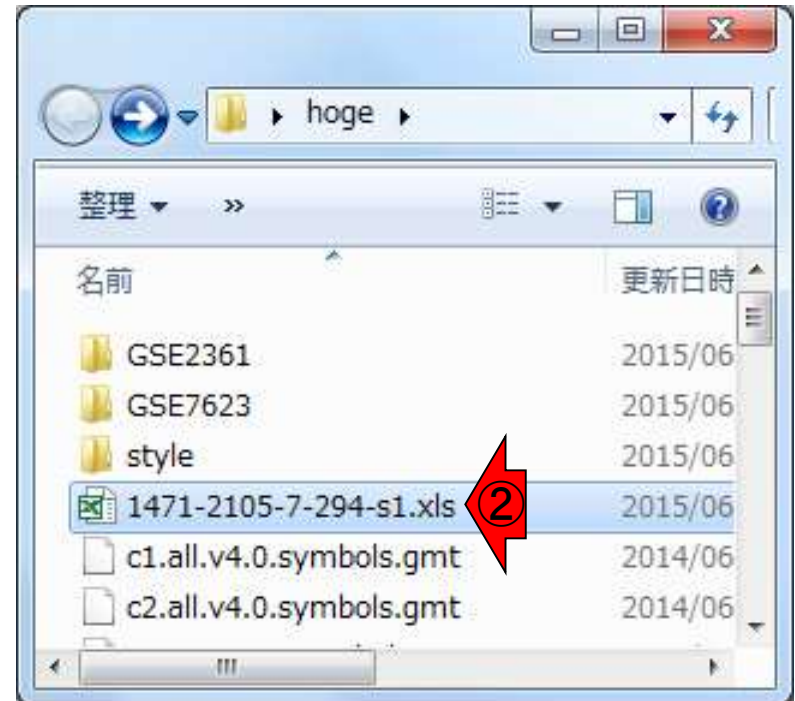
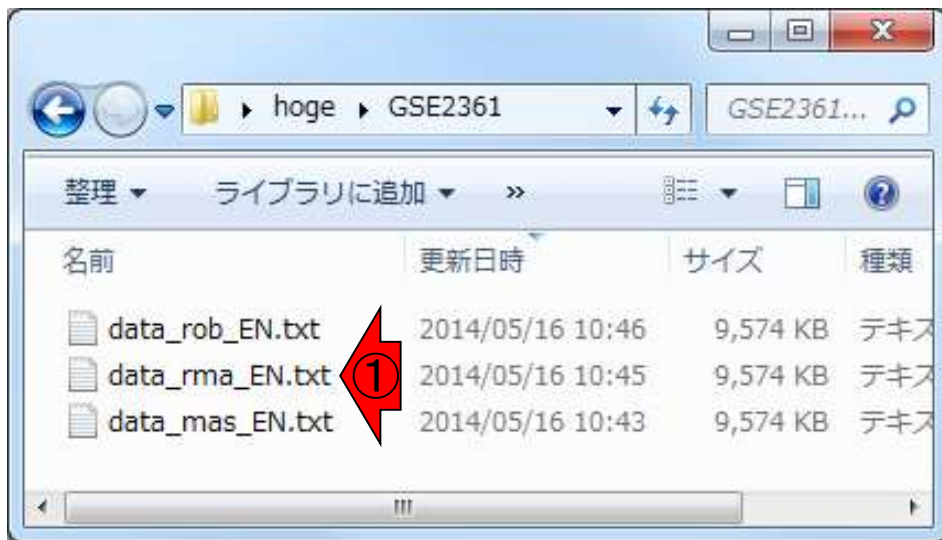
①RMA前処理後のデータを入力としてROKUを実行した結果(の一部)が、②ROKU論文のAdditional file 1と基本的に同じもの

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)、…



①テンプレートスクリプトをテキストエディタにコピペして、②ファイル名部分を変更し、コピペ実行。約2分

ROKUを実行

解析 | 発現変動 | 多群間 | ROKU (Kadota_2006)

TCCパッケージで提供している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"列の数值は、「ROKU論文中の Additional file 1 (Suppl.xls)の "H(x)"列の数值」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となります。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」ということとなります。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と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。

①「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: log2変換後のデータであるという前提です。

```
in_f <- "sample21.txt"
out_f <- "hoge1.txt"

#必要なパッケージをロード
library(TCC)

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

#本番
hoge <- ROKU(data)
outlier <- hoge$outlier
modH <- hoge$modH
ranking <- hoge$rank

#ファイルに保存
tmp <- cbind(row.names(data), outlier, modH, ranking)
write.table(tmp, out_f, sep="\t", quote="")
```

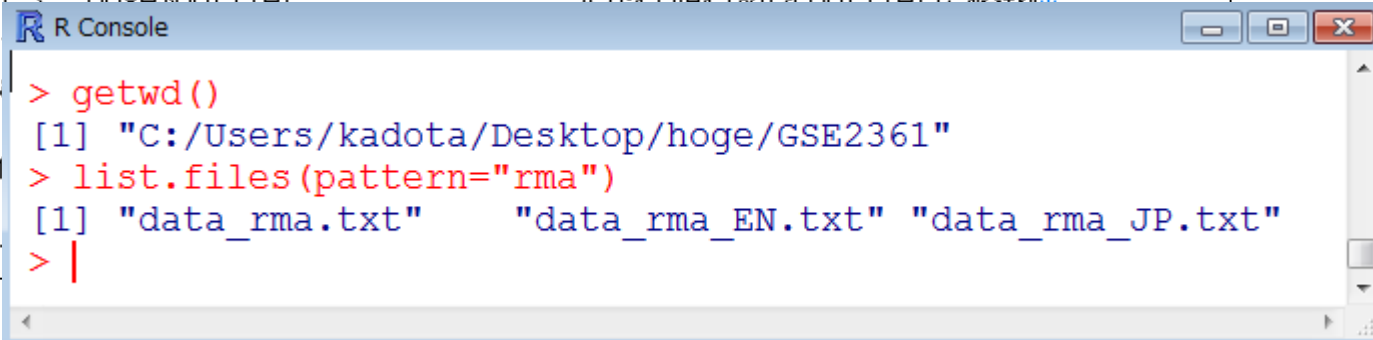
```
in_f <- "data_rma_EN.txt"
out_f <- "hoge_rma_EN.txt"

#必要なパッケージをロード
library(TCC)

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

#本番
hoge <- ROKU(data)
outlier <- hoge$outlier
modH <- hoge$modH
ranking <- hoge$rank

#ファイルに保存
tmp <- cbind(row.names(data), outlier, modH, ranking)
write.table(tmp, out_f, sep="\t", quote="")
```



Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

機能解析の実体は、遺伝子セットの発現変動解析。発現に差のある遺伝子セットを探したい、ということ

機能解析

- 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 Pathway, 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 vs. G2群比較で動いていた
- ...

様々な方法があります

(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

What's new?

- 「解析 | 機能解析 | パスウェイ解析」だけでなく、ネットワーク解析 | 発現変動 | 時系列 | non-periodic genes | [maSigPro \(Conesa 2006\)](#) (last modified 2015/08/16)
- 「解析 | 発現変動 | 2群間算術平均値」に関する [SAM \(Tusher 2001\)](#) (last modified 2015/08/16)
- ① 門田幸二 著 [シリーズ Use R](#) (2006)、WAD法 (Kadoya et al., 2006)、「書籍 | トランスクリプトーム解析」に関する [遺伝子オントロジー\(GO\)解析](#) | [gage \(Luo 2009\)](#) (last modified 2015/05/25)
- ② ③ [GSA \(Efron 2007\)](#) (last modified 2017/03/07) 推奨
- ④ [Category \(Jiang 2007\)](#) (last modified 2014/06/01)
- ⑤ [pcot2 \(Kong 2006\)](#) (last modified 2014/06/01)
- [SAFE \(Barry 2005\)](#) (last modified 2014/06/01)
- [globaltest \(Goeman 2004\)](#) (last modified 2014/06/01)
- [パスウェイ\(Pathway\)解析](#) | [GSAR \(Rahmatallah 2017\)](#) (last modified 2017/03/12) 推奨
- [Pathview \(Luo 2013\)](#) (last modified 2014/06/01)
- [gage \(Luo 2009\)](#) (last modified 2016/05/25)
- [SPIA \(Tarca 2009\)](#) (last modified 2014/06/01)
- [GSA \(Efron 2007\)](#) (last modified 2017/03/08) 推奨
- [sigPathway \(Tian 2012\)](#) (last modified 2014/06/01)

最近はRNA-seqとマイクロアレイどちらにも対応した解析プログラムがあります。一応①GO解析用と②Pathway解析用に分けていますが、大部分重複しています。講義では③GSAを利用してGO解析を行います。これは④Pathway解析用でも使えます。後で⑤GSARも行います

GO解析用

- 解析 | 発現変動 | 時系列 | non-periodic genes | [maSigPro \(Conesa 2006\)](#) (last modified 2015/08/16)
- 解析 | 発現変動 | 時系列 | non-periodic genes | [SAM \(Tusher 2001\)](#) (last modified 2015/08/16)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | について **①** (last modified 2017/03/08)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [gage \(Luo 2009\)](#)(last modified 2015/05/25)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA](#)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [Category](#)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [pcot2](#)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [SAFE](#)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [globaltest](#)
- 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 | [GSAR \(R\)](#)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [Pathview](#)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [gage \(Luo\)](#)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [SPIA \(Tay](#)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSA \(Efron\)](#)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [sigPathway](#)

解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | について

機能解析の実体は遺伝子セット解析です。遺伝子セット解析としてGO termを利用するのがGO解析です。

R用:


- [globaltest: Goeman et al., Bioinformatics, 2004](#)
- [SAFE: Barry et al., Bioinformatics, 2005](#)
- [topGO: Alexa et al., Bioinformatics, 2006](#)
- [pcot2: Kong et al., Bioinformatics, 2006](#)
- [Category: Jiang et al., Bioinformatics, 2007](#)
- [GSA \(作者のGSAウェブページ\): Efron and Tibshirani, Ann. Appl. Stat., 2007](#)
- [dCoxS: Cho et al., BMC Bioinformatics, 2009](#)
- [GAGE: Luo et al., BMC Bioinformatics, 2009](#)
- [GOsemSim: Yu et al., Bioinformatics, 2010](#)
- [Camera \(limmaの一部, 分かりづらい\): We et al., Nucleic Acids Res., 2012](#)
- [RamiGO: Schröder et al., Bioinformatics, 2013](#)
- [LCT: Dinu et al., BMC Bioinformatics, 2013](#)
- [GSNCA法\(GSARで提供されている\): Rahmatallah et al., Bioinformatics, 2014](#)
- [TcGSA\(時系列データ解析用\): Hejblum et al., PLoS Comput Biol., 2015](#)
- [CompGO: Waardenberg et al., BMC Bioinformatics, 2015](#)
- [EnrichmentBrowser: Geistlinger et al., BMC Bioinformatics, 2016](#)
- [GOexpress: Rue-Albrecht et al., BMC Bioinformatics, 2016](#)
- [GSAR: Rahmatallah et al., BMC Bioinformatics, 2017](#)

R以外:

- [GSEA: Subramanian et al., PNAS, 2005](#)
- [GSEAのユーザーガイド](#)

Pathway解析用

②Pathviewなどはプログラム名からも想像できるようにPathway解析専用。以降は、なぜ③GSEAが流行ったかなど、遺伝子セット解析の概念を④の総説をベースに解説

- 解析 | 発現変動 | 時系列 | non-periodic genes | [maSigPro \(Conesa 2006\)](#)
- 解析 | 発現変動 | 時系列 | non-periodic genes | [SAM \(Tusher 2001\)](#)
- 解析 | 機能解析 | 遺伝子オンロジー(GO)解析 | [gage \(Luo 2009\)](#)(last modified 2013)
- 解析 | 機能解析 | 遺伝子オンロジー(GO)解析 | [GSA \(Efron 2007\)](#)
- 解析 | 機能解析 | 遺伝子オンロジー(GO)解析 | [Category \(Jiang 2006\)](#)
- 解析 | 機能解析 | 遺伝子オンロジー(GO)解析 | [pcot2 \(Kong 2006\)](#)
- 解析 | 機能解析 | 遺伝子オンロジー(GO)解析 | [SAFE \(Barry 2005\)](#)
- 解析 | 機能解析 | 遺伝子オンロジー(GO)解析 | [globaltest \(Goeman 2007\)](#)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah 2014\)](#) 
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [Pathview \(Luo 2013\)](#)(last modified 2013)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [gage \(Luo 2009\)](#)(last modified 2013)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [SPIA \(Tarca 2009\)](#)(last modified 2013)
- 解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSA \(Efron 2007\)](#)(last modified 2013)
- 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

機能解析の実体は遺伝子セット解析です。遺伝子セット解析としてKEGGなどのパスウェイ


R用:

- [GSA \(作者のGSAウェブページ\): Efron and Tibshirani, Ann. Appl. Stat., 2007](#)
- [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](#) 
- [GSNCA法\(GSARで提供されている\): Rahmatallah et al., Bioinformatics, 2014](#)
- [TcGSA\(時系列データ解析用\): Hejblum et al., PLoS Comput Biol., 2015](#)
- [EnrichmentBrowser: Geistlinger et al., BMC Bioinformatics, 2016](#)
- [GSAR: Rahmatallah et al., BMC Bioinformatics, 2017](#)

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](#)

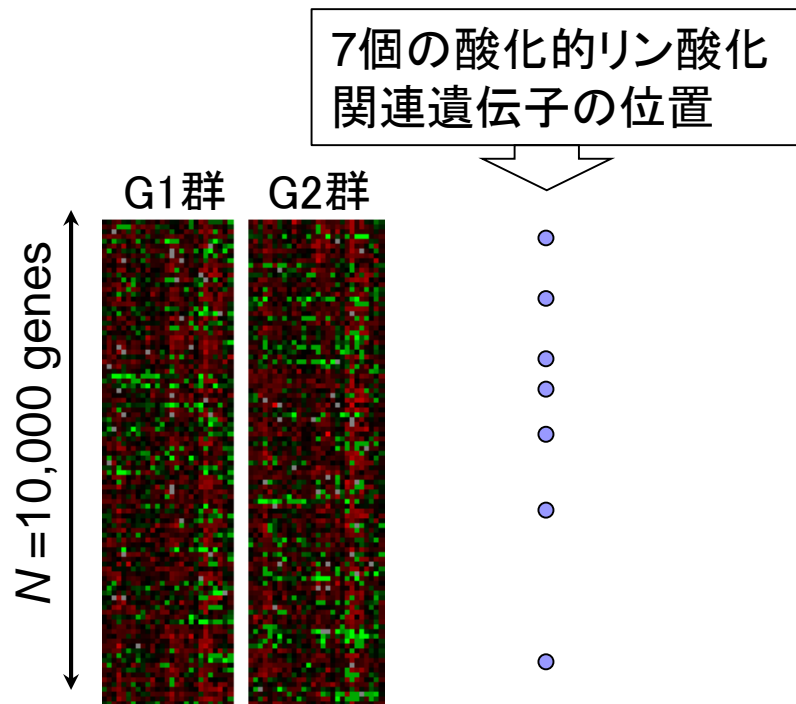
Review、ガイドライン、パイプライン系:

- 手法比較: [Abatangelo et al., BMC Bioinformatics, 2009](#)
- Review: [Khatri H, PLoS Comput Biol., 2012](#)
- Review: [Maciejewski H, Brief Bioinform., 2013](#)
- 手法比較: [Alavi-Majd et al., Gene, 2014](#) 

酸化了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、という問題を考える

機能解析(遺伝子セット解析)

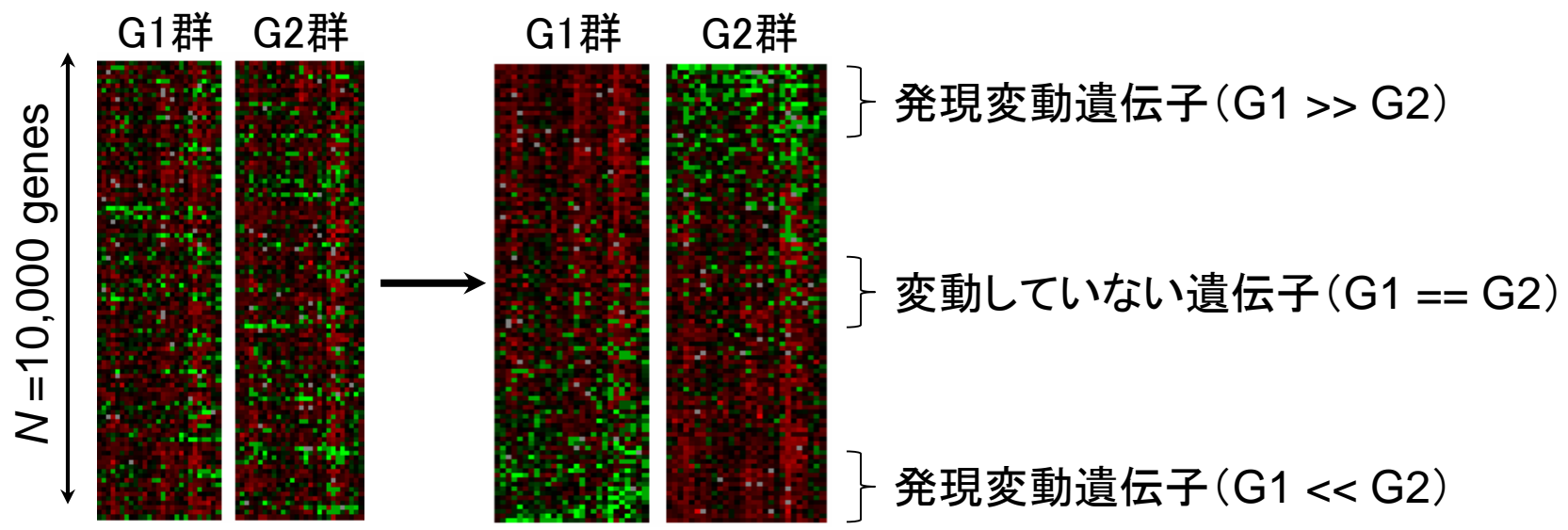
- 発現変動遺伝子セット解析手法(2群間比較用がほとんど)
 - $N=10,000$ 個の遺伝子からなる2群間比較用データ
 - この中に、XXX関連遺伝子が n 個含まれている
 - 例:酸化了的リン酸化(=XXX)関連遺伝子が7($=n$)個含まれている



基本はデフォルトでやるようですが、様々な選択肢があります

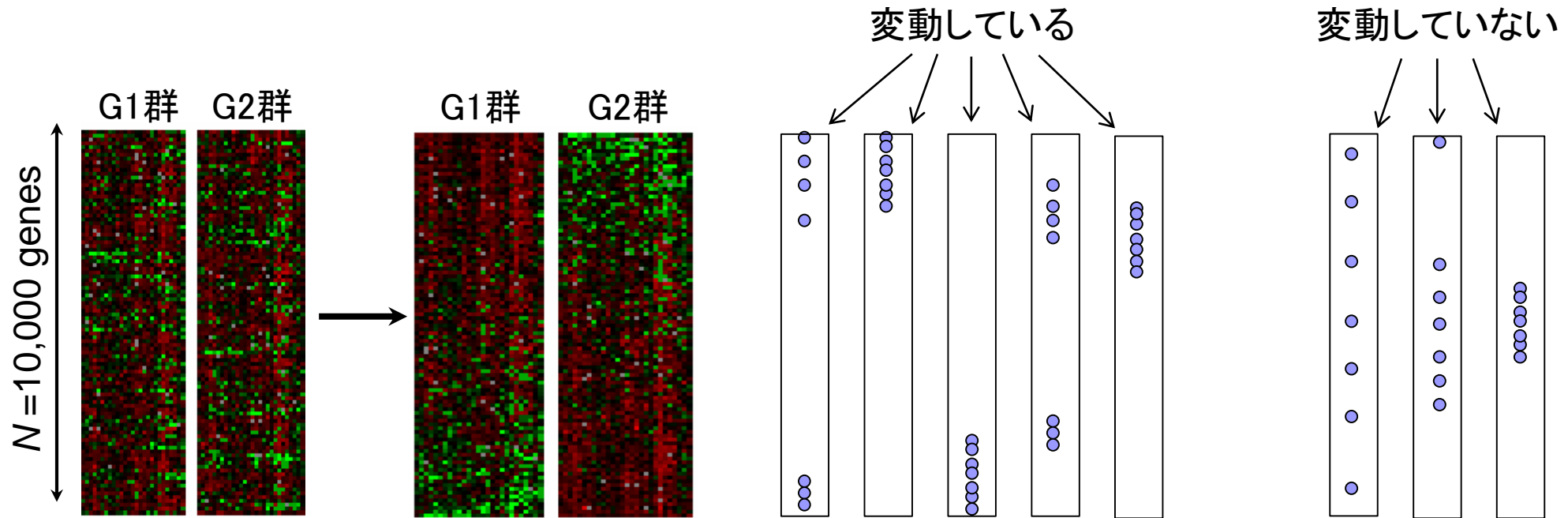
機能解析 (遺伝子セット解析)

- 遺伝子ごとの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、...



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



機能解析 (遺伝子セット解析)

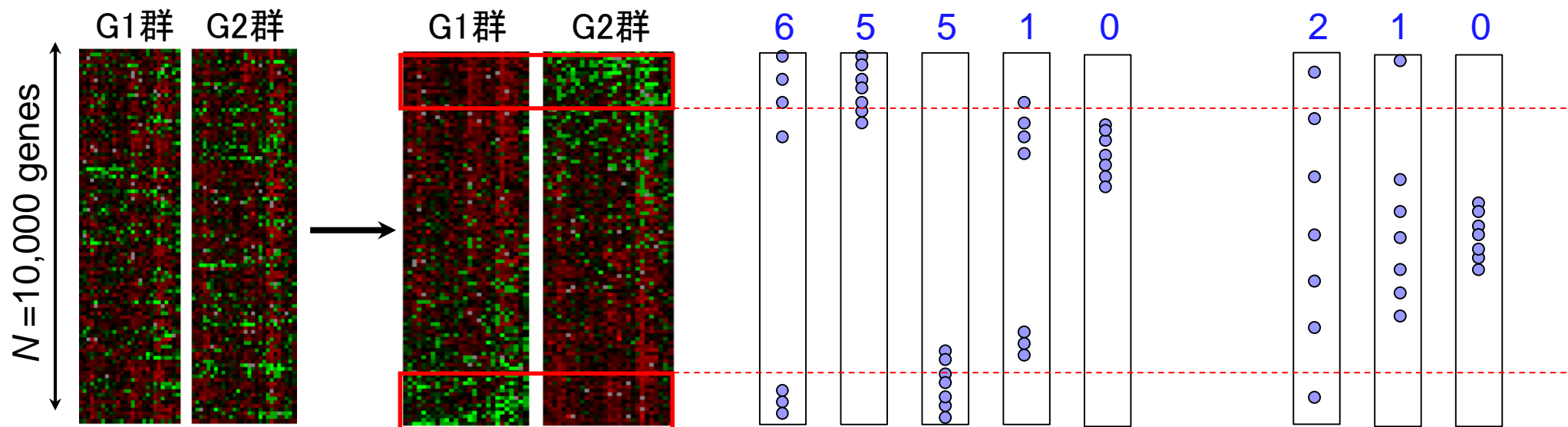
基本的な考え方は、「全遺伝子」と「上位のサブセット」のみで、調べたい遺伝子セットの割合が不変という帰無仮説のもとで検定

Over-Representation Analysis (ORA)

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

酸化リン酸化関連遺伝子セット ($n=7$) が変動していない場合: $x/n \doteq X/N (= 1500/10000)$

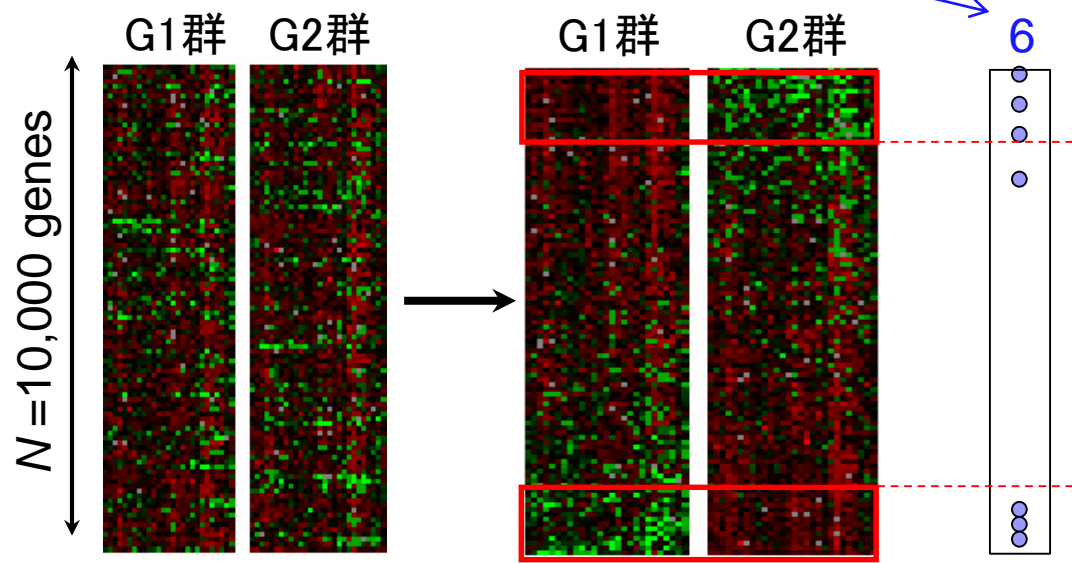
酸化リン酸化関連遺伝子セット ($n=7$) が変動している場合: $x/n \gg X/N (= 15\%)$



2 × 2分割表 (contingency table) に基づく方法。超幾何検定やカイニ乗検定が利用されます

機能解析 (遺伝子セット解析)

- 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	$N-n$	

DEGとして1500個抽出したとき、酸化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

機能解析（超幾何検定）

- $N=10000$ 個の遺伝子発現データ中に XXX =酸化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 $x=6$ 個の酸化リン酸化関連遺伝子が含まれていた
 - 帰無仮説：酸化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

```
R Console
> N <- 10000
> n <- 7
> X <- 1500
> x <- 6
> sum(dhyper(x=x:X, m=n, n=N-n, k=X))
[1] 6.892847e-05
> |
```

機能解析 (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

```
R Console
> ?dhyper
starting httpd help server ... done
> x <- 6
> m <- 7
> n <- 9993
> k <- 1500
> sum(dhyper(x=x:X, m=m, n=n, k=k))
[1] 6.892847e-05
> |
```


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

機能解析(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

直感は重要

- ① N=10000個の遺伝子発現データ中にXXX=酸化リン酸化関連遺伝子はn=7個存在する
- ② 上位X=1500個の発現変動遺伝子(DEG)の中にx=6個の酸化リン酸化関連遺伝子が含まれていた
- ③ 帰無仮説:酸化リン酸化関連遺伝子の割合はDEGとnon-DEG間で差がない

①の段階で、調べたい遺伝子セットは、 $7/10,000 = 0.07\%$ の割合だと考える。②で $6/1,500 = 0.4\%$ の割合に濃縮されていると考える。③今やっているのは、ある2群間比較。もし比較している群間で、この遺伝子セットが全体として発現変動していなかったとしたら…ランダムで1,500個とった時に、この遺伝子セット中の遺伝子が含まれる割合は0.07%なので、個数だと $1500 \times 0.07\% = 1.05$ 個程度しか含まれないはず。実際に得られたのは6個なので、偶然こんな結果が得られたとは考えにくい。起こるとしたら④くらい低い確率なんだね。だから「発現変動遺伝子セット」と考えよう、という思考回路

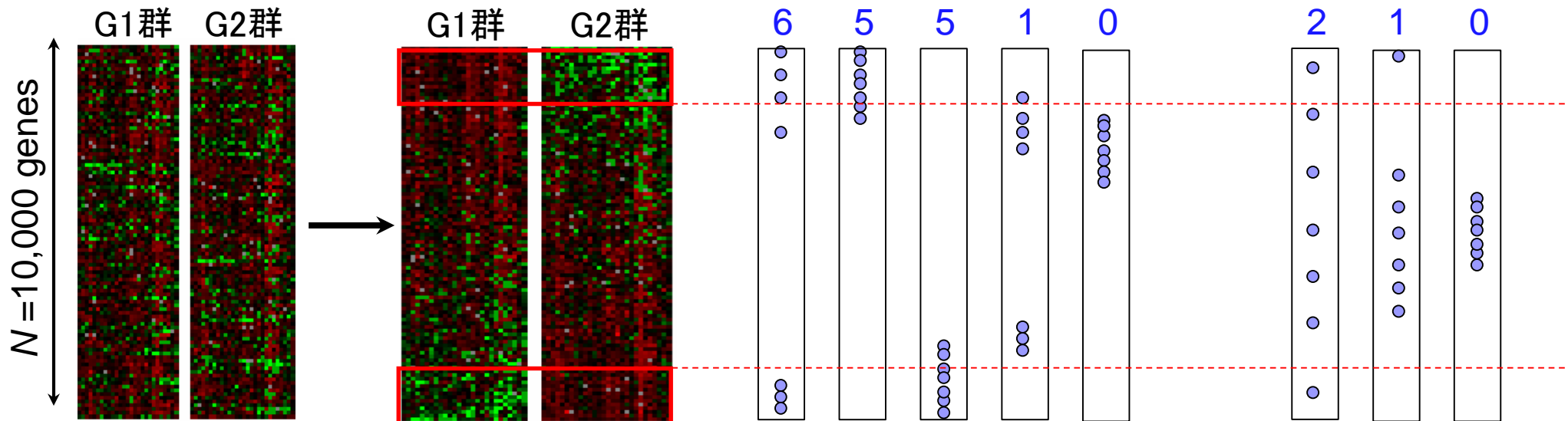
```
R Console
> ?dhype
starting
> x <- 6
> m <- 7
> n <- 9993
> k <- 1500
> sum(dhyper(x=x:X, m=m, n=n, k=k))
[1] 6.892847e-05
> |
```

④

機能解析 (遺伝子セット解)

上位1500個のうち、酸化的リン酸化関連遺伝子が7個中4つ以上含まれていれば $p < 0.05$ で検出可能ということの意味する

Over-Representation Analysis (ORA)



<i>p</i> -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

機能解析（遺伝子セット解析）

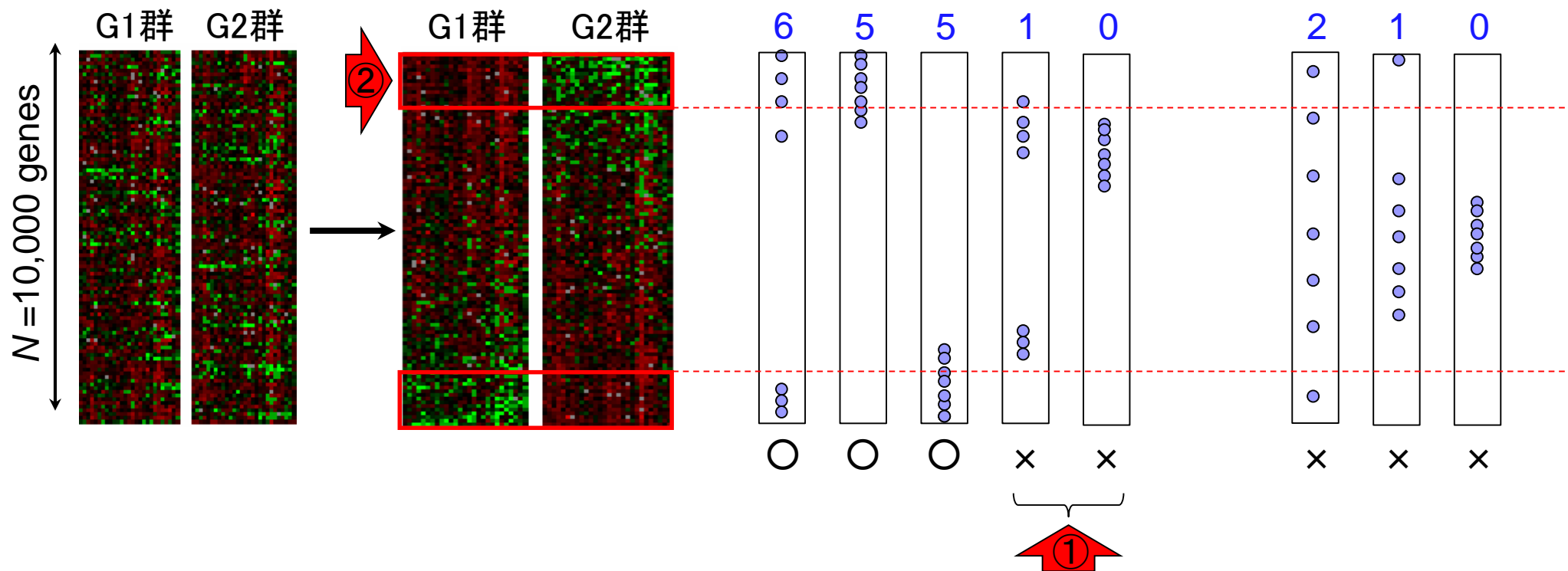
■ Over-Representation Analysis (ORA)

- GenMAPP (Dahlquist et al., *Nature Genet.*, **31**: 19–20, 2002) ①
- 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1世代 (ORA) の短所

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情報量低下(発現変動の度合い→カウント情報)

	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



第2世代 (FCS)

■ 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、 $\log(G2/G1)$ 、相関係数、...
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、 M 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)

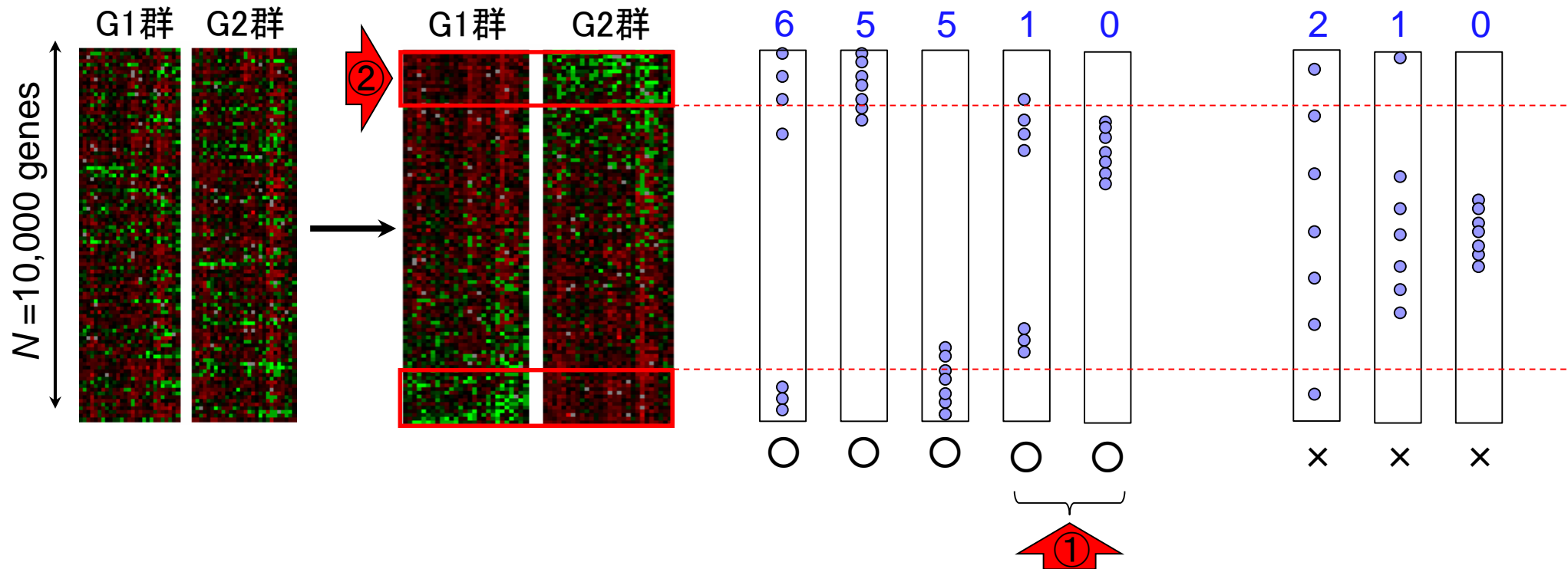
第2世代 (FCS)

第一世代(ORA)の欠点が改善

遺伝子ごとのlog比で考えると、遺伝子を等価に取り扱うのではなく、log比そのものを足し込むことで、発現変動の度合いが大きいほどより大きな重みをかけるようなイメージ

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情報量低下(発現変動の度合い→カウント情報)

	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



第2世代 (FCS)

■ Functional Class Scoring (FCS)

- GSEA (Subramanian et al., *PNAS*, **102**: 15545–15550, 2005) ①
- 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) ②
- GeneTrail (Backes et al., *Nucleic Acids Res.*, **35**: W186–W192, 2007)
- SAM-GS (Dinu et al., *BMC Bioinformatics*, **8**: 242, 2007)
- ...

突っ込みどころは満載だが、ネガティブなことばかりいってもしょうがないし、この種の機能解析が目的の場合も多い

遺伝子セット解析の課題

- (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)
 - アノテーションの解像度を上げる努力も大事
- 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, inferred 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%)

どの遺伝子セットにどの遺伝子が属しているかという情報からなるgmt形式ファイルの取得が第一歩

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
- ...

Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

①MSigDBは、Molecular Signature Databaseの略。様々な遺伝子セット解析を行うために必要な情報を含むgmt形式ファイルをダウンロード可能です

MSigDB ver. 6.0

- H: hallmark gene sets (50 gene sets)
- 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 (4,731 gene sets)
 - CGP: chemical and genetic perturbations (3,402 gene sets)
 - CP: canonical pathways (1,329 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 (5,917 gene sets)
 - BP: biological process (4,436 gene sets) ←
 - CC: cellular component (580 gene sets)
 - MF: molecular function (901 gene sets)
- c6: oncogenic signatures gene sets (189 gene sets)
- c7: immunologic signatures gene sets (4,872 gene sets)

発現変動と関連するKEGGパスウェイを調べたいとき

発現変動と関連するBP中のGO termsを調べたいとき



MSigDB ver. 5.1

- H: hallmark gene sets (50 gene sets)
- 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 (4,726 gene sets)
 - CGP: chemical and genetic perturbations (3,396 gene sets)
 - CP: canonical pathways (1,330 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 (4,872 gene sets)

発現変動と関連するKEGG
パスウェイを調べたいとき

発現変動と関連するBP中
のGO termsを調べたいとき

MSigDB ver. 5.0

- H: hallmark gene sets (50 gene sets)
- 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 (4,725 gene sets)
 - CGP: chemical and genetic perturbations (3,395 gene sets)
 - CP: canonical pathways (1,330 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

発現変動と関連するBP中
のGO termsを調べたいとき

MSigDB ver. 5.0 → 5.1

- H: hallmark gene sets (50 gene sets)
- 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 (4,726 gene sets)
 - CGP: chemical and genetic perturbations (3,396 gene sets)
 - CP: canonical pathways (1,330 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 (4,872 gene sets) ← ①

発現変動と関連するKEGG
パスウェイを調べたいとき

発現変動と関連するBP中
のGO termsを調べたいとき



(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

What'

- 「解
- 「解
- 算術
- 門田
- al.
- 「書
- お知
- の意

- 解析 | 発現変動 | 時系列 | non-periodic genes | [maSigPro \(Conesa 2006\)](#) (last modified 2009/8/3)
- 解析 | 発現変動 | 時系列 | non-periodic genes | [SAM \(Tusher 2001\)](#) (last modified 2009/8/3)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [について](#) (modified 2015/06/07) **NEW**
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GAGE \(Lu 2009\)](#) (last modified 2014/06/01)
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#) (last modified 2014/06/03) 推奨

解析 | 機能解析 | 遺伝子オントロジー(GO)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。遺伝子セット解析としてGO termを利用するのがGO解析です。

R用:

- [globalt](#)
- [SAFE](#):
- [topGO](#)
- [pcot2](#):
- [Catego](#)
- [GSA](#) (
- [dCoxS](#)
- [GAGE](#)
- [GOSer](#)
- [Camer](#)
- [RamiG](#)
- [LCT:T](#)

遺伝子セットDB系:

- [MSigDB: Subramanian et al., PNAS, 2005](#)

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含めた [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、GO解析の場合は biological process (BP) が一般的なようです。2015/06/07現在のバージョンは5.0です。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しておく。
 - 「c5: gene ontology gene sets」の「bp: biological process」を解析する場合: [c5.bp.v5.0.symbols.gmt](#)
 - 「c5: gene ontology gene sets」の「cc: cellular components」を解析する場合: [c5.cc.v5.0.symbols.gmt](#)
 - 「c5: gene ontology gene sets」の「mf: molecular functions」を解析する場合: [c5.mf.v5.0.symbols.gmt](#)

MSigDB

②発現変動と関連するbiological processes (BP)中のGO termsを調べたいときは、③のいずれかのgmtファイルを利用

logged in as kadota@bi.a.u-tokyo.ac.jp
logout BROAD INSTITUTE

Gene Set Enrichment Analysis
GSEA Home Downloads Molecular Signatures Database Documentation Contact

logged in as kadota@bi.a.u-tokyo.ac.jp
logout BROAD INSTITUTE

Gene Set Enrichment Analysis
GSEA Home Downloads Molecular Signatures Database Documentation Contact

2 c5: gene ontology (GO) gene sets

all GO gene sets, gene symbols	c5.all.v5.0.symbols.gmt
all GO gene sets, Entrez IDs	c5.all.v5.0.entrez.gmt
all GO gene sets, original identifiers	c5.all.v5.0.orig.gmt
GO biological processes, gene symbols	c5.bp.v5.0.symbols.gmt
GO biological processes, Entrez IDs	c5.bp.v5.0.entrez.gmt
GO biological processes, original identifiers	c5.bp.v5.0.orig.gmt
GO cellular components, gene symbols	c5.cc.v5.0.symbols.gmt
GO cellular components, Entrez IDs	c5.cc.v5.0.entrez.gmt
GO cellular components, original identifiers	c5.cc.v5.0.orig.gmt
GO molecular functions, gene symbols	c5.mf.v5.0.symbols.gmt
GO molecular functions, Entrez IDs	c5.mf.v5.0.entrez.gmt
GO molecular functions, original identifiers	c5.mf.v5.0.orig.gmt

3

gmtファイル

基本的にどれを使っても自由だが、利用するRパッケージがどの入力形式を受け付けるかにも依存する。経験上gene symbolsを使っておけば間違いないので、私は①*.symbols.gmtをいつも利用しています

	A	B	C					
1	TRNA_PROCESSING	http://www	ADAT1	TRNT1	FARS2	METTL1	SARS	AARS
2	REGULATION_OF_BIOLOGICAL_QUALI	http://www	DLC1	ALS2	SLC9A7	PTGS2	PTGS1	MPV
3	DNA_METABOLIC_PROCESS	http://www	XRCC5	XRCC4	RAD51C	XRCC3	XRCC2	XRC
4	AMINO_SUGAR_METABOLIC_PROCESS	http://www	UAP1	CHIA	GNPDA1	GNE	CSGALNAC	CHS
5	BIOPOLYMER_CATABOLIC_PROCESS	http://www	BTRC	HNRNPD	USE1	RNASEH1	RNF217	ISG2
6	DNA_METABOLIC_PROCESS	http://www	HNRNPF	HNRNPD	SYNCRIP	MED24	POPF	MED

	A	B	C	D	E	F	G	
1	TRNA_PROCESSING	http://www	23536	51095	10667	4234	6301	c5.bp.v5.0.symbols.gmt
2	REGULATION_OF_BIOLOGICAL_QUALI	http://www	10395	57679	84679	5743	←5742	c5.bp.v5.0.entrez.gmt
3	DNA_METABOLIC_PROCESS	http://www	7520	7518	5889	7517	7516	c5.bp.v5.0.orig.gmt
4	AMINO_SUGAR_METABOLIC_PROCESS	http://www	6675	27159	10007	10020	55790	
5	BIOPOLYMER_CATABOLIC_PROCESS	http://www	8945	3184	55850	246243	154214	
6	DNA_METABOLIC_PROCESS	http://www	3185	3184	10492	9862	6096	

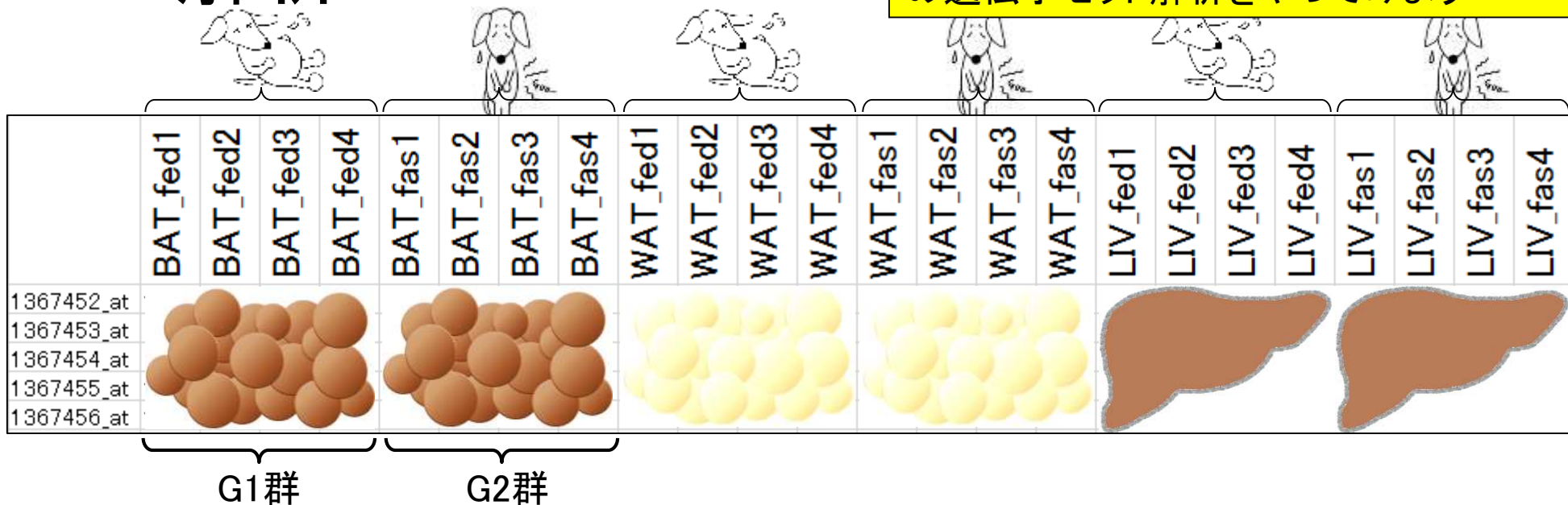
[c5.bp.v5.0.symbols.gmt](#)
[c5.bp.v5.0.entrez.gmt](#)
[c5.bp.v5.0.orig.gmt](#)

	A	B	C	D	E	F	G	
1	TRNA_PROCESSING	http://www	ADAT1	TRNT1	FARS2	METTL1	SARS	AARS
2	REGULATION_OF_BIOLOGICAL_QUALI	http://www	DLC1	ALS2	SLC9A7	PTGS2	PTGS1	MPV
3	DNA_METABOLIC_PROCESS	http://www	XRCC5	XRCC4	RAD51C	XRCC3	XRCC2	XRC
4	AMINO_SUGAR_METABOLIC_PROCESS	http://www	UAP1	CHIA	GNPDA1	G		
5	BIOPOLYMER_CATABOLIC_PROCESS	http://www	BTRC	HNRNPD	USE1	R		
6	DNA_METABOLIC_PROCESS	http://www	HNRNPF	HNRNPD	SYNCRIP	M		

1列目: 遺伝子セット名
 2列目: URL
 3列目以降: gene ID or symbol

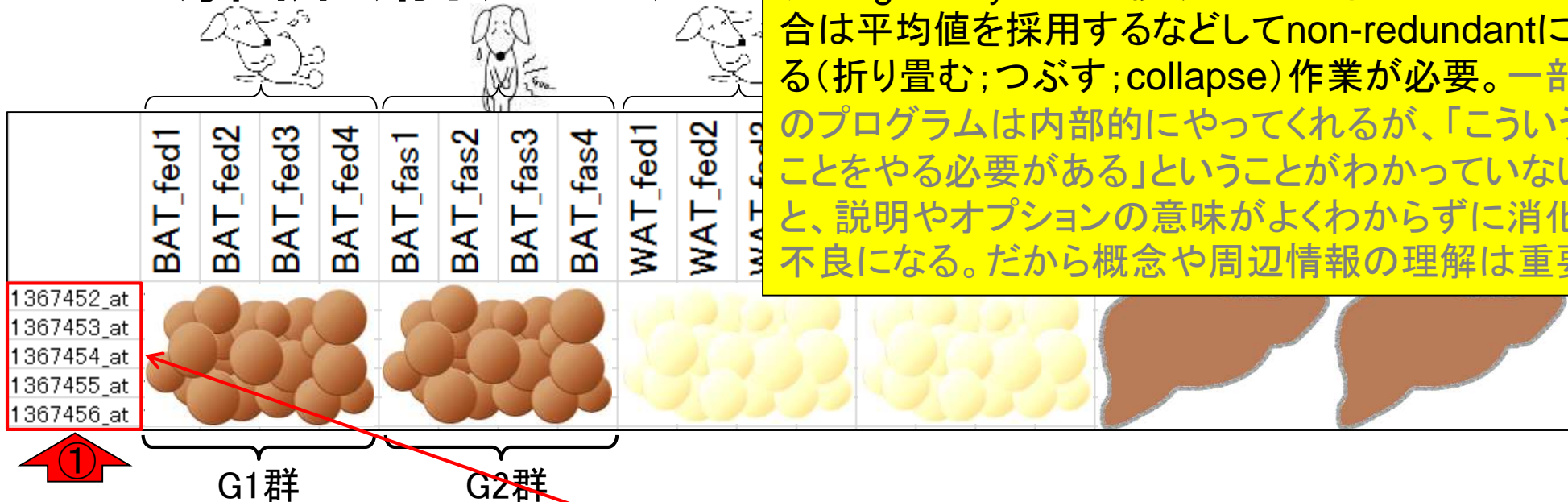
GO解析

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



GO解析 (前処理)

①probeset ID (Affymetrix GeneChipの遺伝子につけられたID)と②gene symbolの対応付けを行い、同じgene symbolに複数の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。一部のプログラムは内部的にやってくれるが、「こういうことをやる必要がある」ということがわかっていないと、説明やオプションの意味がよくわからずに消化不良になる。だから概念や周辺情報の理解は重要



	A	B	C	D	E	F	G	
1	TRNA_PROCESSING	http://www	ADAT1	TRNT1	FARS2	METTL1	SARS	AAR
2	REGULATION_OF_BIOLOGICAL_QUALI	http://www	DLC1	ALS2	SLC9A7	PTGS2	PTGS1	MPV
3	DNA_METABOLIC_PROCESS	http://www	XRCC5	XRCC4	RAD51C	XRCC3	XRCC2	XRC
4	AMINO_SUGAR_METABOLIC_PROCESS	http://www	UAP1	CHIA	GNPDA1	GNE	CSGALNAC	CHS
5	BIOPOLYMER_CATABOLIC_PROCESS	http://www	BTRC	HNRNPD	USE1	RNASEH1	RNF217	ISG2
6	DNA_METABOLIC_PROCESS	http://www	HNRNPD	HNRNPD	SYNPOD	MED24	POPR	MED

A red arrow labeled '2' points to the cell containing 'XRCC5' in row 3, column C.

Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

ID変換

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。ここから、probeset 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。コピペはしなくてよい

- 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
- 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)
- 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 | トランスクリプト | GEOから取得 (last modified 2013/08/16)
- イントロ | トランスクリプト | GEOから取得 (last modified 2013/08/16)
- イントロ | Affymetrix CEL | GEOから取得 (last modified 2013/08/16)

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

公共DB Gene 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」-「ディレクトリの変更」でファイルを保存したいディレクトリに移動し以下をコピペ。

1. Affymetrix Rat Genome

```
out_f <- "hoge1.txt"
param <- "GPL1355"
```

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

```
#前処理
data <- getGEO(param)
```

```
#本番
out <- data@dataTable@
write.table(out, out_f)
```

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。アノテーション情報のバージョンが異なりうるため若干違うかもしれませんが hoge3 GPL1355.txt と酷似したものができていると思います。

```
param1 <- "GSE7623"
param2 <- "hoge3_"
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)
sapply(data, annotation)
```

```
#本番
hoge <- sapply(data, annotation)
```

```
#入手したいGEO IDを指定
#出力ファイル名の最初の部分を指定
#outオブジェクト中のID列とgene symb
```

```
#パッケージの読み込み
```

```
#指定したGEO IDのデータを取得
#用いられたアレイ情報(GPL ID)を表示
```

```
#用いられたアレイ情報(GPL ID)をhoge
```

ID変換

① probeset 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。確認時は2分程度で終わりましたが、hogeフォルダにhoge3_GPL1355.txtを一応置いてあります

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

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。アノテーション情報のバージョンが異なりうるため若干違うかもしれませんがhoge3_GPL1355.txtと酷似したものができていると思います。

```
param1 <- "GSE7623" #入手したいGEO ID
param2 <- "hoge3_" #出力ファイル名
param_posi <- c(1, 11) #outオブジェクトの列番号

#必要なパッケージをロード
library(GEOquery) #パッケージのインストール

#前処理
data <- getGEO(param1) #指定したGEO IDの取得
sapply(data, annotation) #用いられたアレイのID

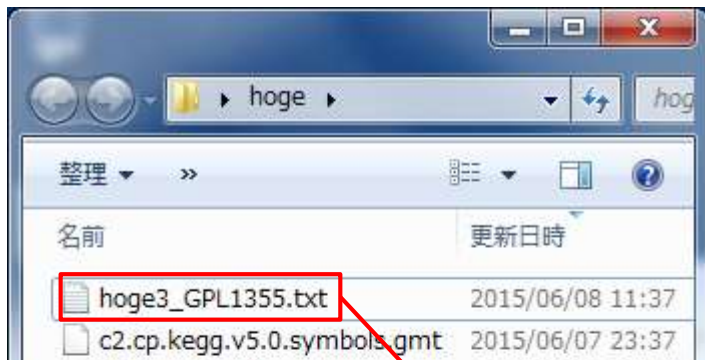
#本番
hoge <- sapply(data, annotation) #用いられたアレイのID
for(i in 1:length(hoge)){ #hogeの要素数
  out_f <- paste(param2, hoge[i], ".txt", sep="")
  out <- data[[i]]@featureData@data #アノテーション情報
  colnames(out) #確認して、必要な列番号
```

```
R Console
> for(i in 1:length(hoge)){ #hogeの要素数
+   out_f <- paste(param2, hoge[i], ".txt", sep="")
+   out <- data[[i]]@featureData@data #アノテーション情報
+   colnames(out) #確認して、必要な列番号
+   out <- out[,param_posi] #サブセット
+   write.table(out, out_f, sep="\t", append=F, $)
+ }
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="hoge3")
[1] "hoge3_GPL1355.txt"
> head(out, n=3)
      ID Gene Symbol
1367452_at 1367452_at Sumo2
1367453_at 1367453_at Cdc37
1367454_at 1367454_at Copb2
> dim(out)
[1] 31099      2
> |
```



ID変換

①エクセルで開くときには注意が必要！1行1列目のところが”ID”から始まる文字列の場合に②や③で示すような現象が起こるようですが、基本無視で構いません（同じ作業を2回やると普通に開けます。後で行います）



Excel

最近使ったファイル

- hoge1.txt
デスクトップ > hoge > GSE7623
- c5.bp.v4.0.symbols.gmt
マイドキュメント > 2015 > Lecture > 機能ゲノム...
- c5.bp.v5.0.symbols.gmt
マイドキュメント > 2015 > Lecture
- 1471-2105-7-294-s1.xls
マイドキュメント > 2015 > Lecture
- hoge_rma_EN.txt
デスクトップ > hoge > GSE2361
- 1471-2105-7-294-s1.xls
デスクトップ

Microsoft Excel

警告 'hoge3_GPL1355.txt' のファイル形式と拡張子が一致しません。ファイルが破損しているか、安全ではない可能性があります。発行元が信頼できない場合は、このファイルを開かないでください。ファイルを開きますか？

はい(Y) いいえ(N) ヘルプ(H)

Microsoft Excel

警告 Excel は 'hoge3_GPL1355.txt' が SYLK ファイルであることを確認しましたが、読み込むことができません。エラーが含まれるか、SYLK ファイル形式でない可能性があります。異なるファイル形式でそのファイルを開くには、[OK] をクリックしてください。

OK キャンセル

ID変換

編集して保存したい場合には、ドラッグ&ドロップで開いてはだめです。「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!

[BMC Bioinformatics](#). 2004 Jun 23;5:80.

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

[Zeeberg BR¹](#), [Riss J](#), [Kane DW](#), [Bussey KJ](#), [Uchio E](#), [Linehan WM](#), [Barrett JC](#), [Weinstein JN](#).

+ Author information

Abstract

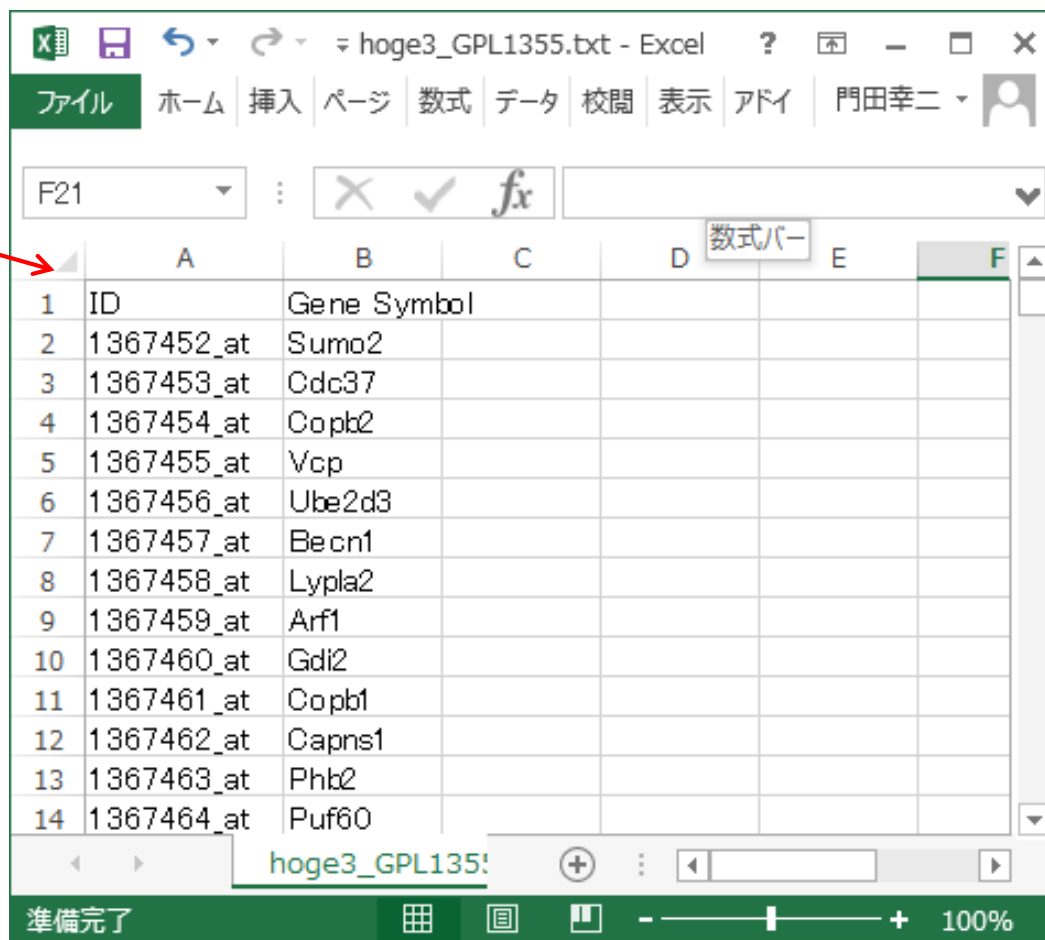
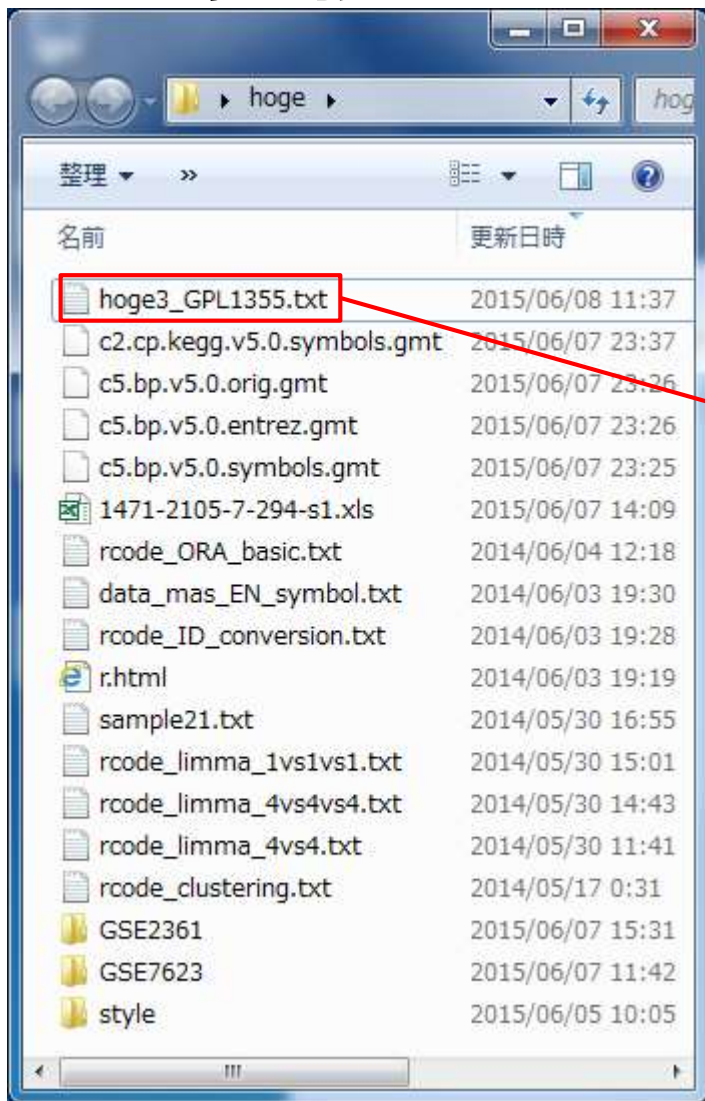
BACKGROUND: When processing microarray data sets, we recently noticed that some gene names were being changed inadvertently to non-gene names.

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.

ID変換

ここでは、ファイルの中身を眺めるだけなので、再度ドラッグ&ドロップ。1回目は失敗しても2回目は普通に開けます



ID変換



	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	Copk2	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

①ソート後の状態。同じgene symbolを持つprobeset IDが複数存在することがわかる。例えば②A2bp1は、6 probeset IDs

ID変換

	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	A1 bg	1369509_a_at	2.283274	2.663464	3.393108	5.581337	4.444774	2.070498	2.366629	1.242299	3.2026
1368784_at	A1 cf	1368784_at	5.844694	5.913832	5.755455	5.509977	4.184778	6.572447	5.657811	5.381227	5.2463
1370027_a at	A1 i3 /// Mug1	1370027_a at	7.186455	6.369903	3.668318	3.498324	7.211794	6.427345	2.480638	3.215938	5.9591
1378371_at	A2bp1	1378371_at	3.395028	5.712191	5.063096	6.231098	6.612678	6.064106	6.833413	6.143807	4.3285
1380516_at	A2bp1	1380516_at	4.359652	4.69428	3.010153	4.315149	4.516653	2.632701	3.892409	5.18021	1.9178
1383130_at	A2bp1	1383130_at	3.081817	4.593572	4.535909	8.811696	9.702333	9.75088	9.105133	8.825036	5.4091
1383257_at	A2bp1	1383257_at	3.501018	2.966239	2.232853	7.283267	8.426518	8.651317	7.625359	7.31545	1.3291
1385235_at	A2bp1	1385235_at	3.847942	4.026388	3.911792	8.330072	9.3733	9.137933	8.885068	8.734736	4.6341
1390594_at	A2bp1	1390594_at	4.641244	6.627334	5.234572	5.89079	2.245885	1.417343	6.138234	6.252581	5.2941
1382945_at	A2ld1	1382945_at	9.974381	10.2494	9.811327	9.831842	9.093722	8.947993	8.82087	9.13077	9.3861
1392725_at	A2ld1	1392725_at	6.696741	5.425293	6.610669	6.104502	6.084417	6.684742	5.49265	2.381073	6.7701



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。①平均化前と②平均化後。③R上でベタ打ちで確認

ID変換

入力1:hoge3_GPL1355.txt

入力2:data_mas_EN.txt

ID	Gene Symbol	BAT_fed1	BAT_fed2	BAT_fed3	BAT_fed4	BAT_fas1	BAT_fas2	BAT_fas3	BAT_fas4	WAT_fe
1369509_a_at	A1 bg	2.283274	2.663464	3.393108	5.581337	4.444774	2.070498	2.366629	1.242299	3.202
1368784_at	A1 cf	5.844694	5.913832	5.755455	5.509977	4.184778	6.572447	5.657811	5.381227	5.246
1370027_a_at	A1 i3 /// Mug1	7.186455	6.369903	3.668318	3.498324	7.211794	6.427345	2.480638	3.215938	5.959
1378371_at	A2bp1	3.395028	5.712191	5.063096	6.231098	6.612678	6.064106	6.833413	6.143807	4.328
1380516_at	A2bp1	4.359652	4.69428	3.010153	4.315149	4.516653	2.632701	3.892409	5.18021	1.917
1383130_at	A2bp1	6.081817	4.93572	4.535909	8.811696	9.702333	9.75088	9.105133	8.825036	5.409
1383257_at	A2bp1	1.501018	4.66239	2.232853	7.283267	8.426518	8.651317	7.625359	7.31545	1.329
1385235_at	A2bp1	3.847942	4.026388	3.911792	8.330072	9.3733	9.137933	8.885068	8.734736	4.634
1390594_at	A2bp1	4.641244	6.627334	5.234572	5.89079	2.245885	1.417343	6.138234	6.252581	5.294
1382945_at	A2ld1	9.974381	10.2494	9.811327	9.831842	9.093722	8.947993	8.82087	9.13077	9.386
1392725_at	A2ld1	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
A2bp1	3.971117	7.70001	3.998062	6.810345	6.812894	6.275713	7.079936	7.075304	3.819
A2ld1	8.335561	7.837348	8.210998	7.968172	7.58907	7.816367	7.15676	5.755922	8.078
A2m	4.201252	5.612429	4.637289	3.967974	3.373227	3.301394	4.6048	3.135392	8.089
A3galt2	5.848709	7.467876	6.437417	6.415588	5.585956	6.751774	7.105526	7.278448	7.548
A4galt	4.71439	1.895269	3.072015	2.174757	5.799177	4.672696	5.229912	5.442959	4.947
AA926063 ///									
Aaas									
Aacs									

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


ID変換

(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

What's new
「解析するだ
「解析算術平
門田幸
al., 200
「書籍
お知ら
の参考

- 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)
- 前処理 | ID変換 | probe ID -> gene symbol (last modified 2014/06/03)
- 前処理 | ID変換 | probe ID -> Entrez ID (last modified 2014/06/03)
- 前処理 | ID変換 | probe ID -> その他 (last modified 2014/06/03)
- 前処理 | ID変換 | 同じ遺伝子名を持つもの
- 解析 | 基礎 | 共通遺伝子の抽出 (last modified 2014/06/03)
- 解析 | 基礎 | ベクトル間の距離 (last modified 2014/06/03)
- 解析 | 基礎 | 遺伝子ごとの各種要約統計量 (last modified 2014/06/03)
- 解析 | 基礎 | 最大発現量を示す組織の同定 (last modified 2014/06/03)
- 解析 | 基礎 | 似た発現パターンを持つ遺伝子 (last modified 2014/06/03)
- 解析 | 基礎 | 平均-分散プロット (last modified 2014/06/03)
- 解析 | クラスタリング | 階層的 | について (last modified 2014/06/03)
- 解析 | クラスタリング | 階層的 | pvclust (last modified 2014/06/03)
- 解析 | クラスタリング | 階層的 | hclust (last modified 2014/06/03)

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が存在する場合には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)、最大値(max)など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、以下をコピー

1. サンプルデータ20の31,099 probesets×24 samplesのRMA-preprocessedデータ(data_rma_2.txt)の場合:

Affymetrix Rat Genome 230 2.0 Arrayを用いて得られたNakai et al., BBB, 2008のデータです。イントロ | アノテーション情報取得 | GEOquery(Davis 2007)の3を行って得られたhoge3_GPL1355.txtの情報も利用しています。data_rma_2.txtの1列目のID情報とhoge3_GPL1355.txtの1列目のID情報の対応がとれる(同じ行の位置でなくてもよい)ことが前提です。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
```

② #入力ファイル名を指定してin_f1に格納(発現データ)
#入力ファイル名を指定してin_f2に格納(Gene symbolとID)
#出力ファイル名を指定してout_fに格納
#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で指定したフ
IDs <- as.vector(sym[,1]) #Gene symbol情報をベクトルに変換し、IDsに格納
names(IDs) <- rownames(sym) #IDsを行名で対応づけられるようにしている
uniqID <- unique(IDs) #non-redundant ID情報を抽出し、uniqIDに格納
uniqID <- uniqID[uniqID != ""] #uniqIDの中から指定したIDがないものを除く
uniqID <- uniqID[!is.na(uniqID)] #uniqIDの中から指定したIDが"NA"のものを除く
uniqID <- uniqID[!is.nan(uniqID)] #uniqIDの中から指定したIDが"NaN"のものを除く
```

#本番

ID変換

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

hogeフォルダ中のdata_mas_EN_symbol.txtは、①のコピペで作成しています(がそれは使わずに自分で作成しましょう)。②作業ディレクトリに入力ファイルがあることを確認してから実行しましょう。③のファイルがないことに気づくはず(思考停止禁止!)

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。gene symbolの対応関係情報が必要です。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が存在する場合には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)、最大値(max)など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

1. サンプルデータ20の31,099 probesets×24 samplesの

Affymetrix Rat Genome 230 2.0 Arrayを用いて得られた情報取得 | GEOquery(Davis 2007)の3を行って得られたdata_rma_2.txtの1列目のID情報とhoge3_GPL1355.txtの1列目のID情報とを比較し、一致していることが前提です。1分程度で終わります。

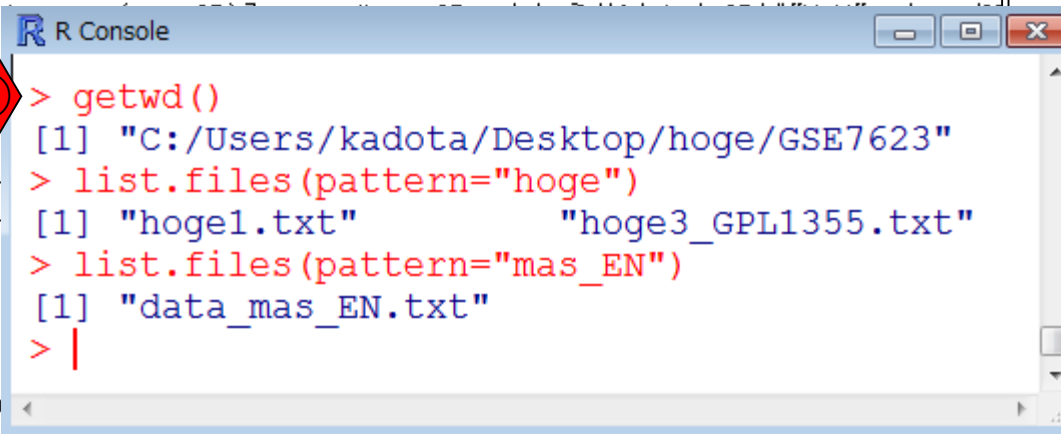
```
in_f1 <- "data_rma_2.txt"
in_f2 <- "hoge3_GPL1355.txt"
out_f <- "hoge1.txt"
param <- mean
```

```
#前処理(IDとGene symbolとの対応関係を含む)
sym <- read.table(in_f2, header=TRUE, row.names=1, sep="¥t", quote="")
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

rancode_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は
names(IDs) <- rownames(sym) #IDsを行名で対応づけられるようにしている
uniqID <- unique(IDs) #non-redundant ID情報を抽出し、uniqIDに格
uniqID <- uniqID[uniqID != ""] #uniqIDの中から指定したIDがないものを除く
uniqID <- uniqID[!is.na(uniqID)] #uniqIDの中から指定したIDが"NA"のものを除
uniqID <- uniqID[!is.nan(uniqID)]
↓
#本番↓
data <- read.table(in_f1, header=TRUE, row.names=1, sep="¥t", quote="")
hoge <- t(apply(data, MARGIN=2, FUN=function(x) {
  param
}, data, rownames(hoge) <- rownames(data))
↓
#ファイルに保存↓
tmp <- cbind(rownames(hoge), as.data.frame(hoge))
write.table(tmp, out_f, sep="¥t", append=F, quote=F, row.names=F)#tmpの中身を1行
```



ID変換

コピー実行後に確認。出力ファイルの① data_mas_EN_symbol.txtは、②14,132個のユニークなgene symbolsからなることがわかる

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) #IDsを行名で対応づけられるようにしている
uniqID <- unique(IDs) #non-redundant ID情報を抽出し、uniqIDに格
uniqID <- uniqID[uniqID != ""] #uniqIDの中から指定したIDがないものを除く
uniqID <- uniqID[!is.na(uniqID)] #uniqIDの中から指定したIDが"NA"のものを除
uniqID <- uniqID[!is.nan(uniqID)] #uniqIDの中から指定したIDが"NaN"のものを除
↓
#本番↓
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t",
hoge <- t(apply(as.matrix(uniqID), 1, function(i, d = data, s
      apply(d[which(s == i), ], 2, p, na.rm = TRUE)
    ), data, IDs, param)) #apply関数でdata, IDs, paramを結合
rownames(hoge) <- uniqID #non-redundant IDをhogeに格納
↓
#ファイルに保存↓
tmp <- cbind(rownames(hoge), hoge) #指定したIDの列を行名として追加
write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F)

```



```

R Console
+      ), data, IDs, param)) # $
> rownames(hoge) <- uniqID # $
>
> #ファイルに保存
> tmp <- cbind(rownames(hoge), hoge) # $
> write.table(tmp, out_f, sep="\t", append=F)
> dim(tmp)
[1] 14132 25
> getwd()
[1] "C:/Users/kadota/Desktop/hoge/GSE7623"
> |

```



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という事実を①のマニュアルを読んでいて偶然発見)。②孫建強氏作は1分、③門田作は2分



Tips: as.matrix

参考

1. サンプルデータ20の31,099 probesets×24 samplesのRMA-preprocessedデータ(data

Affymetrix Rat Genome 230 2.0 Arrayを用いて得られた [Nakai et al., BBB, 2008](#) のデータです。 [イントロ | アノテーション情報取得 | GEOquery \(Davis 2007\)](#) の3を行って得られた [hoge3_GPL1355.txt](#) の情報も利用しています。 [data_rma_2.txt](#) の1列目のID情報と [hoge3_GPL1355.txt](#) の1列目のID情報の対応がとれる(同じ行の位置でなくてもよい)ことが前提です。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との対応関係を含む)
sym <- read.table(in_f2, header=TRUE)
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=TRUE)
hoge <- t(apply(as.matrix(uniqID), 1,
               function(x) apply(d[which(s == x)], data, IDs, param)),
         data, IDs, param))
rownames(hoge) <- uniqID

#ファイルに保存
```



2. サンプルデータ20の31,099 probesets×24 samplesのRMA-preprocessedデータ(data_rma_2.txt)の場合:

Affymetrix Rat Genome 230 2.0 Arrayを用いて得られた [Nakai et al., BBB, 2008](#) のデータです。 [イントロ | アノテーション情報取得 | GEOquery \(Davis 2007\)](#) の3を行って得られた [hoge3_GPL1355.txt](#) の情報も利用しています。 [data_rma_2.txt](#) の1列目のID情報と [hoge3_GPL1355.txt](#) の1列目のID情報の対応がとれる(同じ行の位置でなくてもよい)ことが前提です。2分程度で終わります。

```
in_f1 <- "data_rma_2.txt"
in_f2 <- "hoge3_GPL1355.txt"
out_f <- "hoge2.txt"
param <- mean

#入力ファイル名を指定してin_f1に格納(発現データ)
#入力ファイル名を指定してin_f2に格納(Gene symbol)
#出力ファイル名を指定してout_fに格納
#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で指定した
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=TRUE, row.names=1, sep="\t", quote="")#in_f1で指定し
hoge <- NULL
for(i in 1:length(uniqID)){
  hoge <- rbind(hoge, apply(data[which(IDs == uniqID[i])], 2, param, na.rm=TRUE))#
}
rownames(hoge) <- uniqID

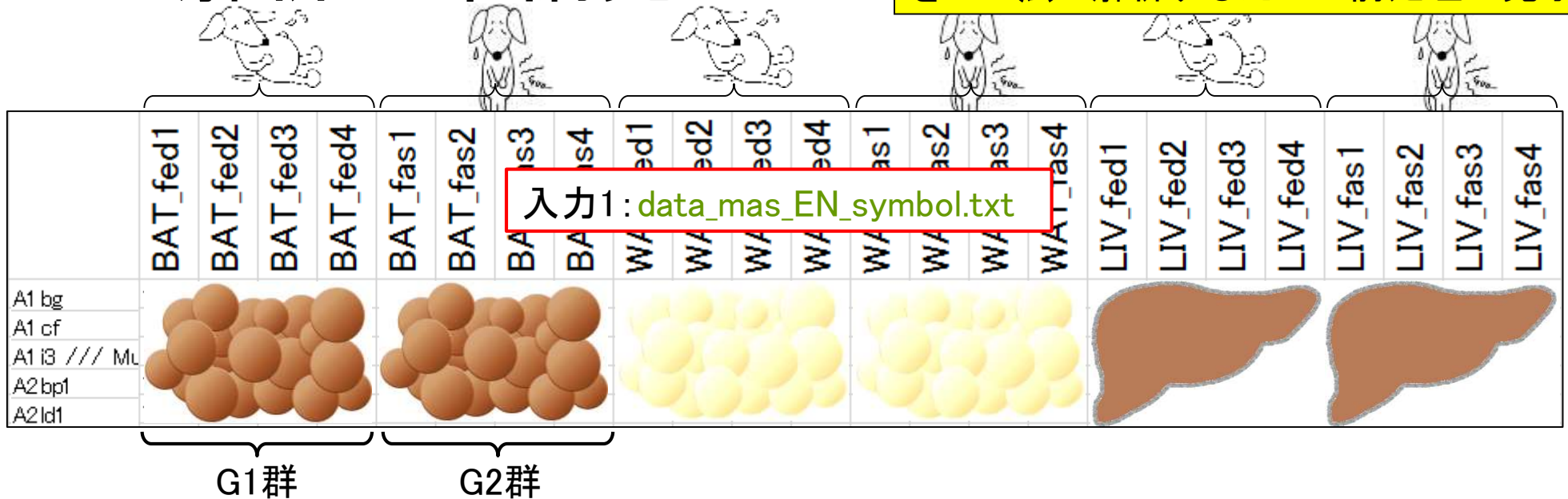
#最終的に欲しい情報を格納するためのプレースホルダ
#non-redundant ID数分だけループを回す
#non-redundant ID数分だけループを回す
#non-redundant IDをhogeの行の名前として利用
```

Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

褐色脂肪「満腹 vs. 空腹」の発現変動に関連したGO Biological Process遺伝子セットをGSA法で解析するための前処理が完了

GO解析の準備完了



Excel spreadsheet showing the GO Biological Process analysis results. The spreadsheet is titled 'c5.bp.v5.0.symbols.gmt - Excel' and shows a list of GO terms and their associated genes. A red box highlights the input file 'c5.bp.v5.0.symbols.gmt'.

	A	B	C	D	E	F	G
1	TRNA_PROCESSING	http://www.ADAT1	TRNT1	FARS2	METTL1	SAR	
2	REGULATION_OF_BIOLOGICAL_QUALITY	http://www.DLC1	ALS2	SLC9A7	PTGS2	PTG	
3	DNA_METABOLIC_PROC		34	RAD51C	XRCC3	XRC	
4	AMINO_SUGAR_METABC			GNPDA1	GNE	CSG	
5	BIOPOLYMER_CATABOLIC_PROCESS	http://www.BTRC	HNRNPD	USE1	RNASEH1	RNF	
6	DNA_METABOLIC_PROCESS	http://www.HNRNPF	HNRNPD	SYNCRIP	MED24	POC	

入力2: [c5.bp.v5.0.symbols.gmt](#)

準備完了

GSAでGO解析

②例題3をテンプレートとして、③入力ファイル名部分のみをdata_mas_EN_symbol.txtに変更してBAT_fed vs. BAT_fasのGO解析をやってみよう。④作業ディレクトリはココ

(Rで)マイクロアレイデータ解析

- (last modified)
- 解析 | 発現変動 | 時系列 | non-periodic genes | [SAM \(Tusher 2001\)](#) (last modified 2009/8/3)
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#) (last modified 2015/06/07) **NEW**
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GAGE \(Luo 2009\)](#) (last modified 2014/06/01)
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#) (last modified 2014/06/03) 推奨
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [Category \(Jiang 2009\)](#) (last modified 2014/06/01)
- What's new?
- 門田幸二先生の最近の講義資料

解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron_2007\)](#) **NEW**

GSAパッケージを使った解析のやり方を示します

3. 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。BAT_fed vs. BAT_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。

1. サンプルデータ

肝臓のみからなるデータの probeset ID

```
in_f1 <- "data_rma_2_nr.txt"
in_f2 <- "c5.bp.v5.0.symbols.gmt"
out_f1 <- "hoge3_G1.txt"
out_f2 <- "hoge3_G2.txt"
param_G1 <- 4
param_G2 <- 4
param_FDR <- 0.1
param_posi <- c(1:4, 5:8)
```

#必要なパッケージをロード

#入力ファイル

```
in_f1 <- "data_rma_2_nr.txt"
in_f2 <- "c5.bp.v5.0.symbols.gmt"
out_f1 <- "hoge3_G1.txt"
out_f2 <- "hoge3_G2.txt"
param_G1 <- 4
param_G2 <- 4
param_FDR <- 0.1
param_posi <- c(1:4, 5:8)
```

#入力ファイル名を指定してin_f1に格納(発現データ)
 #入力ファイル名を指定してin_f2に格納(gmtファイル)
 #出力ファイル名を指定してout_f1に格納(G1群で高)
 #出力ファイル名を指定してout_f2に格納(G2群で高)
 #G1群のサンプル数を指定
 #G2群のサンプル数を指定
 #DEG検出時のfalse discovery rate (FDR)閾値を
 #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)

gmt <- GSA.read.gmt(in_f2)
data <- read.table(in_f1, header=1, as.is=TRUE)
rownames(data) <- toupper(rownames(data))
data.c1 <- c(rep(1, param_G1), rep(2, param_G2))
data <- data[,param_posi]
colnames(data) <- c("G1", "G2")
```

#GSA本番

```
out <- GSA(data, data.c1,
```

```
R Console
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="symbol")
[1] "c2.cp.kegg.v5.0.symbols.gmt"
[2] "c5.bp.v5.0.symbols.gmt"
[3] "data_mas_EN_symbol.txt"
> |
```

#GSAを実行し、結果をoutに格納

GSAでGO解析

①FDR 10%の閾値を満たす有意な遺伝子セット数は、②G1群で高発現のものが24個、G2群で高発現のものが4個だったことがわかる。ヒトによって若干結果が異なります

3. 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。BAT_fed vs. BAT_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。

```

in_f1 <- "data_rma_2_nr.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 <- "c5.bp.v5.0.symbols.gmt" #入力ファイル名を指定してin_f2に格納(gmtファイル)
out_f1 <- "hoge3_G1.txt" #出力ファイル名を指定してout_f1に格納(G1群で高発現のもの)
out_f2 <- "hoge3_G2.txt" #出力ファイル名を指定してout_f2に格納(G2群で高発現のもの)
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を指定
param_posi <- c(1, 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) #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を読み込み
data <- read.table(in_f1, header=TRUE, row.names=1, sep="\t", #in_f1で指定したファイルを読み込み
rownames(data) <- toupper(rownames(data)) #IDを大文字に変換してIDを大文字に変換して
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2で指定
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、結果を出力

```

```

R Console
+ FDRcut=param_FDR) #FDRcutを指定
> #ファイルに保存
> write.table(tmp$negative, out_f1, sep="\t") #G1群で高発現のもの
> write.table(tmp$positive, out_f2, sep="\t") #G2群で高発現のもの
> getwd()
[1] "C:/Users/kadota/Desktop/hoge"
> list.files(pattern="symbol")
[1] "c2.cp.kegg.v5.0.symbols.gmt"
[2] "c5.bp.v5.0.symbols.gmt"
[3] "data_mas_EN_symbol.txt"
> list.files(pattern="hoge3")
[1] "hoge3_G1.txt" "hoge3_G2.txt"
[3] "hoge3_GPL1355.txt"
> nrow(tmp$negative)
[1] 24
> nrow(tmp$positive)
[1] 4
> |

```

GSAでGO解析

出力は、FDR 10%の閾値を満たす有意な遺伝子セットリスト。数値の分布的にG1群で高発現の遺伝子セットである①

STERIOD_BIOSYNTHETIC_PROCESSが最も変動しているのだろう

Gene_set	Gene_set_name	Score	p-value	FDR
22	PHOSPHOLIPID_BIOSYNTHETIC_PROCESS	-0.5692	0	0
56	REGULATION_OF_DNA_BINDING	-0.3102	0	0
92	COFACTOR_METABOLIC_PROCESS	-0.4346	0	0
134	BIOSYNTHETIC_PROCESS	-0.2019	0	0
147	INTRACELLULAR_PROTEIN_TRANSPORT	-0.1555	0	0
205	PROTEIN_IMPORT	-0.2265	0	0
264	MEMBRANE_LIPID_BIOSYNTHETIC_PROCESS	-0.4554	0	0
287	LIPID_METABOLIC_PROCESS	-0.2399	0	0
317	LIPID_BIOSYNTHETIC_PROCESS	-0.7518	0	0
331	REGULATION_OF_TRANSCRIPTION_FACTOR_ACTIVIT	-0.295	0	0
370	MITOCHONDRION_ORGANIZATION_AND_BIOGENESIS	-0.3251	0	0
383	NEGATIVE_REGULATION_OF_DNA_BINDING	-0.6935	0	0
415	NEGATIVE_REGULATION_OF_CELLULAR_COMPONENT	-0.2769	0	0
466	CELLULAR_RESPONSE_TO_STIMULUS	-0.4359	0	0
471	INTERACTION_WITH_HOST	-0.389	0	0
490	RIBONUCLEOTIDE_METABOLIC_PROCESS	-0.6475	0	0
541	CELLULAR_LIPID_METABOLIC_PROCESS	-0.2414	0	0
607	NEGATIVE_REGULATION_OF_BINDING	-0.6935	0	0
609	PHOSPHOLIPID_METABOLIC_PROCESS	-0.2709	0	0
619	REGULATION_OF_CYTOSKELETON_ORGANIZATION_A	-0.2051	0	0
667	NEGATIVE_REGULATION_OF_TRANSCRIPTION_FACTO	-0.804	0	0
669	RIBOSOME_BIOGENESIS_AND_ASSEMBLY	-0.3904	0	0
721	PROTEIN_TRANSPORT	-0.1425	0	0
762	STERIOD_BIOSYNTHETIC_PROCESS	-1.2465	0	0



Gene_set	Gene_set_name	Score	p-value	FDR
39	POSITIVE_REGULATION_OF_SIGNAL_TRANSDUCTION	0.1897	0	0
121	PATTERN_SPECIFICATION_PROCESS	0.4294	0	0
461	PROTEIN_KINASE_CASCADE	0.1781	0	0
783	ACTIVATION_OF_PROTEIN_KINASE_ACTIVITY	0.6607	0	0

Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

GSARパッケージを発見

BMC Bioinformatics. 2017 Jan 24;18(1):61. doi: 10.1186/s12859-017-1482-6.

GSAR: Bioconductor package for Gene Set analysis in R.

Rahmatallah Y¹, Zybailov B², Emmert-Streib F³, Glazko G⁴.

Author information

Abstract

BACKGROUND: Gene set analysis (in a form of functionally related genes or pathways) has become the method of choice for analyzing omics data in general and gene expression data in particular. There are many statistical methods that either summarize gene-level statistics for a gene set or apply a multivariate statistic that accounts for intergene correlations. Most available methods detect complex departures from the null hypothesis but lack the ability to identify the specific alternative hypothesis that rejects the null.

RESULTS: GSAR (Gene Set Analysis in R) is an open-source R/Bioconductor software package for gene set analysis (GSA). It implements self-contained multivariate non-parametric statistical methods testing a complex null hypothesis against specific alternatives, such as differences in mean (location), variance (scale), or net correlation structure. The package also provides a graphical visualization tool, based on the union of two minimum spanning trees, for correlation networks to examine the change in the correlation structures of a gene set between two conditions and highlight influential genes (hubs).

CONCLUSIONS: GSAR provides a set of multivariate non-parametric statistical methods that test a complex null hypothesis against specific alternatives. The methods in package GSAR are applicable to any type of omics data that can be represented in a matrix format. The package, with detailed instructions and examples, is freely available under the GPL (>= 2) license from the Bioconductor web site.

KEYWORDS: Gene set analysis; Kolmogorov-Smirnov; Minimum spanning tree; Non-parametric; Pathways; Wald Wolfowitz

PMID: 28118818 PMCID: [PMC5259853](#) DOI: [10.1186/s12859-017-1482-6](#)

赤下線部分をざっと眺め、①GSARは2群間比較用で、②調べたい遺伝子セット中で最も影響力の大きい遺伝子を示してくれて、③(遺伝子セットの)ネットワーク図を描いてくれる、のだろうと判断

GSAR

(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

What's new?

- 「解析 | 機能解析」するだけでなく、ネ
- 「解析 | 発現変動 (算術平均値)」に
- 門田幸二 著 [シリ](#)
- al., 2006), WAD
- 「書籍 | トランスク
- お知らせは主に [Q](#)
- の [参考資料\(講義\)](#)

- ・ 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)
- ・ 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)
- ・ 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 2017/04/08)
- ・ 解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah_2017\)](#) (last modified 2017/03/12)
- ・ 解析 | 機能解析 | パスウェイ(Pathway)解析 | [Pathview \(Luo_2013\)](#) (last modified 2014/06/01)

解析 | 機能解析 | パスウェイ(Pathway)解析 | GSAR (Rahmatallah_2017)

② GSARパッケージを用いた解析のやり方を示します。GSARは、様々なプログラム群からなるパッケージです。ここでは、GSARで提供されているGene Sets Net Correlations Analysis法(GSNCA; [Rahmatallah et al., 2014](#))を実行するやり方を示します。オリジナルのprobeset IDからgene symbolにID変換がなされた発現データファイルを入力としています。これは、「[イントロ | アンテーション情報取得 | GEOquery\(Davis_2007\)](#)」で入手したprobeset IDとgene symbolの対応表のファイル、およびオリジナルのprobeset IDごとの発現データファイルを用いて「[前処理 | ID変換 | probe ID -> gene symbol](#)」でID変換したものです。gmtファイルは、[MSigDB \(Subramanian et al., PNAS, 2005\)](#)から任意のものをダウンロードしてください。尚、gmtファイルの読み込みにはGSARパッケージ ([Efron and Tibshirani, 2007](#))中のGSA.read.gmt関数を利用しています。「[ファイル | ディレクトリの変更](#)」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

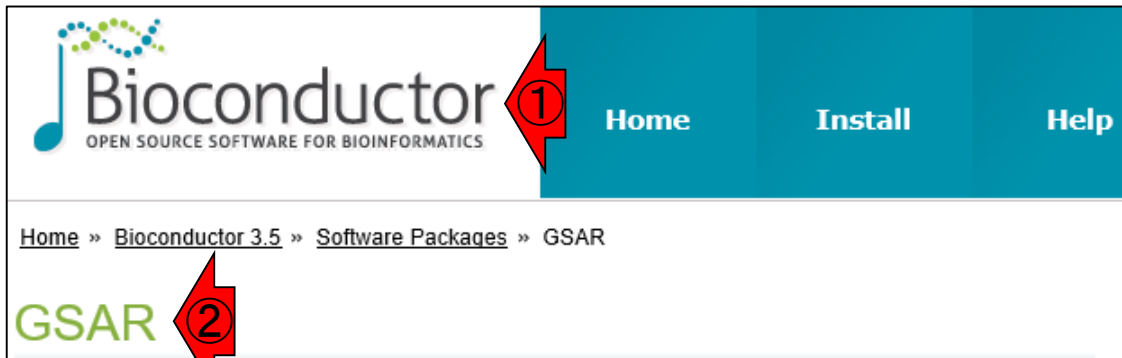
1. GSARの解析例として示されているp53 datasetの場合:

p53DataSetという名前の8,655 genes×50 samplesのデータです。17 wild-type (WT) samples vs. 33 mutant (MUT) samplesの2群間比較です。gmtファイルは、MSigDB (ver. 5.2)の [C2: curated gene sets](#) 中の全データファイル ([c2.all.v5.2.symbols.gmt](#))を利用しています。この中に解析例で取り扱っている遺伝子セット ("[LU_TUMOR_VASCULATURE_UP](#)")が含まれています。ここでは、"[LU_TUMOR_VASCULATURE_UP](#)"を構成する遺伝子群(29 genes)の中から、発現データ中に存在するもの(22 genes)に対してGSNCA法を適用した結果のp-value (= 0.02797)を出力するところまでを示します。p-value = 0.02797という結果は、比較する2群間(WT vs. MUT)でこの遺伝子セットの発現の相関に差がない(帰無仮説)確率が2.797%しかないことを意味します。1つの判断基準として、(建前上)有意水準 $\alpha = 0.05$ を設定していたのなら、帰無仮説が本当は正しいが間違っして棄却してしまう確率を5%に設定したことに相当し、この場合はp-value < 0.05なので帰無仮説を棄却し対立仮説(差がある)を採用することになります。

```
in_f <- "c2.all.v5.2.symbols.gmt" #入力ファイル名を指定してin_flに格納(gmtファイル)
```

まずはマニュアルを眺める

①Bioconductorの、②GSARのサイトに行き、③のPDFファイルを開いて概要や利用例を眺める。hogeフォルダ中のGSAR.pdfでもよい



Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help

Home » [Bioconductor 3.5](#) » [Software Packages](#) » [GSAR](#)



GSAR

platforms all downloads top 20% posts 0 in Bioc 2.5 years
build ok commits 0.50 test coverage unknown

Gene Set Analysis in R

Bioconductor version: Release (3.5)

Gene set analysis using specific alternative hypotheses. Tests for differential expression, so correlation structure.

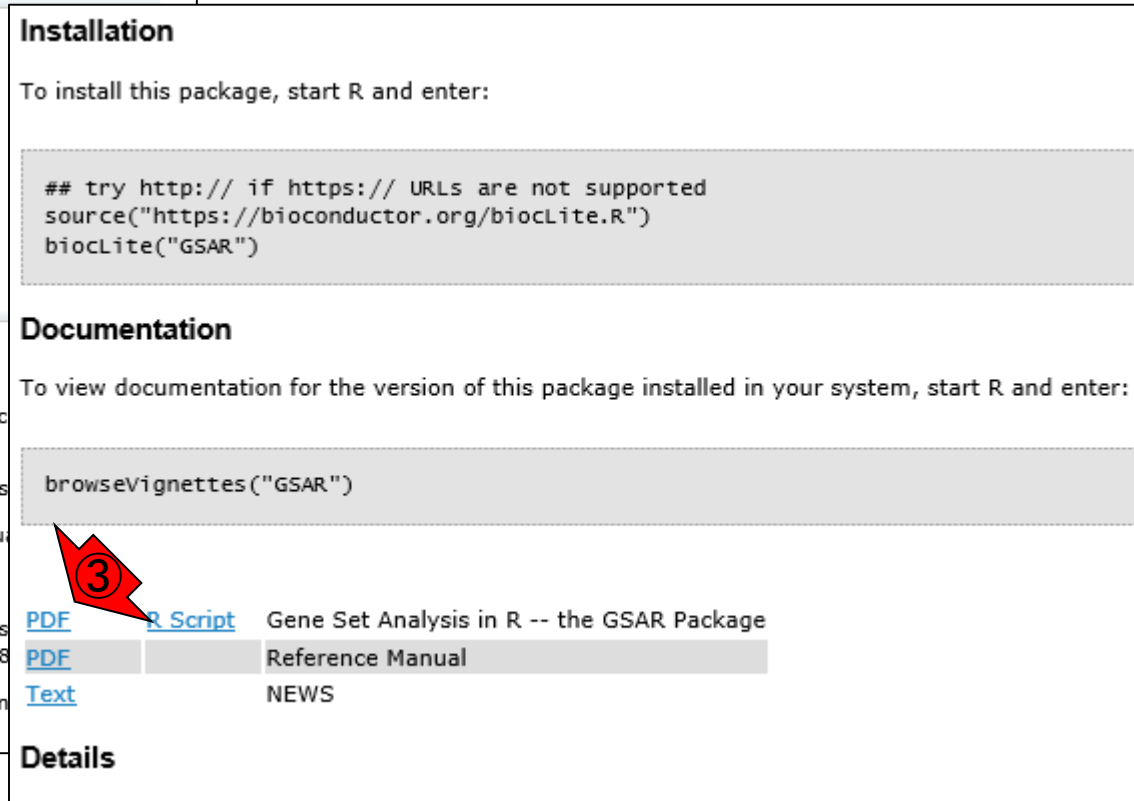
Author: Yasir Rahmatallah <yrahmatallah at uams.edu>, Galina Glazko <gvglazko at uams.edu>

Maintainer: Yasir Rahmatallah <yrahmatallah at uams.edu>, Galina Glazko <gvglazko at uams.edu>

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

Rahmatallah Y, Emmert-Streib F and Glazko G (2014). "Gene sets net correlations analysis multivariate differential coexpression test for gene sets." *Bioinformatics*, **30**, pp. 360-368

Rahmatallah Y, Emmert-Streib F and Glazko G (2012). "Gene set analysis for self-contained complex null and specific alternative hypotheses." *Bioinformatics*, **28**, pp. 3073-3080.



Installation

To install this package, start R and enter:

```
## try http:// if https:// URLs are not supported  
source("https://bioconductor.org/biocLite.R")  
biocLite("GSAR")
```

Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("GSAR")
```

[PDF](#) [R Script](#) Gene Set Analysis in R -- the GSAR Package
[PDF](#) Reference Manual
[Text](#) NEWS

Details

こんな感じのPDFが開きます。①のCase studiesを眺めて、どのように使うのかを学習する。②の部分を次のスライドで示す

まずはマニュアルを眺め

Gene Set Analysis in R – the GSAR Package

Yasir Rahmatallah¹ and Galina Glazko²

Department of Biomedical Informatics,
University of Arkansas for Medical Sciences,
Little Rock, AR 72205.

¹yrahmatallah@uams.edu, ²gvglazko@uams.edu

GSAR version 1.10.0 (Last revision 2017-01-03)

Contents

1	Introduction	2
2	Minimum spanning trees	3
2.1	First MST	3
2.2	MST2 for correlation and PPI networks	4
3	Statistical methods	4
3.1	Wald-Wolfowitz test	4
3.2	Kolmogorov-Smirnov tests	5
3.3	Mean deviation tests	6
3.4	Aggregated F-test	7
3.5	Gene sets net correlations analysis	7
3.5.1	Method	7
3.5.2	The problem of zero standard deviation	8
4	Notes on handling RNA-seq count data	8
5	Case studies	8
5.1	The p53 dataset	8
5.1.1	Introduction	8
5.1.2	Filtering and normalization	9
5.1.3	GSA	9
5.2	The ALL dataset	12
5.2.1	Introduction	12
5.2.2	Filtering and normalization	12
5.2.3	Selected gene set	13
5.3	The Pickrell dataset	14
5.3.1	Introduction	14
5.3.2	Filtering and normalization	16
5.3.3	Testing selected pathways	17
6	Session in	--

①

②

p53データセット解析例

①p53データセットです。②Affymetrix GeneChipを用いて得られた発現情報で、③log変換後のデータです。④「17 WT vs. 33 MUT」の2群間比較

5 Case studies

This Section illustrates the typical procedure for applying the methods available in package *GSAR* to perform GSA. Two microarray and one RNA-seq datasets are used.

5.1 The p53 dataset



5.1.1 Introduction

p53 is a major tumor suppressor protein. The p53 dataset comprises 50 samples of the NCI-60 cell lines differentiated based on the status of the TP53 gene: 17 cell lines carrying wild type (WT) TP53 and 33 cell lines carrying mutated (MUT) TP53 [8, 9]. Transcriptional profiles obtained from microarrays of platform hgu95av2 are available from the Broad Institute's website (<http://www.broadinstitute.org/gsea/datasets.jsp>).



5.1.2 Filtering and normalization

A preprocessed version of p53 dataset is available in package *GSAR* as a *matrix* object. The p53 dataset was downloaded from the Broad Institute's website. Probe level intensities were quantile normalized and transformed to the log scale using $\log_2(1 + intensity)$. Probes originally had Affymetrix identifiers which are mapped to unique gene symbol identifiers. Probes without ③ mapping to entrez and gene symbol identifiers were discarded. Probes with duplicate intensities were assessed and ② probe with the largest absolute value of t-statistic between WT and MUT conditions was selected as the gene match. Finally, genes were assigned gene symbol identifiers and columns were assigned names indicating whether they belong to WT or MUT group. The columns were sorted such that the first 17 columns are WT samples and the next 33 columns are the MUT samples. This processed version of the p53 dataset was used in the analysis presented in [1].



p53データセット解析例

①「5.1.3 GSA」のところ。Rコードがば
 ーっと書かれていて、私でも解読に
 時間がかかるところ。②遺伝子セット
 (“LU_TUMOR_VASCULATURE_UP”)
 に対して、③GSNCA法を適用して云
 々、のところを書き直したのが…

5.1.3 GSA

GSA is performed on selected C2 curated gene sets (pathways) of the *molecular signatures database* (MSigDB) 3.0 [10]. This list of gene sets is available in package *GSVAdata*. We start by loading the required data

```
> library(GSVAdata)
> data(p53DataSet)
> data(c2BroadSets)
```

c2BroadSets is an object of class *GeneSetCollection* supported by package *GSEABase*. Its object have *entrez* identifiers. Package *org.Hs.eg.db* is used to convert the *entrez* identifiers. Genes without unique mapping to gene symbol identifiers or that do not exist from the C2 pathways. This insures proper indexing of genes in the dataset by the *entrez* identifiers. Finally, we keep only pathways with $10 \leq p \leq 500$ where p is the number of genes in the pathway.

```
> library(org.Hs.eg.db)
> library(GSEABase)
> C2 <- as.list(geneIds(c2BroadSets))
> len <- length(C2)
> genes.entrez <- unique(unlist(C2))
> genes.symbol <- array("",c(length(genes.entrez),1))
> x <- org.Hs.egSYMBOL
> mapped_genes <- mappedkeys(x)
> xx <- as.list(x[mapped_genes])
> for (ind in 1:length(genes.entrez)){
+   if (length(xx[[genes.entrez[ind]]])!=0)
+     genes.symbol[ind] <- xx[[genes.entrez[ind]]]
+ }
> ## discard genes with no mapping to gene symbol identifiers
> genes.no.mapping <- which(genes.symbol == "")
> if(length(genes.no.mapping) > 0){
+   genes.entrez <- genes.entrez[-genes.no.mapping]
+   genes.symbol <- genes.symbol[-genes.no.mapping]
+ }
```

```
> names(genes.symbol) <- genes.entrez
> ## discard genes in C2 pathway
> p53genes <- rownames(p53DataSet)
> remained <- array(0,c(1,len))
> for (k in seq(1, len, by=1)) {
+   remained[k] <- sum((genes.symbol[C2[[k]]] %in% p53genes) &
+   (C2[[k]] %in% genes.entrez))
+ }
> ## discard C2 pathways which have less than 10 or more than 500 genes
> C2 <- C2[(remained>=10)&&(remained<=500)]
> pathway.names <- names(C2)
```

```
> c2.pathways <- list()
> for (k in seq(1, length(C2), by=1)) {
+   selected.genes <- which(p53genes %in% genes.symbol[C2[[k]])
+   c2.pathways[[length(c2.pathways)+1]] <- p53genes[selected.genes]
+ }
> names(c2.pathways) <- pathway.names
> path.index <- which(names(c2.pathways) == "LU_TUMOR_VASCULATURE_UP")
```

c2.pathways is now a list with each entry being a named list of the genes (gene symbol identifiers) forming one C2 pathway. To demonstrate the use of different tests, we consider the C2 pathway LU TUMOR VASCULATURE UP used in [1] to illustrate GSNCA.

```
> target.pathway <- p53Data[[c2.pathways[["LU_TUMOR_VASCULATURE_UP"]],]
> group_label <- c(rep(1, length(target.pathway)), rep(2, 33))
```

①例題1。私のウェブページの記述形式に慣れているヒトは、この中の記述を眺めるほうが、より理解しやすいのでは…

p53データセット解析例

解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah 2017\)](#)

[GSAR](#)パッケージを用いた解析のやり方を示します。GSARは、様々なプログラム群からなるパッケージです。ここでは、GSARで提供されている

Gene Sets Net CoreにID変換がなされ、[probeset ID](#)と[gene symbol](#)でID変換された[gmtファイル](#)の読み込み「ファイル」→「ディレクトリ」

①

1. [GSAR](#)の解析例として示されているp53 datasetの場合:

p53DataSetという名前の8,655 genes×50 samplesのデータです。17 wild-type (WT) samples vs. 33 mutant (MUT) samplesの2群間比較です。gmtファイルは、[MSigDB](#) (ver. 5.2)の [C2: curated gene sets](#) 中の全データファイル([c2.all.v5.2.symbols.gmt](#))を利用しています。この中に解析例で取り扱っている遺伝子セット("LU_TUMOR_VASCULATURE_UP")が含まれています。ここでは、"LU_TUMOR_VASCULATURE_UP"を構成する遺伝子群(29 genes)の中から、発現データ中に存在するもの(22 genes)に対してGSNCA法を適用した結果のp-value (= 0.02797)を出力するところまでを示します。p-value = 0.02797という結果は、比較する2群間(WT vs. MUT)でこの遺伝子セットの発現の相関に差がない(帰無仮説)確率が2.797%しかないことを意味します。1つの判断基準として、(建前上)有意水準 $\alpha = 0.05$ を設定していたのなら、帰無仮説が本当は正しいが間違っして棄却してしまう確率を5%に設定したことと相当し、この場合はp-value < 0.05なので帰無仮説を棄却し対立仮説(差がある)を採用することになります。

```
in_f <- "c2.all.v5.2.symbols.gmt" #入力ファイル名を指定してin_fに格納(gmtファイル)
param_G1 <- 17 #G1群のサンプル数を指定
param_G2 <- 33 #G2群のサンプル数を指定
param <- "LU_TUMOR_VASCULATURE_UP" #解析したい遺伝子セット名を指定

#必要なパッケージをロード
library(GSAR) #パッケージの読み込み
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
hoge <- GSA.read.gmt(in_f) #in_f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ました、となって)
gmt <- hoge$genesets #フォーマット変換
names(gmt) <- hoge$geneset.names #フォーマット変換

data(p53DataSet) #GSARパッケージ中のp53DataSetオブジェクトの呼び出し
data <- p53DataSet #オブジェクト名の変更(dataとして取り扱う)
rownames(data) <- toupper(rownames(data)) #gene symbolを大文字に変換している (gmtファイルに合わせるため)
dim(data) #確認してるだけです
data.c1 <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2としたベクトルdata.c1を作成

#GSAR本番(正確にはGSARから提供されているGSNCA法の実行)
set.seed(1053) #おまじない (同じ乱数になるようにするため)
```

p53データセット解析例

①のgmtファイルが存在するディレクトリ上で、コピペ実行した結果。②GSARマニュアルと同じく、低いp-value

1. GSARの解析例として示されているp53 datasetの場合:

p53DataSetという名前の8,655 genes×50 samplesのデータです。17 wild-type (WT) samples vs. 33 mutant (MUT) samplesの2群間比較です。gmtファイルは、MSigDB (ver. 5.2)の C2: curated gene sets中の全データファイル(c2.all.v5.2.symbols.gmt)を利用しています。この中に解析例で取り扱っている遺伝子セット("LU_TUMOR_VASCULATURE_UP")が含まれています。ここでは、"LU_TUMOR_VASCULATURE_UP"を構成する遺伝子群(29 genes)の中から、発現データ中に存在するもの(22 genes)に対してGSNCA法を適用した結果のp-value (= 0.02797)を出力するところまでを示します。p-value = 0.02797という結果は、比較する2群間(WT vs. MUT)でこの遺伝子セットの発現の相関に差がない(帰無仮説)確率が2.797%しかないことを意味します。1つの判断基準として、(建前上)有意水準 $\alpha = 0.05$ を設定していたのなら、帰無仮説が本当は正しいが間違っただけで棄却してしまう確率を5%に設定したことに相当し、この場合はp-value < 0.05なので帰無仮説を棄却、対立仮説(差がある)を選択することになります。

```
in_f <- "c2.all.v5.2.symbols.gmt" #入力ファイル
param_G1 <- 17 #G1群のサンプル数
param_G2 <- 33 #G2群のサンプル数
param <- "LU_TUMOR_VASCULATURE_UP" #解析したい遺伝子セット名

#必要なパッケージをロード
library(GSAR) #GSARパッケージ
library(GSA) #GSA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
hoge <- GSA.read.gmt(in_f) #in_fで指定したgmtファイルを読み込み
gmt <- hoge$genesets #フォーマットされた遺伝子セット
names(gmt) <- hoge$geneset.names #遺伝子セット名をラベルとして設定

data(p53DataSet) #GSARパッケージからデータを読み込み
data <- p53DataSet #オブジェクトとして取得
rownames(data) <- toupper(rownames(data)) #gene symbolsに統一
dim(data) #確認してデータサイズを確認
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #GSAR本番(正確にはGSARから提供されているGSNCA法の実行)

#GSAR本番(正確にはGSARから提供されているGSNCA法の実行)
set.seed(1053) #おまじない
```

```
R Console
> data(p53DataSet) #GSARパッケージ
> data <- p53DataSet #オブジェクト
> rownames(data) <- toupper(rownames(data)) #gene symbols
> dim(data) #確認してデータサイズを確認
[1] 8655 50
> data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #GSAR本番(正確にはGSARから提供されているGSNCA法の実行)
> #GSAR本番(正確にはGSARから提供されているGSNCA法の実行)
> set.seed(1053) #おまじない
> hoge <- gmt[[param]] #解析したい遺伝子セット
> length(hoge) #遺伝子数を数える
[1] 29
> obj <- is.element(rownames(data), hoge) #条件判定
> sum(obj) #遺伝子数を数える
[1] 22
> p.value <- GSNCAtest(data[obj, ], data.cl) #GSNCA法を実行
> p.value #p-valueを出力
[1] 0.02797203
```


GSARマニュアル再訪

GSARマニュアルのp10。①GSARは多くの発現変動遺伝子セット検定法を提供している。②検定結果(p-value)。この値とその後の記述内容から、③GSNCA法を使ってもらいたいと忖度する

```
> target.pathway <- p53DataSet[c2.pathways[["LU_TUMOR_VASCULATURE_UP"]],]  
> group.label <- c(rep(1,17), rep(2,33))  
> WW_pvalue <- WWtest(target.pathway, group.label)  
> KS_pvalue <- KStest(target.pathway, group.label)  
> MD_pvalue <- MDtest(target.pathway, group.label)  
> RKS_pvalue <- RKStest(target.pathway, group.label)  
> RMD_pvalue <- RMDtest(target.pathway, group.label)  
> F_pvalue <- AggrFtest(target.pathway, group.label)  
> GSNCA_pvalue <- GSNCAtest(target.pathway, group.label)
```

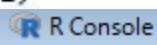


```
> WW_pvalue  
[1] 0.12  
> KS_pvalue  
[1] 0.049  
> MD_pvalue  
[1] 0.0529  
> RKS_pvalue  
[1] 0.127  
> RMD_pvalue  
[1] 0.192  
> F_pvalue  
[1] 0.877  
> GSNCA_pvalue  
[1] 0.032
```



GSARマニュアル再訪

マニュアル中の値と似た結果になっていることを確認して安心する。p値が低いという結果は、①遺伝子セット“LU_TUMOR_VASCULATURE_UP”が「17 WT vs. 33 MUT」の2群間比較で発現変動しているっぽいことを意味する。値の違いは、プログラム内部での乱数の振り方の違いや、gmtファイルのバージョンの違いなどが考えられる



```
> target.pathway <- p53DataSet[c2.paths[[ "LU_TUMOR_VASCULATURE_UP"
> group.label <- c(rep(1,17), rep(2,33))
> WW_pvalue <- WWtest(target.pathway, group.label)
> KS_pvalue <- KStest(target.pathway, group.label)
> MD_pvalue <- MDtest(target.pathway, group.label)
> RKS_pvalue <- RKStest(target.pathway, group.label)
> RMD_pvalue <- RMDtest(target.pathway, group.label)
> F_pvalue <- AggrFtest(target.pathway, group.label)
> GSNCA_pvalue <- GSNCAtest(target.pathway, group.label)
> WW_pvalue
[1] 0.12
> KS_pvalue
[1] 0.049
> MD_pvalue
[1] 0.0529
> RKS_pvalue
[1] 0.127
> RMD_pvalue
[1] 0.192
> F_pvalue
[1] 0.877
> GSNCA_pvalue
[1] 0.032
```

```
R Console
> data(p53DataSet) #GSARパッケージ
> data <- p53DataSet #オブジェクト
> rownames(data) <- toupper(rownames(data)) #gene sym
> dim(data) #確認してる
[1] 8655 50
> data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #GS
>
> #GSAR本番 (正確にはGSARから提供されているGSNCA法の$
> set.seed(1053) #おまじない$
> hoge <- gmt[[param]] #解析したい$
> length(hoge) #遺伝子数を$
[1] 29
> obj <- is.element(rownames(data), hoge) #条件判定
> sum(obj) #遺伝子数を$
[1] 22
> p.value <- GSNCAtest(data[obj, ], data.cl) #GSNCAを$
> p.value #p-valueを$
[1] 0.02797203
> |
```

相関ネットワーク図

Pathway: LU_TUMOR_VASCULATURE_UP

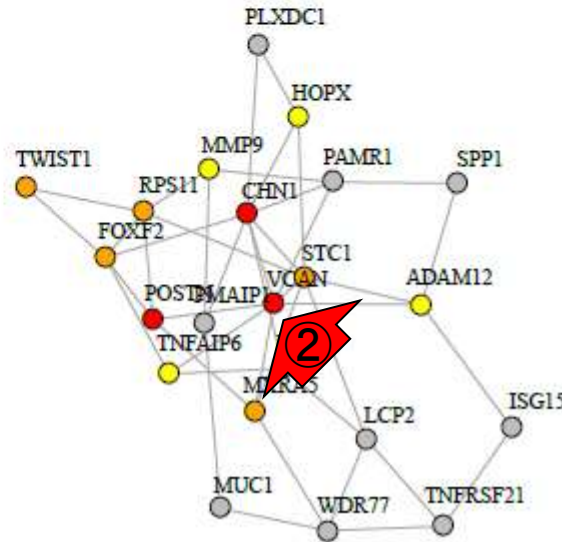
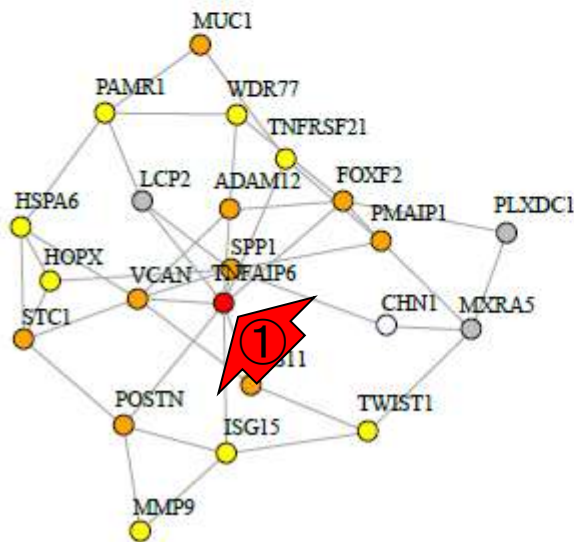
There are 22 genes in this pathway

Group 1
 Hub Gene (Group 1): TNFAIP6 ①
 Weight Factor: 1.362
 Hub Gene (Group 2): VCAN ②
 Weight Factor: 1.165

Group 2
 Hub Gene (Group 2): VCAN ②
 Weight Factor: 1.465
 Hub Gene (Group 1): TNFAIP6 ①
 Weight Factor: 0.979



■ $w > 1.5$ ■ $1.25 < w < 1.5$ ■ $1 < w < 1.25$ ■ $0.75 < w < 1$ ■ $0.5 < w < 0.75$ □ $w < 0.5$



GSARマニュアルのp11。この遺伝子セットの中で、発現変動に最も寄与している?!遺伝子 (hub genes) が示されるのがGSNCA法のポイント。この場合は、①Group1(WT)がTNFAIP6、②Group2(MUT)がVCANであることがわかる

相関ネットワーク図

解析 | 機能解析 | パスウェイ(Pathway)解析 | GSAR (Rahmatallah_2017)

[GSAR](#)パッケージを用いた解析のやり方を示します。GSARは、様々なプログラム群からなるパッケージです。ここでは、GSARで提供されているGene Sets Net Correlations Analysis法(GSNCA; [Rahmatallah et al., 2014](#))を実行するやり方を示します。オリジナルのprobeset IDからgene symbolにID変換がなされた発現データファイルを入力としています。これは、「イントロ | アンテーション情報取得 | [GEOquery\(Davis 2007\)](#)」で入手したprobeset IDとgene symbolの対応表のファイル、およびオリジナルのprobeset IDごとの発現データファイルを用いて「前処理 | ID変換 | [probe ID -> gene symbol](#)」でID変換したものです。gmtファイルは、[MSigDB \(Subramanian et al., PNAS, 2005\)](#)から任意のものをダウンロードしてください。尚、gmtファイルの読み込みには[GSA](#)「ファイル」-「ディレクトリの変更」



3. [GSAR](#)の解析例として示されているp53 datasetの場合:

2.と基本的に同じですが、相関ネットワーク(correlation network)図をpngファイルに保存しています。

```

in_f <- "c2.all.v5.2.symbols.gmt" #入力ファイル名を指定してin_fに格納(gmtファイル)
out_f <- "hoge3.png" #出力ファイル名を指定してout_fに格納
param_G1 <- 17 #G1群のサンプル数を指定
param_G2 <- 33 #G2群のサンプル数を指定
param <- "LU_TUMOR_VASCULATURE_UP" #解析したい遺伝子セット名を指定
param_fig <- c(700, 650) #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(GSAR) #パッケージの読み込み
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
hoge <- GSA.read.gmt(in_f) #in_fで指定したファイルの読み込み(不完全な最終行が)
gmt <- hoge$genesets #フォーマット変換
names(gmt) <- hoge$geneset.names #フォーマット変換

data(p53DataSet) #GSARパッケージ中のp53DataSetオブジェクトの呼び出
data <- p53DataSet #オブジェクト名の変更(dataとして取り扱う)
rownames(data) <- toupper(rownames(data)) #gene symbolを大文字に変換している (gmtファイル)
dim(data) #確認してるだけです
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
    
```



相関ネットワーク図

3. [GSAR](#)の解析例として示されているp53 datasetの場合:

2.と基本的に同じですが、相関ネットワーク(correlation network)図をpngファイルに保存しています。

```
in_f <- "c2.all.v5.2.symbols.gmt" #入力ファイル名を指定してin_fに格納(gmtファイル)
out_f <- "hoge3.png" #出力ファイル名を指定してout_fに格納
param_G1 <- 17 #G1群のサンプル数を指定
param_G2 <- 33 #G2群のサンプル数を指定
param <- "LU_TUMOR_VASCULATURE_UP" #解析したい遺伝子セット名を指定
param_fig <- c(700, 650) #ファイル出力サイズを指定
```

#必要なパッケージをロード

```
library(GSAR) #パッケージをロード
library(GSA) #パッケージをロード
```

#入力ファイルの読み込みとラベル情報の作成

```
hoge <- GSA.read.gmt(in_f) #in_fで指定したgmtファイルを読み込み
gmt <- hoge$genesets #フォーマットした遺伝子セット
names(gmt) <- hoge$geneset.names #フォーマットした遺伝子セット名
```

```
data(p53DataSet) #GSARパッケージのデータセットを読み込み
data <- p53DataSet #オブジェクトとして格納
rownames(data) <- toupper(rownames(data)) #gene symbolを大文字に変換
dim(data) #確認してデータサイズを確認
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #条件判定用のベクトル
```

```
[1] 29
> obj <- is.element(rownames(data), hoge) #条件判定
> sum(obj) #遺伝子数を$
[1] 22
> p.value <- GSNCAtest(data[obj, ], data.cl) #GSNCAを$
> p.value #p-valueを$
[1] 0.02797203
> #ファイルに保存(相関ネットワーク(correlation network)図をpngファイルに保存)
> png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])
> plotMST2.pathway(object=data[obj, ], #相関ネットワーク図を生成
+ group=data.cl, name=param, #相関ネットワーク図のグループと名前
+ legend.size=1.2, leg.x=-1.2, #相関ネットワーク図の凡例のサイズと位置
+ leg.y=2, label.size=1, #相関ネットワーク図の凡例のラベルのサイズ
+ label.dist=0.8, cor.method="pearson") #相関ネットワーク図の凡例のラベルの距離と相関方法
> dev.off() #おまじない
null device
      1
> |
```


相関ネットワーク図

①pngファイル。②ネットワークのトポロジーが若干異なっているが、③数値は同じ。こんな感じでマニュアルと同じような結果が得られることを確認した後で、自分のデータに適用する

3. GSARの解析例として示されているp53 datasetの場合:

2.と基本的に同じですが、相関ネットワーク(correlation network)図をpngファイルに保存しています。

```
in_f <- "c2.all.v5.2.symbols.gmt" #入力
out_f <- "hoge3.png" #出力
param_G1 <- 17 #G1値
param_G2 <- 33 #G2値
param <- "LU_TUMOR_VASCULATURE_UP" #解析
param_fig <- c(700, 650) #フ
```



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

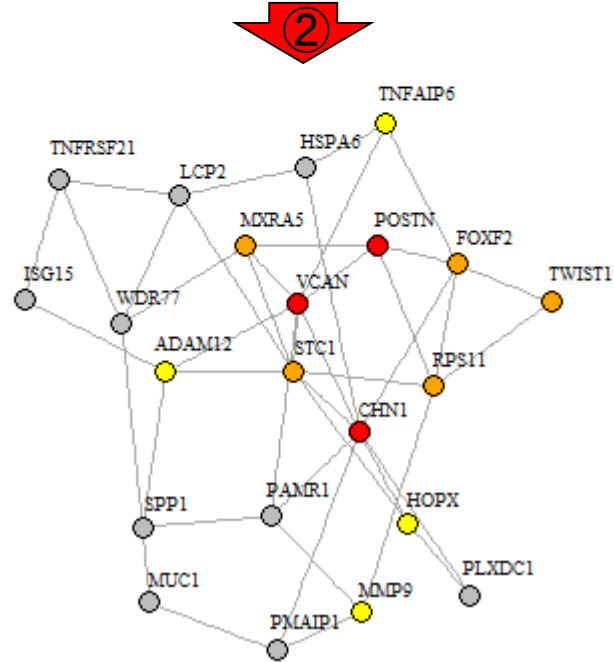
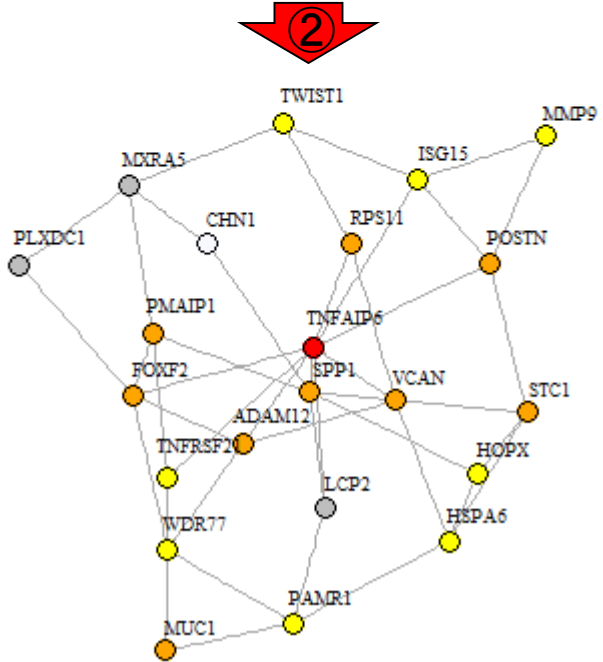
```
#入力ファイルの読み込みとラベル情報の作成
hoge <- GSA.read.gmt(in_f) #in_
gmt <- hoge$genesets #フォ
names(gmt) <- hoge$geneset.names #フォ
```

```
data(p53DataSet) #GSA
data <- p53DataSet #オ
rownames(data) <- toupper(rownames(data)) #g
dim(data) #確認
data.cl <- c(rep(1, param_G1), rep(2, param
```

Pathway: LU_TUMOR_VASCULATURE_UP

There are 22 genes in this pathway

Group 1	Group 2
Hub Gene (Group 1): TNFAIP6	Hub Gene (Group 2): VCAN
Weight Factor: 1.362	Weight Factor: 1.465
Hub Gene (Group 2): VCAN	Hub Gene (Group 1): TNFAIP6
Weight Factor: 1.165	Weight Factor: 0.979



Contents

- 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(反復あり)
- 反復なし多群間比較(教科書p182-188)
 - limmaパッケージを用いた3群間比較(反復なし)
 - 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(probeset ID → gene symbol)
 - GSAパッケージを用いた遺伝子セット解析
 - GSARパッケージを用いた遺伝子セット解析とネットワーク図の作成
 - パッケージのマニュアル中にある例題データの解析
 - GSA実行結果として得られた発現変動遺伝子セットの解析

復習。②例題3をテンプレートとして、③入力ファイル名部分のみをdata_mas_EN_symbol.txtに変更してBAT_fed vs. BAT_fasのGO解析をやった結果が

GSAでGO解析

(Rで)マイクロアレイデータ解析

- (last modified 2015/06/07) NEW
- 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#) (last modified 2015/06/07) NEW
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GAGE \(Luo 2009\)](#) (last modified 2014/06/01)
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#) (last modified 2014/06/03) 推奨
 - 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [Category \(Jiang 2009\)](#) (last modified 2014/06/01)

What's new?

- 門田幸二先生の最近の講義資料

解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron_2007\)](#) NEW

GSAパッケージを使った解析のやり方を示します

3. 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。BAT_fed vs. BAT_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。

1. サンプルデータ

肝臓のみからなるデータのprobeset ID:

```
in_f1 <- "data_rma_2_nr.txt"
in_f2 <- "c5.bp.v5.0.symbols.gmt"
out_f1 <- "hoge3_G1.txt"
out_f2 <- "hoge3_G2.txt"
param_G1 <- 4
param_G2 <- 4
param_FDR <- 0.1
param_posi <- c(1:4, 5:8)
```

```
#必要なパッケージをロード
library(GSA)
```

```
#入力ファイルを読み込み
gmt <- GSA.read.gmt(in_f2)
```

```
in_f1 <- "data_rma_2_nr.txt"
in_f2 <- "c5.bp.v5.0.symbols.gmt"
out_f1 <- "hoge3_G1.txt"
out_f2 <- "hoge3_G2.txt"
param_G1 <- 4
param_G2 <- 4
param_FDR <- 0.1
param_posi <- c(1:4, 5:8)
```

```
#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)
```

```
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み込み (不完全な最終行は除外)
```

```
rownames(data) <- toupper(rownames(data)) #IDを大文字に変換している(gmtファイルに合わせた)
```

```
data.c1 <- c(rep(1, param_G1), rep(2, param_G2)) #G1群を1、G2群を2としたベクトルdata.c1
```

```
data <- data[,param_posi]
```

```
colnames(data)
```

```
#GSA本番
```

```
out <- GSA(data, data.c1,
```

```
#入力ファイル名を指定してin_f1に格納(発現データ)
#入力ファイル名を指定してin_f2に格納(gmtファイル)
#出力ファイル名を指定してout_f1に格納(G1群で高発現)
#出力ファイル名を指定してout_f2に格納(G2群で高発現)
#G1群のサンプル数を指定
#G2群のサンプル数を指定
#DEG検出時のfalse discovery rate (FDR)閾値を指定
#G1群およびG2群の位置情報を指定
```

```
#パッケージの読み込み
```

```
#in_f2で指定したファイルの読み込み (不完全な最終行は除外)
```

```
#in_f1で指定したファイルの読み込み (不完全な最終行は除外)
```

```
#G1群を1、G2群を2としたベクトルdata.c1
```

```
#サブセットを抽出
#確認してるだけです
```

```
#GSAを実行し、結果をoutに格納
```

GSAでGO解析

これだった。出力は、FDR 10%の閾値を満たす有意な遺伝子セットリスト。数値の分布的にG1群で高発現の遺伝子セットである①**STEROID_BIOSYNTHETIC_PROCESS**が最も変動しているのだから、ということで、この相関ネットワーク図を描いてみる

Gene_set	Gene_set_name	Score	p-value	FDR
22	PHOSPHOLIPID_BIOSYNTHETIC_PROCESS	-0.5692	0	0
56	REGULATION_OF_DNA_BINDING	-0.3102	0	0
92	COFACTOR_METABOLIC_PROCESS	-0.4346	0	0
134	BIOSYNTHETIC_PROCESS	-0.2019	0	0
147	INTRACELLULAR_PROTEIN_TRANSPORT	-0.1555	0	0
205	PROTEIN_IMPORT	-0.2265	0	0
264	MEMBRANE_LIPID_BIOSYNTHETIC_PROCESS	-0.4554	0	0
287	LIPID_METABOLIC_PROCESS	-0.2399	0	0
317	LIPID_BIOSYNTHETIC_PROCESS	-0.7518	0	0
331	REGULATION_OF_TRANSCRIPTION_FACTOR_ACTIVITY	-0.295	0	0
370	MITOCHONDRION_ORGANIZATION_AND_BIOGENESIS	-0.3251	0	0
383	NEGATIVE_REGULATION_OF_DNA_BINDING	-0.6935	0	0
415	NEGATIVE_REGULATION_OF_CELLULAR_COMPONENT	-0.2769	0	0
466	CELLULAR_RESPONSE_TO_STIMULUS	-0.4359	0	0
471	INTERACTION_WITH_HOST	-0.389	0	0
490	RIBONUCLEOTIDE_METABOLIC_PROCESS	-0.6475	0	0
541	CELLULAR_LIPID_METABOLIC_PROCESS	-0.2414	0	0
607	NEGATIVE_REGULATION_OF_BINDING	-0.6935	0	0
609	PHOSPHOLIPID_METABOLIC_PROCESS	-0.2709	0	0
619	REGULATION_OF_CYTOSKELETON_ORGANIZATION_AND_BIOGENESIS	-0.2051	0	0
667	NEGATIVE_REGULATION_OF_TRANSCRIPTION_FACTOR_ACTIVITY	-0.804	0	0
669	RIBOSOME_BIOGENESIS_AND_ASSEMBLY	-0.3904	0	0
721	PROTEIN_TRANSPORT	-0.1425	0	0
762	STEROID_BIOSYNTHETIC_PROCESS	-1.2465	0	0



Gene_set	Gene_set_name	Score	p-value	FDR
39	POSITIVE_REGULATION_OF_SIGNAL_TRANSDUCTION	0.1897	0	0
121	PATTERN_SPECIFICATION_PROCESS	0.4294	0	0
461	PROTEIN_KINASE_CASCADE	0.1781	0	0
783	ACTIVATION_OF_PROTEIN_KINASE_ACTIVITY	0.6607	0	0

GSARでGSNCAを実行

(Rで)マイクロアレイデータ解析

(last modified 2017/04/19, since 2005)

What's new?

- 「解析 | 機能解析」するだけでなく
- 「解析 | 発現算術平均値」
- 門田幸二 著, al., 2006). W「書籍 | トラン」
- お知らせは主の参考資料

- ・ 解析 | 発現変動 | 時系列 | non-periodic genes | [maSigPro \(Conesa 2006\)](#) (last modified 2015/08/16)
- ・ 解析 | 発現変動 | 時系列 | non-periodic genes | [SAM \(Tusher 2001\)](#) (last modified 2015/08/16)
- ・ 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSAR \(Rahmatallah 2017\)](#) (last modified 2017/04/19) **NEW**
- ・ 解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [gage \(Luo 2009\)](#) (last modified 2015/05/25)

解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | GSAR (Rahmatallah_2017) NEW

GSARパッケージを用いた解析のやり方を示します。GSARは、様々なプログラム群からなるパッケージです。ここでは、GSARで提供されているGene Sets Net Correlations Analysis法(GSNCA; [Rahmatallah et al., 2014](#))を実行するやり方を示します。オリジナルのprobeset IDからgene symbolにID変換がなされた発現データファイルを入力としています。これは、「イントロ | アノテーション情報取得 | [GEOquery\(Davis 2007\)](#)」で入手したprobeset IDとgene symbolの対応表のファイル、およびオリジナルのprobeset IDごとの発現データファイルを用いて「前処理 | ID変換 | [probe ID -> gene symbol](#)」でID変換したものです。gmtファイルは、[MSigDB \(Subramanian et al., PNAS, 2005\)](#)から任意のものをダウンロードしてください。尚、gmtファイルの読み込みにはGSAパッケージ ([Efron and Tibshirani, 2007](#))中のGSA.read.gmt関数を利用しています。「ファイル」-「ディレクトリの変更」で解析したいファイルを置いてあるディレクトリに移動し以下をコピー。

1. data_mas_EN_symbol.txtの場合:

14,132 genes×24 samplesのデータです。BAT_fed vs. BAT_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。「解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah 2017\)](#)」の例題7と「解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#)」の例題3を合わせたものです。ここでは、「STERIOD_BIOSYNTHETIC_PROCESS」という遺伝子セットのネットワーク図をpngファイルに出力させています。

```
in_f1 <- "data_mas_EN_symbol.txt"
in_f2 <- "c5.bp.v5.0.symbols.gmt"
out_f <- "hoge1.png"
param_G1 <- 4
param_G2 <- 4
param_FDR <- 0.1
param_posi <- c(1:4, 5:8)
param <- "STERIOD_BIOSYNTHETIC_PROCESS"
param_fig <- c(700, 650)
```

- #入力ファイル名を指定してin_f1に格納(発現データファイル)
- #入力ファイル名を指定してin_f2に格納(gmtファイル)
- #出力ファイル名を指定してout_fに格納
- #G1群のサンプル数を指定
- #G2群のサンプル数を指定
- #false discovery rate (FDR)閾値を指定
- #G1群およびG2群の位置情報を指定
- #解析したい遺伝子セット名を指定
- #ファイル出力時の横幅と縦幅を指定(単位はピクセル)

GSARでGSNCAを実行

実行結果。①p-valueが非常に高い(1に近いので有意とはとてもいえない)ことがわかる。GSAでは有意(10% FDR)であっても、GSNCAでは有意ではないということなのだろう

1. data_mas_EN_symbol.txtの場合:

14,132 genes×24 samplesのデータです。BAT_fed vs. BAT_fastedの2群間比較を行うべく、それらの位置情報から「解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah 2017\)](#)」の例題7と「解析 | 機能解析 | トロロジー(GO)解析 | [GSA \(Efron 2007\)](#)」の例題3を合わせたものです。ここでは、「**STEROID_BIOSYNTHETIC_PROCESS**」という遺伝子セットのネットワーク図をpngファイルに出力させています。

```
in_f1 <- "data_mas_EN_symbol.txt" #入力ファイル名を指定してin_f1に格納(発現データファイル)
in_f2 <- "c5.bp.v5.0.symbols.gmt" #入力ファイル名を指定してin_f2に格納(gmtファイル)
out_f <- "hoge1.png" #出力ファイル名を指定してout_fに格納(pngファイル)
param_G1 <- 4 #G1群のサンプル数を指定してparam_G1に格納
param_G2 <- 4 #G2群のサンプル数を指定してparam_G2に格納
param_FDR <- 0.1 #false discovery rateを指定してparam_FDRに格納
param_posi <- c(1:4, 5:8) #G1群およびG2群の遺伝子位置を指定してparam_posiに格納
param <- "STEROID_BIOSYNTHETIC_PROCESS" #解析したい遺伝子セットの名前を指定してparamに格納
param_fig <- c(700, 650) #ファイル出力時の幅と高さを指定してparam_figに格納

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

#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_f1, header=TRUE, row.names=colnames(data)) #IDを大文字に変換
rownames(data) <- toupper(rownames(data)) #IDを大文字に変換
data <- as.matrix(data) #型変換(Te
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #サブセットを指定してdata.clに格納
data <- data[,param_posi] #サブセットを指定してdataに格納
colnames(data) #確認して
```

```
[1] 23
> obj <- is.element(rownames(data), hoge) #条件判定
> sum(obj) #遺伝子数を$
[1] 19
> p.value <- GSNCAtest(data[obj, ], data.cl) #GSNCAを$
> p.value #p-valueを$
[1] 0.7462537 ①
> #ファイルに保存(相関ネットワーク(correlation network)を生成)
> png(out_f, pointsize=13, width=param_fig[1], height=param_fig[2])
> plotMST2.pathway(object=data[obj, ], #相関ネット$
+ group=data.cl, name=param, #相関ネット$
+ legend.size=1.2, leg.x=-1.2, #相関ネット$
+ leg.y=2, label.size=1, #相関ネット$
+ label.dist=0.8, cor.method="pearson") #相関ネット$
> dev.off() #おまじない
null device
      1
> |
```

GSARでGSNCAを実行

① 相関ネットワーク図。p53データセットのときと違って、② weight factorが赤色のものがない。③ それでもhub genesは表示されるようだ

1. [data_mas_EN_symbol.txt](#)の場合:

14,132 genes×24 samplesのデータです。BAT_fed vs. BAT_fastedの2群間比較を行うべく、それらの位置情報で指定されています。「解析 | 機能解析 | パスウェイ(Pathway)解析 | [GSAR \(Rahmatallah 2017\)](#)」の例題7と「解析 | 機能解析 | 遺伝子オントロジー(GO)解析 | [GSA \(Efron 2007\)](#)」の例題3を合わせたものは、「**STERIOD_BIOSYNTHETIC_PROCESS**」という遺伝子

```
in_f1 <- "data_mas_EN_symbol.txt" #入力
in_f2 <- "c5.bp.v5.0.symbols.gmt" #入力
out_f <- "hoge1.png" #出力
param_G1 <- 4 #G1群
param_G2 <- 4 #G2群
param_FDR <- 0.1 #false
param_posi <- c(1:4, 5:8) #G1群
param <- "STERIOD_BIOSYNTHETIC_PROCESS" #解析
param_fig <- c(700, 650) #ファ
```



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

#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_f1, header=TRUE, row.names=rownames(data) <- toupper(rownames(data))) #II
data <- as.matrix(data) #型変
data.cl <- c(rep(1, param_G1), rep(2, param_G2)) #サブ
data <- data[,param_posi] #確認
colnames(data)
```

Pathway: STERIOD_BIOSYNTHETIC_PROCESS

There are 19 genes in this pathway

Group 1	Group 2
Hub Gene (Group 1): ADM	Hub Gene (Group 2): NSDHL
Weight Factor: 1.204	Weight Factor: 1.202
Hub Gene (Group 2): NSDHL	Hub Gene (Group 1): ADM
Weight Factor: 1.129	Weight Factor: 1.175



■ w>1.5 ■ 1.25<w<1.5 ■ 1<w<1.25 ■ 0.75<w<1 ■ 0.5<w<0.75 □ w<0.5

