ゲノム・オミックスのデータ解析教室(R版)

ryamada

2021年8月21日

データ解析上の特徴

ポイント(データ解析上の特徴)

- 解析されるデータ(入力データ)の特徴
- ハイスループット
- 高次元データ
- 実験の一次データ(生データ)を加工してデータ解析に用いることが多い
- 解析スタイルの特徴
- 多変量解析
- 変量の組合せに興味がある
- 多段階解析
- データ駆動型

はじめに

ゲノム・オミックス解析に用いられるデータは、ハイスループット実験による標本のゲノム・エピゲノム・トランスクリプトーム等のオミックスデータと、標本に関するデータに分けて整理すると理解しやすい。

標本に関するデータの例を挙げる。 標本が個体由来であれば、個体の特徴を表すデータであり、年齢・性別・疾病の有無・臨床検査値などがこれに相当する。

標本がシングルセルであれば、シングルセルについて実験とは別に得られるデータが相当し、シングルセルの由来元である個体の情報や、シングルセルの採取時期・場所等に関するデータ、シングルセルの形態等のデータが含まれる。

なぜ、ゲノム・オミックス実験のデータとそれ以外のデータを分けるとわかりやすいと言うのかの理由は次の通りである。

ゲノム・オミックス以外のデータは、いわゆるデータサイエンス・統計解析における多変量解析の変数のデータとして得られ、データ解析に用いる際に、通常の解析フローにすぐに乗せることができる。 それに対して、ゲノム・オミックス実験のデータは、データの取得に際して多数の変数に関して一塊として得られるという特徴があり、データ解析に用いる前に、データ全体の評価を行い、その全体の記述統計結果に基づいて、調整するステップが必要となる点で、それ以外のデータと異なるからである。

また、データ解析は大きな括りで言えば、多変量解析・多変量データマイニングであると言えるが、ほぼすべてのデータ解析において、一次出力を行った後、その結果の適正な解釈のために、さらなる処理を要するという特徴がある。 この点が、ゲノム・オミックス研究のデータ解析に比較的、特有である。

以下に、ゲノム・オミックスのデータ解析に特徴的な点を順次説明する。

解析されるデータ(入力データ)の特徴

ゲノム・オミックス実験データの特徴

ハイスループット実験という名称がその特徴をよく表している。 一度に、１標本について、非常にたくさんの変数のデータが一塊として得られるのがハイスループット実験であり、ゲノム・オミックス実験データの特徴である。

例えば、数十万超の一塩基多型のジェノタイプや万の単位の遺伝子の発現量を、一標本について一度に得るチップ実験がそれである。 また、パラレル・シークエンシング技術によりゲノム・エピゲノム・トランスクリプトームに関する定量データを、一標本について一度に得る実験もその例である。

このハイスループット実験から得られる情報の特徴は、個々のデータに質の良し悪しのばらつきがあること、一塊で得られるデータ全体からデータの質の良し悪しの全体としての傾向の情報が得られること、標本ごとに一塊で得られるデータセットを標本間で比較可能にするためには、個々の標本のデータセットの全体の傾向を標本間で比較し、その比較結果に基づいて調整するステップが必要になることである。

この解析前の処理は、標本ごとのデータセットの質的記述統計と、質的記述統計に基づくデータの採否決定と値の標準化とを含む。

また、ゲノム・オミックス実験データは、その一次出力データをそのままデータ解析に使えないことが多いことも特徴的である。 パラレル・シークエンシングの一次出力データは多数のショートリードであるが、ショートリードの情報そのものはゲノム・オミックス的な情報としては不十分である。 レファレンス配列との比較照合(ショートリードのマッピング・アラインメント)を経て、ショートリードの分子遺伝学的意味付けを行う必要がある。 さらに、ショートリードがレファレンス配列上に付与した情報を基に、ゲノム配列の状態(多型性)やゲノム・クロマチンの化学修飾状態・アクセスしやすさ、遺伝子発現量・トランスクリプトバリアント別発現量などの、データ解析するための変数のデータに変換される。

このデータ解析用の変数データへの変換に当たって、統計学的に推定処理が施される。

このようにデータ解析の前段階において、記述統計、アラインメント処理、推定などが行われるわけだが、本テキストでは、これらの処理について、データ解析の前段階の一部として説明することはしない。

これらの処理は、統計学的処理・機械学習であり、いわゆるデータの本解析の考え方が同一であるので、それらと共に扱うこととする。 そのような扱いをすることで、統計学的処理・機械学習の立場で考慮するべき基本的な考え方を押さえれば、前処理も本解析も解析結果の解釈も行えることを強調したい。

ゲノム・オミックス以外のデータの特徴

標本の属性は、個体の属性にしろシングルセルの属性にしろ、標本ごと・特徴量ごとにそれぞれ観測する場合が多く、データ値そのものが多変量解析にそのまま使えることが通例である。従って、一塊での観測に伴う記述統計や標本間での標準化は不要であることが普通である。

また、一次観測データの変換も不要であることが多い。

もし、観測が一斉に行われたり、観測が複合的なものであって、そこから何らかの特徴量を抽出する必要がある場合などは、前項で説明したのと同様の考え方を適用すればよい。 そのような例としては、画像情報からの形態特徴量抽出などが含まれるかもしれない。

解析スタイルの特徴

データの本解析は、入力データを統計学的・機械学習的に評価し、メッセージを引き出すことである。 その意味では、ゲノム・オミックス解析が特殊な手法を用いているわけではない。

ゲノム・オミックスのデータ解析では、評価対象になる変数の数が膨大であり、その膨大な変数の組合せや相互作用を積極的に考慮する姿勢が強いことが特徴となっている。

変数の組合せ・相互作用を取り扱うのは、オミックス(-ome)という概念自身が、多数の要素の総体としての働きの理解を目指すことと定義されているから当然である。

また、多数の要素(多数の遺伝子多型、多数の遺伝子の発現量など)を同時に取り扱うことから、解析の過程で、その多数の要素が作る分布に関する情報を得ながら、その情報を解釈に活用していくことも多い。 これは、データ駆動型にデータ全体からメッセージを読み取るという特徴であると言い換えることができる。 多数の要素をいちどきに評価することで得られる情報は、変数の観測値の分布だけでなく、各要素ごとに得られる統計量や推定量の分布も含まれる。

以下では、多変量解析と多段階解析とデータ駆動型解析いう３つの視点から、ゲノム・オミックスデータ解析の特徴を概観する。

多変量解析と多段階解析

ゲノム・オミックスデータ解析は、多数の変数を用いるという点では多変量解析と言える。 しかしながら、要素の一次線形和での説明を目標としたり、変数選択によって寄与変数の数を減らすことで良しとするという形式の多変量解析ではない。

より複雑で巧妙な仕組みを理解することを目標に、説明変数の組合せをさらに階層的に組み合わせたり、非線形性を用いた解釈を目指す解析となっている。 この点で、従来型の多変量解析とは異なる。

説明変数の組合せ解析は、要素の組合せ爆発の問題があるために、統計・機械学習的な組合せ探索にはおのずと限界がある。 ゲノム・オミックス解析では、要素の組合せを考慮するにあたり、生物学的な事前知識を投入することが多い。 このような生物学的な事前知識としては、オントロジー情報やパスウェイ情報が良く用いられる。

また、同一遺伝子・同一分子に紐づく複数の要素を組合せとして用いることもある。

このような生物学的な情報を用いて複数の要素の組合せを評価する場合、個々の要素についてデータ解析上の出力をいったん行い、その一次出力を要素の組合せとして統合するという手続きを踏むアプローチが取られることが多い。 また、組合せ評価をしたうえで、複数の組合せ同士の関係性の評価をすることで、階層的な組合せの評価を行うことができる。

このようなアプローチでは、多段階の処理が施されていることがわかる。

非線形性を持ち込んだ解釈をすることは、現象自体に非線形性があるので重要である。 しかしながら、非線形モデルを設定し、最適なモデルパラメタの推定をするというアプローチは得策ではないことが多い。 なぜなら、ゲノム・オミックスの解析では変数の数が膨大であるため、線形モデルであっても自由度・次元が十二分に高く、それが解析の実行上のハードルになっており、そこに、変数の寄与として非線形性を取り込むと、解析の自由度・次元がさらに高くなり、非現実的であるからである。

そのような直接的な非線形モデルの活用ではなく、手法の中にうまく非線形処理を取り入れるアプローチが取られることが多い。 詳細は後述するが、例えばニューラルネットワークを用いたり、サポートベクタマシンを用いれば自然と非線形性を持ったデータマイニングになることが、その例として挙げられる。 また、ノンパラメトリックなモデルフィッティングを行うことでデータ駆動型のモデル推定が可能となるが、これも、そのような例と考えられる。

他の例としては、次元削減の手法の中に非線形次元削減法として分類されるものがあるが、これも非線形な処理結果をもたらす手法の例である。

データ駆動型解析と多段階解析

ゲノム・オミックス研究では、多数の要素について一度にデータが得られるので、その要素の分布についての情報が得られる。

その分布に関する情報は既知の場合もあるかもしれないが、実験をして初めて得られる情報であることも多い。

そのような新たに得られた情報を利用することで、個々の統計的・機械学習的出力を再評価することが適切であることもある。 データを集めてみてから、そのデータの解析・解釈をどのようにするかの方針が決まる部分があると言うことである。 このようなアプローチはデータ駆動型なアプローチと言える。

このアプローチでは実質的に２段階の解析(多段階解析)が行われていることがわかる。

１段階目で得られる結果が、多数の帰無仮説検定結果のセットであるときに、多数の検定を行ったことに留意して、棄却の判定を調整する場合には、２段階目は多重検定補正と呼ばれる処置に相当する。

１段階目の結果を用いて２段階目を行い、２段階目の結果の解釈に当たって１段階目の結果が得られたということの影響を加味することとすれば、Selective inferenceに相当する。

この１段階目と２段階目とをそれぞれベイズ流に行えば、階層型ベイズと呼ばれるアプローチに相当する。

２段階処理の形を取らないものの、多数の要素の分布の情報を用いて、個々の要素に関する推定の個別最適化ではなく、全要素の推定の包括的最適化を行うことで、個々の要素の推定結果が変わるような方法もある。これも、２段階アプローチと考え方を共有しているとみなせるだろう。

例

以下の例では、いくつかの具体的な解析手法が登場するが、その詳細についてこだわることなく、ゲノム・オミックスデータ解析の特徴との関係に主眼を置いて、理解することが望ましい。

多変量の組合せ爆発

n個の要素の効果を調べるとき、個々の効果を調べれば、n通りの評価をすることになる。 組合せを考える場合には、ペアであれば$\frac{n(n-1)}{2} = \left(\begin{array}{c} n \\ 2 \end{array} \right)$通り。 トリオを考えれば、$ = \begin{pmatrxi}n\ 3 \end{pmatrix}。

n個の要素のすべてが集まって生まれる効果について考えると、$\frac{n!}{0!n!} = \left(\begin{array}{c} n \\ n \end{array} \right) = 1$。

これらのすべての場合を足し合わせると、

$$2^n - 1 = (1+1)^n - 1$$

組合せは2^nという大きな値になる。 特にnが少しばかり大きくなると、手に負えない組み合わせ数になる。これが組合せ爆発である。

具体的には以下のようになる。

$$
\begin{aligned}
1 = 1 &= 2^1 - 1 \\
2 + 1 = 3 &= 2^2 - 1 \\
3 + 3 + 1 = 7 &= 2^3 - 1 \\
&\cdots \\
\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} &= 2^n - 1
\end{aligned}
$$

このような組合せを、いわゆる多変量回帰で行うとき、次のようにする。

100標本、4説明変数のデータを作ってやってみる。

```
n.sample <- 100
n.feature <- 4
X <- matrix(rnorm(n.sample*n.feature),ncol=n.feature)
# Y はXを用いて適当に作った値とする
Y <- apply(sin(X),1,sum)

my.data <- data.frame(Y,X)
```

単純な線形回帰では、長さが説明変数のベクトル$(a) = (a_1, \ldots, a_{n.feature})$を使って、以下のように表される。

$$
y_i \sim b + \sum_{j=1}^{n.feature} x_{i,j} \times a_j
$$

この回帰は以下のように実施する。 切片項の値(Intercept)と、４つの説明変数の係数が推定される。

```
lm.out <- lm(Y~X1+X2+X3+X4,data=my.data)
lm.out$coefficients
```

```
##  (Intercept)           X1           X2           X3           X4
## -0.007324244  0.619219088  0.597828798  0.556099984  0.691340471
```

この方式で、説明変数のペアも考慮するとすれば、以下のようにする。こうすることで、ペア項($X_1 \times X_j$)の係数も推定される。

```
lm.out2 <- lm(Y~(X1+X2+X3+X4)^2,data=my.data)
lm.out2$coefficients
```

```
## (Intercept)          X1          X2          X3          X4       X1:X2
## -0.01488428  0.67344343  0.58374555  0.65348866  0.74918604  0.13118017
##       X1:X3       X1:X4       X2:X3       X2:X4       X3:X4
##  0.01312179 -0.11759796 -0.04064968  0.11747433 -0.16336744
```

n個の組合せまですべて評価することにするなら、以下のようにすることができる。

```
lm.out4 <- lm(Y~(X1+X2+X3+X4)^4,data=my.data)
lm.out4$coefficients
```

```
##  (Intercept)           X1           X2           X3           X4        X1:X2
## -0.001730685  0.687659531  0.562631394  0.637766329  0.776733408  0.093886889
##        X1:X3        X1:X4        X2:X3        X2:X4        X3:X4     X1:X2:X3
##  0.018316857 -0.150514493 -0.047070405  0.136649397 -0.163496347 -0.047890045
##     X1:X2:X4     X1:X3:X4     X2:X3:X4  X1:X2:X3:X4
## -0.083930101 -0.016756171 -0.026490995  0.100591019
```

以上のように、組合せを考慮すると、非常に多くの項が現れ、推定する係数の数も非常に多くなる。

しかしながら、線形回帰法を適用する限りは、項の数が多くなっても、同じ枠組みで解析の処理が行われていることに注意しておこう。

線形回帰は線形代数計算により係数の推定値が決定論的に求められるが、線形代数計算は、計算機が最も得意とする処理であり、その意味では、この例では、項数の増大のみが解析上の問題になるっている。

パスウェイ解析とGene-set enrichment analysis(GSEA)の考え方

トランスクリプトーム解析により、すべての遺伝子の発現量を観測し、その発現量が２群間(疾病群と健常群など)で異なるかどうかを、個々の遺伝子について比較する解析があります。

この比較のやり方には幾つかの方法がありますが、例えば、２群間の発現量を平均値の差の検定により、平均値に差がないという帰無仮説が棄却されるか否かでラベル付けをする、という方法があります。

全遺伝子の数がN、そのうち、帰無仮説が棄却された遺伝子の数がMとします。

ここで、N個の遺伝子のうち、K個の遺伝子のセットを取り出すことにします。

このK個の遺伝子は、ある『パスウェイ』に乗っている遺伝子であるとしましょう。 この『パスウェイ』とは、過去の色々な研究の結果、遺伝子同士・遺伝子がコードする分子同士に相互作用があることが判明した結果、データベースなどに登録されているものです。

このK個の遺伝子は、帰無仮説を棄却したものと棄却しないものとに分けられるので、その個数をk_m個と$K - k_m$個と書くことにします。

そうすると以下のような2×2分割表が得られます。

	Rejected	**Not-Rejected**
In Pathway	k_m	$K - k_m$
Not-In Pathway	$M - k_m$	$(N - M) - (K - k_m)$

このパスウェイ上の遺伝子が、棄却遺伝子群にエンリッチしている(非棄却群に比べて棄却群に偏っている)かどうかは、この分割表の独立性の検定によって評価することが可能です。

以下は、その例を示しています。

フィッシャーの正確確率検定によるp値が10^{-6}オーダーと小さいので、このパスウェイの遺伝子は棄却群にエンリッチしていると言えそうです。 このことは、このパスウェイの働きと２群間の発現量の違いとに関連があることを示唆していると考えても良さそうです。

```
N <- 25000
M <- 1000
K <- 60
km <- 12

tab <- matrix(c(km, K - km, M - km, (N-M)-(K-km)),byrow=TRUE,2,2)
print(tab)
```

```
##      [,1]  [,2]
## [1,]   12    48
## [2,]  988 23952
```

```
f.test.out <- fisher.test(tab)
print(f.test.out)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tab
## p-value = 3.731e-06
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   2.920329 11.625808
## sample estimates:
## odds ratio
##   6.059151
```

この例では、多数の要素を個別に評価し、その結果を用いて、要素の集合について、さらなる評価を行うという２段階処理をしている。

また、要素の集合は生物学的な知識を活用して作成している点も、ゲノム・オミックスデータ解析のバイオインフォマティクス的特徴を表している。

非線形次元削減の例 UMAP

ゲノム・オミックスデータは非常に高次元である。トランスクリプトームで言えば、遺伝子の数が次元になるので、二万超次元の空間に、標本がたなびく雲のように広がっている。

シングルセル・トランスクリプトーム解析では、多数のシングルセルの高次元の広がりの様子を２次元に次元縮減し、互いに似通った発現パターンの細胞のクラスタを存在と、クラスタの相互の遠近関係を視覚化することがある。

高次元空間の広がりを低次元空間に押し込むためには、高次元空間で曲がって広がっている細胞を平たくして低次元空間に押し込めることになるので、「非線形」な処理となる。

このことから「非線形次元削減」と呼ぶ。

その一手法であるUMAPの例を示す。

200細胞、90遺伝子のシングルセル発現量データを用いて、UMAP法による2次元非線形埋め込みを実施している。

UMAP法の出力が２次元座標であることが出力から見て取れる。

その座標を用いて、200個の細胞を２次元平面に埋め込んだ様子を視覚化して見せる。

特定の遺伝子の発現量の多寡で色を変えてプロットすることで、細胞のクラスタリングの具合と、その遺伝子の発現量との関係も視覚的に示すことができる。

また、UMAP法では実施条件のパラメタがあり、その値ごとに出力が変わる様子も示している。

```
# install.packages("umap")
# install.packages("iCellR")
library(umap)
```

```
## Warning: package 'umap' was built under R version 4.0.5
```

```
library(iCellR) # single cell data set "demo.obj@raw.data"のため
```

```
## Warning: package 'iCellR' was built under R version 4.0.5
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Loading required package: plotly
```

```
## Warning: package 'plotly' was built under R version 4.0.5
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

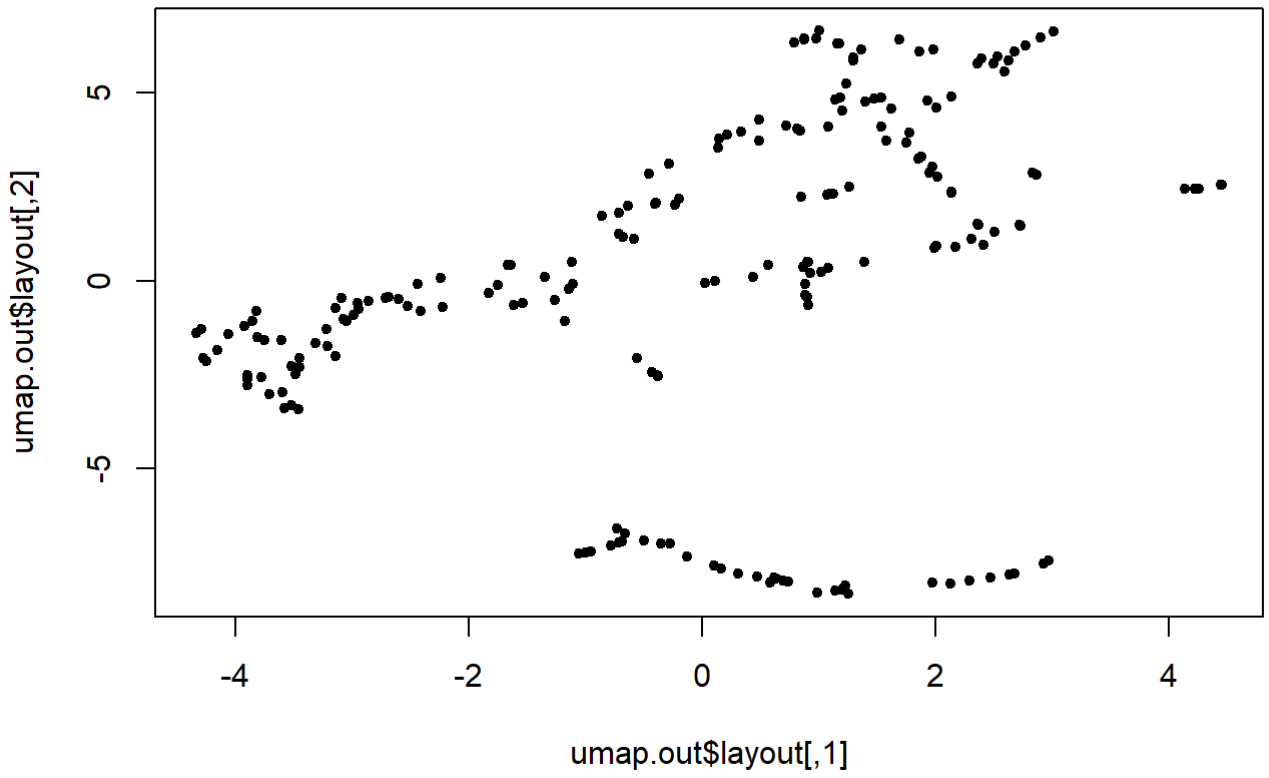
```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

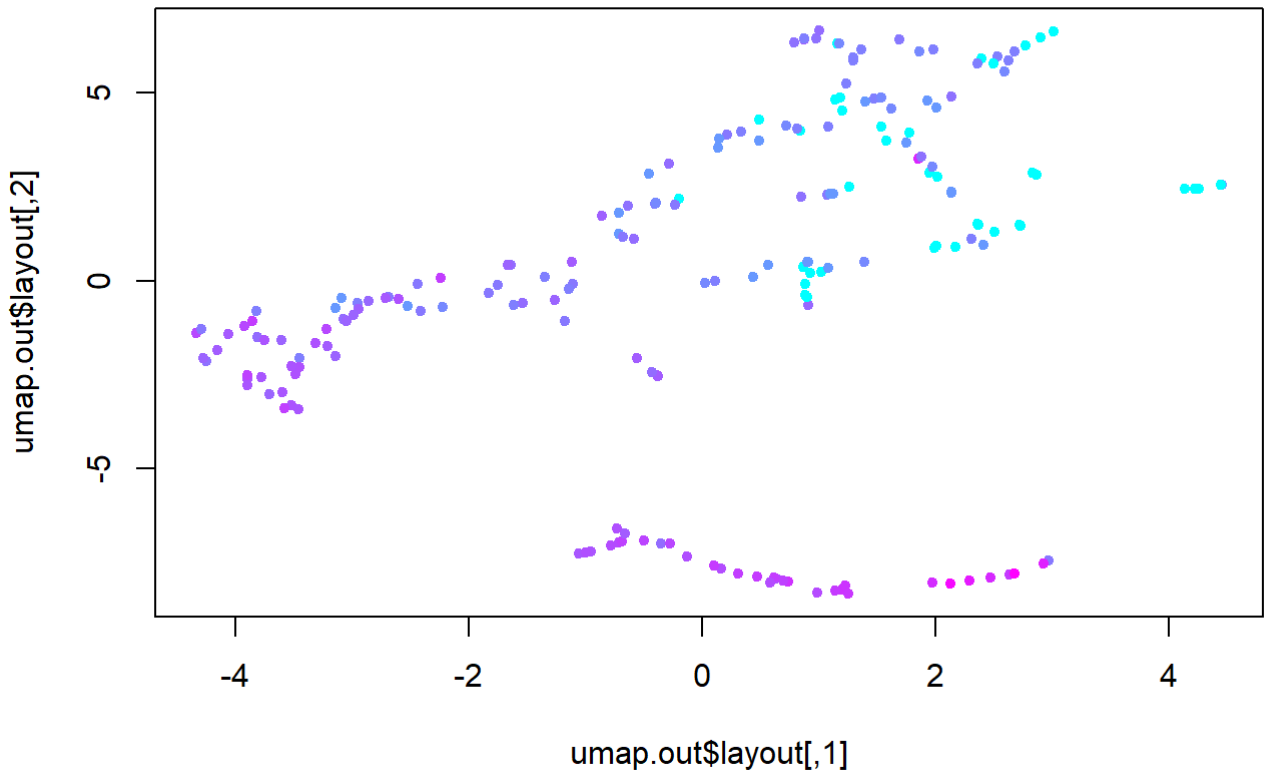
```
data <- demo.obj@raw.data
custom.settings = umap.defaults
custom.settings$n_neighbors = 3

umap.out <- umap(data,config=custom.settings)
head(umap.out$layout)
```

```
##               [,1]       [,2]
## ACTB    -0.3829263 -2.5289519
## ACTG1   -0.4042539  2.0525558
## AIF1     1.9947418  0.8686044
## ALDOA    2.1359478  2.3625201
## ANXA1    1.1430706  4.8172555
## ARHGDIB -0.6373846  2.0015010
```

```
plot(umap.out$layout,pch=20)
```



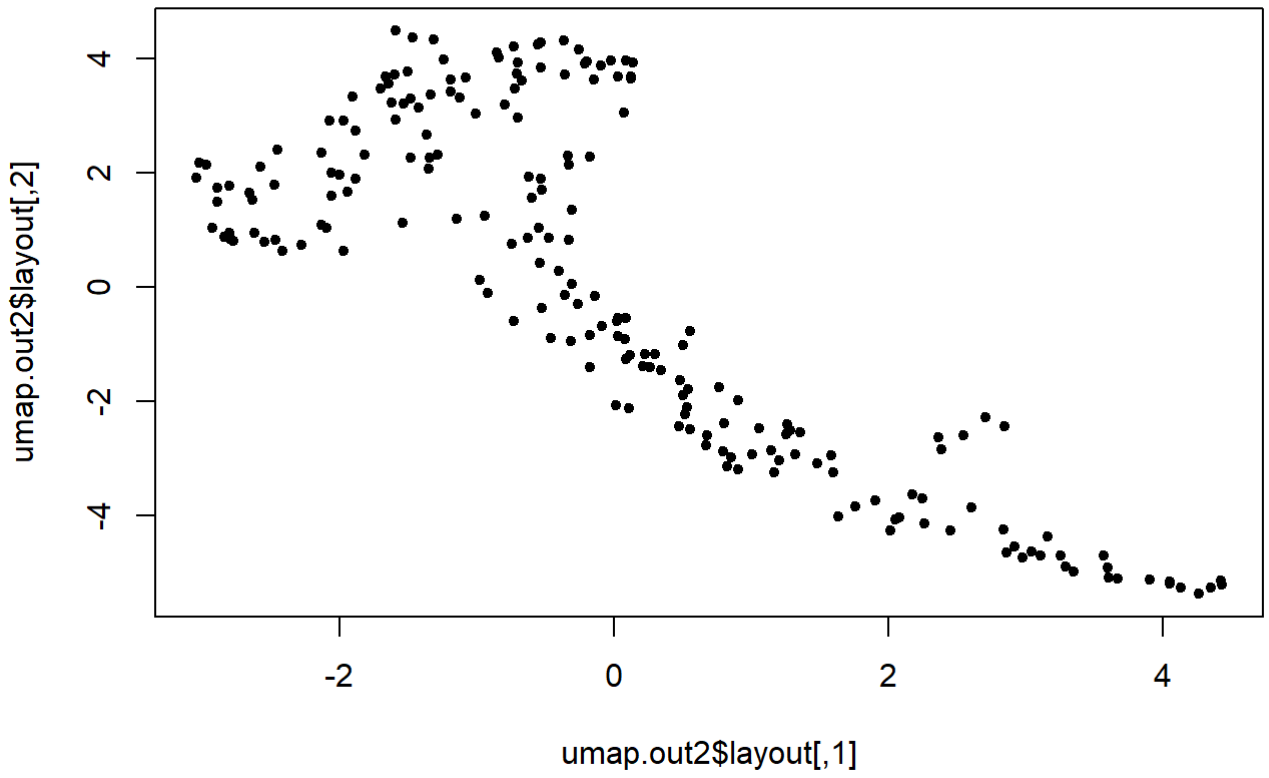

```
# 特定の遺伝子発現量の多寡で色塗りしてみる
# 色塗りの具合の見栄えのために、無意味な値変換をしているので、その点は無視してほしい
col <- data[,12]
col <- (col-min(col))/(max(col)-min(col))
col <- col^0.2
plot(umap.out$layout,pch=20,col=rgb(col,1-col,rep(1,length(col))))
```



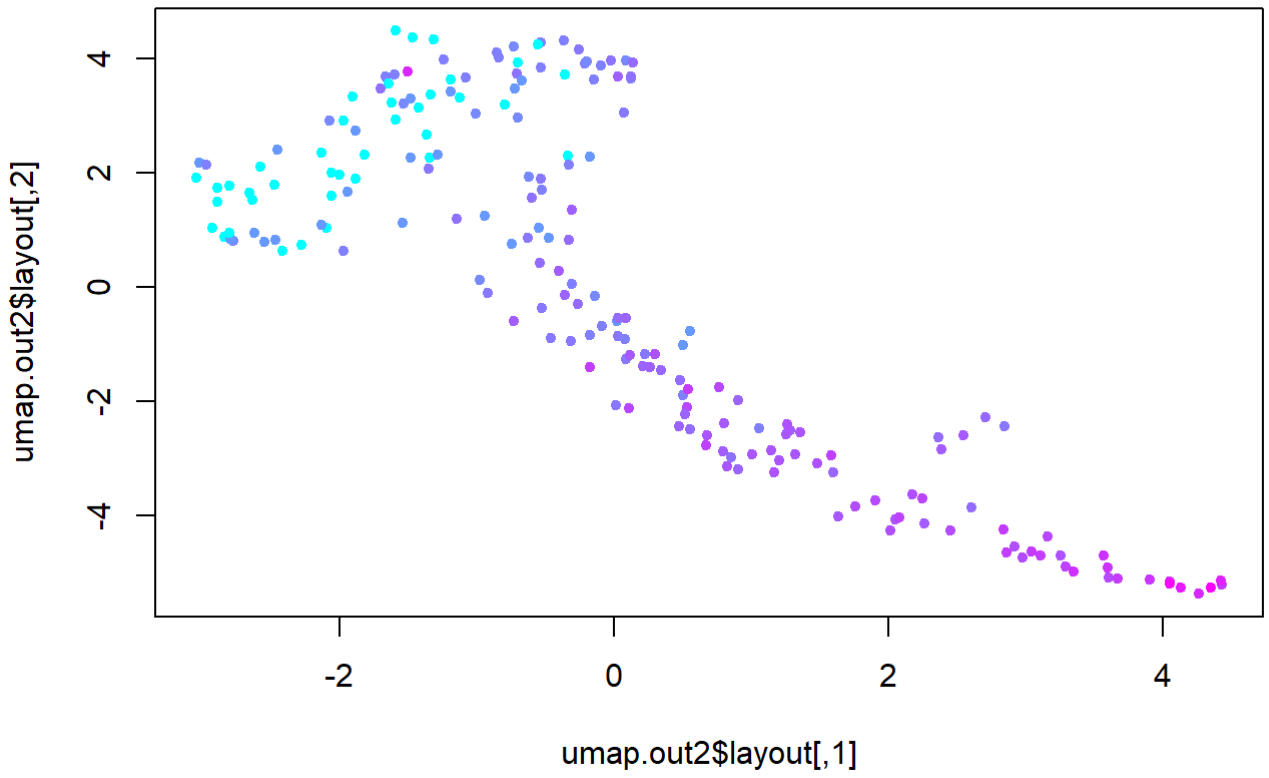
UMAP実行条件パラメタを変えてみる。

２次元プロットの様子は随分、変化することが見て取れます。

```
custom.settings = umap.defaults
custom.settings$n_neighbors = 10

umap.out2 <- umap(data,config=custom.settings)

plot(umap.out2$layout,pch=20)
```




```
# 特定の遺伝子発現量の多寡で色塗りしてみる
# 色塗りの具合の見栄えのために、無意味な値変換をしているので、その点は無視してほしい
col <- data[,12]
col <- (col-min(col))/(max(col)-min(col))
col <- col^0.2
plot(umap.out2$layout,pch=20,col=rgb(col,1-col,rep(1,length(col))))
```



この例は非線形性の手法を説明している。 非線形性の手法は、柔軟な評価ができることを特徴の１つとする。 柔軟な評価を可能にする手法の多くは、実行条件をパラメタで指定する方式を取る。 それには次のような理由が挙げられる。 極めて柔軟な手法では、出力を選ぶ自由度が高くなっているが、出力を選ぶ範囲を広げ過ぎると、出力を選び出すアルゴリズの計算量が多くなりすぎるという問題が生じる。 その問題を緩和するために、実行条件をパラメタで指定することで、パラメタ依存的に、出力の範囲を絞るという方法が取れる。 このようにして、出力の自由度を大きくしつつ、実際の出力決定に当たっては、ある程度の制約を施すことで、処理として現実的なものとしている。 また、出力がパラメタ依存になることから、線形手法の決定論的特性に比べて、非決定論的側面も有することがわかる。

マルチプルテスティング補正の考え方 FWER

帰無仮説検定で算出されるp値は、「帰無仮説が真」であるとの仮定に立った時、得られたデータと同じかそれよりも帰無仮説から逸脱したデータが得られる確率として定義される。

その定義を満足するように検定手法は設計され、その結果、帰無仮説が真であるとき、p値は0から1の一様分布を取る。

今、N個の独立した検定を行うことを考える。 N個すべての検定で帰無仮説が真であるとする。 それぞれの検定のp値は一様分布に従う。

独立な確率事象が同時に起きる確率は、個々の事象の確率の積となるから、N個の検定が相互に独立ならば、すべてのp値がある値 α未満である確率は、α^N となる。

逆に、すべてのp値がαより大きい確率は、$(1-\alpha)^N$となる。

従って、N個の検定の全てで帰無仮説が真なのに、「たまたま」棄却してしまう検定が１個もない確率は、検定閾値をαとしたとき、

$$(1-\alpha)^N$$

となる。

逆に言うと、１個以上の仮説を棄却してしまう確率(１個以上の偽陽性を出してしまう確率)は

$$1-(1-\alpha)^N$$

となる。

従って、偽陽性を１個以上、出してしまう確率を、t(例えば、0.05)にしたければ、

$$1-(1-\alpha)^N = t = 0.05$$

を満足するようなαを閾値にして、個々の検定のp値の棄却の判断をすればよい。

そのような棄却水準は、以下の式で与えられる。

このようにして設定する棄却水準で個々の検定のp値(素のp値、nominal p-値)の棄却を再評価する方法を、Family-Wise error rate法による多重検定補正法と言う。

$$\alpha = 1-(1-t)^{\frac{1}{N}}$$

今、

$$f(\alpha) = (1-\alpha)^N = 1-t$$

を$\alpha = 0$で級数展開し、その第２項までで近似すると

$$f(\alpha) = \sum_{i=0}^{\infty} \frac{1}{i!} f^{(i)}(0)\alpha^i$$

$$\begin{aligned}\hat{f}\,(\alpha) &= f^{(0)}(0)\alpha^0 + f^{(1)}(0)\alpha^1 \\ &= 1 - N\alpha = 1-t\end{aligned}$$

従って

$$\alpha \sim \frac{t}{N}$$

これは、素のp値が棄却水準 t (=0.05)を検定数Nで割って小さくした値より小さいときに初めて、帰無仮説を棄却するということになる。

この基準はBonferroni法と呼ばれる方法に相当している。

Bonferroni法は単純で保守的(偽陽性が出にくくなる)方法なので、頻用される多重検定補正法である。

$$f^{(1)}(t) = \frac{d}{dt}f(t) = \frac{1}{N}(1-t)^{\frac{1}{N}-1}$$

$$f^{(2)}(t) = \frac{d}{dt}f^{(1)}(t) = (-1)\frac{1}{N}(\frac{1}{N}-1)(1-t)^{\frac{1}{N}-2}$$

$$\cdots$$

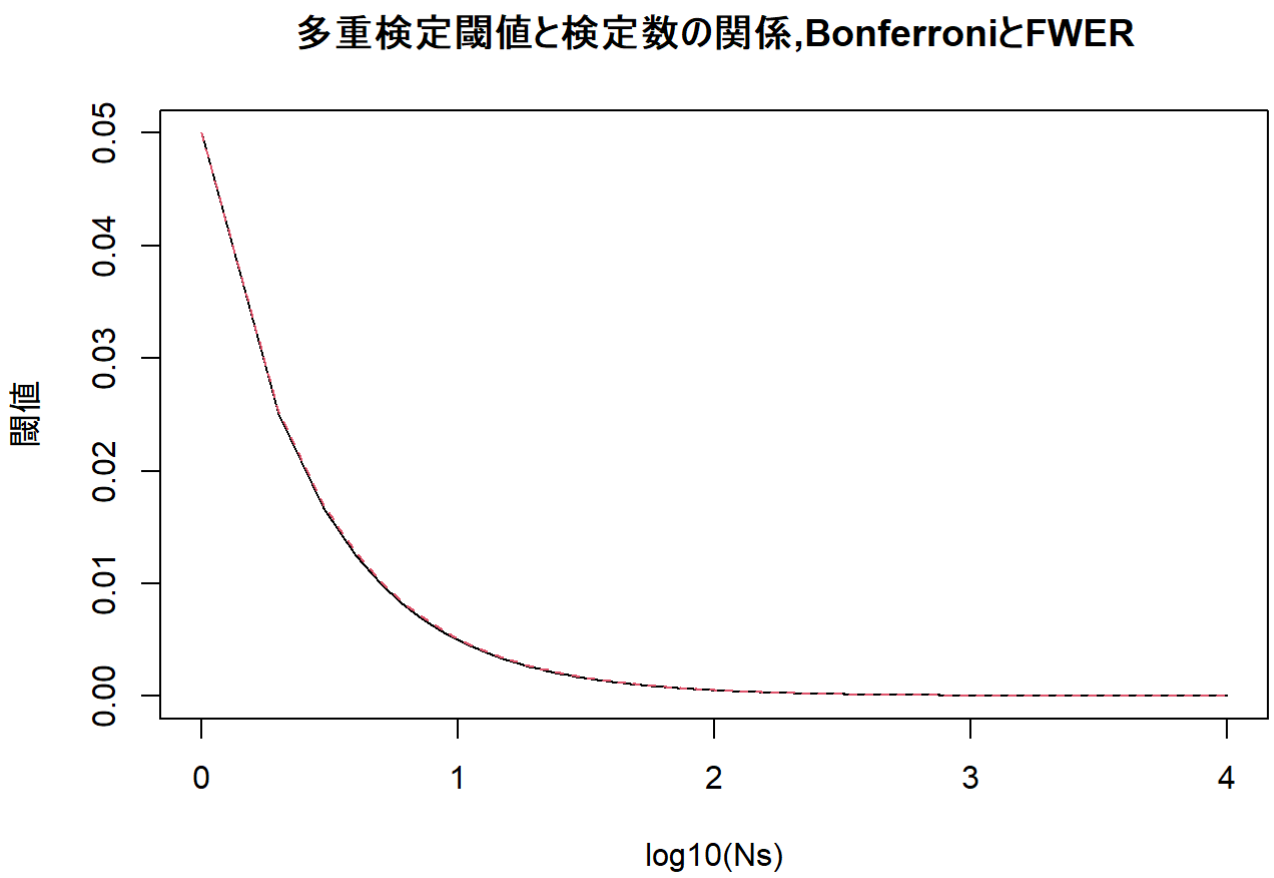
$$f^{(k)}(t) = \frac{d}{dt}f^{(k-1)}(t) = (-1)^{k-1}\prod_{i=1}^{k}(\frac{1}{N}-(i-1)) \times (1-t)^{\frac{1}{N}-k}$$

$$\alpha = \sum_{i=0}^{\infty} \frac{1}{i!} f^{(i)}(t=0)$$

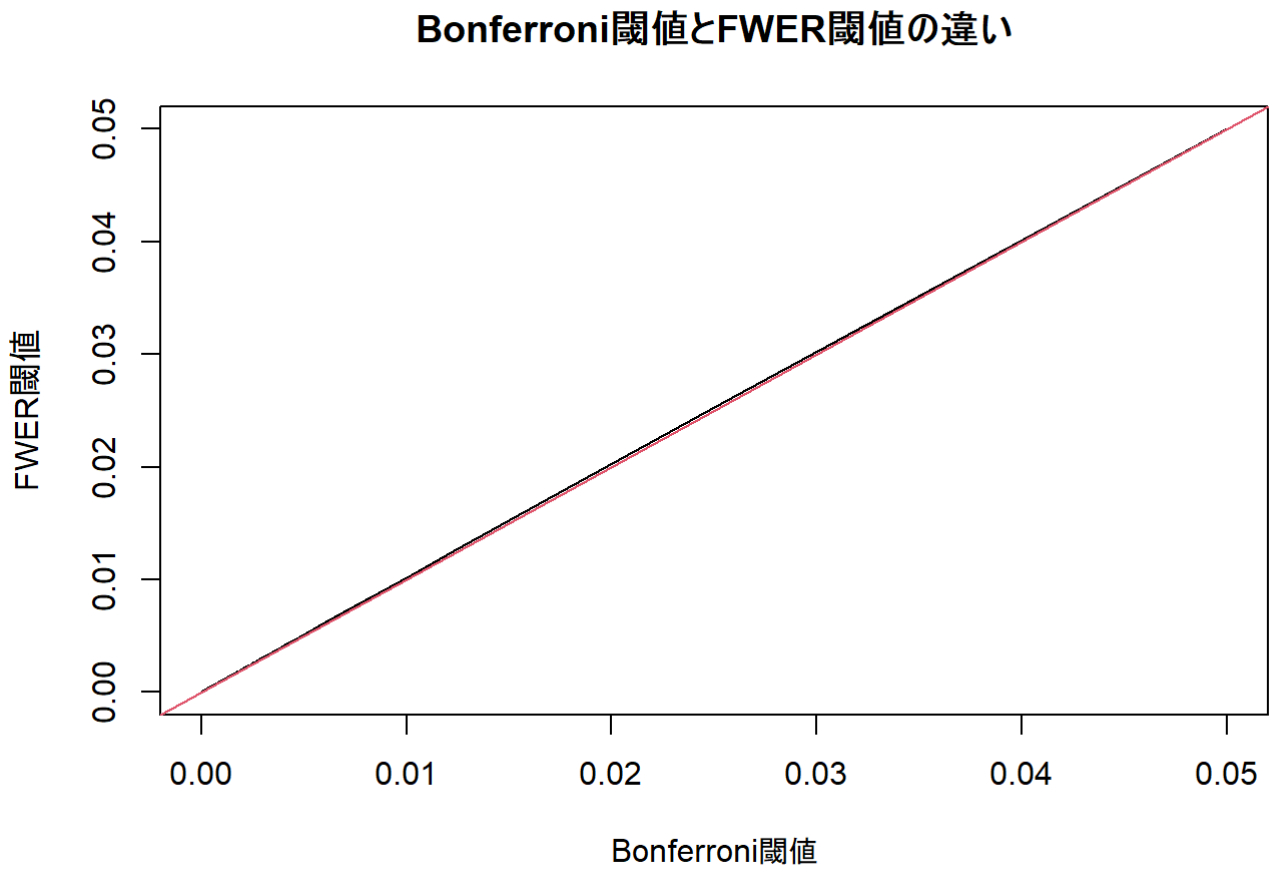
Nの値を変えて、$1-(1-\alpha)^{\frac{1}{N}}$の値と$\frac{\alpha}{N}$の値を比較してみる。

タイプ１エラーを0.05で比べると、両法にほとんど差がないことが判る。

```
Ns <- 1:10000
alpha <- 0.05
t.bonferroni <- alpha/Ns
t.FWER <- 1-(1-alpha)^(1/Ns)
matplot(log10(Ns),cbind(t.bonferroni,t.FWER),type="l",ylab="閾値",main="多重検定閾値と検定数の関係,BonferroniとFWER")
```




```
plot(t.bonferroni,t.FWER,xlab="Bonferroni閾値",ylab="FWER閾値",main="Bonferroni閾値とFWER閾値の違い",type="l")
abline(0,1,col=2)
```

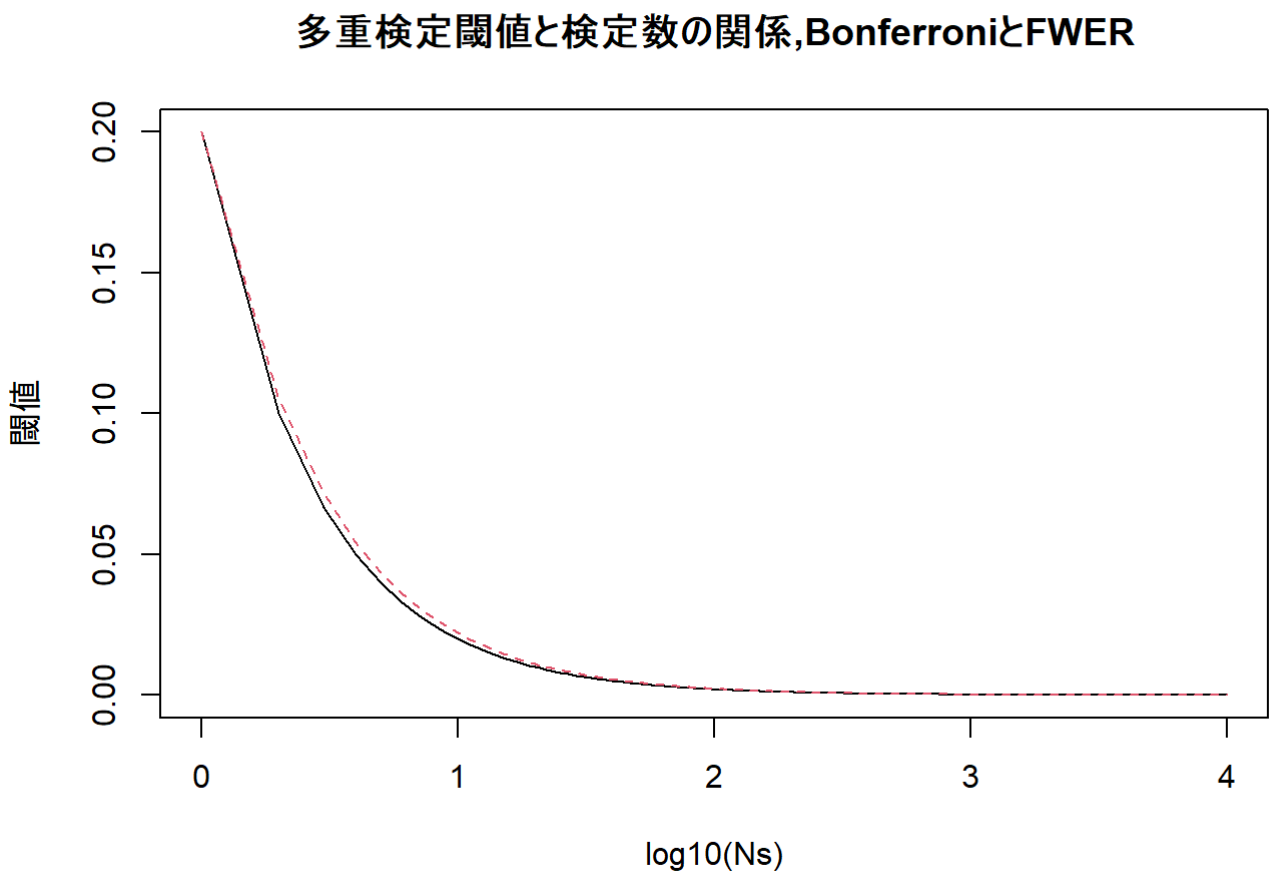


両法の違いを際立たせるために、タイプ１エラーの値を大きくして視覚化すると以下のようになる。

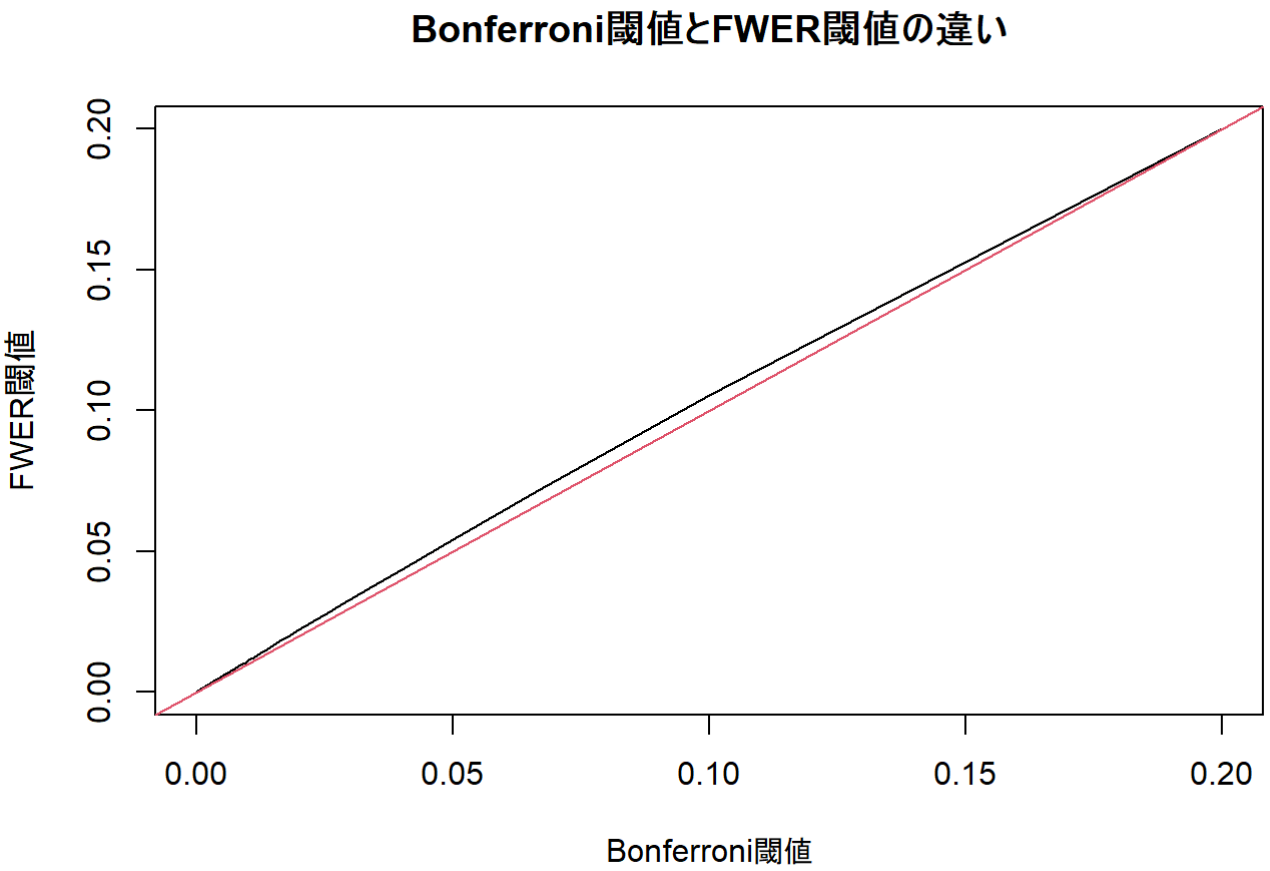
黒のライン(Bonferroni法)の閾値の方が赤のライン(FWER法)より小さめになることがかろうじて見て取れる。

両法の差がわずかだが、Nが大きくなると、１個、数個の偽陽性の有無への影響は増幅されることに注意する。

```
Ns <- 1:10000
alpha <- 0.2
t.bonferroni <- alpha/Ns
t.FWER <- 1-(1-alpha)^(1/Ns)
matplot(log10(Ns),cbind(t.bonferroni,t.FWER),type="l",ylab="閾値",main="多重検定閾値と検定数の関係,BonferroniとFWER")
```




```
plot(t.bonferroni,t.FWER,xlab="Bonferroni閾値",ylab="FWER閾値",main="Bonferroni閾値とFWER閾値の違い",type="l")
abline(0,1,col=2)
```



多重検定補正においては、１段階目で個々の検定の素のp値を出し、２段階目で、そのp値の解釈を多重検定の文脈で調整していると言う意味で、２段階処理になっている。

FWER、Bonferroni法においては、いくつの検定を行ったかという情報のみを用いて２段階目の処理が行われている。

マルチプルテスティング補正の考え方2 FDR

FWER法とBonferroni法では、すべての帰無仮説が真であることを想定してタイプ１エラーをコントロールしていました。

FDR(False Discovery Rate)法では、N個の仮説のうち、ある割合βでは帰無仮説が偽で、残りの仮説で帰無仮説が真だと言う前提で考えます。

このとき、N個の検定のp値を小さい順に並べ、i番目に小さいp値に対して、$\beta \times \frac{i}{N}$という値を棄却水準にすることにします。

この棄却水準を下回る小さなp値のうち、最も大きいp値がk番目に小さいp値だったとき、小さい方からk番目までのp値を持つ仮説の帰無仮説を棄却します。

このようにすることで、帰無仮説が棄却される仮説のうち、割合がおよそβの割合の仮説が偽陽性になります。

この偽陽性の割合のことを、FalseにDiscoveryされたRateと称して、FDR法と称します。

ここで書いた、p値の大小順別の棄却水準の決め方はFDR法の１つで、Benjamini＆Hochberg法(BH法)と呼ばれれます。

わかりやすく、よく使われる方法です。

以下のように考えます。

仮説のうちβの割合の仮説は、対立仮説が真であるので、標本数が無限大だと、p値は0とみなせます。残りの$1-\beta$の割合の仮説のp値は一様分布を取ります。

従って、p値を小さい順に並べると、横軸xの値が0からβまでは、縦軸$y=0$、xがβから1までは、$y=\frac{1}{1-\beta}(x-\beta)$の直線になります。

他方、検定棄却水準は、横軸xに応じて変わりますが、その値は$y_{thres}=\beta x$という直線です。

この交点を計算すると

$$(x,y)=(\frac{\beta}{\beta^2-\beta+1},\frac{\beta^2}{\beta^2-\beta+1})$$

となる。

検定棄却直線よりも昇順p値の折線が下に来る割合は、$\frac{\beta}{\beta^2-\beta+1}$、そのうち、対立仮説が真なのは$\beta$、帰無仮説が真なのは$\frac{\beta}{\beta^2-\beta+1}-\beta$になる。

棄却される仮説に占める、帰無仮説が真である仮説の割合(FDR)は

$$FDR=\frac{\frac{\beta}{\beta^2-\beta+1}-\beta}{\frac{\beta}{\beta^2-\beta+1}}=\beta-\beta^2$$

βの値が小さめだとするとβ^2は無視できるくらい小さいとすれば、

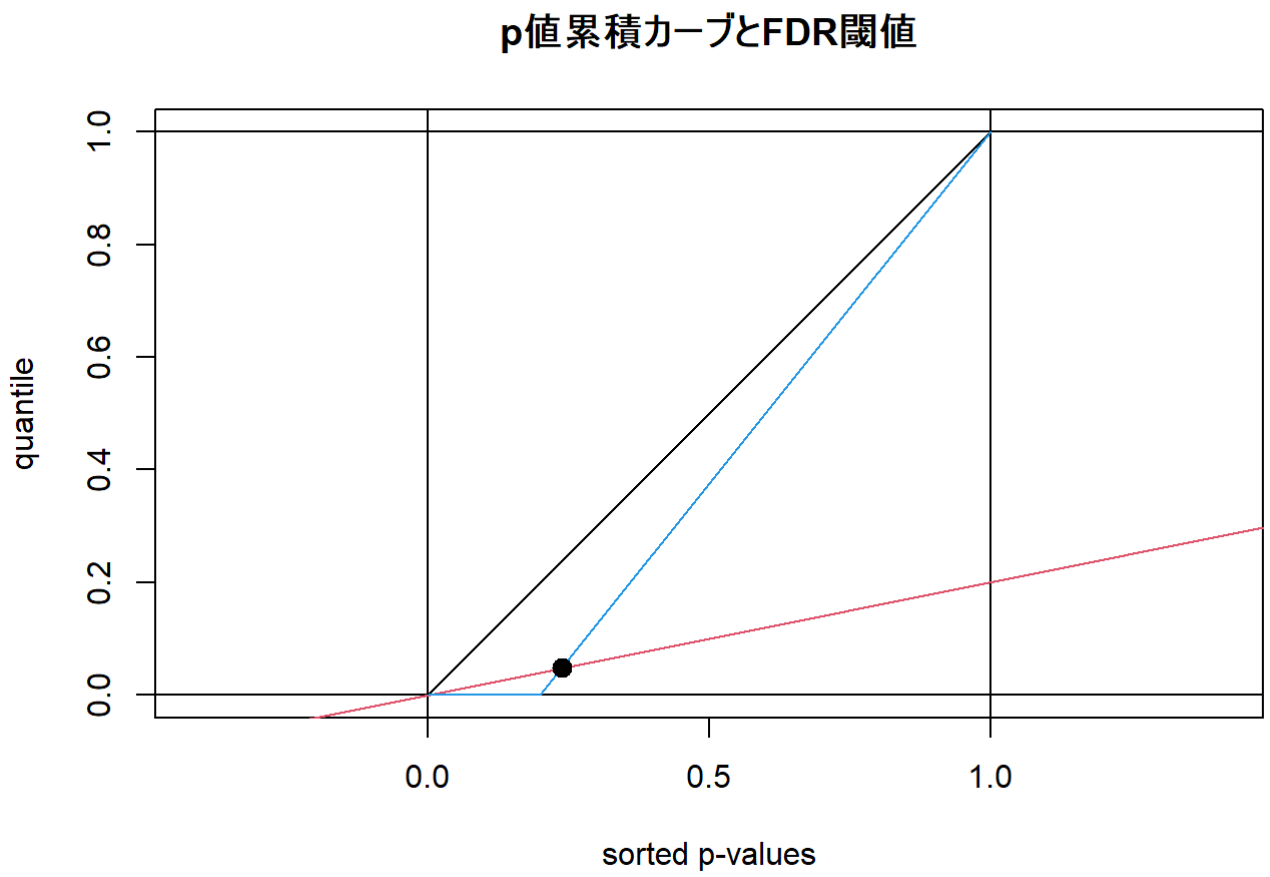
$$FDR\sim\beta$$

となり、棄却仮説に占める偽陽性の割合がβくらいにコントロールできることになる。

昇順p値の折線と棄却水準の線とを描く。

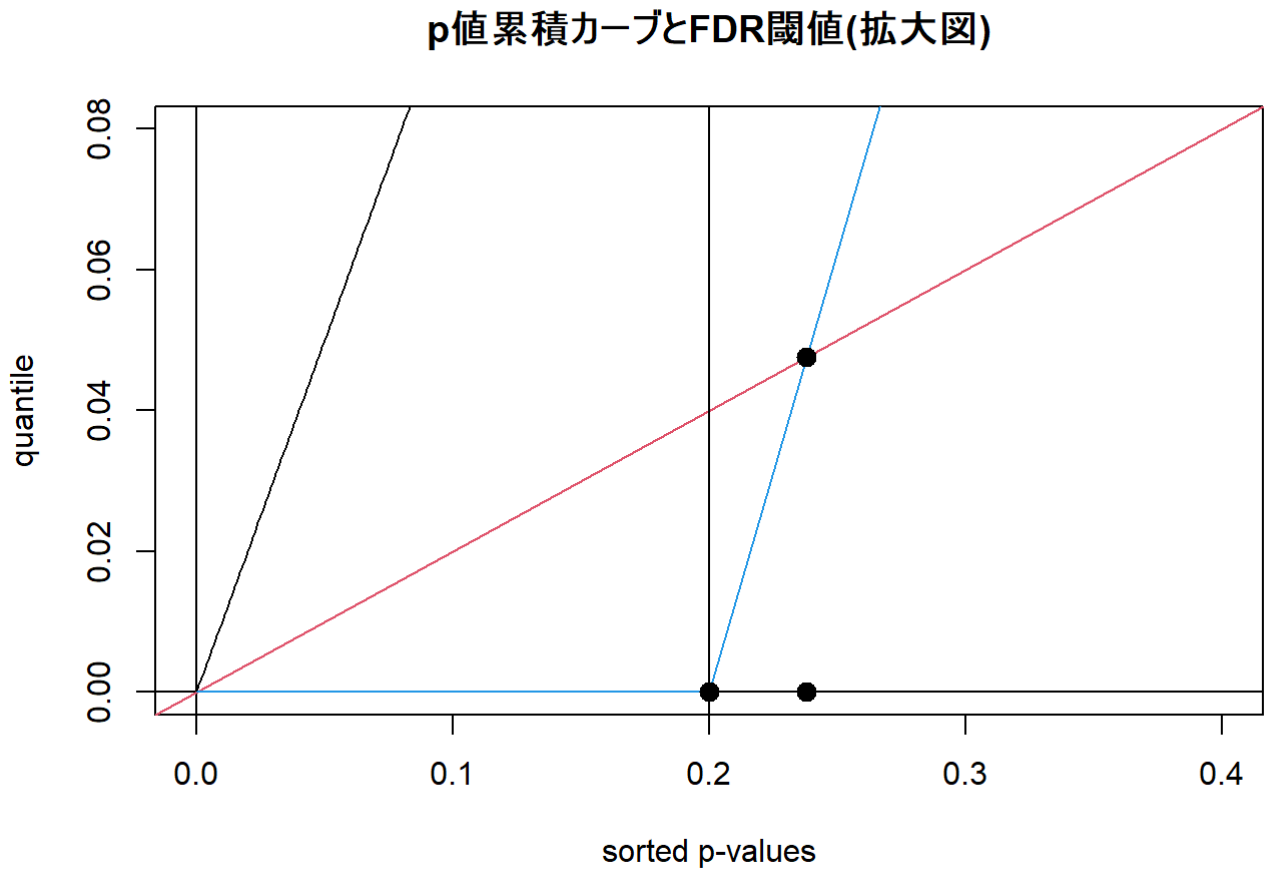
黒線は$y=x$の線、青線は、理論的・理想的なp値の累積分布に相当する折線。 赤線は、p値の昇順ごとに異なる帰無仮説棄却閾値を示す。

```
t <- seq(from=0,to=1,length=1000)
beta <- 0.2
thres.x <- t
thres.y <- 1 * t
plot(thres.x,thres.y,type="l",main="p値累積カーブとFDR閾値",xlab="sorted p-values",ylab="quanti
le",asp=TRUE)
abline(v=c(0,1))
abline(h=c(0,1))
abline(0,beta,col=2)
segments(0,0,beta,0,col=4)
segments(beta,0,1,1,col=4)
X <- beta/(beta^2-beta+1)
Y <- beta^2/(beta^2-beta+1)
points(X,Y,pch=20,cex=2)
```



青線と赤線との交点付近を拡大した図を描く。

```
t <- seq(from=0,to=1,length=1000)
beta <- 0.2
thres.x <- t
thres.y <- 1 * t
plot(thres.x,thres.y,type="l",main="p値累積カーブとFDR閾値(拡大図)",xlab="sorted p-values",ylab
="quantile",xlim=c(0,beta*2),ylim=c(0,beta^2*2))
abline(v=c(0,1))
abline(h=c(0,1))
abline(0,beta,col=2)
segments(0,0,beta,0,col=4)
segments(beta,0,1,1,col=4)
X <- beta/(beta^2-beta+1)
Y <- beta^2/(beta^2-beta+1)
points(X,Y,pch=20,cex=2)
points(X,0,pch=20,cex=2)
points(beta,0,pch=20,cex=2)
abline(v=beta)
```

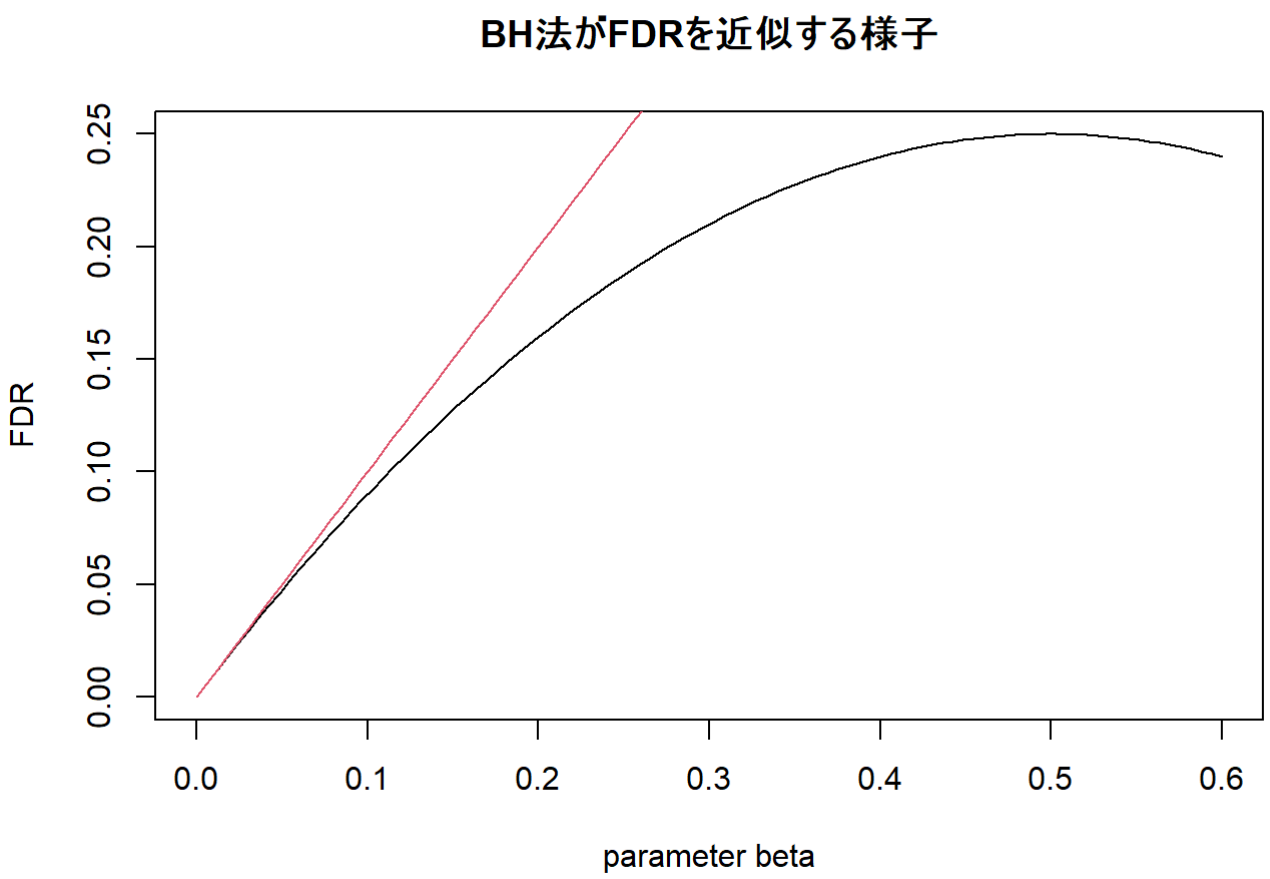


次に、βの値と、FDR($\beta - \beta^2$)の値との関係を描く。

赤の直線がBH法で目指しているFDRの値。 黒の曲線が、BH法で実現されると予想されるFDRの値。

β値が小さめのときには、両者に大差がないことが判る。

```
betas <- seq(from=0,to=0.6,length=100)
fdrs <- betas - betas^2

plot(betas,fdrs,type="l",main="BH法がFDRを近似する様子",xlab="parameter beta",ylab="FDR")
points(betas,betas,type="l",col=2)
```



多重検定補正においては、１段階目で個々の検定の素のp値を出し、２段階目で、そのp値の解釈を多重検定の文脈で調整していると言う意味で、２段階処理になっている。

FDR法においては、いくつの検定を行ったかという情報だけではなく、個々のp値が全p値の中で何番目に小さいかの情報も使って２段階目の処理を行っている。

Selective inference の考え方

多段階解析の例としてSelective inferenceを取り上げる。

Selective inferenceを個々の問題に適用して、適切な判断をするための手法は発展途上の部分が多いのが現状である。

従って、ここでは、その考え方の枠組みを紹介することとする。

そして、この問題への対処が、医学・生命科学の研究成果報告の再現性の乏しさの改善に資すると期待されていることを理解することを、本書での目標とする。

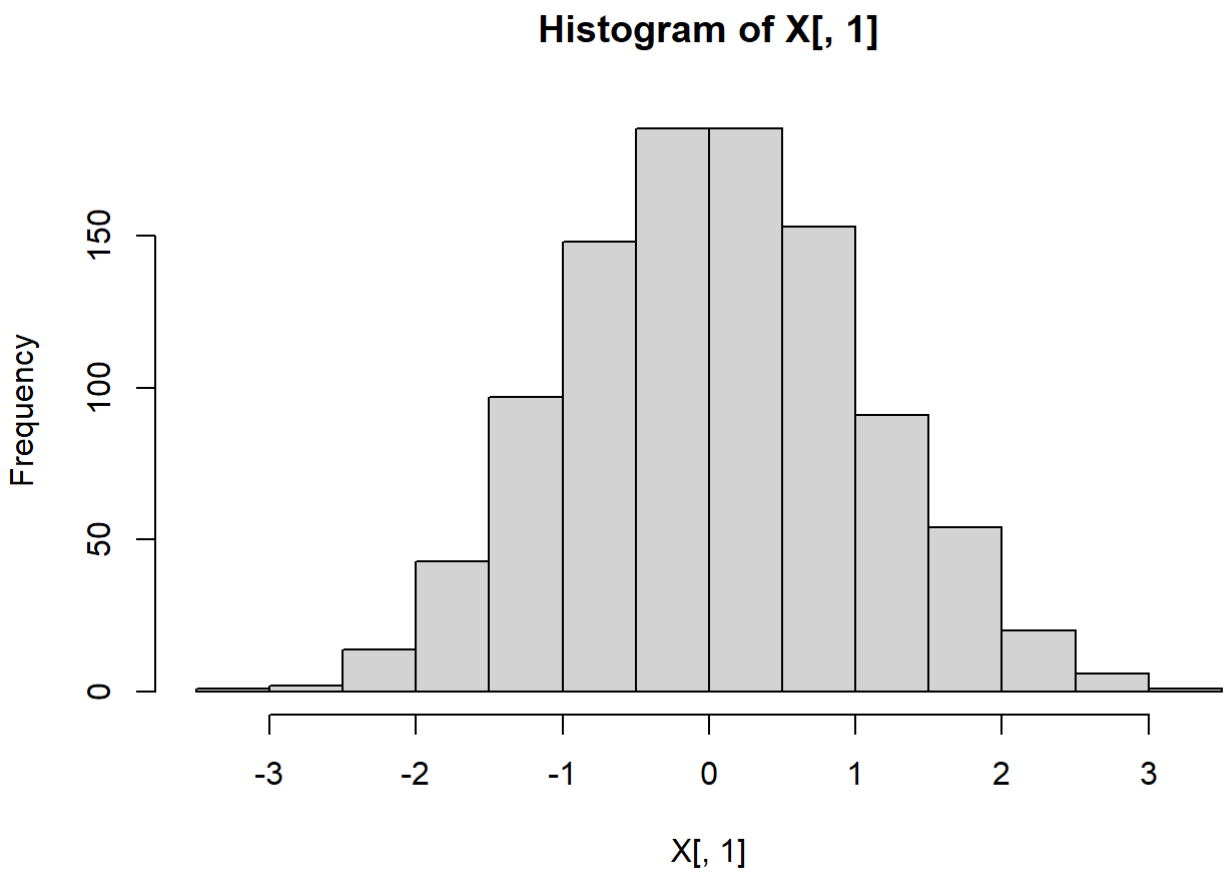
以下に例を挙げる。

きれいな正規分布に従う２個の変数があり、多数の標本についてその変数の値を観測したとする。

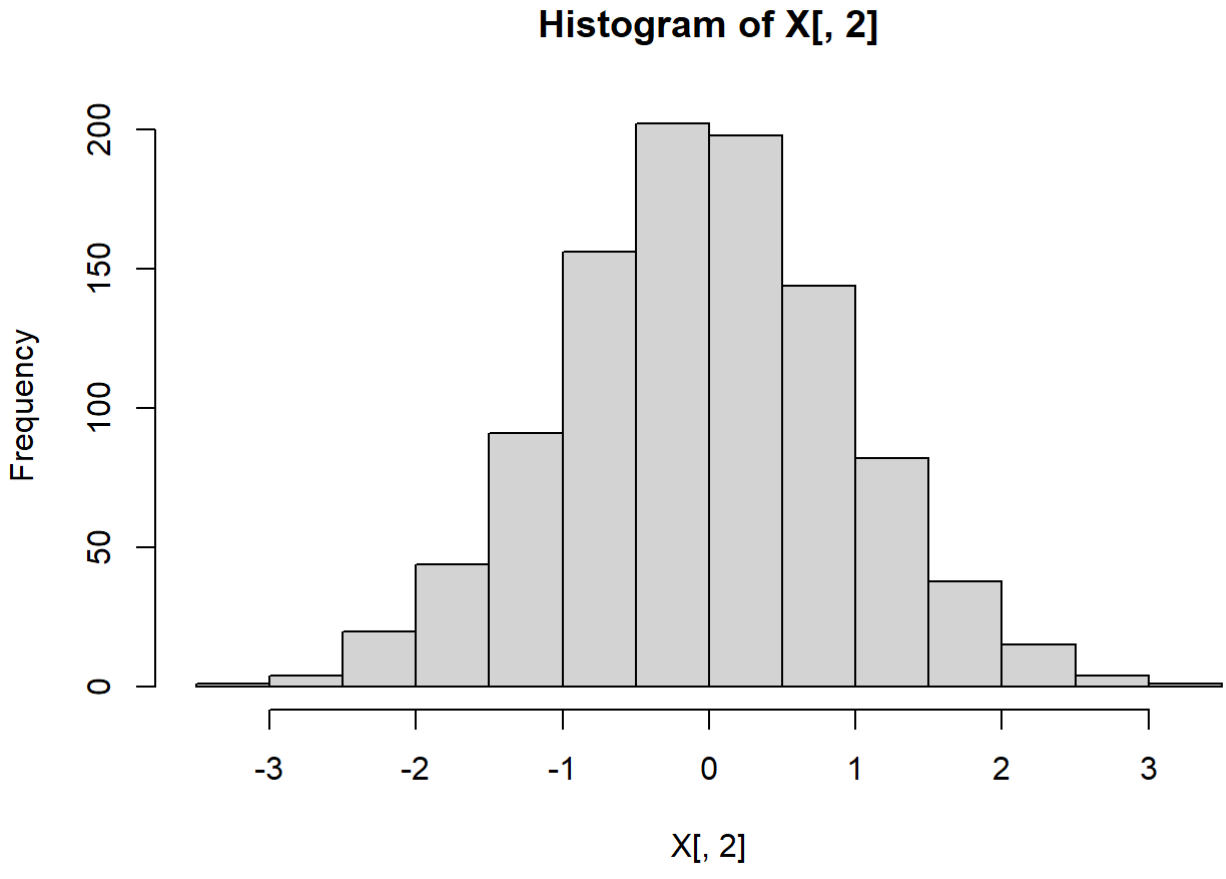
今、２群にクラスタリングする。これを第一ステップとする。

そのようにして出来た２群に対して、平均値の差の検定をしてみる。

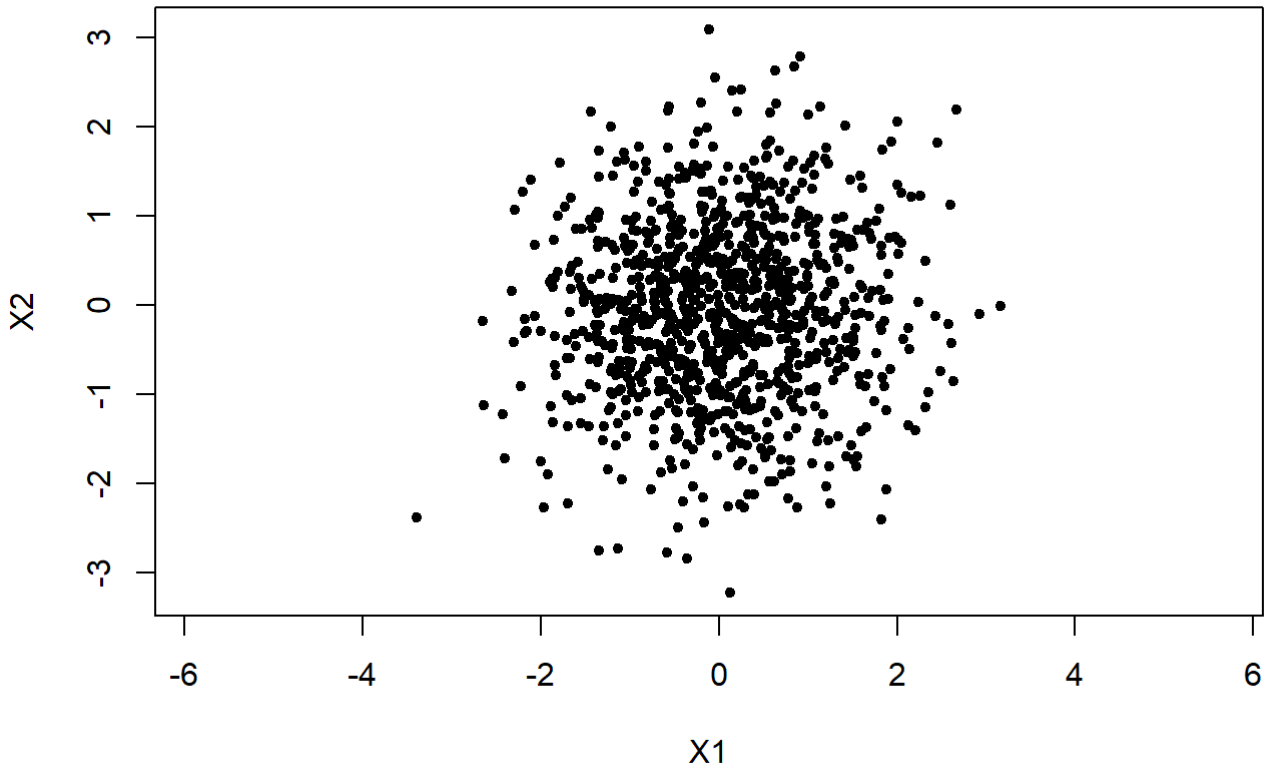
```
n.sample <- 1000
n.feature <- 2
X <- matrix(rnorm(n.sample * n.feature),ncol=n.feature)
hist(X[,1])
```


Histogram of X[, 1]
Frequency
0
50
100
150
-3
-2
-1
0
1
2
3
X[, 1]

```
hist(X[,2])
```


Histogram of X[, 2]
Frequency
0
50
100
150
200
-3
-2
-1
0
1
2
3
X[, 2]

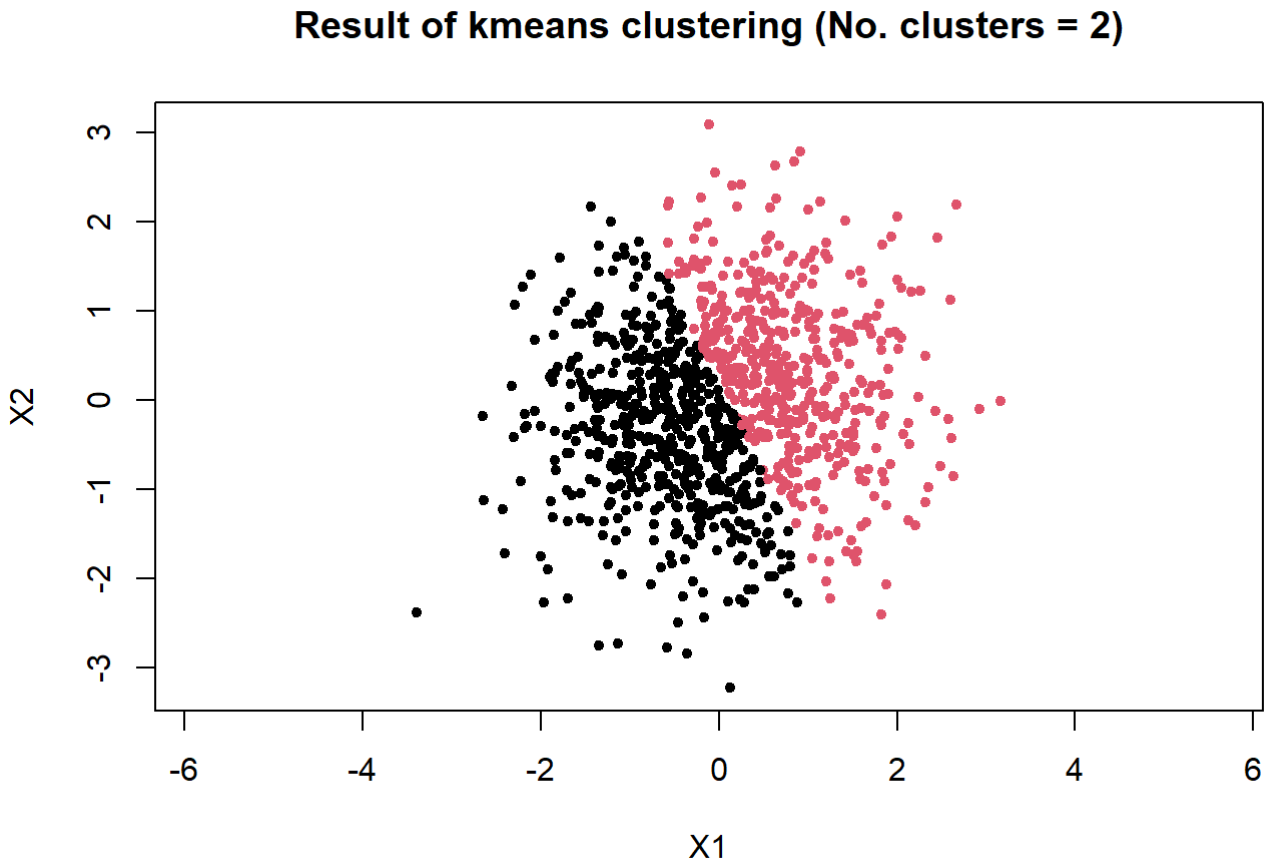
```
plot(X,pch=20,xlab="X1",ylab="X2",asp=TRUE)
```



kmeans法で２群に非階層クラスタリングする。

２群を色分けして表示する。

```
km <- kmeans(X,2)
plot(X,asp=TRUE,pch=20,col=km$cluster,xlab="X1",ylab="X2",main = "Result of kmeans clustering
 (No. clusters = 2)")
```



第一段階で、標本が２群に分かれたので、その情報を使って、第一の説明変数について、２群間で平均値の差があるかどうかをt-検定してみる。 このt-検定が２段階目の処理である。

```
my.data <- data.frame(cl=km$cluster,X)
head(my.data)
```

```
##   cl          X1          X2
## 1  2  1.17934117 -0.42220742
## 2  2  0.44237176  0.02689393
## 3  2  0.94889486 -0.17634291
## 4  1 -0.34859792 -0.98357515
## 5  2  0.02749754  0.76676101
## 6  1 -0.69396598 -0.64359345
```

```
t.test(X1 ~ cl, data=my.data)
```

```
##
##  Welch Two Sample t-test
##
## data:  X1 by cl
## t = -33.891, df = 966.31, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.552376 -1.382436
## sample estimates:
## mean in group 1 mean in group 2
##      -0.6314022       0.8360037
```

２群間の平均値の差がないという帰無仮説は非常に小さなp値で棄却される。

この結果の解釈について考えよう。

そもそも、標本の２変数の分布はキレイな二次元正規分布であり、２群に分ける根拠は全く無かった。

しかしながら、kmeans法を適用することによって、無理やり２群に分けてやった。 分けるに際して、２変数を使って分けたので、その２群の間で２変数の値に違いがあるのは、当然のことであり、２群間でその変数について差があるかどうかを検定したときに、統計的有意差を持って差があると判定されることは驚くに値しない。

p値は「そんなに違うとは驚きだ」という気持ちを数値にしたものなので、「驚くに値しない」のであれば、このp値は「驚き」の程度を適切に数値化しているとは考えがたい。

Selective inferenceの基本的な考え方は、このようなものである。

この例では、クラスタリングをしたことがそもそも不適切であったので、この結果の解釈のおかしさを、そのせいだと考えるかもしれない。

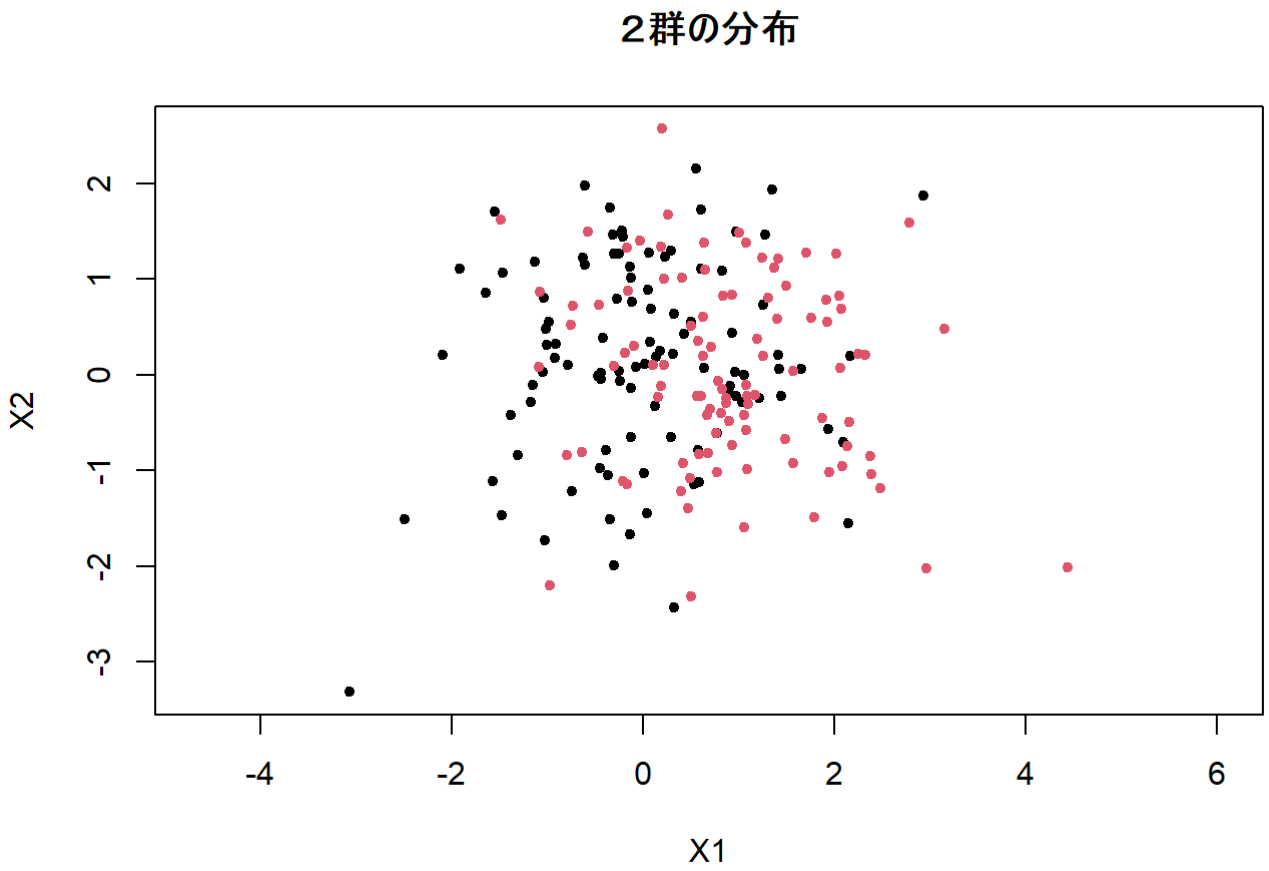
次の例をみてみよう。

明らかに母集団が２群に分かれているとする。

但し、母集団からサンプルすると、その２群の標本の取る値にはある程度の重なりがあるものとする。

以下に図で示す。

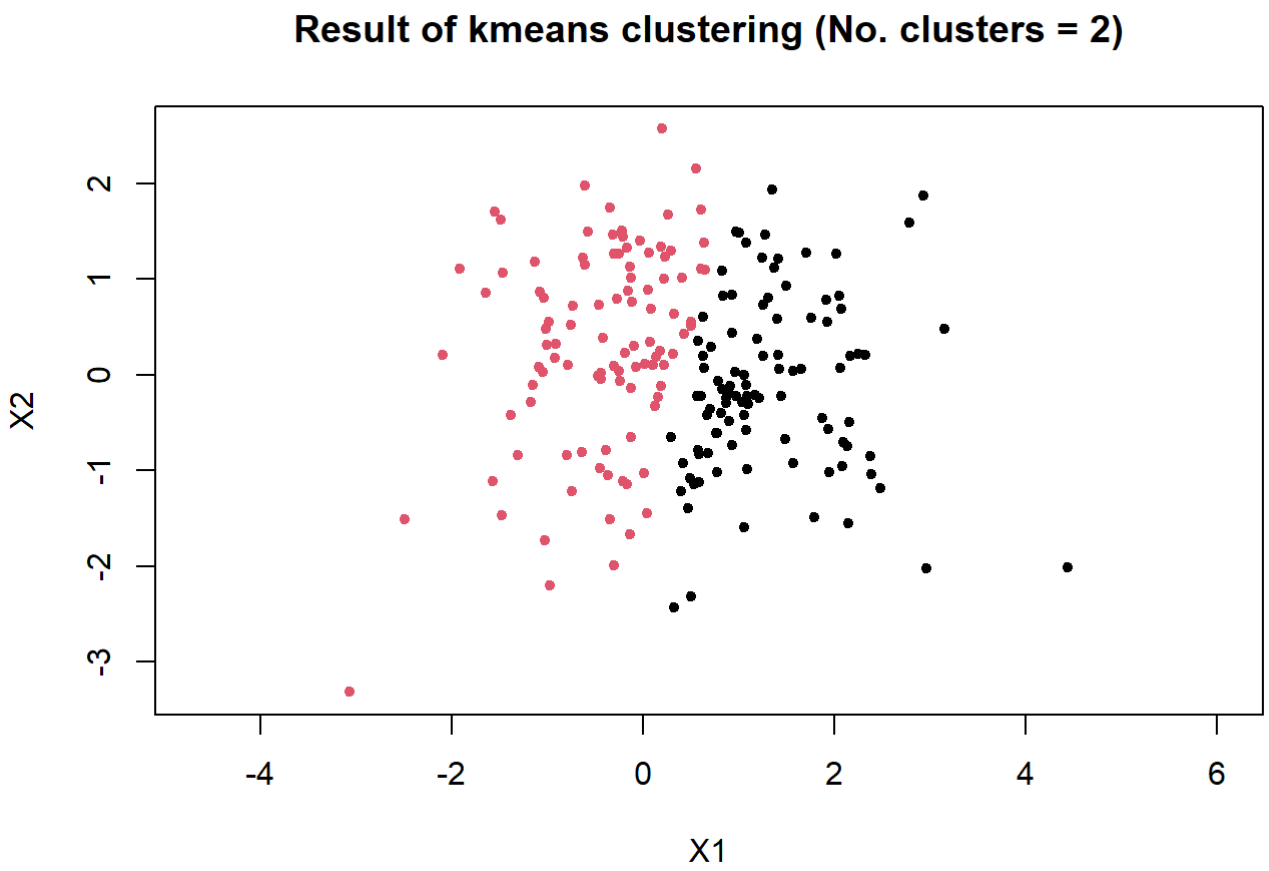
```
n.sampleA <- 100
n.sampleB <- 100
n.feature <- 2
XA <- matrix(rnorm(n.sampleA * n.feature,0,1),ncol=n.feature)
XB <- matrix(rnorm(n.sampleB * n.feature,0,1),ncol=n.feature)
XB[,1] <- XB[,1] + 1

AB.label <- c(rep(1,n.sampleA),rep(2,n.sampleB))
X.AB <- rbind(XA,XB)

plot(X.AB,asp=TRUE,pch=20,col=AB.label,xlab="X1",ylab="X2",main="２群の分布")
```



標本が２群のどちらの由来かが判ったものとして、上図は描いたが、由来群が不明であったとする。

群をkmeans法で推定してみる。

```
km2 <- kmeans(X.AB, 2)
plot(X.AB, asp=TRUE, pch=20, col=km2$cluster, xlab="X1", ylab="X2", main = "Result of kmeans clusteri
ng (No. clusters = 2)")
```



色分けの具合を見ると、スパっと２群に分かれている。 このkmeans法による２群分けは、真の群とは異なっていることも見て取れる。

では、 ２群の帰属情報が、真の情報の場合と、kmeans法で推定した場合とで、 ２群間の平均値の差の検定をしてみる。

どちらも、非常に小さいp値を返してるが、kmeans法のクラスタリングの結果を利用した場合のp値の方が極端に小さくなっているのがわかる。

観測データに基づいて群分けをすると、真の群分けから逸脱するが、その逸脱の影響が群間の差を大きくしていると理解できる。

クラスタリングと、それに引き続く群間比較におけるSelective inferenceの考え方は、「群分けに使われた変数を、群別に比較すると、群間差が過大評価されるので、その分を調整しなければ、群間差があるという主張は言い過ぎになる。その結果、研究の再現性が下がる」と言うものである。

```
my.data.original <- data.frame(cl=AB.label,X.AB)
t.test(X1 ~ cl, data=my.data.original)
```

```
##
##  Welch Two Sample t-test
##
## data:  X1 by cl
## t = -6.3453, df = 197.81, p-value = 1.483e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.2170484 -0.6399241
## sample estimates:
## mean in group 1 mean in group 2
##     -0.02353168      0.90495456
```

```
my.data.kmeans <- data.frame(cl=km2$cluster,X.AB)
t.test(X1 ~ cl, data=my.data.kmeans)
```

```
##
##  Welch Two Sample t-test
##
## data:  X1 by cl
## t = 17.321, df = 197.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.553825 1.953098
## sample estimates:
## mean in group 1 mean in group 2
##        1.326209       -0.427252
```

データの解析と確率事象

ポイント(データの解析と確率事象)

- 確率事象、確率変数、確率密度分布
- 検定統計量とp値
- 確率と尤度
- 尤度関数、最尤推定、最適化

はじめに

データを取るとき、複数の(たくさんの)観察をします。 データを取るたびに、値が変わるからです。

同じものを同じ方法で計測しても、計測のたびに値が変わります。 ある人の身長を続けて複数回、測定すると、わずかですが測定値は変わるように。 測定誤差が入るからです。

似たようなものについて、同じ変数を計測すると、計測のたびに値が変わります。複数の人の身長を測ると、身長の値がばらつくように。 個人差があるからです。測定値には測定誤差によるばらつきも入りますが、この場合は個人差の方が測定誤差より圧倒的に大きいでしょう。

このように、測定するたびに、異なる値が返って来るような現象が、データ解析の対象になります。

ですから、データ解析は複数の観測値を対象にするのです。

測定ごとに値がばらつくときに、値たちに対して施す処理がデータ解析の処理です。

ゲノム・オミックス解析の場合には、生データを本解析のために加工するときの処理も、この処理に相当しますし、本解析自体も、この処理に相当します。

ゲノム・オミックスのデータ解析は、多変量・多段階・データ駆動型の解析ですが、そのすべてに共通するのが「値がばらつくことへの対処」と言えます。

本章では、この点を取り扱います。

確率事象

確率変数と確率密度関数・確率質量関数

測定するたびに、異なる値が返って来るような現象を 『あなたの値は何ですか？』と問いかけるたびに、『異なる値を返す』ブラックボックスと見立てることにします。

このとき、このブラックボックスが『次に何の値が返すか』は判らないが、『返して来る値の分布』には『決まり』があるとします。

このとき、このブラックボックスは確率変数であり、『返してくる値の分布』は確率密度関数・確率質量関数として『決ま』っていると考えます。

例 Hardy-Weinberg平衡検定とカイ二乗分布

一塩基多型(SNP)は、遺伝子多型の一種で、DNAの塩基配列の１箇所が分子ごとに異なっている場所のことを指します。

塩基は４種(A,T,G,C)あるので、ある箇所が取りうる塩基の種類も４種類ですが、大多数の場合は、そのうちの２種類のいずれかを取ります。 ２種類の塩基をM,mで表し、M,mをそれぞれアレルと呼びます。

常染色体は母由来・父由来の２本ずつあるので、常染色体のSNPの塩基の具合を観察すると、Mが２つ、Mとmとが１つずつ、mが２つの３通りのタイプ(遺伝子型、ジェノタイプ)がありえます。

MM,Mm,mmと書いたり、Mのホモ接合体、(Mとmの)ヘテロ接合体、mのホモ接合体と呼んだりします。

今、MMが50人($N_{MM} = 50$)、Mmが38人($N_{Mm} = 38$)、mmが12人($N_{mm} = 12$)観察されたとします。

全部で50+38+12 = 100 人の遺伝子型を観察したことになります。 各人、2本の染色体を持っているので、200本の染色体のアレルを観察したことにもなります。

Mの染色体とmの染色体は、それぞれ、$2 \times 50 + 1 \times 38 = 138$本、$1 \times 38 + 2 \times 12 = 62$本、観測されています。

138のMと62のmとをごちゃまぜにして、100ペアを作ることにします。

ごちゃまぜににしてペアを作って３遺伝子型の人数を観測することは、『何人になるか？』を問合せて、そのたびに『異なる値を返す』ことなので、確率事象です。

この確率事象をでは、 Mの割合が0.79、mの割合が0.31なので、MMの割合は$0.79^2 = 0.6241$、Mmの割合は$2 \times 0.79 \times 0.31 = 0.4898$、mmの割合は$0.31^2 = 0.0961$と「期待」されます。

100ペアを作るので、MM,Mm,mmの人数は$0.6241 \times 100 = 62.41$人、$0.4898 \times 100 = 48.98$人、$0.0961 \times = 100 = 9.61$人と「期待」されます。

「期待」される、とは、何度もペアを作っては壊し、作っては壊しを繰り返して、人数を観測したとき、その平均人数がこれらの値になる、ということです。

この期待人数を$\hat{N}_{MM}, \hat{N}_{Mm}, \hat{N}_{mm}$と書くことにすると、 ごちゃまぜにしてペアを作るたびに、３遺伝子型の人数が得られる確率事象ですが、次のような計算式で得られる値を返す確率事象である、ともみなせます。

$$\chi^2 = \frac{(N_{MM} - \hat{N}_{MM})^2}{\hat{N}_{MM}} + \frac{(N_{Mm} - \hat{N}_{Mm})^2}{\hat{N}_{Mm}} + \frac{(N_{mm} - \hat{N}_{mm})^2}{\hat{N}_{mm}}$$

このχ^2の値は、ペアを作り直すたびに異なる値になりますが、その値は、おおまかに『自由度１のカイ二乗分布』に従うことが知られています。

(自由度１の)カイ二乗分布は確率密度分布の１つです。

そして式からも判るように、$(N_{MM}, N_{Mm}, N_{mm}) = (\hat{N}_{MM}, \hat{N}_{Mm}, \hat{N}_{mm})$のときに$\chi^2 = 0$となり、期待値からずれると0より大きくなります。

期待値からずれればずれるほど、χ^2の値は0からずれて大きな値を取ります。

従って、ある$(N_{MM}, N_{Mm}, N_{mm}) = (50, 38, 12)$に対して計算した
$\chi^2((N_{MM}, N_{Mm}, N_{mm}) = (50, 38, 12))$の値が大きいほど、「期待」された値から遠いことを意味します。

いくら遠くても、起きえないわけではないですが、あまりに遠いと、「ごちゃまぜにしてペアを作ってみた」という前提を疑いたくなることもあるでしょう。

なので、自由度１のカイ二乗分布に照らして、カイ二乗値が$\chi^2((N_{MM}, N_{Mm}, N_{mm}) = (50, 38, 12))$以上の値になる確率を求め、それをp値とし、このp値が棄却水準(例えば0.05)未満になったら、「ごちゃまぜにしてペアを作ってみた」という前提(帰無仮説)を棄却することにするという使い方をすることがあります。

ここで検定しているのは、「ごちゃまぜにしてペアを作った」かどうか、でしたが、「100人を取ってきた母集団で、このSNPのアレルが十分にごちゃまぜになっているか」どうかを検定したことの代わりとして使うことにすると、これをHardy-Weinberg平衡検定と呼びます。

ある多型が集団中に生じて、その集団がランダムメイティングしていれば、アレルのペアの取り方は「ごちゃまぜにしてペアを作る」ことに相当するから、集団がランダムメイティングしているかどうかを知るすべとして使います。 実際、集団でランダムメイティングが起きて、染色体ペアがシャッフルするには、何世代もの時間がかかるので、ランダムメイティングしていて、しかも十分に時間が経過していることに関して知るすべとして使います。

遺伝子多型のアレルのペアの具合が、集団のランダムメイティングを反映していることを「Hardy-Weinberg平衡」と言うので、この検定は、Hardy-Weinberg平衡検定と呼ばれます。

Rでやってみる

```
N_MM <- 50
N_Mm <- 38
N_mm <- 12

N <- N_MM + N_Mm + N_mm # 100

N_M <- 2 * N_MM + N_Mm # 138
N_m <- N_Mm + 2 * N_mm # 62

Frac_M <- N_M / (2 * N) # 0.69
Frac_m <- N_m / (2 * N) # 0.31

Ex_N_MM <- Frac_M^2 * N # MM期待人数
Ex_N_Mm <- 2 * Frac_M * Frac_m * N # Mm期待人数
Ex_N_mm <- Frac_m^2 * N # mm期待人数

# カイ二乗値
chi2_ori <- (N_MM - Ex_N_MM)^2/Ex_N_MM + (N_Mm - Ex_N_Mm)^2/Ex_N_Mm + (N_mm - Ex_N_mm)^2/Ex_N_m
m

print(chi2_ori)
```

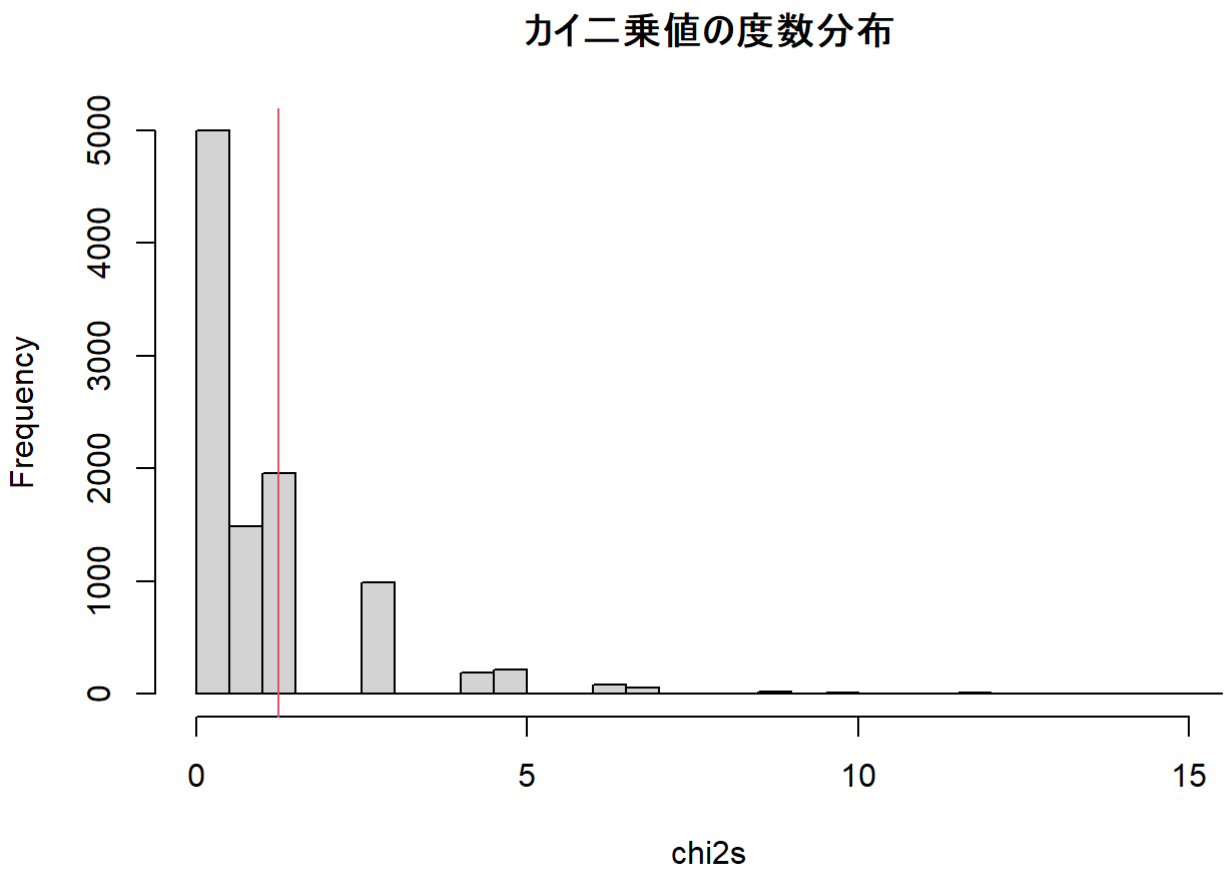
```
## [1] 1.248459
```

```
# p値
p <- pchisq(chi2_ori,df=1,lower.tail=FALSE)
print(p)
```

```
## [1] 0.263847
```

200個のアレルをごちゃまぜにしてペアを作ることを繰り返してみる。

```
n_iter <- 10000 # 繰り返し回数

# ごちゃまぜにしてペアを作って３遺伝子型の人数を求める
Mm <- c(rep(0,N_M),rep(1,N_m))
genotype_counts <- matrix(0,3,n_iter)
for(i in 1:n_iter){
  shuffled <- matrix(sample(Mm),ncol=2)
  g_type <- apply(shuffled,1,sum)
  genotype_counts[1,i] <- length(which(g_type==0))
  genotype_counts[2,i] <- length(which(g_type==1))
  genotype_counts[3,i] <- length(which(g_type==2))
}

# カイ二乗値を計算する
chi2s <- rep(NA,n_iter)
for(i in 1:n_iter){
  N_MM_ <- genotype_counts[1,i]
  N_Mm_ <- genotype_counts[2,i]
  N_mm_ <- genotype_counts[3,i]
  chi2_ <- (N_MM_ - Ex_N_MM)^2/Ex_N_MM + (N_Mm_ - Ex_N_Mm)^2/Ex_N_Mm + (N_mm_ - Ex_N_mm)^2/Ex_N
_mm
  chi2s[i] <- chi2_
}

hist(chi2s,breaks=50,main="カイ二乗値の度数分布")
abline(v=chi2_ori,col=2)
```



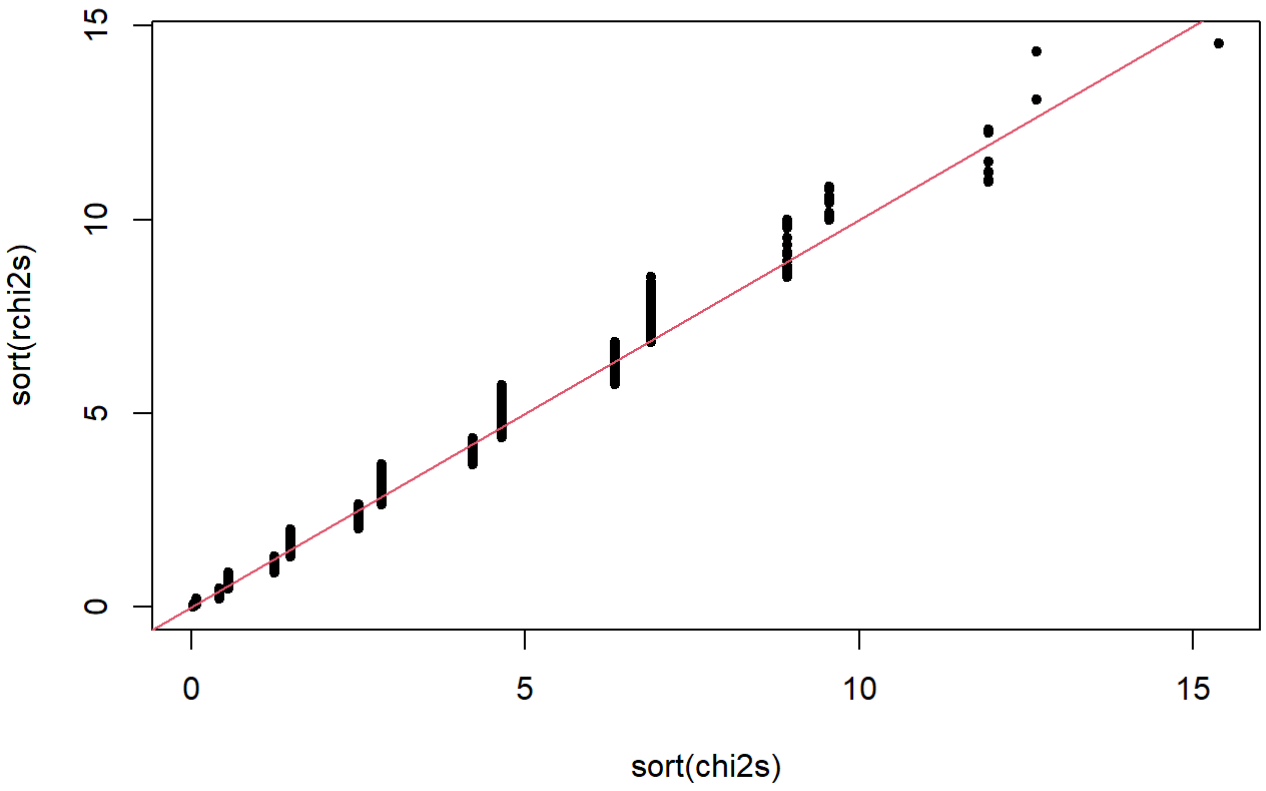
算出されたカイ二乗値と、自由度１のカイ二乗分布に従う乱数との関係は、おおまかに$y = x$の直線に乗ります。 しかしながら、総人数100人なので、発生しうる３遺伝子型人数のパターンには限りがあり、それに対応するカイ二乗値も特定の値しか取れない様子が見て取れます。

これが、おおまかに『自由度１のカイ二乗分布』に従うというときの「おおまかに」の意味です。

ぴたりと合うようにするためには、総人数を無限大にする必要があります。 総人数を増やすとどんどんカイ二乗分布に近づいていくので、漸近近似を利用した検定になっています。

このことを実感するには、サンプル数を大きくして同様のプロットをしてみると良いでしょう。

```
rchi2s <- rchisq(n_iter,df=1)
plot(sort(chi2s),sort(rchi2s),pch=20,cex=1)
abline(0,1,col=2)
```



３ジェノタイプの人数と言う観測データと、漸近的にカイ二乗分布に近づく確率事象とを結びつけて、Hardy-Weinberg平衡に関して帰無仮説検定をする例でした。

カイ二乗値に限らず、正規分布、t-分布、F-分布など、色々な分布に従う統計量を用いて、観測データから各種統計量を計算し、その統計量が従う分布に照らしてp値を算出し、検定に利用します。

また、この例では、観測データから統計量が計算され、その統計量がある確率密度分布に従うという単純な例を示しましたが、「問いかけると、そのたびに異なる値を返す」という仕組みで説明できる現象ならば、込み入った考え方をする必要はありますが、基本的な考え方は同じです。

込み入った考え方を含めて、確率事象を説明するモデル(確率モデル)を立てることができれば良いのです。

確率モデルを立てると、ある事象を観測する「確率」が計算できるようになります。

確率と尤度 DNA鑑定を例に

確率

ある男性M1とある女性Fとの間に子Cが生まれたとします。

あるSNPの遺伝子型は、M1がAA、FがAGであるとき、Cの遺伝子型がAAとなる確率は、0.5です。

別の男性M2と同じ女性Fとの間に子Cが生まれたとします。

M2の遺伝子型がAGであるとき、Cの遺伝子型がAAとなる確率は、0.25です。

FとCとは同じで、父親がM1かM2かの部分だけが違います。

従って、以下の２つのモデルを想定し、

- 「父親がM1である」というモデル
- 「父親がM2である」というモデル

子Cの遺伝子型がAAとなる『確率』が計算されました。

子Cの遺伝子型は、「生まれるたびに異なった型が観測される」ので確率事象です。 Cの遺伝子型はAA,AG,GGの３通りがあり得て、それぞれの確率が定まっています。

具体的には、父親がM1というモデルの下では、Cの遺伝子型がAA,AG,GGとなる確率は、$0.5, 0.5, 0$となります。 父親がM2というモデルの下では、$0.25, 0.5, 0.25$となります。

尤度

確率を計算するときには、モデルが与えられ、Cの遺伝子型が変わりうるので、遺伝子型に対して確率を計算しました。

似たような状況ながら、ちょっと異なる設定にしてみます。

子Cの遺伝子型はすでにAAであると判っているとします。データがすでに得られているのです。

Cの母親がFであることも判っていて、Fの遺伝子型がAGであることも判っています。

M1,M2の遺伝子型がAA,AGであることも判っていますが、Cの父親がM1なのかM2なのかが不明とします。

このとき、「もしも父親がM1だとしたときに、Fとの間に子が生まれたとすると、その子の遺伝子型がAGである確率」は0.5です。これは先に計算した通りです。

このことを「Fと誰かの子Cの遺伝子型AGであるというデータが得られているとき、父親がM1であるとのモデル(仮説)の尤度」は0.5である、とも言います。

従って、父親がM2であるとのモデル(仮説)の尤度」は0.25となります。

これだけの情報で、父親をM1にするかM2にするかを決めるのは勇気の要ることですが、それでも、「父親がM1であるモデル(仮説)の尤度はM2であるモデル(仮説)の尤度の２倍」である、とも言えます。

「M1を父親と思うか、M2を父親と思うか、と言われたら、M1を２倍強く、父親と思うのがよさそう」と言い換えることも出来ます。

このように尤度は、データが与えられたときにモデル(仮説)を信じる強さを示し、２つのモデル(仮説)を比較するときには、尤度の比(尤度比)を計算して、どちらを信じるかについての判断材料とすることができます。

この例では、尤度比が２と小さめだったので、決めかねましたが、たくさんのSNPを使うなどすれば、尤度比が1000倍になるかもしれません。 そうなれば、かなり自信を持って、父親を選ぶことができます。

DNA鑑定は、この尤度を基本とした計算に基づきます。 父親鑑定や犯罪捜査における現場試料DNAとの照合の鑑定などがDNA鑑定に含まれます。

また、特定の遺伝性疾患の原因遺伝子変異を同定する解析なども、この尤度の計算に基づいています。

家系図が複雑になると、家系図上の各個人の遺伝子型の場合分けが非常に多岐にわたり、その計算も一筋縄ではいかなくなるため、計算を可能にするアルゴリズムを用いるなどの工夫が必要になりますが、その部分を除けば、考え方の基本は尤度の計算に帰着します。

DNA鑑定・原因遺伝子同定のための連鎖解析と言った、遺伝子型特有の解析には尤度が用いられてきました。その影響もあり、ゲノム・オミックスの解析では尤度・尤度比の考え方を使うものが多いです。

尤度はベイズ推定にも用いますので、ゲノム・オミックス解析にはベイズ流のアプローチが織り込まれることが多いのも、こうした経緯と無関係ではありません。

最尤推定

尤度比から、最大の尤度へ

２人の父親候補が居る場合には、それぞれを父親と仮定し、それぞれの仮定に対して尤度を計算し、尤度比の大きさを考えることが適切であると説明しました。

父親候補が100万人居たとしたら、100万通りの仮定に対して尤度を計算することが出来ます。 一番父親らしいのは、尤度が最大の候補者になります。 もしかすると、尤度が最大の候補者の尤度と尤度が2番目に大きい候補者の尤度の比はそれほど大きくないかもしれませんので、父親を選ぶ決定打にはならないかもしれません。 決定打ではないけれども、「尤度を最大にする候補者」には何らかの意味があるでしょう。

父親を決めるためのDNA鑑定では、尤度比が大きくならなければ決断しにくいですが、DNA鑑定とは異なる状況では、「尤度を最大にする」仮説を探すことに意味があることもあります。

尤度を最大にする仮説(モデル)を推定することを最尤推定と言います。

例　2SNPのハプロタイプ推定

同じ常染色体上に２つのSNPがあり、相互に比較的近いところに位置しているとします。

１つ目のSNPの２つのアレルをA/a、もう一つのSNPのそれをB/bとします。

それぞれのSNPの遺伝子型はAA,Aa,aaの３種類とBB,Bb,bbの３種類あります。

ある個人について２つのSNPの遺伝子型をそれぞれ観察すると(AA,BB),(AA,Bb),(AA,bb),(Aa,BB),(Aa,Bb),(Aa,bb),(aa,BB),(aa,Bb),(aa,bb)の９通りの「組合せ遺伝子型」のいずれかになります。

この９通りの組合せ遺伝子型の人数が
$\mathbf{N} = (n_{AABB}, n_{AABb}, n_{AAbb}, n_{AaBB}, n_{AaBb}, n_{Aabb}, n_{aaBB}, n_{aaBb}, n_{aabb})$とします。

同一の染色体上にSNP A/aのアレルとSNP B/bのアレルが並んで乗るパターンはA---B,A---b,a---B,a---bの４通りあります。

この４種類のアレルの並び方をハプロタイプと呼びます。 ４種類のハプロタイプの頻度が判らないので、それを推定する殊にします。

観察データをサンプリングした母集団の４ハプロタイプの頻度を$\mathbf{H} = (h_{AB}, h_{Ab}, h_{aB}, h_{ab})$とすると、

$$h_{AB} + h_{Ab} + h_{aB} + h_{ab} = 1$$
$$h_{AB}, h_{Ab}, h_{aB}, h_{ab} \geq 0$$

が成り立っています。

今、Hardy-Weinberg平衡を仮定すると、９種類の組合せ遺伝子型の母集団頻度$\mathbf{G} = (g_{AABB}, \ldots, g_{aabb})$は$\mathbf{H}$から計算できます。

$$
\begin{aligned}
g_{AABB} &= h_{AB}^2 \\
g_{AABb} &= 2h_{AB}h_{Ab} \\
g_{AAbb} &= h_{Ab}^2 \\
g_{AaBB} &= 2h_{AB}h_{aB} \\
g_{AaBb} &= 2h_{AB}h_{ab} + 2h_{Ab}h_{aB} \\
g_{AaBB} &= 2h_{AB}h_{aB} \\
g_{aaBB} &= h_{aB}^2 \\
g_{aaBb} &= 2h_{aB}h_{ab} \\
g_{aabb} &= h_{ab}^2
\end{aligned}
$$

ここで、注意が必要なのはg_{AaBb}の計算方法です。２つのSNPがいずれもヘテロ接合体となる場合には、A---BというハプロタイプとA---bというハプロタイプがペアになる場合とA---bというハプロタイプとa---Bというハプロタイプとがペアになる場合との２通りのハプロタイプペアを考慮する必要があるので、上記の式になります。

この$\mathbf{G}$が決まれば、$\mathbf{N}$を観測する確率の計算が可能になるので、どんな$\mathbf{G}$が尤度を最大にするかも判断出来ます。 $\mathbf{G}$は$\mathbf{H}$によって決まりますから、$h_{AB}, h_{Ab}, h_{aB}, h_{ab}$の値が、$\mathbf{N}$という観測の下で最大の尤度をもたらすかも判ります。

このようにして求めた$\mathbf{H} = (h_{AB}, h_{Ab}, h_{aB}, h_{ab})$の値を、「$\mathbf{N}$の下、Hardy-Weinberg平衡を仮定したときのハプロタイプ頻度の最尤推定値」と呼びます。

尤度関数・対数尤度関数

尤度の計算は、確率 g_{****} で起きる事象がn_{****}回起きたことを、すべての組合せ遺伝子型について掛け合わせるので、尤度関数は以下のようになります。

$$
L(\mathbf{H}|\mathbf{N}) = \frac{(\sum n_{****})!}{\prod n_{****}!} g_{****}^{n_{****}}
$$

$$
**** \in \{AABB, AABb, AAbb, AaBB, AaBb, Aabb, aaBB, aaBb, aabb\}
$$

尤度関数$L(\mathbf{H}|\mathbf{N})$の組合せの場合の数の項$C = \frac{(\sum n_{****})!}{\prod n_{****}!}$は、推定したい$\mathbf{H}$と無関係なので、この項の計算は実際の推定過程では省略されることが多いです。

尤度関数$L(\mathbf{H}|\mathbf{N}) = C \times g_{****}^{n_{****}}$ は、確率の掛け算の形になることが多いので、対数を取るとさらに簡単になります。

$$
\log L(\mathbf{H}|\mathbf{N}) = \log C + \sum n_{****} \times \log g_{****}
$$

最適化

尤度関数・対数尤度関数が決まれば、その関数の値を最大にする変数($\mathbf{H}$)を探せば良いことになります。

尤度関数に限らず、関数の最大値・最小値を与える変数のセットを探索する課題は「最適化問題」と呼ばれ、数理工学の基本的な課題の１つです。

様々なアルゴリズムが知られており、最適化する関数の特徴、変数の制約などに応じて使い分けられます。

データを用いて判断するとは

ポイント(データを用いて判断するとは)

- 検定、推定、学習
- 尤度比検定
- 最尤推定

- 最小二乗法
- コスト関数の最適化
- 過剰適合
- バイアス-バリアンス トレードオフ
- ペナルティ項

検定と推定と学習

データを集めて、メッセージを読み取る作業は大きく分けて、

- 検定
- 推定
- 学習

に分けることが出来ます。

検定は単純なモデルと複雑なモデルとの比較します。比較にあたって、観測されたデータが、単純なモデルを仮定すると起きにくいとき、単純なモデルを信じることを取りやめて(棄却して)、より複雑なモデル(対立仮説)を信じてもよいのかもしれない、という考え方です。

推定は、あるモデルを過程したときに、そのモデルを構成するパラメタの値がいくつだと、観測データがうまく説明できるか、という考え方で行います。 観測データの説明のうまさを尤度関数で測るとき、パラメタの値を最尤推定するかもしれません。 線形回帰直線を推定するときは、$y = ax + b$という直線を定める係数a, bの値を推定します。最小二乗法では二乗誤差という関数を最小にするように推定します。

学習は教師あり学習と教師なし学習とに分けられますが、教師あり学習の場合には、教師データをうまく説明するモデルを見つけるのが仕事になります。モデルを構成するパラメタの値をいくつにすればよいかという問題設定で考えれば、パラメタ値の推定問題と同じことになります。

教師なし学習は、標本の集まりにどのようなパターンがあるかをかいつまんで説明するのが課題です。ごく単純な例では、標本の平均値を算出して、標本の代表値としてかいつまむ記述統計も一種の教師なし学習的な作業です。標本が 2 群に分かれていて、それぞれの平均値によって 2 群を代表させれば、クラスタリングという教師なし学習とその学習に認められた 2 群に代表値を添えることになりますし、クラスタリングによって、 1 群だと判断して、全標本の平均値を代表値とすれば、単なる記述統計に戻ります。

本章では、以下、検定・推定・学習の相互関係を見ることによって、検定・推定・学習が「データを用いて判断する」という共通の仕事であることを確認し、用途に応じて使い分けられるアプローチの違いであることに慣れることにします。

ゲノム・オミックスのデータ解析では、ときに検定し、ときに推定し、ときに学習しますが、データからどのようなタイプのメッセージを引き出したいかによって、どのアプローチを取るのかが選ばれます。 慣れないうちは「同じようなゲノム・オミックス実験データを使っているのに、どうしてこの論文では検定をしていて、別の論文では機械学習をしているのか」と、戸惑うこともありますが、著者・解析者がデータから何を読み取りたいのかというスタンスの違いだと捉えれば、そのような戸惑いは解消すると思います。

尤度関数で検定

次世代シークエンサーを使った遺伝子発現解析とRPKM

RPKM

ある組織の健常細胞と癌細胞との遺伝子発現パターンの違いを解析しているとします。 全遺伝子の発現量を対象にするのでトランスクリプトーム解析と言います。

次世代シークエンサー(NGS)を用いた実験では、細胞から抽出したmRNA分子を鋳型として、短いDNA分子(ショートリード)を大量に生成します。

生成されたショートリードの塩基配列をシークエンサーで読み、ATGCの文字列として出力します。

この文字列はmRNA分子を鋳型として作られたので、ゲノム上のどこかにうまくマッピングできます。場合によっては、ショートリードが異なるエクソンにまたがっていることもありますが、それも、ゲノム配列とよく比較すれば、分離してゲノム上にマッピングすることができます。

ゲノム配列にマッピングする代わりに、データベース化されたmRNA配列の中にうまくマッピングすることもあります。

ショートリードによっては、実験の質が悪く、マッピング出来ないこともあります。

このようにして、ある細胞C_1では、ある遺伝子g_1とg_2の配列上に、それぞれ$n_1^{(1)}$本,$n_2^{(1)}$本のショートリードがマッピングされたとします。

また、別の細胞C_2では、g_1, g_2に$n_1^{(2)}, n_2^{(2)}$本のショートリードがマッピングされたとします。

マッピングされたショートリードの本数は抽出したmRNA分子量を反映しているので、$n_1^{(1)}, n_2^{(1)}, n_1^{(2)}, n_2^{(2)}$を比較したいところですが、補正をする必要があります。

まず、細胞C_1の２つの遺伝子のショートリード数$n_1^{(1)}, n_2^{(1)}$ですが、遺伝子が長ければ長いほど、鋳型として使われる頻度が上がり、マッピングされるショートリード数も増えるので、遺伝子の長さで補正した上で、g_1, g_2のどちらの発現量が多いのかを解釈することが適切です。

遺伝子(mRNA)の長さをL_1, L_2としたなら、 $\frac{n_1^{(1)}}{L_1}, \frac{n_2^{(1)}}{L_2}$を比較するということです。

この事情はC_2でも同様なので、$\frac{n_1^{(2)}}{L_1}, \frac{n_2^{(2)}}{L_2}$と補正します。

この補正により、同一サンプルの異なる遺伝子の発現量の比較は可能になりました。

では、異なるサンプル・異なる細胞でg_1の発現量を比較するのに、$\frac{n_1^{(1)}}{L_1}, \frac{n_1^{(2)}}{L_1}$で比較してよいかと言うと、さらに補正するのが良いです。

なぜなら、NGS実験のために大量のショートリードを生成するとき、そのショートリードの質と量は、実験ごとに大きくばらつくからです。 ですので、実験ごとのショートリード数のばらつきを補正した上で比較するのが良いことになります。

この補正をする際、各サンプルで、マッピングされたショートリードの総本数$M^{(1)}, M^{(2)}$で補正する方法が提唱されています。

$\frac{n_1^{(1)}}{L_1 \times M^{(1)}}, \frac{n_1^{(2)}}{L_1 \times M^{(2)}}$によって、異なるサンプル・異なるサンプルのある遺伝子発現量を比較するというわけです。

実施には、遺伝子の長さとしては、1000塩基対を１単位とし、ショートリードの総本数としては100万本を１単位として補正する方法が使われることが多いです。

補正後の値が扱いやすい値になるように桁の調整がなされた値で、RPKM (Reads Per Kilobase of exon per Million mapped reads)と呼ばれます。

$\frac{n_1^{(1)}}{L_1/1000 \times M^{(1)}/1000000}$のように計算するというわけです。

２群間の遺伝子発現量のt検定と尤度比検定

T-検定

健常部の細胞 N_1 個、非健常部の細胞N_2 個のNGS-トランスクリプトーム実験を行い、ある遺伝子の発現量がRPKM単位で$\mathbf{X}^{(1)} = (x_1^{(1)}, \ldots, x_{N_1}^{(1)}), \mathbf{X}^{(2)} = (x_1^{(2)}, \ldots, x_{N_2}^{(2)})$

と得られたとします。

２群間の平均値の差の検定としてt-検定を実施するのに適切な値が得られていたとすると、t-統計量を算出し、しかるべくp-値に変換することで、２群間の平均値に差がないという帰無仮説の棄却検定を行うことができます。

ここでは、帰無仮説が真な場合にt-統計量がT分布に従うことを利用しています。

尤度比検定

他方、尤度の考え方を使って(見方を変えて)検定することもできます。

２つの仮説を以下のように設定します。

- ２群の値は同一の正規分布からの乱数とします。その平均をm_0、標準偏差をs_0とします。
- ２群の値は、異なる２つの正規分布からの乱数とします。２群の平均はm_1, m_2、標準偏差はs_1, s_2とします。

この帰無仮説と対立仮説の比較においては、「２群の平均が異なるか、標準偏差が異なるか、平均と標準偏差の両方が異なるか、そのいずれの異なり方かは問わず」「同一の正規分布から得られたという仮説では説明できない」ならば、帰無仮説を棄却します。

単に、２群の平均が同じか違うかの比較ではありません。

さて。 それぞれの仮説につき、$m_0, s_0, m_1, m_2, s_1, s_2$を最尤推定し、 $\hat{m}_0, \hat{s}_0, \hat{m}_1, \hat{m}_2, \hat{s}_1, \hat{s}_2$ になったとします。

このとき、２つの仮説の尤度を次のように書くとします。

$$L_1 = L_1(\hat{m}_0, \hat{s}_0 | \mathbf{X}_1, \mathbf{X}_2)$$
$$L_2 = L_{2,1}(\hat{m}_1, \hat{s}_1 | \mathbf{X}_1) \times L_{2,2}(\hat{m}_2, \hat{s}_2 | \mathbf{X}_2)$$

L_1, L_2の比(尤度比)を取り、その対数を取って２倍してみよう。

$$K = 2 \log \frac{L_2}{L_1}$$

この値Kを自由度2のカイ二乗分布に照らして、p値に換算すると、２群の平均が同じであるとの帰無仮説に関する帰無仮説検定になります。

尤度の比を使った検定なので、尤度比検定と言います。

自由度を2としたのは、第一の仮説では、パラメタの数が２個で、第二の仮説では、パラメタの数が４個であり、二つの仮説のパラメタの数の差が2だからです。

Rでやってみる

t-検定。

```
# 細胞数
N1 <- 50
N2 <- 60
# ２群の平均
m1 <- 20
m2 <- 20.5
# ２群の標準偏差
s1 <- 3
s2 <- 3.3

# データ生成
X1 <- rnorm(N1,m1,s1)
X2 <- rnorm(N2,m2,s2)

X <- c(X1,X2)
Cell <- c(rep(1,N1),rep(2,N2))

df <- data.frame(Cell,X)

# t-検定
t.test(X ~ Cell, var.equal = FALSE,data=df)
```

```
##
##  Welch Two Sample t-test
##
## data:  X by Cell
## t = -0.75524, df = 105.82, p-value = 0.4518
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.6468148  0.7382707
## sample estimates:
## mean in group 1 mean in group 2
##        19.93126        20.38553
```

尤度比検定。 最尤推定の部分をちょっと端折って、正規分布の平均の最尤推定値は、標本平均で、標準偏差の最尤推定値も、標本標準偏差で代用して、尤度比に基づく統計量Kを出すことにします。

```
est_m_0 <- mean(X)
est_s_0 <- sd(X)

est_m_1 <- mean(X1)
est_s_1 <- sd(X1)
est_m_2 <- mean(X2)
est_s_2 <- sd(X2)

L1 <- dnorm(X,est_m_0,est_s_0)
LL1 <- sum(log(L1))

L2_1 <- dnorm(X1,est_m_1,est_s_1)
L2_2 <- dnorm(X2,est_m_2,est_s_2)
LL2 <- sum(log(L2_1)) + sum(log(L2_2))

K <- 2 * (LL2 - LL1)
print(K)
```

```
## [1] 0.6587055
```

```
p_LRtest <- pchisq(K,df=2,lower.tail=FALSE)
p_LRtest
```

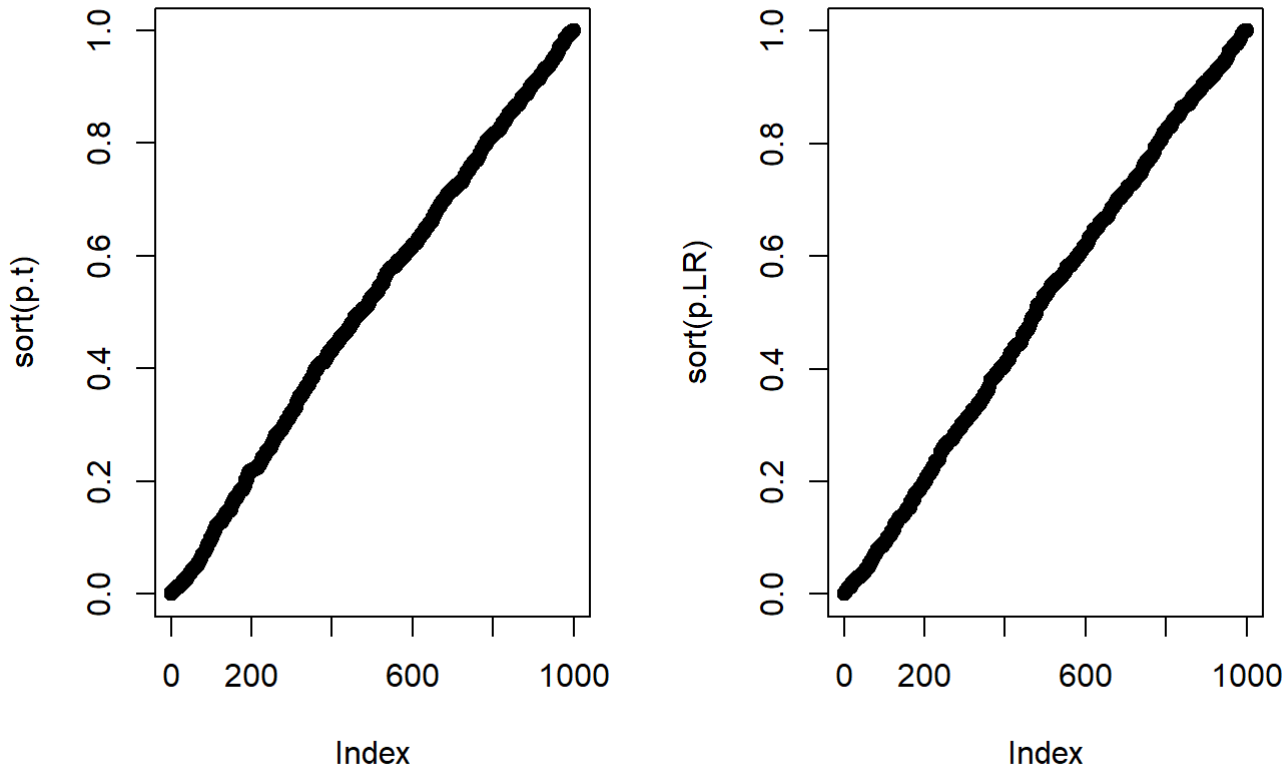
```
## [1] 0.7193892
```

t-検定と尤度比検定では「逸脱」の方向性が異なりますが、どちらも「帰無仮説検定」としての性質である、「帰無仮説が成り立っているときのp値の分布が一様分布になる」という性質は共通です。

その様子を見てみます。

一様乱数を小さい順に並べてプロットすると、0から1へと直線状に並びますが、その様子が見て取れます。

```
# 細胞数
N1 <- 50
N2 <- 60
# 帰無仮説では、２群の平均も分散も同じ
# ２群の平均
m1 <- 20
m2 <- 20
# ２群の標準偏差
s1 <- 3
s2 <- 3

n.iter <- 1000
p.t <- p.LR <- ts <- Ks <- rep(NA,n.iter)

for(i in 1:n.iter){
  X1 <- rnorm(N1,m1,s1)
  X2 <- rnorm(N2,m2,s2)

  X <- c(X1,X2)
  Cell <- c(rep(1,N1),rep(2,N2))

  df <- data.frame(Cell,X)

  # t-検定
  t.out <- t.test(X ~ Cell, var.equal = FALSE,data=df)
  p.t[i] <- t.out$p.value
  ts[i] <- t.out$statistic
  # LR-test
  est_m_0 <- mean(X)
  est_s_0 <- sd(X)

  est_m_1 <- mean(X1)
  est_s_1 <- sd(X1)
  est_m_2 <- mean(X2)
  est_s_2 <- sd(X2)

  L1 <- dnorm(X,est_m_0,est_s_0)
  LL1 <- sum(log(L1))

  L2_1 <- dnorm(X1,est_m_1,est_s_1)
  L2_2 <- dnorm(X2,est_m_2,est_s_2)
  LL2 <- sum(log(L2_1)) + sum(log(L2_2))

  K <- 2 * (LL2 - LL1)

  p_LRtest <- pchisq(K,df=2,lower.tail=FALSE)
  p.LR[i] <- p_LRtest
  Ks[i] <- K
}

par(mfcol=c(1,2))
plot(sort(p.t))
plot(sort(p.LR))
```




```
par(mfcol=c(1,1))
```

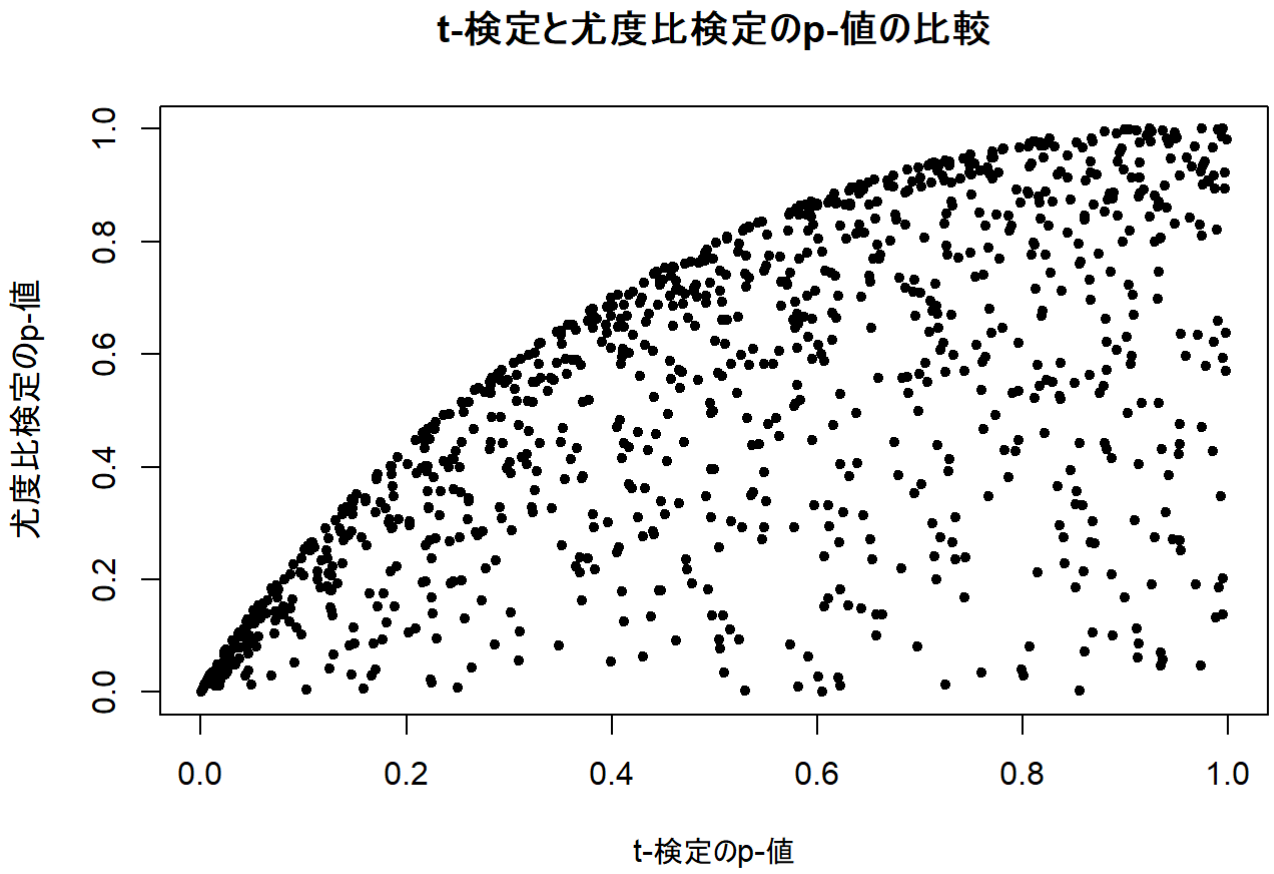
２つの検定の違い

t-検定も尤度比検定も、帰無仮説が真であるときに、p値が一様分布になりました。

では、ある観測データについて、t-検定と尤度比検定とを行ったら、そのp-値は一致するのでしょうか？

比較してみます。

```
plot(p.t,p.LR,pch=20,main="t-検定と尤度比検定のp-値の比較",xlab="t-検定のp-値",ylab="尤度比検定のp-値")
```



２つの検定が同じならば、２種のp-値の散布図のは直線$y = x$となるはずですが、そうはなっていません。

だからと言って、２種のp-値に全く関係がないのかと言えば、そうでもありません。

ある２群の観測データが得られたとき、２群の違いについて、「平均値の差」に着目したのか、「同一の正規分布から得られたのか、異なる２つの正規分布から得られたのか」を比較したのかという、比較の違いがp-値の違いになって現れています。 この２つは比較の仕方は違いますが、両者は完全に独立なわけではなく、ある程度の関連があることも、散布図から読み取れます。

SNPの分割表検定の場合

常染色体上のSNPの３種類の遺伝子型MM,Mm,mmの割合について、疾患群と非疾患群とで同じか違うかを検定することがあります。

ゲノムワイド関連研究(GWAS)では、数十万個のSNPについて、このような検定をして、SNPの数だけのp-値を算出します。 そのような多数のp-値は、多重検定補正をして評価される必要があるという話は、冒頭の例に出てきました。

遺伝形式

今、疾患群の３遺伝子型の人数をA_{MM}, A_{Mm}, A_{mm}、非疾患群のそれをB_{MM}, B_{Mm}, B_{mm}とします。

この2×3分割表の検定をするに当たって、遺伝形式を考慮します。

メンデル型遺伝病と呼ばれる遺伝病の場合には、病気の原因となる、アレル(例えばM)を１本以上持つと、発病する場合と、アレルを2本持って初めて発病する場合とがあります。

優性遺伝形式、劣性遺伝形式と呼びます。

優性遺伝形式の場合には、MMとMmとはどちらも同じく病気を引き起こしますので、検定するときには、$A_{MM} + A_{Mm}, A_{mm}$と$B_{MM} + B_{Mm}, B_{mm}$を比較します。

2×2分割表に直して、独立性の検定を行うのです。

他方、劣性遺伝形式の場合には、Mmとmmとはどちらも同じく、病気を引き起こさないので、検定するときには、$A_{MM}, A_{Mm}+A_{mm}$と$B_{MM}, B_{Mm}+B_{mm}$を比較します。 やはり、2×2分割表に直して、独立性の検定を行うのです。

メンデル遺伝病の場合は、このように2×3分割表から2×2分割表を作成して独立性の検定を行いますが、GWASの対象となる複合遺伝性疾患の場合には、SNPの影響は、優性・劣性のいずれの遺伝形式にも合致しません。 発病のリスクを高めるアレル(例えばM)を持つ本数が多いと、発病するリスクも高くなる、というような影響の仕方をします。

Mを持つ本数が0,1,2本の３通りがあり得ますが、３通りの場合で、どのようなリスクの増加具合になるのかは、調べる前にはわかりません(GWASを行っても、はっきりしたことはわからないままです)。

したがって、Mの本数に「線形に応じて」３遺伝子型の人数比が変わると「仮定」して検定を行うことが通例です。 優性・劣性遺伝形式ではなく「相加」形式を想定して検定する言います。

Rでやってみる

優性・劣性遺伝形式は次のようになります。

```
A <- c(49,42,9)
B <- c(36,48,16)

# 優性モデル
A.dom <- c(A[1]+A[2],A[3])
B.dom <- c(B[1]+B[2],B[3])
dom.table <- rbind(A.dom,B.dom)
print(dom.table)
```

```
##       [,1] [,2]
## A.dom   91    9
## B.dom   84   16
```

```
chisq.test(dom.table,correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dom.table
## X-squared = 2.24, df = 1, p-value = 0.1345
```

```
# 劣性モデル
A.rec <- c(A[1],A[2]+A[3])
B.rec <- c(B[1],B[2]+B[3])
rec.table <- rbind(A.rec,B.rec)
print(rec.table)
```

```
##       [,1] [,2]
## A.rec   49   51
## B.rec   36   64
```

```
chisq.test(rec.table,correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  rec.table
## X-squared = 3.4578, df = 1, p-value = 0.06295
```

相加モデルは次のように検定します。

```
# 相加モデル
S <- A + B # ２群の和

prop.trend.test(A,S,score=c(0,1,2))
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  A out of S ,
##  using scores: 0 1 2
## X-squared = 4.3478, df = 1, p-value = 0.03706
```

相加モデルでは、３遺伝子型に引数scoreとして(0,1,2)を与えています。

優性・劣性遺伝形式では、この引数scoreを(1,1,0),(1,0,0)としても検定できます。

2×2分割表を作成してカイ二乗検定をした結果と同じになっていることを確認しましょう。

```
# 優性
prop.trend.test(A,S,score=c(1,1,0))
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  A out of S ,
##  using scores: 1 1 0
## X-squared = 2.24, df = 1, p-value = 0.1345
```

```
#劣性
prop.trend.test(A,S,score=c(1,0,0))
```

```
##
##  Chi-squared Test for Trend in Proportions
##
## data:  A out of S ,
##  using scores: 1 0 0
## X-squared = 3.4578, df = 1, p-value = 0.06295
```

カイ二乗検定と尤度比検定

優性遺伝形式の検定(2×2表の独立性の検定)を、カイ二乗検定と尤度比検定との２つの方法で実施してみます。

尤度比検定では、以下の２つの仮説について、比率の最尤推定値を求め、それに基づいて、尤度を計算してその比を取ります。

- 帰無仮説：$(A_{MM}+A_{Mm},A_{mm})$の比率と$(B_{MM}+B_{Mm},B_{mm})$の比率は同一

- 対立仮説：$(A_{MM}+A_{Mm}, A_{mm})$の比率と$(B_{MM}+B_{Mm}, B_{mm})$の比率は異なる

帰無仮説のリスク群の割合の最尤推定値は$\frac{A_{MM}+A_{Mm}+B_{MM}+B_{Mm}}{A_{MM}+A_{Mm}+A_{mm}+B_{MM}+B_{Mm}+B_{mm}}$です。

対立仮説のそれはA,Bで異なり、$\frac{A_{MM}+A_{Mm}}{A_{MM}+A_{Mm}+A_{mm}}$、$\frac{B_{MM}+B_{Mm}}{B_{MM}+B_{Mm}+B_{mm}}$です。

やってみます。

```
f.null <- (A[1]+A[2]+B[1]+B[2])/(sum(A)+sum(B))
f.A <- (A[1]+A[2])/sum(A)
f.B <- (B[1]+B[2])/sum(B)

# 組み合わせ項はキャンセルされるので計算に入れないで対数尤度を計算する
logL.null <- (A[1]+A[2]+B[1]+B[2])*log(f.null) + (A[3]+B[3])*log(1-f.null)

logL.alt <- (A[1]+A[2])*log(f.A) + A[3]*log(1-f.A) + (B[1]+B[2])*log(f.B) + B[3] * log(1-f.B)

chi_LR <- 2 * (logL.alt - logL.null)

print("statistics of LR")
```

```
## [1] "statistics of LR"
```

```
print(chi_LR)
```

```
## [1] 2.266524
```

```
p_LR <- pchisq(chi_LR,df=1,lower.tail=FALSE)
print("p-value")
```

```
## [1] "p-value"
```

```
print(p_LR)
```

```
## [1] 0.1321961
```

いわゆるカイ二乗検定の結果は以下の通りです。

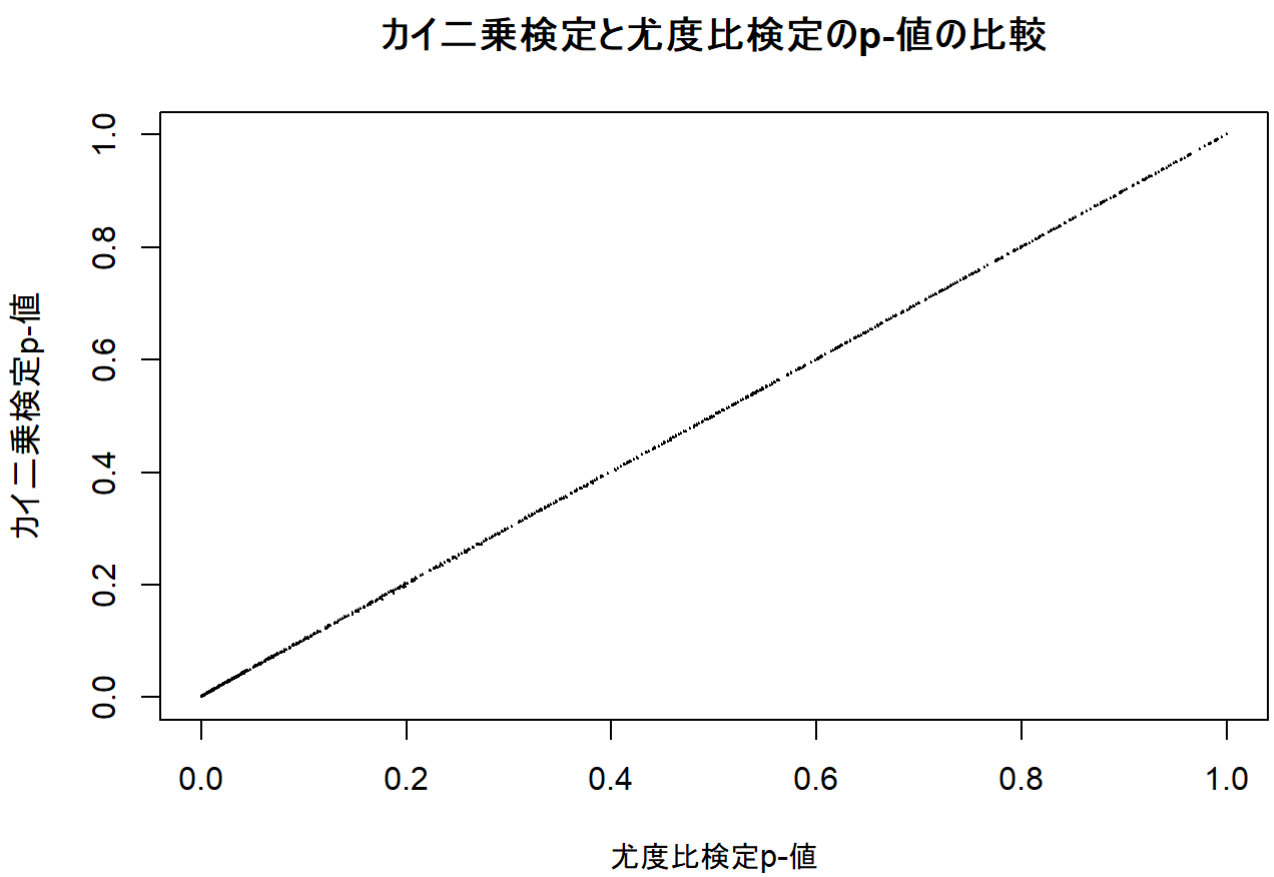
```
chisq.test(dom.table,correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dom.table
## X-squared = 2.24, df = 1, p-value = 0.1345
```

結果はごく近いです。

たくさんの2×2分割表を作成して２法の結果を比較してみます。

```
n.iter <- 1000
p.chisq <- p.LR <- rep(NA,n.iter)
for(i in 1:n.iter){
  tab <- matrix(sample(10:30,4,replace=TRUE),2,2)
  chisq.out <- chisq.test(tab,correct=FALSE)
  p.chisq[i] <- chisq.out$p.value

  f.null <- (tab[1,1]+tab[2,1])/sum(tab)
  f.A <- tab[1,1]/(tab[1,1]+tab[1,2])
  f.B <- tab[2,1]/(tab[2,1]+tab[2,2])

  logL.null <- (tab[1,1]+tab[2,1]) * log(f.null) + (tab[1,2]+tab[2,2]) * log(1-f.null)
  logL.alt <- tab[1,1]*log(f.A) + tab[1,2] * log(1-f.A) + tab[2,1] * log(f.B) + tab[2,2] * log(
1-f.B)

  chi_LR <- 2 * (logL.alt - logL.null)

  p.LR[i] <- pchisq(chi_LR,df=1,lower.tail=FALSE)

}
plot(p.LR,p.chisq,pch=20,cex=0.1,main="カイ二乗検定と尤度比検定のp-値の比較",xlab="尤度比検定p-
値",ylab="カイ二乗検定p-値")
```



２法のp-値はほぼ同じです。２ × 2分割表検定をカイ二乗検定で行う場合と、尤度比検定で行う場合とでは、「帰無仮説からの逸脱の測り方」が同じなので、両者は実質的に同じ検定になっているからです。

t-検定による平均値の差の検定と、同一正規分布からの標本か、異なる２つの正規分布からの標本かを比較した尤度比検定の場合では、「逸脱の測り方」が異なっていたので、結果が違っていたことを確認しておきましょう。

このように、帰無仮説検定と尤度関数・最尤推定とは繋がっています。

最小二乗法と最尤推定

線形回帰

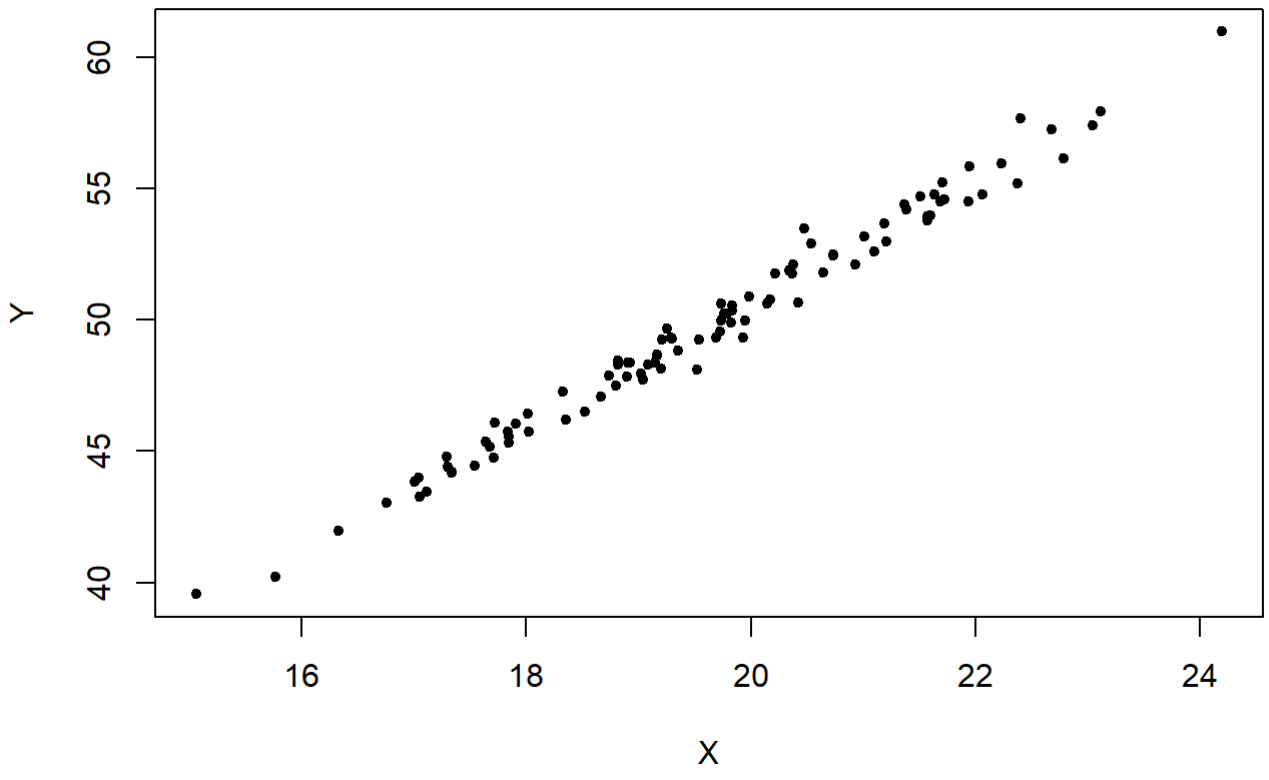
引き続き、トランスクリプトーム解析を例に取ることにします。

ある組織から多数の細胞を採取し、シングルセル・トランスクリプトーム解析をしています。

２つの遺伝子g_X, g_Yの発現量をRPKM単位で観測したところ、強い正の相関が認められたので、回帰直線を引くことにしたとします。

g_X, g_Yの発現量が$\mathbf{X} = (x_1, \ldots, x_N), \mathbf{Y} = (y_1, \ldots, y_N)$とします。

```
N <- 100
X <- rnorm(N,20,2)
a <- 2.3
b <- 4.5
Y <- a * X + b + rnorm(N,0,0.5)
plot(X,Y,pch=20)
```

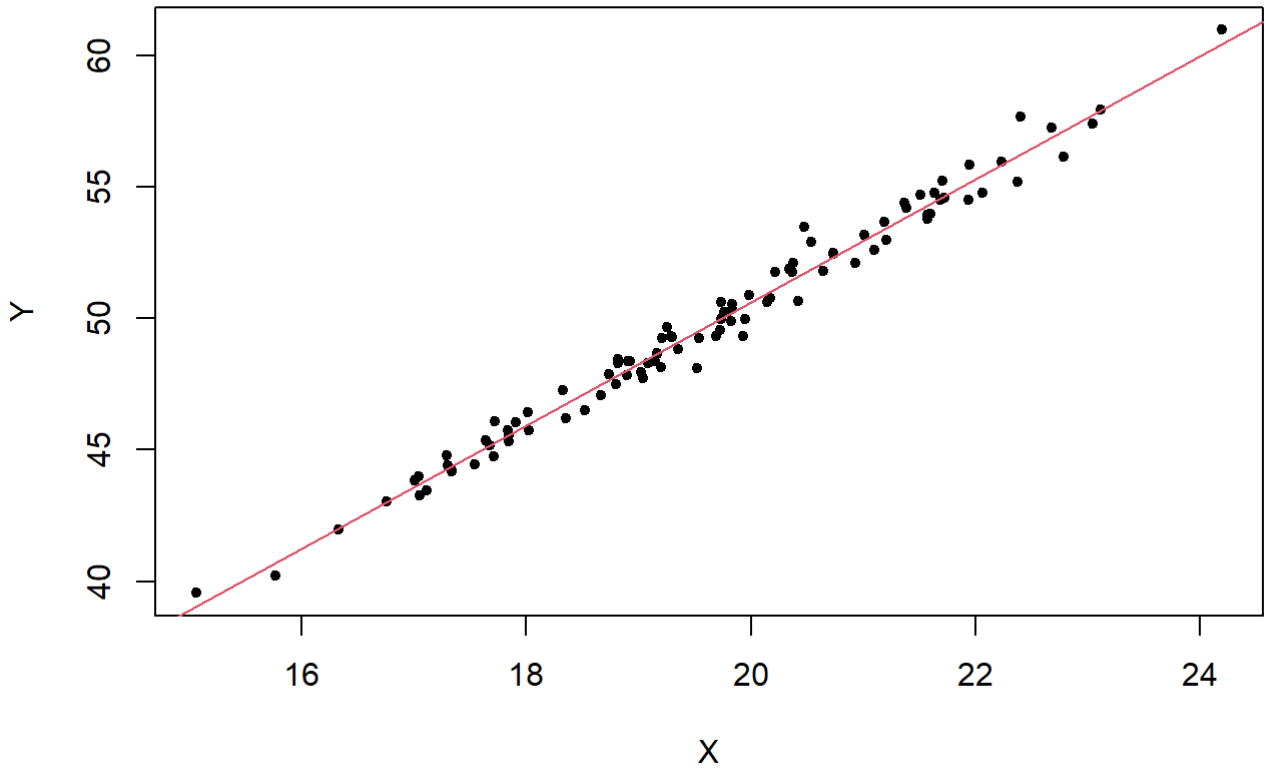


線形回帰直線の推定をアプリケーションを使って行うことは、ごく簡単に出来ます。

```
# 線形回帰
lm.out <- lm(Y~X)
lm.out$coefficients
```

```
## (Intercept)           X
##    3.825387    2.339421
```

```
plot(X,Y,pch=20)
abline(lm.out$coefficients[1],lm.out$coefficients[2],col=2)
```



最小二乗法

この回帰直線の傾きaと切片bは、次の式で表される関数$f(a,b|\mathbf{X},\mathbf{Y})$が最小になるような値となっている。

$$f(a,b|\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\hat{y}_i = ax_i + b$$

関数を最小化する係数の探索問題です。

$\hat{y}_i - y_i$は、推定回帰モデルでの推定値と観測値との差です。従って、「推定値と観測値の差」の二乗(を足し合わせた値)の最小化、という意味で「最小二乗法」と呼ばれます。

線形回帰のありがたさは、この最小化探索問題が線形代数を使って、スッキリと解けることです。

多くの最小化・最大化問題にはアルゴリズムが必要であるのに比べて、とても便利です。

正規乱数の誤差

二つの遺伝子の発現量は、$Y = aX + b$という関係にあるものの、実際には、Yの値は$aX + b$の値からばらついてしまうというモデル(仮説)に立ってみることにします。

そのばらつき(誤差)が平均0の正規分布に従うものとします。

実際、上の描図では、そのようにしてYの値をXとa, bとから生成しました。

このモデルでは、回帰直線から垂直方向へと大きくずれるような観察は起きにくく、そのずれが小さい観察は起きやすくなります。

この正規乱数の分散をσ^2とすれば、ある観察データ$\mathbf{X}, \mathbf{Y}$が与えられたときに、a, bに対して尤度が計算できます。

正規分布の確率密度を掛け合わせた関数になっています。

$$L(a, b|\mathbf{X}, \mathbf{Y}, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\hat{y}_i - y_i)^2}{2\sigma^2}}$$
$$\hat{y}_i = ax_i + b$$

対数尤度関数に書き換えると、

$$\begin{aligned}\log L(a, b|\mathbf{X}, \mathbf{Y}, \sigma^2) &= \sum_{i=1}^{N} (\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(\hat{y}_i - y_i)^2}{2\sigma^2}) \\ &= -N \log \sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2\end{aligned}$$

正規乱数の分散σ^2がどんな値であろうとも、特定の値であるならば、この対数尤度関数が最大になるときとは、右辺の第二項に現れる次の関数が最小になるときであると判ります。

$$g(\mathbf{X}, \mathbf{Y}, a, b) = \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$
$$\hat{y}_i = ax_i + b$$

これは、最小二乗法で最小化した関数に他なりません。

従って、最小二乗法による線形回帰のモデル直線の推定は、$\mathbf{Y}$が$\mathbf{X}$の線形式で表される値に正規乱数誤差が加わったモデルの下での最尤推定になっていることがわかります。

最適化で学習

コスト関数

線形回帰は、最小二乗法が定義する関数の最適化によって推定できることがわかりました。

そして、それは、ある設定の下での最尤推定でもありました。

しかも、その推定は線形代数を用いて、推定値が確定するという特徴も持っていました。

多くの推定・教師アリ学習では、次のようなアプローチをします。

- あるモデルで考えることにする
- そのモデルはパラメタを持っていて、パラメタの値を変えることで、具体的な学習の結果を変えることができる
- 観測データにうまく適合するパラメタ値を探すことが学習である
- 「うまく適合する」という部分は、何かしらの「関数の最適化」として設定される

線形回帰の例で言えば、「$y = ax + b$という式で考える」ことが「あるモデルで考える」ことに相当します。

このモデルでは、具体的に(a, b)の値を変えることで、色々な直線を表すことができます。

観測データが与えられたら、それと「うまく適合」する(a, b)を探しますが、「うまく適合する」ことを観測値と予測値の差の二乗の和の最小化として設定しています。

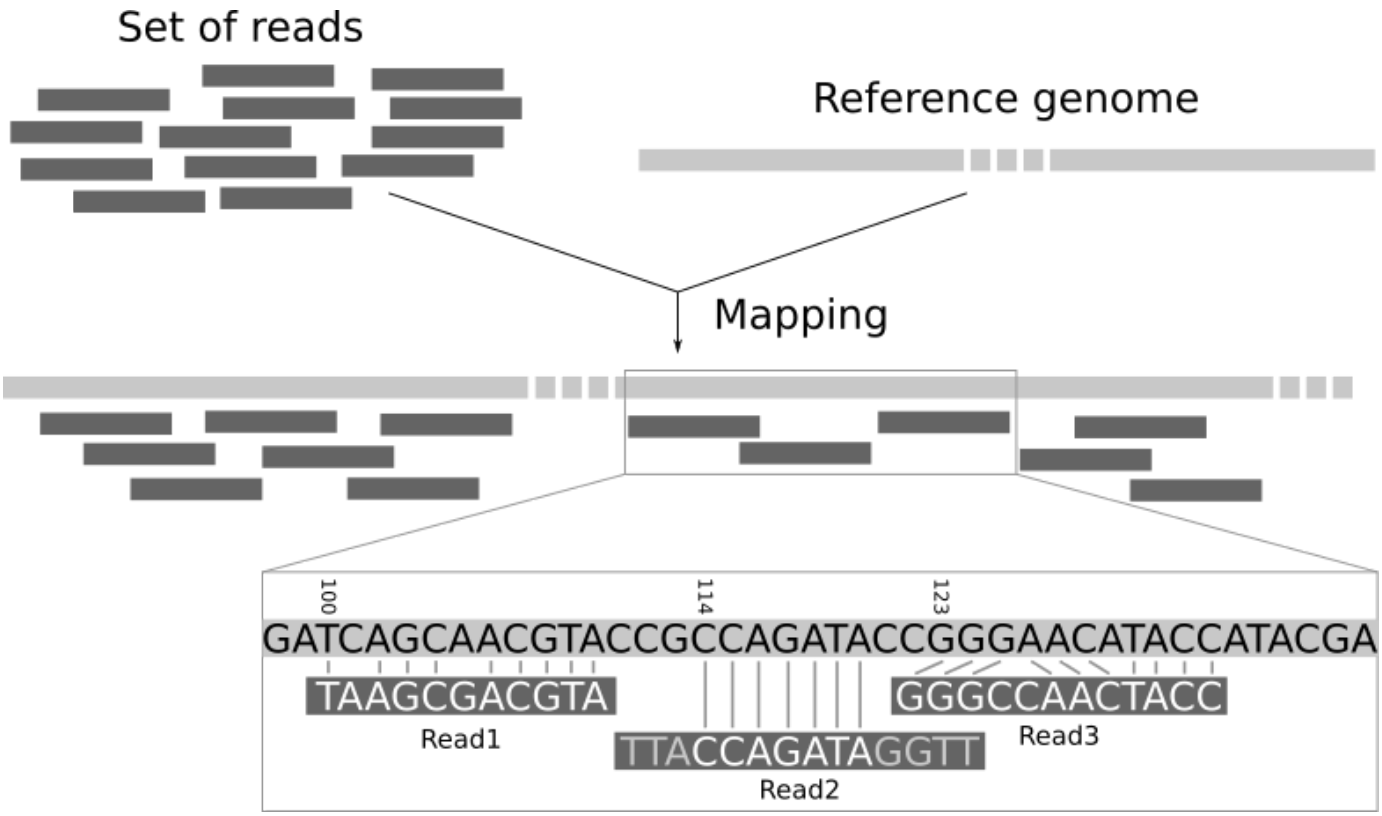
「うまく適合する」ことを表す関数は「最小化」することが多いので、「コストの最小化」をする、という意味で、コスト関数と呼びます。

最尤推定は、尤度関数・対数尤度関数を最大化する探索ですが、尤度関数・対数尤度関数に(-1)を掛けた関数をコスト関数とした学習である、とも言い換えられます。

尤度関数・対数尤度関数に(-1)を掛けた関数をコスト関数にした学習は、学習の出力が表す仮説の尤度が、観測データの下で最大になるという、「確率・尤度」的な「意味」を持ちますが、コスト関数は必ずしも「確率・尤度」によって定まるものではありません。

「あれよりはこれの方が良さそう」というアイディアがあれば、そのアイディアを関数として表して、それをコスト関数として最適化アルゴリズムを回して、パラメタを決めてやれば、そのアイディアの下での学習がなされたことになります。

ショートリードのマッピングにおけるコスト関数



(https://training.galaxyproject.org/training-material/topics/sequence-analysis/tutorials/mapping/tutorial.html#map-reads-on-a-reference-genome (https://training.galaxyproject.org/training-material/topics/sequence-analysis/tutorials/mapping/tutorial.html#map-reads-on-a-reference-genome))

次世代シークエンサーを使ってゲノム配列を読んだり、トランスクリプトーム解析を行うと、大量のショートリードが得られます。

得られたショートリードはレファレンス配列にマッピングされます。

今、ショートリードの長さが40塩基分あるとします。

ヒトのレファレンスゲノム配列(約30億塩基対あります)を探し回って、どこかにピタリと合う場所が 1 箇所だけ見つかったとします。

この場合は、このショートリードはこの場所にめでたくマッピングされたことになります。

残念ながら、そのような場所が見つからなかったものの、39塩基分はピタリと合い、1塩基だけ合わなかったとします。

このショートリードはこの場所にマッピングするのがよいのか、しないのがよいのか、決めなければ、次に進むことができません。

ショートリードは実験の産物ですから、1塩基くらいは読み誤りをするかもしれません。

また、たまたま、そこがSNPであって、レファレンスゲノム配列では標本のアレルとは別のアレルが登録されているのかもしれません。

読み間違いにしろ、SNPにしろ、1塩基分くらいは「許容」しよう、と考えても良いかもしれません。

または、ショートリードの各塩基には、精度情報がついているので、「精度が悪い塩基でたまたまレファレンスゲノム配列と合わな」かったのなら、「許容」することにする、という方針もアリでしょう。その場合は、逆に、「精度が良い塩基」だったら、「許容しない」ことにするという基準を決めないといけません。

では「精度の良し悪しの弁別基準はどうするのか」も決めないといけません。

SNPかもしれないから許容しようという場合にも考慮するべき事柄はあります。

SNPのデータベースに「そこはSNPである」と登録があるのであれば「許容」する、という方針もアリでしょう。 さらに厳しくして、「データベースに登録されているSNPのアレルと一致」していたら「許容」する、という方針もアリです。

この考慮・基準値によって、「許容する」「許容しない」という判定をするわけですが、「許容する・許容しない」ことを「判定関数が値1・0を返す」として関数を定義すれば、これは一種のコスト関数です。

1塩基だけ合わない場合にも色々と考慮することがありました。基準値も定めないといけませんでした。

合わない箇所が2か所、3か所…と増えて言ったら、どうするのかも定義しないとマッピングは行えません。

ここまでの例では、マッピングしてもよいかどうか迷う場所がゲノム上で一か所しかない場合として書いてきました。

実際には、 あるゲノム上の場所では、「2塩基分だけ合わない」が、別の場所では、レファレンスゲノム上に「長さ2塩基対分の欠失があったと見做せば、完璧に合う」が、どちらにマッピングしたと考えたら良いだろうか、というような問題も出て来ます。

配列の長さには問題がないが、塩基が入れ替わっている場合も、挿入・欠失を考慮すれば合致する場合も、ショートリードの実験上の読み間違いであると考えて、どちらが正しそうかを数値化するという方針もアリでしょう。

逆に、ショートリードは実験的には正しいと考えられる場合には、2個のSNP型突然変異が起きたのか、長さ2塩基分の欠失型突然変異が起きたのかのいずれかと考えて、どちらが、生物学的に尤もらしいかとして、 2つの可能性を比較することが適切かもしれません。

このように考えると、ショートリードをレファレンスゲノム配列にマッピングする処理は、30億か所のすべてについて、マッピングするためのコストを計算し、最もコストが小さいか所を見つける処理であることがわかります。

大多数の場所では、ショートリードのほぼすべてのか所にSNP型突然変異が起きたというコストを想定しないといけないので、「高コスト」であることになって脱落し、ごく限られた場所のみが、「低コスト」の場所としてマッピング候補場所となります。

このコスト関数は、ショートリードの質や、合致しないパターン(置換型、挿入・欠失型など)、その数など、複数の要素からなる多変量関数になっていますし、様々なコスト関数の定め方があり得ます。

そして、このコスト関数は、確率・尤度的に説明できる部分も含みますが、かなり恣意的な定義になることの方が多いです。

いわゆる機械学習で用いるコスト関数も、統計学的な意味・計算しやすさなど、複数の要因によって決められるものですし、同一の学習器に複数のコスト関数を使う余地があり、選択されたコスト関数の違いが、学習結果の違いとなって現れてきます。

複雑なモデルと過剰適合

togetter 記事 キーワード・ユーザ名を入力 検索

ランキング 注目まとめ 今週の人気 新着まとめ イチオシ 編集部厳選！ まとめ簡単 大好評Togetter Clip!

トップ > 2016年 > 1月 > 25日 2016年1月25日

「ビッグデータ解析で薬剤副作用予測がほぼ100%可能に」というニュースへの反応

京大のプレスリリースの怪しさを指摘するツイートを収集してみました。
http://pc.watch.impress.co.jp/docs/news/yajiuma/20160125_740480.html

国内 副作用 京大 ビッグデータ 過学習

litulon 40070 16 263 B! 51 109 52

PC Watch @pc_watch
【やじうまPC Watch】ビッグデータ解析で薬剤副作用予測がほぼ100%可能に～京大発表。今後はSoC活用で処理を高速化 pc.watch.impress.co.jp/docs/news/yaji... pic.twitter.com/4RES8fcj6g
2016-01-25 12:33:08

まろやか🐟ついに来た浅利773戦国時代 @MaroYakaZ
この分野でそこまで言い切るのは大した度胸 / “ビッグデータの解析で薬の副作用予測がほぼ100%可能に ― 京都大学” htn.to/8AXbWX
2016-01-25 07:20:44

™ @tmaehara
うーん？？？違和感が強いぞ．ポジネガ判定を線形回帰でやって，正答率100%ってなんかおかしくないか．
2016-01-25 12:30:29

™ @tmaehara
予測とかそのへんの記述がめっちゃ薄いのに，ビッグデータバズーンなポンチ絵とか，オブジェクト指向設計のクラス図とか，その辺の記述が盛りだくさんで，かなり警鐘鳴ってるぞこれ．

kyoto-u.ac.jp/ja/research/re...
2016-01-25 12:37:22

(https://togetter.com/li/930035 (https://togetter.com/li/930035))

ゲノム・オミックス解析ではたくさんの要素(たくさんの遺伝子多型・たくさんの遺伝子・たくさんの分子)を用います。

説明変数が非常にたくさんあると言うことです。

たくさんの説明変数を使いさえすれば、すべての標本を100%正確に分類・回帰等、することができます。

説明変数の数がモデルの自由度になるからです。 自由度の高いモデルは観測データをうまく説明する(フィットする)ことができてしまうのです。

説明変数の数を増やすことは、モデルの自由度を大きくする一つの方法ですが、説明変数の数を増やさずに、モデルの自由度を大きくする方法があります。

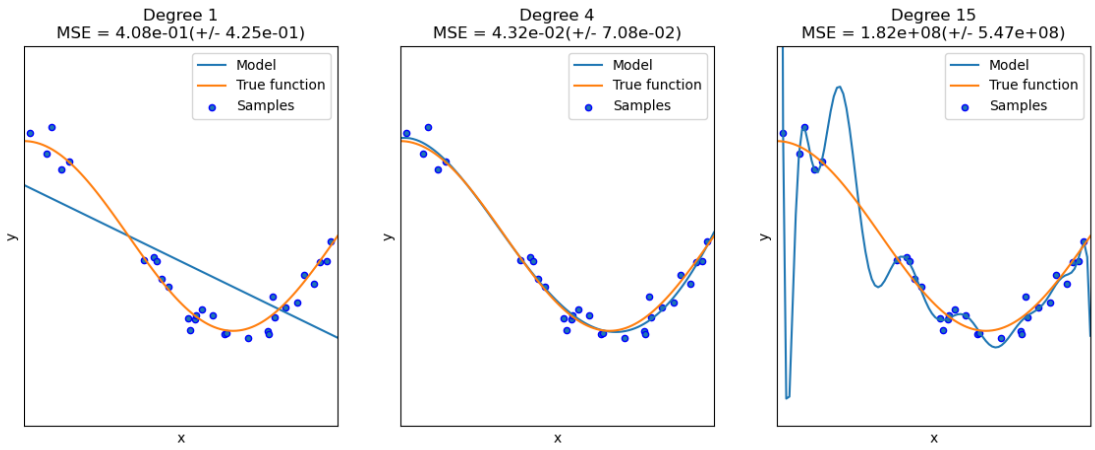
一つの変数から、いくつもの変数を作り出すと、作り出した変数の数だけ、自由度が上がります。

$y = ax + b$というモデルは自由度が2(a, b)ですが、$y=a_2 x^2 + a_1 x +b$ のように、xからx^2という新たな変数を作ると、自由度が3(a_2,a_1,b)に上がります。

以下の図はpythonでの多項式近似です。多項式の次数を上げる(モデルの自由度を上げる)と、カーブの増減は複雑になり、観測点とのフィッティングが良くなる様子が見て取れます。

ただし、観測点とのフィッティングは良くなりますが、観測値には誤差項が含まれているのにも関わらず、その誤差にもフィッティングしようとして、背後に隠れている真の曲線の予測としては、適切でなくなっていることが判ります。

観測値に(その誤差も含めて)、寄り添い過ぎたフィッティングという意味で、「過剰適合(overfitting)」と呼ばれます。



多項式近似

(https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html#sphx-glr-auto-examples-model-selection-plot-underfitting-overfitting-py (https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html#sphx-glr-auto-examples-model-selection-plot-underfitting-overfitting-py) より)

ほどよくフィットし、ほどよく外れるモデル

GWASのリスクSNPとPolygenic Risk Score

多重検定補正による偽陰性とリスクSNP数の制御

観測データをうまく説明することを目指すことは、データの活用の原則ですが、過剰適合という問題があることが判りました。

過剰適合を避けるためには、モデルの自由度を下げるのが一方法です。

GWASで帰無仮説棄却の閾値を厳しくして(多重検定補正)、疾患と関連しているとみなすSNPの数を多くし過ぎないことは、偽陽性SNPを増やさないという意味が強いですが、偽陰性を増やしていることも確かです。

リスクが小さいSNPは捨てているのです。

これにより、疾患リスクを説明する変数の数を減らしています。 説明変数の数を減らせば、それらを用いてリスクの説明をする場合、過剰適合が起きにくくなります。

複合遺伝性疾患では、疾患によって数十、百超個のSNPがリスクSNPとして同定されます。

これらのリスクSNPを変数とした線形モデルにより、疾患リスクが決める方法にPolygenic risk scoreと呼ばれる手法があります。

以下のような単純な式になります。m個のSNPを用いて、i-番目のSNPのリスクアレルの保有本数をX_iとし、そのSNPアレルに$\hat{\beta}_i$なる重みを与えて和を取るものです。

$$\hat{S} = \sum_{i=1}^{m} X_j \hat{\beta}_j$$

Lasso回帰とPolygenic risk scoreを組み合わせる

多変量線形回帰において、説明変数の数を減らす手法にLasso回帰(least absolute shrinkage and selection operator regression)と呼ばれる手法があります。

Lasso回帰によって得られる学習モデルがどんなものかを端的に説明すると、次のようになります。

- 全部でM個の説明変数を学習に用いられるが
- 手法の特性により、$m < M$個の説明変数のみが選択されて
- 選択された説明変数のみによって線形回帰式が出力される

説明変数を絞り込んでいるので、過剰適合を防ぐ力があります。

実際、Lasso回帰によって説明変数SNPを絞り込み、Polygenic risk score算出式を推定する手法が提案されています(R package "SummaryLasso")。

Lasso 回帰による変数選択とペナルティ

Lasso 回帰では、次のような式を最小化します。

標本数はNで、説明変数の数はMです。

$$A + B$$

ただし、

$$A = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$B = \lambda \sum_{j=1}^{M} |\beta_j|$$

$$\hat{y}_i = \sum_{j=1}^{M} x_{i,j} \beta_j$$

Aは最小二乗法と同じで、観測値と予測値との差の二乗の和です。 このAを小さくすることは、最小二乗法で回帰をしていることと同じですから、いわゆる普通の線形回帰です。

Lasso 回帰では、AにBを加えています。Bは各説明変数の重み係数の絶対値の和に比例して大きくなる項です。

これを小さくするには、係数β_jが0に近くする必要があります。係数が0になるということは、その説明変数を回帰モデルから外すことに相当するので、モデルに取り入れる説明変数の数を減らす力があります。

観測データへのフィッティングをよくするためにはβ_jに値を与えて、変数を働かせる必要がありますが、そうすると、働く変数の数が増えてしまうので、Aを小さくするとBが大きくなり、Bを小さくするとAが大きくなるというジレンマを抱えた式になっています。

このようなジレンマを抱えた式の最小化をするβを答えとすることで、観測データに寄り添い過ぎない(Aのみの最小化ではない)回帰式が得られます。

ここまでBの項に含まれていたλについて説明をしませんでした。

このλは0以上の実数として与えなければなりません。λを大きくすればBの寄与が大きくなるので、捨てられる説明変数の数が大きくなり、採択された説明変数の重み係数の絶対値は小さめになります。 逆にλを小さくすれば、より多くの説明変数が採択され、観測データへのフィッティングは良くなります。$\lambda = 0$の場合は、単なる線形回帰になります。

このBの項のように過剰適合を防ぐ働きを持つ項をペナルティ項と呼びます。

最適化関数をフィッティングを良くする項とペナルティ項の和にするアプローチは広く機械学習で用いられています。

ペナルティ項をどのように設定するかによって、Lasso 回帰のように、説明変数の数が少なくなる場合もありますし、説明変数の数は減らさずに、係数の値が小さめになる場合もあります。

過剰適合の防止とクロスバリデーション

ペナルティ項を導入することで過剰適合を防ぐ方向性は見えてきました。

しかしながら、λがペナルティ項には含まれていたように、「過剰適合が本当に防げていたのか」「少々のペナルティでは、いまだに過剰適合しているのか」逆に「過剰適合を恐れるあまり、過少適合になっていないか」という心配が残ります。

この問題に対処する方法の１つがクロスバリデーションという方法です。

過剰適合とは、「学習をする際に用いたデータセット」に寄り添い過ぎた出力であることでした。

従って、「学習をする際に用いたデータセット」のフィッティングの程度と、「学習に用いていないデータセット」のフィッティングの程度とが、同程度であれば、「フィッティングの程度」は「学習をする際に用いたデータセット」に寄り添い「過ぎ」ていないと考えることが出来そうです。 逆に、「学習に用いたデータセット」のフィッティングが「用いていないデータセット」のフィッティングより良ければ、「過剰」に適合していると考えることが出来るでしょう。

従って、過剰適合についての評価をするためには、「学習する際に用いるデータセット」と「学習する際に用いないデータセット」との２タイプのデータセットを持つことが有用になります。

２種類のデータセットを完全に独立に持っていれば、そのようにすることは容易です。

学習に用いるデータセットを訓練データセット(training data set)、学習結果を試すデータセットをテストデータセット(testing data set)と呼びます。

もし、２セットを独立に持てないならば、工夫をして２セットを作る必要があります。 そして、多くの場合、２セットを独立に用意するのは難しいので、工夫をする必要が出てきます。 なぜなら、２セット分の標本を集めることは難しいことが多く、また、標本数が多くなれば実験費用もかさむからです。

ある１つのデータセットを分割しつつ、その分割パターンを変える、分割パターンを変えるときにリサンプリングの手法を使うなど、訓練・テストのデータセットを作ってクロスバリデーションをする方法が幾つか知られています。

データ駆動型とノンパラメトリック手法

はじめに

ゲノム・オミックス研究は、「いったい、どういう風になっているのだろう？」と検討がついていない状況で、「実験技術が進んだから、ひとまず取れるだけデータをとってみよう」、その上で、「データから何が言えるかを引き出そう」というタイプの研究です。

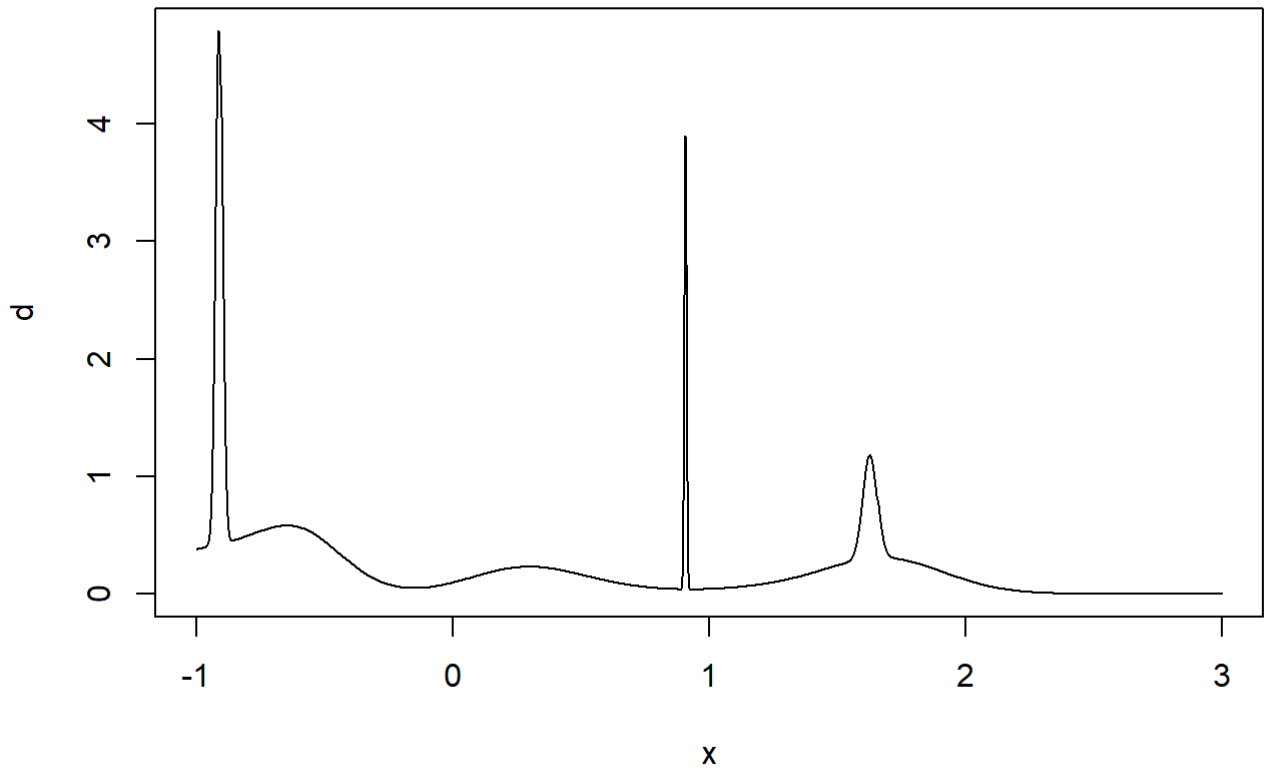
そのような大規模なデータに対して、モデルを設定して、モデルのパラメタを推定する方法(検定、推定、教師アリ機械学習など、ここまで述べてきたアプローチ)を取ることも出来ますが、データそのものに無前提で向き合って、データが得られるままに、解釈をして行く、というアプローチもあります。

このアプローチの場合には、データ量が増えれば増えるほど、細かな読み取りが可能になり、得られる出力は複雑になっていきます。

ノンパラメトリックなアプローチ：標本が増えれば増えるほど複雑な出力が得られる

複雑な分布があるとしましょう。

```
n.dist <- 10
ms <- runif(n.dist) * 3 -1
sds <- runif(n.dist)*0.3
fs <- runif(n.dist)
fs <- fs/sum(fs)

x <- seq(from=-1,to=3,length=1000)
d <- rep(0,length(x))
for(i in 1:n.dist){
  d <- d + dnorm(x,ms[i],sds[i]) * fs[i]
}
plot(x,d,type="l")
```

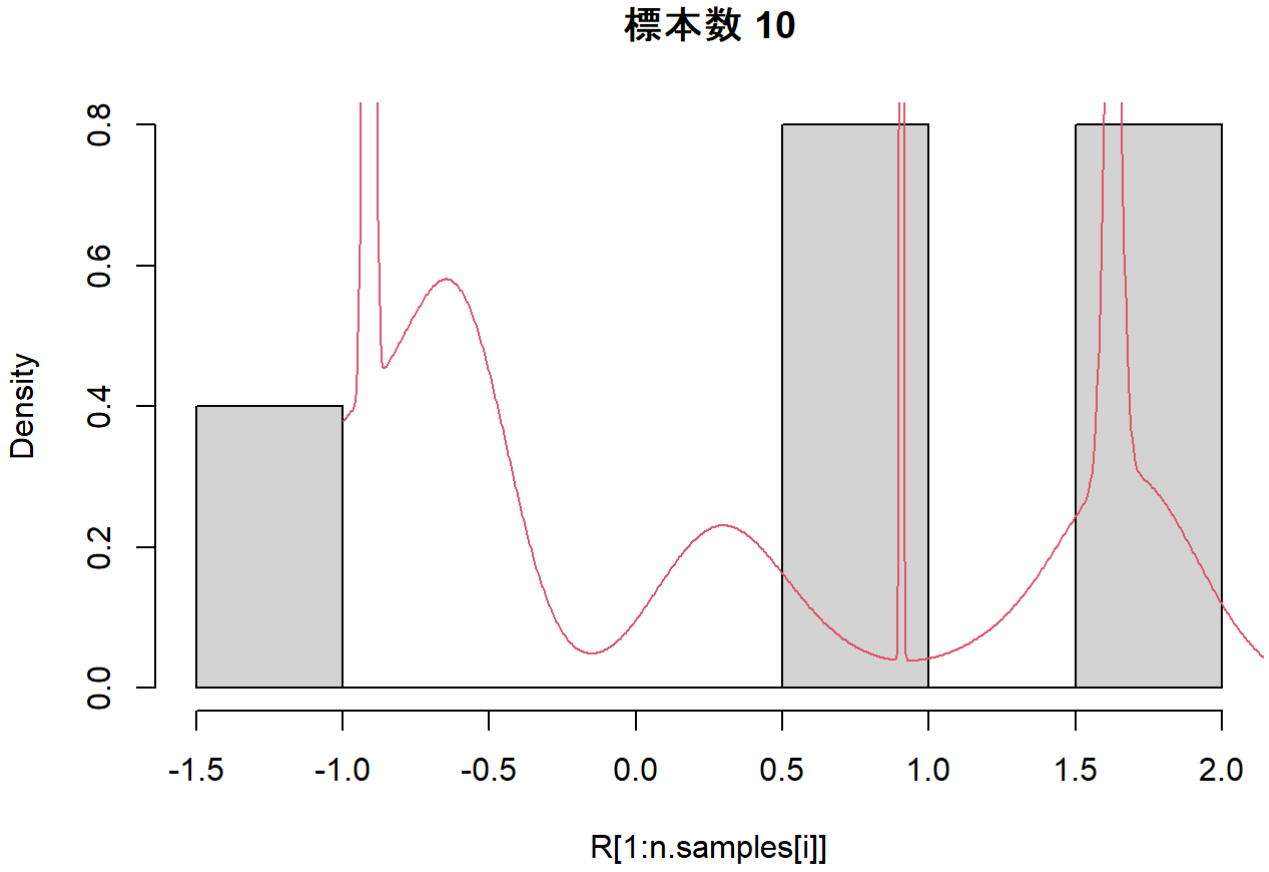


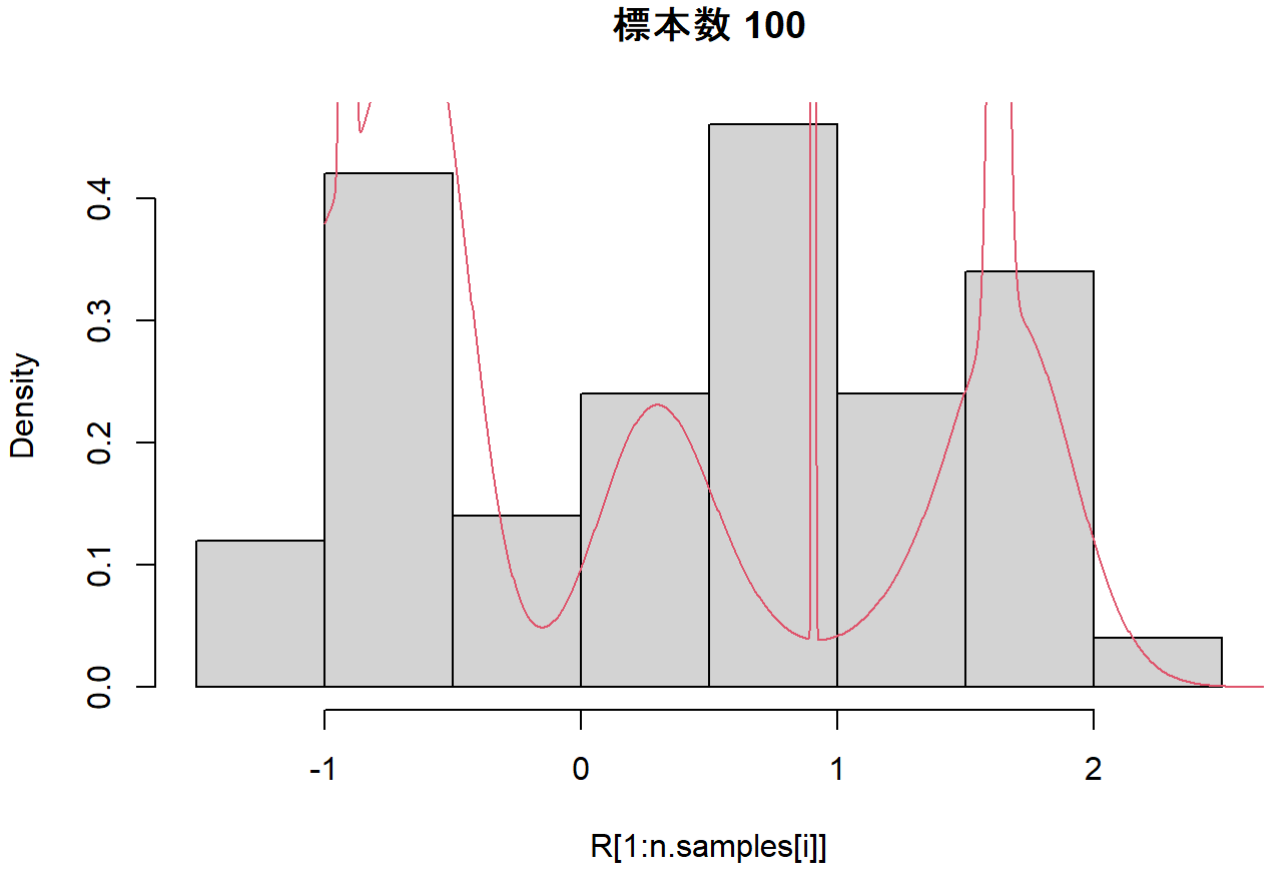
この分布から標本をランダムサンプリングし、だんだんに標本数を増やして行くとします。

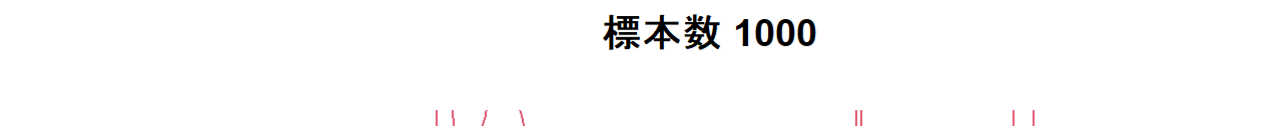
その標本に対してヒストグラムを描いてみます。

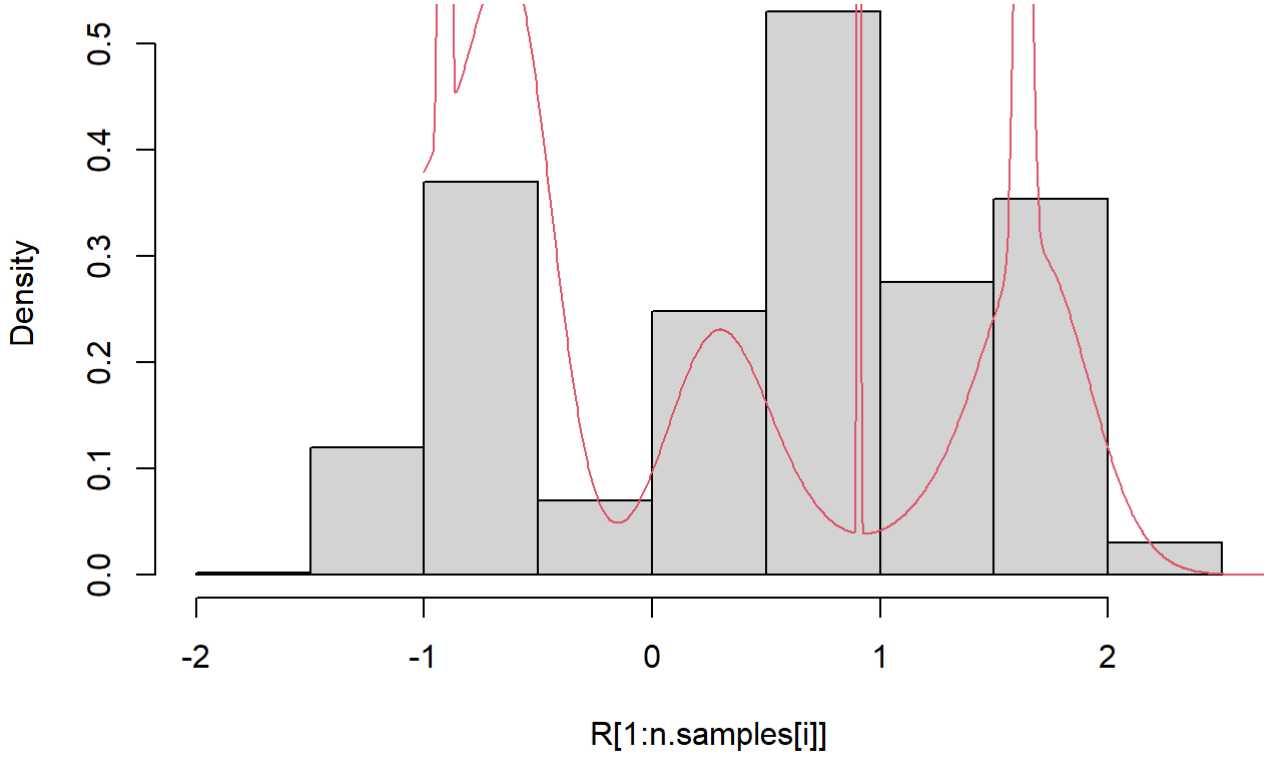
```
# 乱数を発生
max.sample <- 10^5
r <- sample(1:n.dist,max.sample,replace=TRUE)
R <- rep(NA,max.sample)
for(i in 1:max.sample){
  R[i] <- rnorm(1,ms[r[i]],sds[r[i]])
}
```

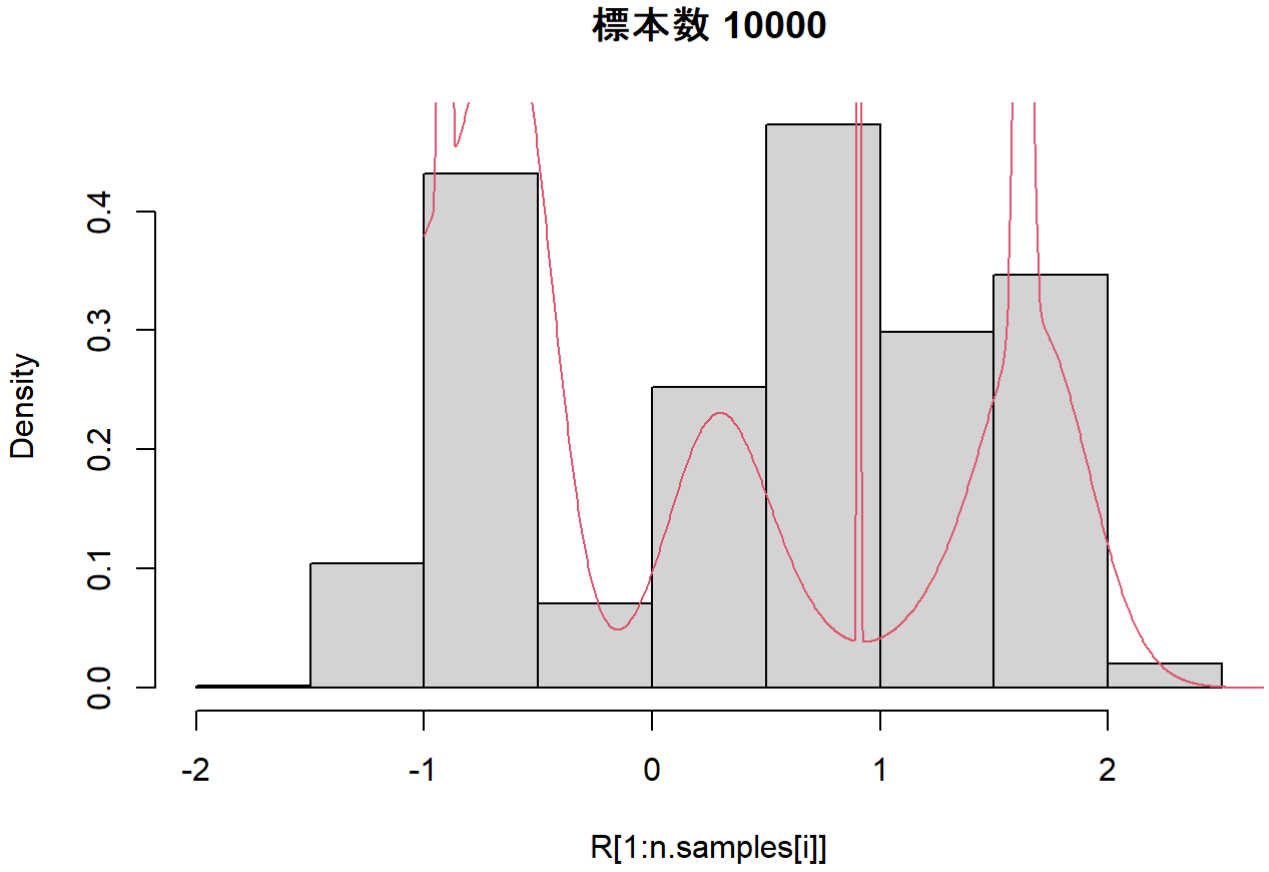
```
n.samples <- c(10,100,1000,10000,100000)
for(i in 1:length(n.samples)){
  hist(R[1:n.samples[i]],freq=FALSE,main=paste("標本数",n.samples[i]))
  points(x,d,type="l",col=2)
}
```

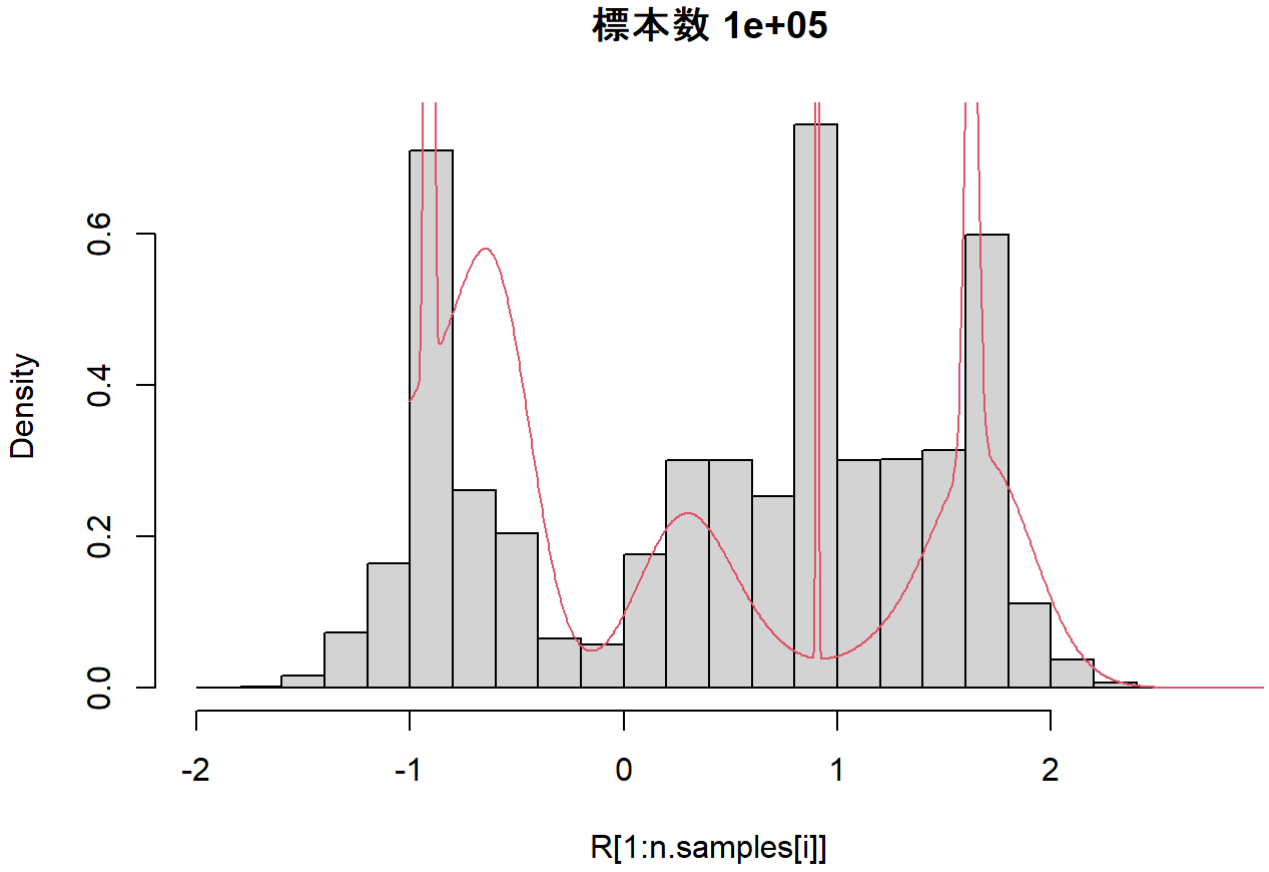

標本数 10
Density
0.0
0.2
0.4
0.6
0.8
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
R[1:n.samples[i]]


標本数 100
Density
0.0
0.1
0.2
0.3
0.4
-1
0
1
2
R[1:n.samples[i]]


標本数 1000


Density
0.0
0.1
0.2
0.3
0.4
0.5
-2
-1
0
1
2
R[1:n.samples[i]]





標本数が増えるほど、ヒストグラムは複雑になり、「真の分布」に近づいていく様子が見えます。

標本数が十分に大きいならば、複雑なモデルを考えることが適当なことが判ります。 そして、ヒストグラムを描くというアプローチは、特段の処置を施さずに、標本数の増加に応じて結果が複雑になっています。

これが、ノンパラメトリックな手法の特徴です。

過剰適合とノンパラメトリックアプローチ

前の章では、過剰適合を防ぐためには複雑なモデルは避けるべきでした。

データ駆動型では、標本数に応じて複雑なモデルが得られて、それなりのメリットがある例が示されています。

モデルは複雑にしない方がよいのか、複雑にする方がよいのか、どちらなのでしょうか？

過剰適合を避けるために複雑なモデルを避けた場合と、標本数に応じて複雑なモデルを採用した場合とでは、フィッティングに当たっての立場が違うので、あたかも、相反する主張をしているようになるのです。

過剰適合を避けるために複雑なモデルを避けた場合には、多項式を使ったり、既知の確率密度分布を使ったりして、その重み係数のフィッティングをしました。

その際に採用した多項式や既知の分布が、背景にある真実にピタリと当て嵌まる場合には、標本数さえ増やせば過剰適合の心配は要りません。

しかしながら、多項式や既知の分布は単純な部品なので、それを使っても、「真実の複雑さ」を完全に説明することはできない(ことが多い)です。

そもそも完全に説明することができない前提なので、無理にフィッティングすると、過剰適合の問題が起きるのです。

この多項式や既知の分布はパラメターを用いて定義されているので、このアプローチはパラメトリックアプローチと言われます。 多項式の係数や正規分布の平均と標準偏差などがパラメタに当たります。

他方、ノンパラメトリックな手法では、パラメタを使って定義された部品を想定していません。データ量が増えるに従って、「自由に」複雑さを変えることが可能です。 この特徴は、背景に隠された「真実の複雑さ」を完全に説明する力の源となっています。

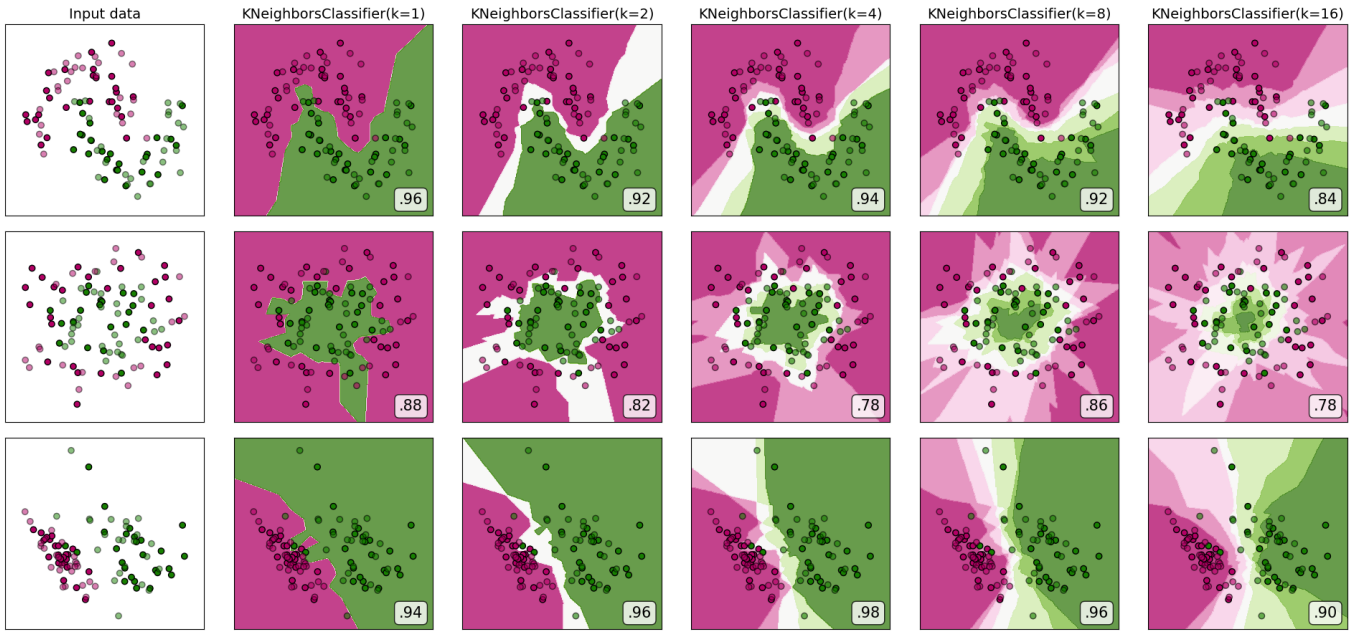
ゲノム・オミックス解析では、複雑な仕組みの解明を目指しているので、「真実の複雑さ」に迫るために、ノンパラメトリックな手法が活躍する場面が増えてきます。

では、ノンパラメトリックな手法であれば、過剰適合を心配しなくて良いのでしょうか？

答えは「否」です。真実の複雑さを説明する力が強いということは、それだけ柔軟にデータに寄り添うことができるということでもあり、過剰適合は問題になります。

kNN法の例

ノンパラメトリックな手法の１つにkNN法(k-近傍法)というものがあります。



knn法による分類

(https://tomdlt.github.io/decks/2018_pydata/#1 (https://tomdlt.github.io/decks/2018_pydata/#1) より)

左端に２色の点が観測データとして表示されています。

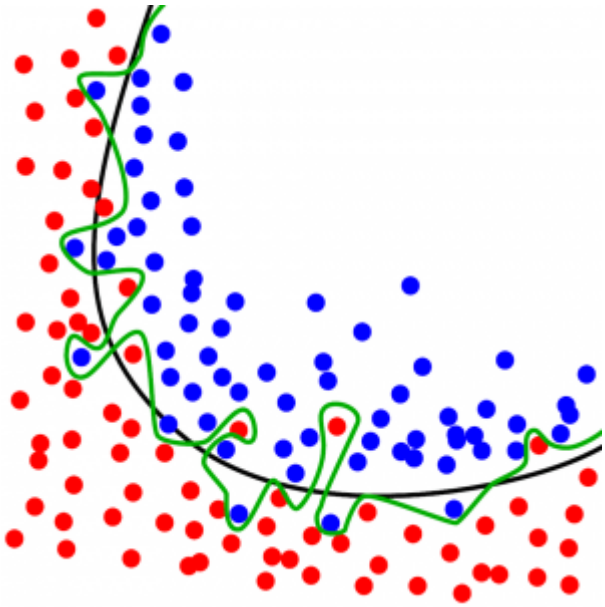
kNN法により、空間(この場合は２次元平面)のそれぞれの場所が、赤らしいか、緑らしいかで色分けしています。

kNN法では、空間の各点(観測データがある場所も無い場所も)についての「赤らしさ・緑らしさ」に点をつけます。

- その点から1番,2番,...,k番目に近い観測データ点(k個の「近傍」の点)を確認します
- k個の「近傍」の点の赤・緑情報を確認し、赤ばかりなら、その点は赤の領域に属し、緑ばかりなら、その点は緑の領域に属すものとします
- 赤と緑が混在していたら、その割合に応じて、赤っぽいか・緑っぽいかを決めます

図では、kの値が左から1,2,4,8,16となっています。kの値が増えるほど、曖昧な領域(白っぽい領域)が増えています。

この例では、k=1ではっきり分けるのが良さそうに見えますが、次の例では、緑の線で細かくわけるのは「やりすぎ」で、黒い線で大雑把に分ける方が良さそうに見えます。



(https://towardsdatascience.com/model-selection-tuning-and-evaluation-in-k-nearest-neighbors-6d3024d78745 (https://towardsdatascience.com/model-selection-tuning-and-evaluation-in-k-nearest-neighbors-6d3024d78745) より)

kNN法ではkの値の選択によって複雑さの調整をすることができるので、それを過剰適合回避に利用することができます。

非線形次元削減 UMAP法でのkNN法

冒頭で紹介した非線形次元削減 UMAP法では、処理の一部にkNN法を使っています。

冒頭のRコードの中で以下のような実行条件の指定を行っていました。

```
custom.settings$n_neighbors = 3
custom.settings$n_neighbors = 10
```

このn_neighborsというのが、近傍の数という意味でkNNのkに相当します。

シングルセル・トランスクリプトーム解析をするにあたって、遺伝子発現のパターンを細胞同士で比較し、似ている方からk番目までの細胞を、「お隣」とみなすことにする、という設定です。

「お隣同士」とみなされた細胞ペアの間に線を引くことで、各細胞を頂点とするグラフが得られます。

グラフが与えられると、任意の２頂点間には、グラフ距離(グラフ上をたどって測る距離)が定義できるので、細胞同士の間に、グラフの形を考慮した距離が定まります。

UMAP法では、このグラフ距離を「細胞間の発現パターンの違いを表す値」とみなします。

グラフ上をたどる道筋は、うねうね・くねくねと曲がったものになるので、「違いの評価」が「非線形〜曲がったもの」になっています。

kNN近傍法を使ってグラフを作ることから、同手法は非線形性を持った手法になっているのです。

UMAP法によりシングルセルを平面に描図する差異には、「グラフ視覚化アルゴリズム」が使われています。

グラフ視覚化とは、グラフは高次元にのたくった隣接関係を表しているが、それを２次元(や３次元)に表示するには、どうするとうまく視覚化できるかに関する研究分野で、色々な手法が開発されています。 高次元空間にあるグラフを無理に低次元に押し込めているので、このグラフ視覚化アルゴリズムも非線形な処理をしています。

終わりに

ゲノム・オミックスデータ解析の具体的な話題を例に取りながら、ゲノム・オミックスデータ解析の特徴を、データ自身の特徴と解析上の特徴に分けて説明しました。

また、ゲノム・オミックスのデータ解析は多段階になっており、また、データから読みだすメッセージに応じて、検定・推定・学習を使い分けているため、スタディで用いられる手法は、スタディごとにバリエーションを持って来ます。

したがって、特定の論文に使われる手法のフローをなぞって確認すると、その論文の手法についての理解は深まるものの、別の論文を開くと、全く別の手法が使われていることに気づき、また一から勉強し直しとなって、やる気がそがれるかも知れません。

ですので、この「解析教室」では、解析フローを構成するパーツは一見すると違った姿をしていますが、すべてに共通する考え方があると言う点を強調して説明することを試みました。

その試みが成功したか失敗したかは判りませんが、少しでもゲノム・オミックスデータ解析手法の理解の助けになればと思います。