ハッケン コウケン
Tokyo University of Agriculture and Technology

2015年度 S-PLUS & Visual R Platform 学生研究奨励賞応募論文

大容量メモリサーバを用いた 判別分析による変動遺伝子グループ分けの最適化

東京農工大学大学院
連合農学研究科 遺伝子機能制御学研究室
博士課程２年　小林拓嗣

研究の目的

大きな目的：

麹菌* の遺伝子欠損株を解析し、欠損した遺伝子の機能を明らかにする

*麹菌：日本酒や醤油、味噌などの日本の伝統的醸造産業に欠かせない真核微生物
日本の “国菌” であると認定されている

解析手法の一つとして、**RNA-Seq 解析**（次スライドで説明）を行っている

麹菌のある遺伝子を欠損した株と Control 株の遺伝子の発現量データを取得

↓

遺伝子欠損株で発現が変動した遺伝子を調べることで、
欠損した遺伝子の機能を明らかにする

RNA-Seq 解析とは

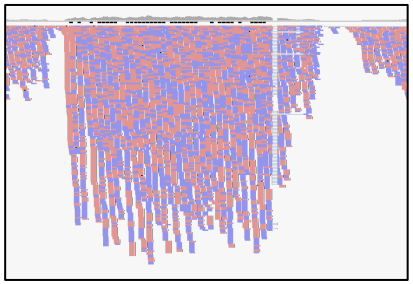
Tokyo University of Agriculture and Technology

生物体内の遺伝子の転写量 (RNA 量) を網羅的に定量する方法
遺伝子組換え体や薬剤処理群などでの転写量の増減を調べるのに使われる

次世代シーケンサー

(Genome Analyzer IIx, Illumina)
http://genome.lab.tuat.ac.jp/~genome/

読まれた配列をマッピング



カウントデータを取得
→転写量 (RPKM) を算出

算出された転写量 (RPKM など) は、
転写量の変動比 (fold change) や統計処理により発現変動遺伝子を抽出する

Gene Ontology (GO) 解析や Pathway 解析などにより、
生物学的意義付けを行う (どのような機能を持つ遺伝子が変動していたのか)

生物学的意義付け

Tokyo University of Agriculture and Technology

生物学的意義付けには、

・発現変動遺伝子がどのような機能を持つのか
・何のタンパク質をコードしているのか

} という情報が必要

↓しかし

生物種によっては、この情報が充実していないものも・・・

麹菌も “Predicted protein” となっていてどのような遺伝子かわからない遺伝子が多く、詳しく調べるために別の解析が必要 (時間がかかる)

“Predicted protein” が多くても、重要なものから解析したい・・・

↳ 発現変動比や **統計解析** により絞り込む

統計解析による絞り込み

例えば・・・

Student's *t*-test などの *p*-value の順位で絞り込み

↓しかし

Student's *t*-test の *p*-value 順 = 重要な遺伝子順？？

ある遺伝子が関わる生命現象や表現型には、
複数の遺伝子がグループとして働くと考えられる

↓

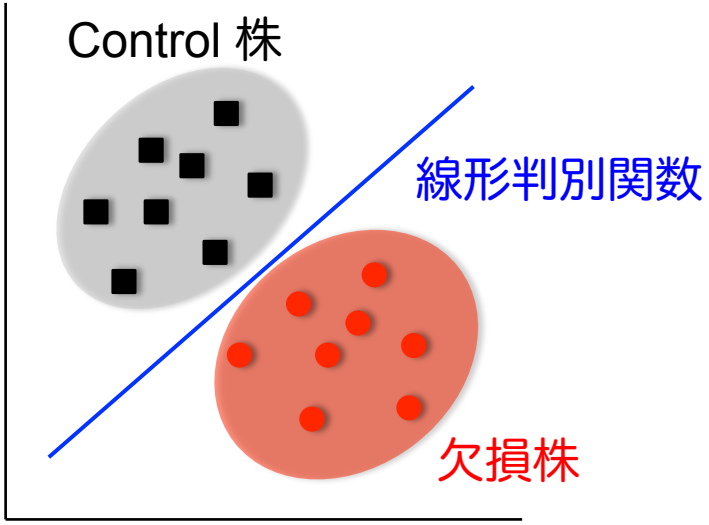
遺伝子の絞り込みもグループとして行う必要があるのでは？

判別分析

Tokyo University of Agriculture and Technology

判別分析：

属する群がわかっている標本を “**ある項目**” でどの群に属するかを判定する方法



		判別結果		数
		Control 株	欠損株	
実際の株	Control 株	A	B	Nc
	欠損株	C	D	Nd

感度　：欠損株を欠損株と判別する確率 = D / Nd

特異度：Control 株を Control 株と判別する確率 = A / Nc

的中率：正しく判別する確率 = (A + D) / (Nc + Nd)

“ある項目” = ある遺伝子 が欠損株と Control 株を区別することができるか判定できる

➡ 区別に重要な遺伝子（グループ）は、欠損した遺伝子と関わりが深いのでは？？

しかし ➡ “**ある項目**” がわかっている場合に使われる

判別分析（逆のアプローチ）

Tokyo University of Agriculture and Technology

“**ある項目**” がわかっていない場合は？？

全ての項目で判別分析を行い、上手く判別できる項目を探索すれば良いのでは？

膨大な計算が必要

＊発現変動遺伝子が 1,000 個ある場合・・・

説明変数の数 ＝遺伝子のグループ内の要素数	組合せの数 =判別分析の回数
2	499,500
3	166,167,000
4	41,417,124,750
5	8.25×10^{12}
6	1.37×10^{15}

膨大な計算を可能にする環境とそれを動かす手法が必要

逆のアプローチを可能にする環境とそのための工夫

Tokyo University of Agriculture and Technology

大容量メモリサーバ

HP Integrity Superdome X

(16プロセッサ 240コア, CPUs12TB メモリ)

http://h50146.www5.hp.com/products/servers/integrity/superdome_x/overview.html

コマンドライン上でバッチ処理

```
[genome@isei C5]$
[genome@isei C5]$ head ABC_C5.sh
R --vanilla --slave --args result_ABC_combn389_C5_100000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_200000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_300000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_400000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_500000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_600000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_700000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_800000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_900000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_1000000 < bootstrap100000_ABC_C5.R &
[genome@isei C5]$
[genome@isei C5]$
[genome@isei C5]$ tail ABC_C5.sh
R --vanilla --slave --args result_ABC_combn389_C5_24200000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24300000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24400000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24500000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24600000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24700000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24800000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_24900000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_25000000 < bootstrap100000_ABC_C5.R &
```

http://www.msi.co.jp/splus/

https://www.r-project.org

大量の判別分析を行うことが可能

解析するデータ

Tokyo University of Agriculture and Technology

生物種：*Aspergillus oryzae*（麹菌）

サンプル（株）：

Control, AB（遺伝子 A, B を欠損）, A（遺伝子 A を欠損）, B（遺伝子 B を欠損）

RNA-Seq のレプリケート：n = 5

全遺伝子を対象とすると、あまりに数が多いため、
Student's *t*-test で有意差が認められた遺伝子に絞った

ABC：Control vs AB で Student's *t*-test *p*-value <0.05 となった遺伝子, 389 個

AC：Control vs A で Student's *t*-test *p*-value <0.05 となった遺伝子, 946 個

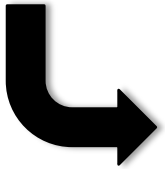
BC：Control vs B で Student's *t*-test *p*-value <0.05 となった遺伝子, 826 個

*ABC, AC, BC はあらかじめ同値データを抜いてある
（同値データが同じ組合せ内に存在するとランク落ちのエラーが出るため）

解析の流れ

Tokyo University of Agriculture and Technology

sh ○○○.sh ←シェルスクリプトでバッチ処理を自動化

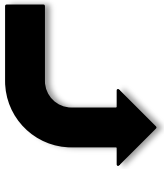


○○○.sh

△△△.R (or .Splus)
△△△.R (or .Splus)
⋮
△△△.R (or .Splus)
△△△.R (or .Splus)

←最大 250 個のバッチ処理

△△△.R (or .Splus)



```
⋮
mylda <- function(x) { }
⋮
```

判別分析のための関数を含むプログラムを実行→

判別分析のための関数を含むプログラム

Tokyo University of Agriculture and Technology

ABC_C3.R

```
args <- commandArgs(trailingOnly = T)                    # バッチ処理の引数を指定
data1 <- read.table("ABC_STT_005_RPKM_1_nr.csv",sep="\t",quote="",row.names=1,header=T)
data.log <- log((data1), base=2)
data2 <- data.log

inf <- args[1];                                          # 1 つ目の引数を inf に格納
num <- as.integer(args[2]);                              # 2 つ目の引数を num に格納
num2 <- num - 99999;                                     # num から 99999 を引いた値を num2 に格納

mylda <- function(x){                                    # 判別分析の関数
library(genefilter);                                     # 発現量 RPKM のサンプル間の補正用のライブラリを読み込み
library(MASS);
data3 <- data2[x,];                                      # 判別分析に用いるデータを指定
data.z <- genescale(data3, axis=1, method="Z");          # 発現量 RPKM のサンプル間の補正
data4 <- t(data.z);
grouping1 <- matrix(c(rep("1",5),rep("0",5)),nrow=10,ncol=1);        # コントロール株か欠損株かのラベルを作成
rlt1_1 <- lda(as.matrix(data4), grouping1);              # 判別分析の結果を rlt1 に格納
rlt1_2 <- predict(rlt1_1);                               # predict 関数で予測した結果を rlt1_2 に格納
table( grouping1, rlt1_2$class );                        # 予測の結果を表示
sensi <- (table( grouping1, rlt1_2$class )[1,1])/sum(table( grouping1, rlt1_2$class )[1,]);          # 感度を求め、結果を sensi に格納
speci <- (table( grouping1, rlt1_2$class )[2,2])/sum(table( grouping1, rlt1_2$class )[2,]);          # 特異度を求め、結果を speci に格納
acc <- (table( grouping1, rlt1_2$class )[1,1]+table( grouping1, rlt1_2$class)[2,2])/sum(table( grouping1, rlt1_2$class ));
                                                         # 的中率を求め、結果を acc に格納
cl <- c(rep(0,5),rep(1,5));                              # ラベルを作成
data5 <- cbind(t(data.z), cl);
data5 <- as.data.frame(data5);
wilks <- summary(manova(cbind(data5[,c(1)], data5[,c(2)], data5[,c(3)])~cl, data=data5), test="Wilks")$stats[3];
                                                         # wilks' lambda を求め、wilks に格納
return(t(c(x, sensi, speci, acc, wilks)))                # 結果(遺伝子の組合せ, 感度, 特異度, 的中率, wilks' labmda)を返す
}
dim(combn389<-combn(389,3))                              # 389 遺伝子から 3 遺伝子を選ぶ全組合せのリストを作成
result <- t(apply(t(combn389)[c(num2:num),],1,mylda));   # 組合せリストのうち、引数で指定した 10 万個の組合せについて判別分析の関数を実行
colnames(result) <- c("gene1","gene2","gene3","sensitivity","specificity","accuracy","Wilks-lambda");
write.table(result, inf, sep="\t", append=F, quote=F, row.names=F)          # 結果の出力
```

バッチ処理

Tokyo University of Agriculture and Technology

ABC_C5.sh

```
R --vanilla --slave --args result_ABC_combn389_C5_rep1_100000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_200000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_300000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_400000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_500000 < bootstrap100000_ABC_C5.R &
                                                    ⋮
R --vanilla --slave --args result_ABC_combn389_C5_rep1_24600000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_24700000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_24800000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_24900000 < bootstrap100000_ABC_C5.R &
R --vanilla --slave --args result_ABC_combn389_C5_rep1_25000000 < bootstrap100000_ABC_C5.R &
```

説明変数 2, 3 個の判別分析の結果

Tokyo University of Agriculture and Technology

説明変数 2、3 個は、組合せの数が少ないので、全組合せの判別分析を行った

説明変数 2、3 個の判別分析の結果

説明変数の数	組合せの数	判別分析の数	Wilks' lambda の最小値
2	75,466	75,466	0.006279669
3	9,735,114	9,735,114	0.000375859

Wilks' lambda：平均値の差の検定の多変量版
0 に近いほど 2 群間の差が大きい

Wilks' lambda の値を低い順に並べると、
上位の組合せの感度・特異度・的中率は、ほぼ "1"

↓

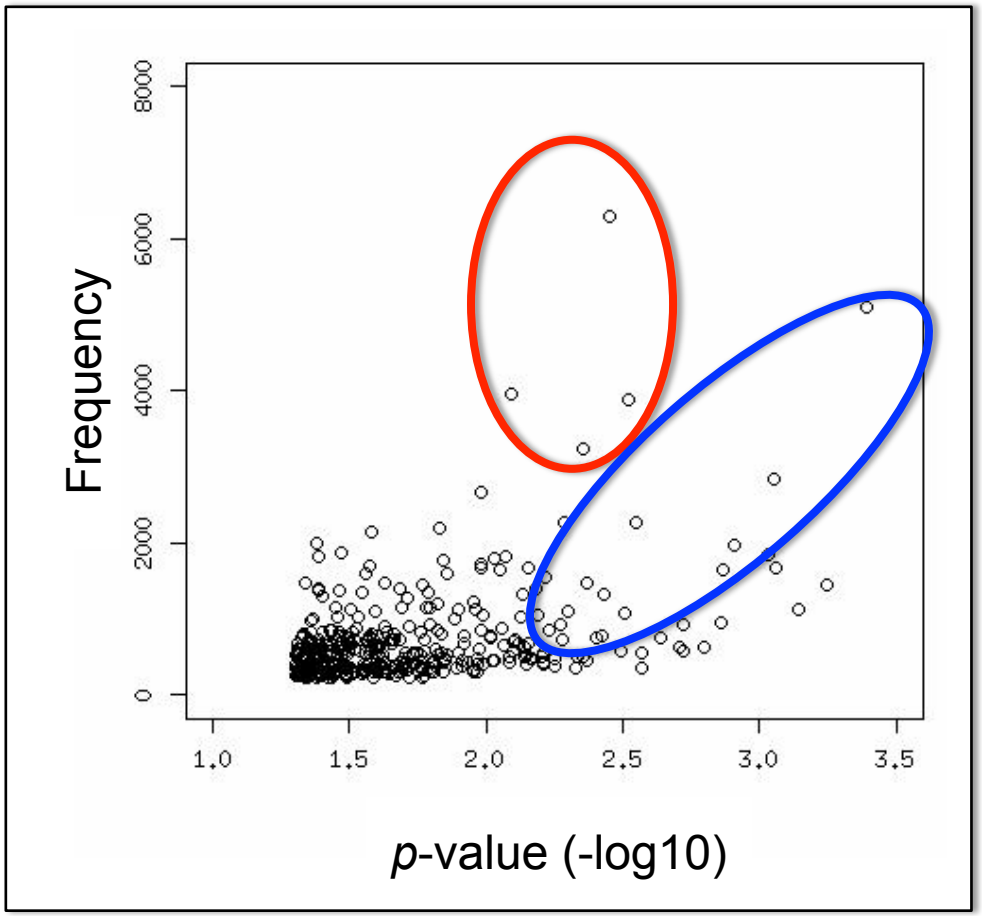
ほぼ 100% の確率でグループ分けに成功する

↓

以降は Wilks' lambda の値を基に、
よりはっきりとグループ分けが可能な組合せを探索する

判別分析の結果と Student's *t*-test の相関

説明変数 3 個（判別分析 約 970 万回 ）のうち、
Wilks' lambda の値が低い順に 10 万個の組合せ = のべ 30 万遺伝子の頻度 (Frequency) と Student's *t*-test の *p*-value(-log10) をプロット



p-value が小さい（-log10 の値が大きい）遺伝子の Frequency は高い傾向にあったが、

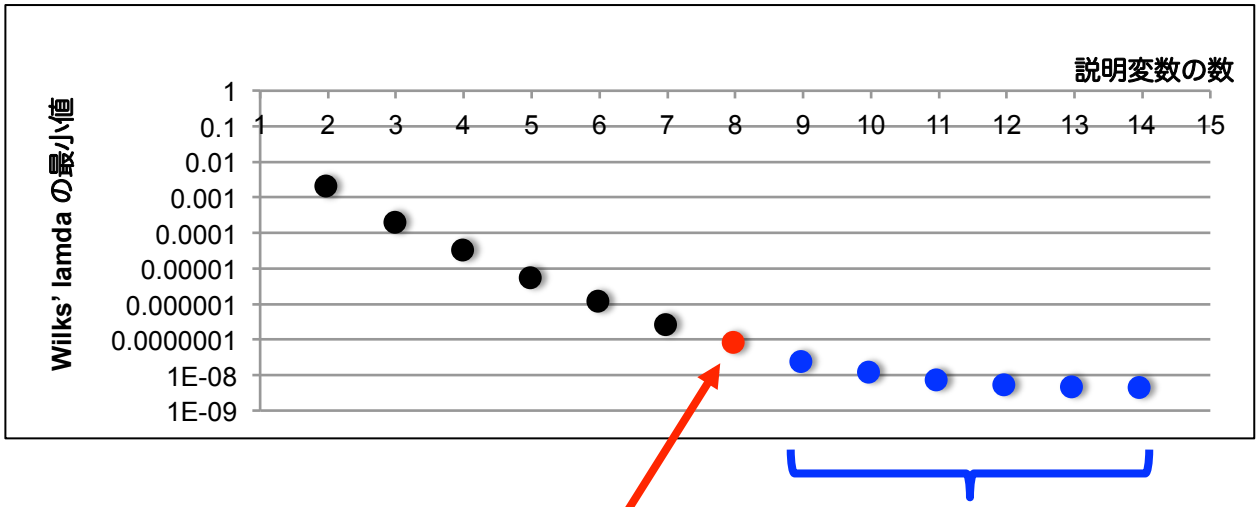
p-value が大きい（-log10 の値が小さい）遺伝子の Frequency が低いとは限らない

最適な説明変数の数を決める方法

Tokyo University of Agriculture and Technology

説明変数の数は多ければ多いほど良いのか？？

説明変数の数を 4, 5, 6・・・と増やしていくと



Wilks' lambda の値が下がらなくなる
→過学習（どの組合せでもグループ分けに成功する）
と思われる

過学習が起こる手前と思われる説明変数の数を選択し、重点的に判別分析を行う
（説明変数の数が増えると、組合せ数が増加し、全組合せの解析が行えないため）
＝ 最適と思われる説明変数でなるべく多く判別分析を行う

方法：説明変数が多い時の判別分析

Tokyo University of Agriculture and Technology

説明変数が 4 個以上では、組合せの総数が多いため、総当たりではなく 10 万個の乱数を発生させ、250 回分を 1 度にバッチ処理 = 2,500 万回分の判別分析

説明変数 4~7 個の組合せ数

説明変数の数	組合せの数	判別分析の数
4	939,438,501	25,000,000
5	72,336,764,577	25,000,000
6	4.63×10^{12}	25,000,000
7	2.53×10^{14}	25,000,000

乱数の発生方法

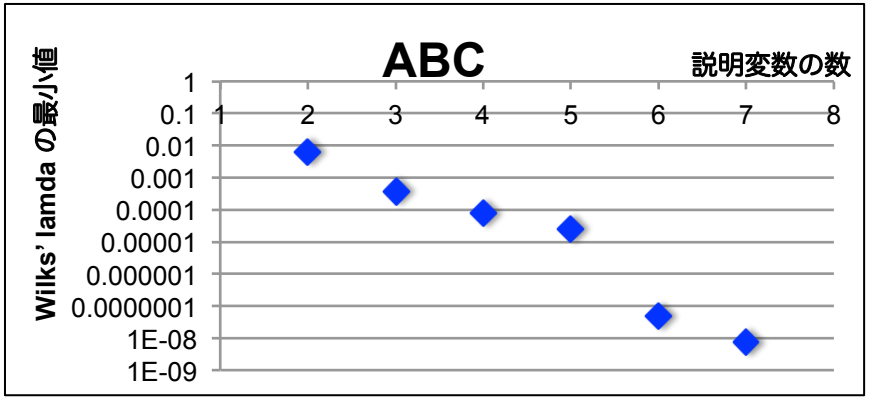
```
combnlist <- list()
for(i in 1:100000){combnlist[[i]] <- sample(389,4,replace=F)}
combn100000 <- t(as.data.frame(combnlist))
result <- t(apply(combn100000,1,mylda));
```

結果：説明変数が多い時の判別分析

説明変数 4~7 個の判別分析の結果

説明変数の数	組合せの数	判別分析の数	Wilks' lambda の最小値
4	939,438,501	25,000,000	7.52639×10^{-5}
5	72,336,764,577	25,000,000	2.46592×10^{-5}
6	4.63×10^{12}	25,000,000	4.61173×10^{-8}
7	2.53×10^{14}	25,000,000	7.09812×10^{-9}

説明変数 4~7 個の Wilks' lambda の最小値のプロット



説明変数 8 個以上では、
エラーが続出し解析ができなかった
（ランク落ちと多重共線性）

過学習が起こる説明変数の数をはっきりと決めることができなかったので、
過学習が起きていないと考えられる説明変数 5 個を選択した

重点的な判別分析

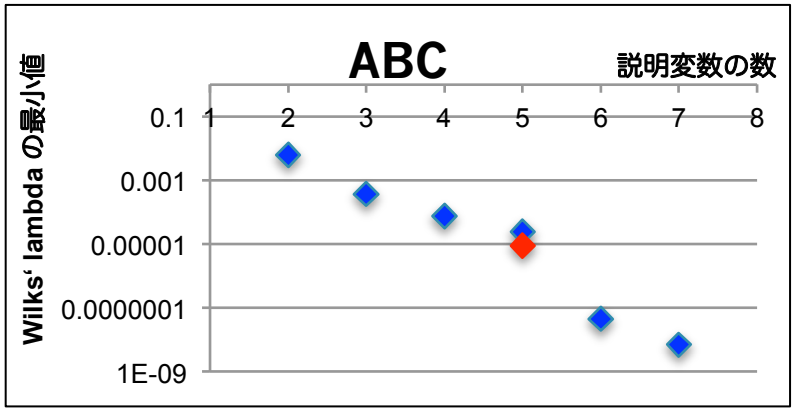
Tokyo University of Agriculture and Technology

説明変数 5 個について、さらに 4 回のバッチ処理を行い、
計 1 億 2,500 万回の判別分析を行った

判別分析の結果のまとめ

説明変数の数	組合せの数	判別分析の数	Wilks' lambda の最小値
2	75,466	75,466	0.006279669
3	9,735,114	9,735,114	0.000375859
4	939,438,501	25,000,000	7.52639×10^{-5}
5	72,336,764,577	25,000,000	2.46592×10^{-5}
6	4.63×10^{12}	25,000,000	4.61173×10^{-8}
7	2.53×10^{14}	25,000,000	7.09812×10^{-9}
5	72,336,764,577	125,000,000	8.97643×10^{-6}

Wilks' lambda の最小値のプロット



2,500 万回分の結果よりも Wilks' lambda の最小値が小さいものが含まれていた

区別のためにより良い組合せが含まれている

組合せの絞り込み①

Tokyo University of Agriculture and Technology

説明変数 5 個の結果を絞る前に、全組合せの計算を行った説明変数 3 個の結果を用いて、重要と思われる遺伝子の選抜を行うことを考えた

方法：説明変数 3 個のうち、Wilks' lambda の値が低い順に 10 万、1 万、1 千個の組合せの遺伝子の頻度 (Frequency)のそれぞれ上位 5 個ずつを選抜する

上位 10 万

Gene	Frequency
237	29,948
73	6,286
319	5,102
79	3,946
253	3,888
134	3,236
4	2,839

上位 1 万

Gene	Frequency
237	3,956
73	595
117	417
51	415
120	410
12	404
359	402

上位 1 千

Gene	Frequency
237	595
120	388
12	113
117	78
359	75
73	35
134	28

Gene 237, 73, 120, 319, 117, 12, 79, 51, 253, 359 を選抜

組合せの絞り込み②

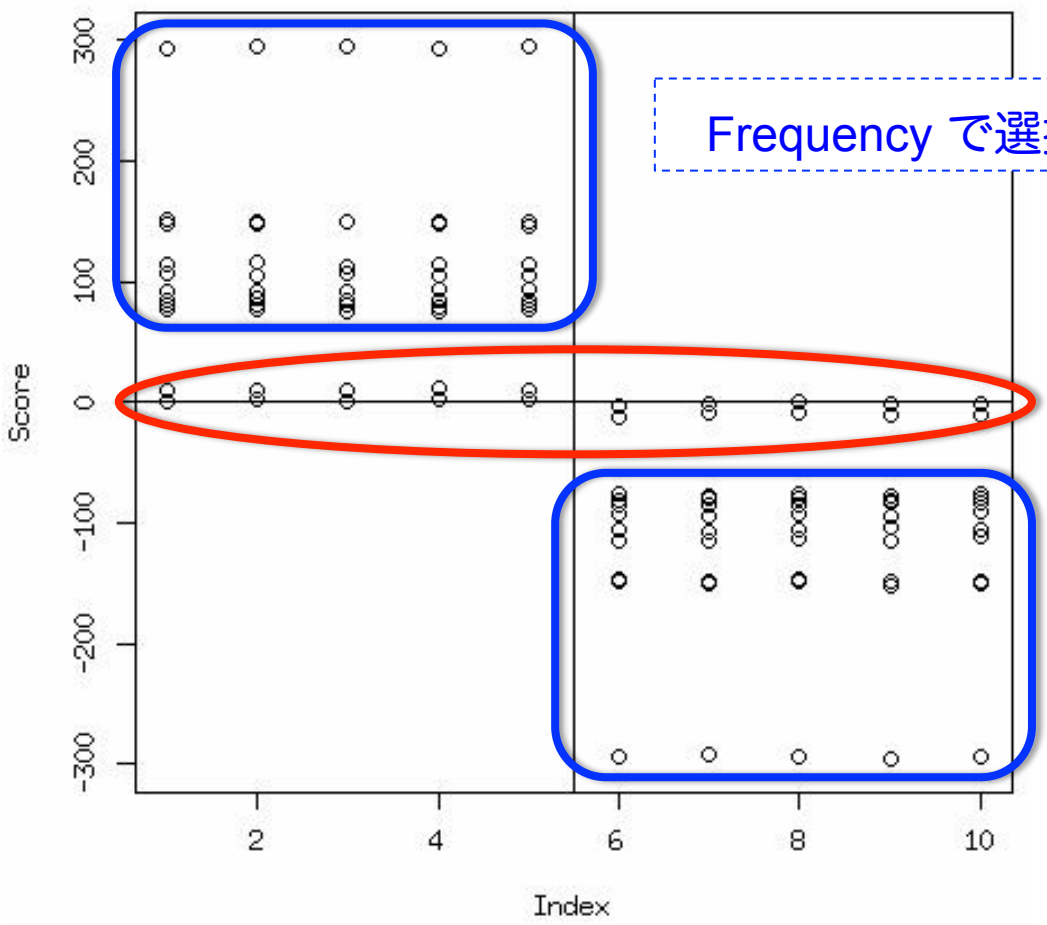
Tokyo University of Agriculture and Technology

選抜した Gene 237, 73, 120, 319, 117, 12, 79, 51, 253, 359 それぞれを含む組合せについて、説明変数 5 個の 1 億 2,500 万回の判別分析の結果を Wilks' lambda の小さい順に並べる

遺伝子の組合せ	Wilks' lambda の値
237,185, 26, 148, 306	3.66111675008352e-05
212, 42, 237, 53, 338	6.40126561402781e-05
117, 42, 53, 338, 237	6.72759297517926e-05
191, 237, 12, 3, 55	7.1513427442044e-05
359, 327, 4, 237, 378	7.42064950463597e-05
⋮	⋮

Gene 237 を含む組合せのうち、Wilks' lambda が最も小さい組合せを選び、
判別分析の結果をプロット（選抜した 10 個の遺伝子について行った）

絞り込んだ遺伝子の組合せの判別分析結果のプロット



Frequency で選抜した遺伝子の判別分析の結果

Student's *t*-test の
p-value 上位 or 下位 5 個の遺伝子
の判別分析の結果

Frequency で選抜した遺伝子の組合せの方が、単に Student's *t*-test で選んだ組合せよりも欠損株とコントロール株をより明確にグループ分けすることができた

方法：判別関数（数理モデル）の作成

Tokyo University of Agriculture and Technology

```
(rlt1_1 <- lda(as.matrix(data4[,x]), grouping1))
Call:
lda(as.matrix(data4[, x]), grouping = grouping1)

Prior probabilities of groups:
  0   1
0.5 0.5

Group means:
       GeneA      GeneB      GeneC      GeneD      GeneE
0 -0.7755834  0.7996515  0.5910129 -0.8376868  0.6256248
1  0.7755834 -0.7996515 -0.5910129  0.8376868 -0.6256248

Coefficients of linear discriminants:
              LD1
GeneA   139.008572
GeneB   -45.685510
GeneC   -58.397485
GeneD   141.975012
GeneE     5.454758
> apply(rlt1_1$means%*%rlt1_1$scaling,2,mean)
         LD1
-8.526513e-14
> rlt1_2 <- predict(rlt1_1)
> plot(rlt1_2$x)
```

$$y = ax1 + bx2 + cx3 + dx4 + ex5 - f$$

各遺伝子の係数
(= a, b, c, d, e)

定数項 (= f)

結果：判別関数（数理モデル）の作成

Tokyo University of Agriculture and Technology

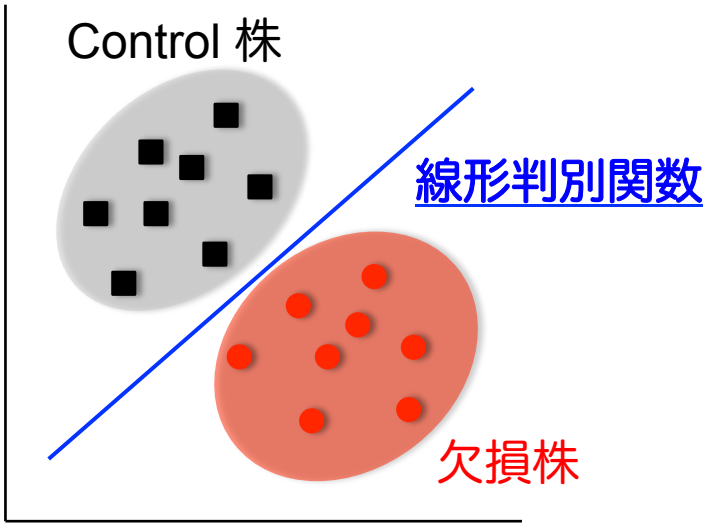
絞り込んだ遺伝子の組合せについて判別関数（数理モデル）を作成した

y = ax1 + bx2 + cx3 + dx4 + ex5 - f

x <- c(79,155,272,4,352)　　#判別関数を作成した遺伝子の組合せ

y= 139.008572*Gene79 -45.685510*Gene155 -58.397485*Gene272
+141.975012*Gene4 +5.454758*Gene352 + 8.526513e-14

各遺伝子の Student's *t*-test での *p*-value の順位：60, 23, 243, 6, 96



欠損株と Control 株を区別する判別関数（数理モデル）を作成することができた

絞り込んだ遺伝子の組合せには、必ずしも Student's *t*-test の順位が高いものばかりではないことも明らかになった

検証：S-PLUS での判別分析

Tokyo University of Agriculture and Technology

これまでの方法が他の環境でも使うことができるかを検証するために、
S-PLUS を用いて検証した

全く同じプログラムでは動かないので、
R のプログラムを S-PLUS 用に置き換えた

S-PLUS での大きな変更点①

Tokyo University of Agriculture and Technology

R

```
:
mylda <- function(x){
library(genefilter);
library(MASS);
data3 <- data2[x,];
data.z <- genescale(data3, axis=1, method="Z");
data4 <- t(data.z);
:
```

S-PLUS

```
:
mylda <- function(x){
library(MASS);
data3 <- data2[x,];
data4 <- scale(t(data3));
data5 <- as.matrix(t(t(data4)));
:
```

S-PLUS で library(genefilter) が使えなかった
→ library(genefilter) の代わりに scale()

S-PLUS での大きな変更点②

Tokyo University of Agriculture and Technology

R

```
:
data5 <- cbind(t(data.z), cl);
data5 <- as.data.frame(data5);
wilks <- summary(manova(cbind(data5[,c(1)], data5[,c(2)],
                              data5[,c(3)])~cl, data=data5), test="Wilks")$stats[3];
return(t(c(x, sensi, speci, acc, wilks)))
}
:
```

S-PLUS

data=data6 無し、$stats[3]→$Stats[2]

```
:
data6 <- cbind(t(data3), cl);
data6 <- as.data.frame(data6);
wilks <- summary(manova(cbind(data6[,c(1)], data6[,c(2)],
                              data6[,c(3)])~cl), test="wilks")$Stats[2];
return(t(c(x, sensi, speci, acc, wilks)))
}
:
```

S-PLUS での大きな変更点③

Tokyo University of Agriculture and Technology

R

```
⋮
dim(combn389<-combn(389,3))
result <- t(apply(t(combn389)[c(num2:num),],1,mylda));
colnames(result) <-
c("gene1","gene2","gene3","sensitivity","specificity","accuracy","Wilks-lambda");
write.table(result, inf, sep="\t", append=F, quote=F, row.names=F)
⋮
```

S-PLUS

```
⋮
combn389<-combn(389,3)
result <- t(apply(t(combn389)[c(3100001:3200000),],1,mylda))
colnames(result) <-
c("gene1","gene2","gene3","sensitivity","specificity","accuracy","Wilks-lambda”)
write.table(result, file="result_ABC_C3_3200000", sep="\t", append=F,
                                                  quote.strings=F,dimnames.write=F)
⋮
```

S-PLUS で使える引数が見つからなかった
→行数を指定するために個別の ~.Splus が必要

S-PLUS で用いるプログラムのまとめ

ハッケン コウケン
Tokyo University of Agriculture and Technology

ABC_C3_3200000.Splus

```
data1 <- read.table("ABC_STT_005_RPKM_1_nr.csv",sep="\t",row.names=1,header=T)
data2 <- log2(data1)

mylda <- function(x){
library(MASS);
data3 <- data2[x,];
data4 <- scale(t(data3));
grouping1 <- matrix(c(rep("1",5),rep("0",5)),nrow=10,ncol=1);
data5 <- as.matrix(t(t(data4)));
y <- lda(data5, grouping1);
z <- predict(y);
table( grouping1, z$class );
sensi <- (table( grouping1, z$class )[1,1])/sum(table( grouping1, z$class )[1,]);
speci <- (table( grouping1, z$class )[2,2])/sum(table( grouping1, z$class )[2,]);
acc <- (table( grouping1, z$class )[1,1]+table( grouping1, z$class)[2,2])/sum(table( grouping1, z$class ));
cl <- c(rep(0,5),rep(1,5));
data6 <- cbind(t(data3), cl);
data6 <- as.data.frame(data6);
wilks <- summary(manova(cbind(data6[,c(1)], data6[,c(2)], data6[,c(3)])~cl), test="wilks")$Stats[2];
return(t(c(x, sensi, speci, acc, wilks)))
}


combn389<-combn(389,3)

result <- t(apply(t(combn389)[c(3100001:3200000),],1,mylda))

colnames(result) <- c("gene1","gene2","gene3","sensitivity","specificity","accuracy","Wilks-lambda")

write.table(result, file="result_ABC_C3_3200000", sep="\t", append=F, quote.strings=F,dimnames.write=F)
```

方法：S-PLUS での説明変数 2 個の判別分析

ABC：combn(389,2) = 75,466 (10 万×1)

AC：combn(946,2) = 446,985 (10 万×5)

BC：combn(826,2) = 340,725 (10 万×4)

＝ 10 プロセス < 250 なので、バッチ処理 1 回分

バッチ処理

C2.sh

```
Splus BATCH ABC_C2.Splus ABC_C2.Splus.out &
Splus BATCH AC_C2_100000.Splus AC_C2_100000.Splus.out &
Splus BATCH AC_C2_200000.Splus AC_C2_200000.Splus.out &
Splus BATCH AC_C2_300000.Splus AC_C2_300000.Splus.out &
Splus BATCH AC_C2_400000.Splus AC_C2_400000.Splus.out &
Splus BATCH AC_C2_446985.Splus AC_C2_446985.Splus.out &
Splus BATCH BC_C2_100000.Splus BC_C2_100000.Splus.out &
Splus BATCH BC_C2_200000.Splus BC_C2_200000.Splus.out &
Splus BATCH BC_C2_300000.Splus BC_C2_300000.Splus.out &
Splus BATCH BC_C2_340725.Splus BC_C2_340725.Splus.out &
```

結果：S-PLUS での説明変数 2 個の判別分析

Tokyo University of Agriculture and Technology

各結果のはじめの 2 行を表示させた

```
==> result_ABC_C2 <==
1   2   0.6  0.8  0.7  0.512720050394183
1   3   0.6  1    0.8  0.488611742303872

==> result_AC_C2_100000 <==
1   2   1    1    1    0.268054582845524
1   3   1    1    1    0.267657343426462

==> result_AC_C2_200000 <==
113 490 1    1    1    0.167925256588485
113 491 1    1    1    0.235008430727051

==> result_AC_C2_300000 <==
243 715 1    1    1    0.0415817774576416
243 716 1    1    1    0.164622722080882

==> result_AC_C2_400000 <==
404 573 0.8  0.8  0.8  0.358965793584428
404 574 0.8  1    0.9  0.38963846061802

==> result_AC_C2_446985 <==
639 933 1    1    1    0.257676265451078
639 934 1    1    1    0.361236318419852
```

```
==> result_BC_C2_100000 <==
1   2   1    1    1    0.268054582845524
1   3   1    1    1    0.267657343426462

==> result_BC_C2_200000 <==
132 573 1    0.8  0.9  0.372994216548674
132 574 0.8  1    0.9  0.400118701982971

==> result_BC_C2_300000 <==
295 817 1    1    1    0.261376250653406
295 818 1    0.8  0.9  0.269849005269729

==> result_BC_C2_340725 <==
541 572 0.8  1    0.9  0.152439245758582
541 573 0.8  1    0.9  0.249826870078678
```

result_AC_C2_100000 と
result_BC_C2_100000 の結果が一致

方法：S-PLUS での説明変数 4 個の判別分析

Tokyo University of Agriculture and Technology

一致しないはずの結果が一致したのはなぜか？？

説明変数が 4 個以上では、組合せの総数が多いため、
全組合せのリストではなく乱数のリストを作成する

乱数の発生方法

```
combnlist <- list()
for(i in 1:100000){combnlist[[i]] <- sample(389,4,replace=F)}
combn100000 <- t(as.data.frame(combnlist))
result <- t(apply(combn100000,1,mylda));
```

プログラムが若干異なる（乱数発生の分）説明変数 4 個でも
同様の現象が見られるか調べた

結果：S-PLUS での説明変数 2 個の判別分析

各 result 1 行目の wilks-lambda 値がユニークかどうかで判断した

```
head -1 result* | cut -f8 | grep -v ">" | sort | uniq -c | awk '{print$1}' | sort | uniq -c
```

Wilks' lambda の値の種類を数える

数えた結果を数える

すべての結果がユニークであれば、
250　1
となるはずである

結果

```
      1   113
      1   131
      1     3
```

113 個が同じ結果、131 個が別の同じ結果、
3 個がさらに別の同じ結果であった

残り 3 つは、
エラーが出て結果が出力されなかった

プロセス数や組合せの行数を減らしても同様の結果だった

研究内容のまとめ

Tokyo University of Agriculture and Technology

・大容量サーバーを用いて、大量の判別分析を行うことができた

・Wilks' lambda の値を基に、
グループ分けに適した変動遺伝子の組合せを絞り込んだ
（必ずしも Student's *t*-test の *p*-value と
相関があるわけではないことも明らかになった）

・絞り込んだ変動遺伝子の組合せについて、判別関数（数理モデル）を作成した

遺伝子発現変動を基に、欠損株と Control 株を区別するのに適した遺伝子のグループを見つける方法を確立した
➡ 欠損した遺伝子の機能を調べる足掛かりを得ることができた

・S-PLUS で大量の判別分析を行ったが、バッチ処理が上手くいかなかった

考察・課題

Tokyo University of Agriculture and Technology

ランク落ち、多重共線性

RNA-Seq のデータはカウントデータがもとになっているため、
要素の独立性の問題が生じやすいと考えられる。

S-PLUS でのバッチ処理による大量計算

一致しないはずの結果が一致したのはなぜか？？

→ S-PLUS では R に比べ、conbn() の計算に時間がかかっていた

→ sample() のリスト作成にも時間がかかった

http://www.msi.co.jp/splus/support/useFaq/programming.html#p-8
でも説明されているように、for(){} は S-PLUS では工夫が必要

時間がかかるだけでは結果は同じにならないはず・・・
→おそらく、メモリの使い方の問題で他のプロセスの計算過程が持ち込まれたのでは？

参考文献

・新井仁之 (2006). 線形代数ー基礎と応用, 日本評論社.

・小西貞則 (2010). 多変量解析入門ー線形から非線形へー, 岩波書店.

・石村貞夫 (1992). すぐわかる多変量解析, 東京図書.

・内田治, 菅民郎, 高橋信 (2003). 文系にもよくわかる多変量解析, 東京図書.

・久米均, 飯塚悦功 (1987). 回帰分析 シリーズ 入門 統計的方法 2, 岩波書店

・Watanabe SY, Iga J, Ishii K, Numata S, Shimodera S, Fujita H, Ohmori T (2015). Biological tests for major depressive disorder that involve leukocyte gene expression assays. *Journal of Psychiatric Research*, **66-67**, 1-6

・https://cran.r-project.org/manuals.html

・W.N.ヴェナブルズ, B.D.リプリー 著, 伊藤幹夫, 大津泰介, 戸瀬信之, 中東雅樹 訳 (2012). S-PLUS による統計解析 第 2 版, 丸善出版.

謝辞

東京農工大学　文部科学省特別経費による「農学系ゲノム科学領域における実践的先端研究人材育成プログラム」

石井一夫 特任教授

古崎利紀 特任助教

東京農工大学

高橋信弘 教授

有江力 教授

この場をお借りして、御礼申し上げます