2015 年 11 月 19-20 日 遺伝研

新学術 「ゲノム支援」
平成 27 年度 情報解析講習会
R を用いた統計解析

東京工業大学 大学院生命理工学研究科
森 宙史（Hiroshi Mori）

R は、コンピュータのプログラム言語の名前でもあり、R 言語を用いたオープンソースの統合解析ソフトウェアの名前でもある。ただし、ほとんどの場合 R は後者の意味で用いられており、本実習でも後者の意味で用いる。統合解析ソフトウェアである R は、多彩な統計解析手法や視覚化手法が実装されているにも関わらず無料で利用可能なため、バイオインフォマティクスの分野に限らず世界中で広く利用されている。ここでは、R の操作の基本と、大規模オミックスデータの解析に必須な基本的な統計解析を R で行う方法について解説する。

全体の流れは以下の通りである。

[1] R と RStudio のインストール
[2] RStudio を用いたパッケージのインストール
[3] RStudio を用いた基本的な統計量の算出およびグラフ描画
[4] 分割表データを用いた統計検定
[5] 多重性の問題
[6] 多変量解析

[1] R と RStudio のインストール

(1) R のインストール

R のプログラムは、CRAN (Comprehensive R Archive Network)の Web サイトからダウンロード可能である。URL は http://cran.r-project.org/ となる。2015 年 11 月時点での最新版は version 3.2.2 となる。

Windows の場合は Download R for Windows、Mac の場合は Download R for (Mac) OS X を選択してダウンロードする。ダウンロード速度が遅い場合は、

Web サイトの左側にある Mirrors から日本のミラーサイトを選び、そこからダウンロードした方が高速である。ファイルサイズは 70MB ほどであり、展開後は 150MB ほどのサイズになる。ダウンロードした後、ファイルをダブルクリックしてインストールを行う。

(2) R の起動

使用許諾に同意等の操作を行い R のインストールを終えたら、R アイコンをダブルクリックして R を起動する。R コンソールというコマンド入力を受け付ける Window が起動したら、R のインストールは完了である。

(3) RStudio のインストール

R は標準では R コンソールのみのシンプルなユーザインターフェース(UI)であるが、コマンドを入力して実行という形式に慣れていない場合は使いにくく感じることもある。そこで、マウスクリック等を併用してより使いやすい UI で R を操作可能にする環境である、RStudio をインストールする。RStudio は以下の RStudio の Web ページの、Products から選択可能である（図 1)。

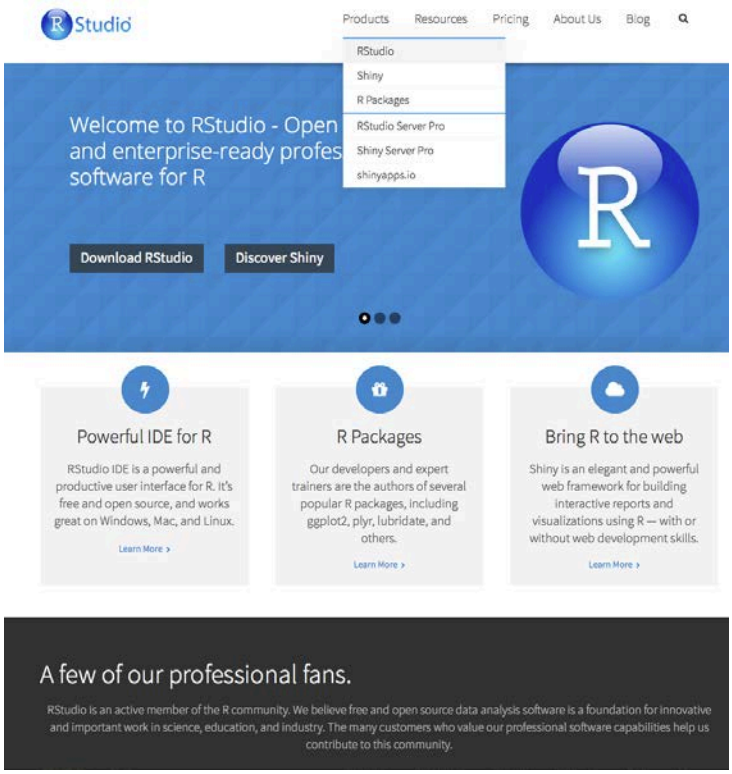


図 1. http://www.rstudio.com/　の画面

Desktop 版と Server 版があるが、Desktop 版を選択。Desktop 版を選択後、左側の Download RStudio Desktop を選択（図 2)。

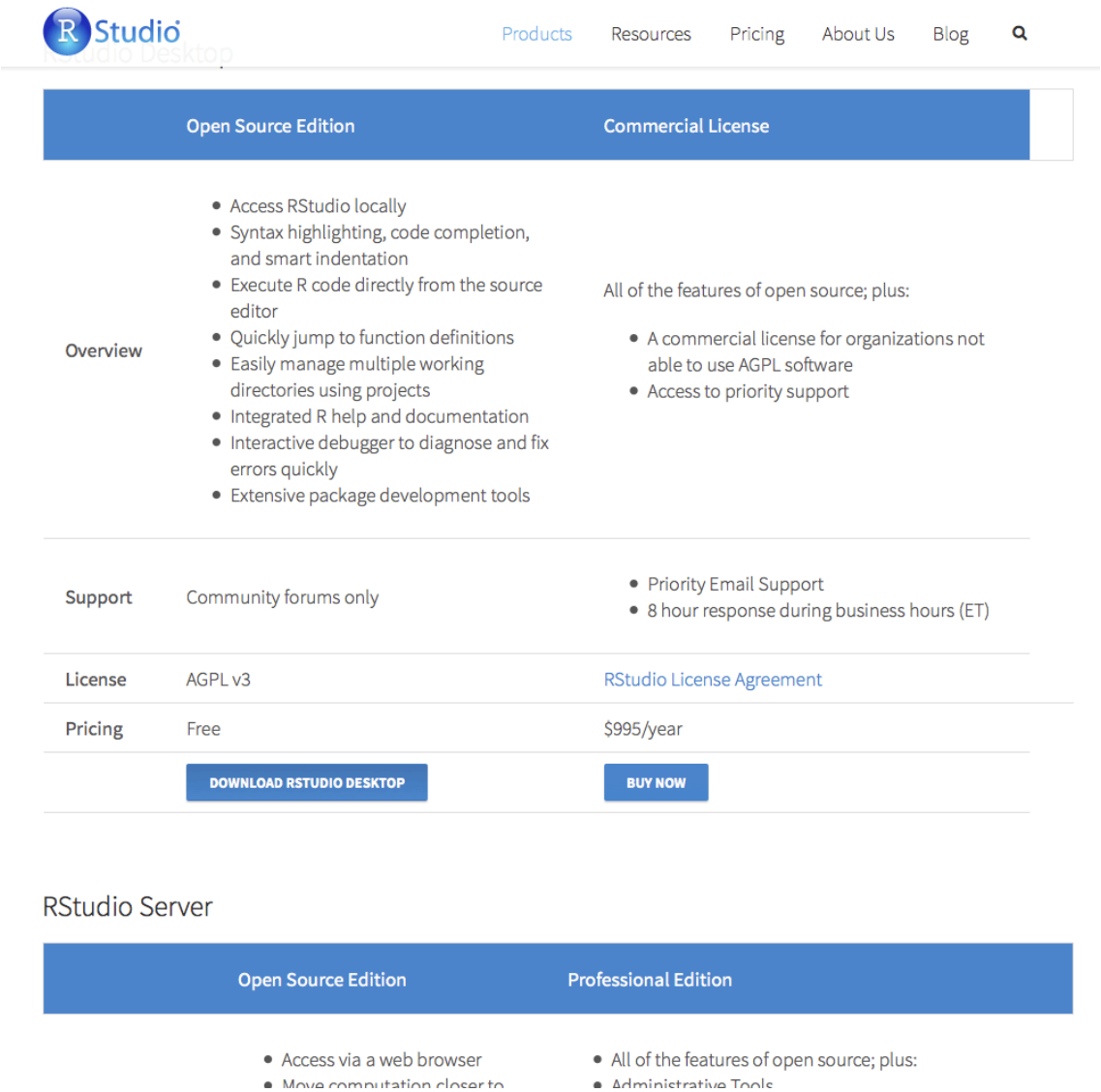


図 2. Desktop 版を選択後の画面

Installers の、該当する OS を選択して RStudio の Installer をダウンロードする。ダウンロード後、ファイルをダブルクリックして RStudio をインストールする。Installer のファイルサイズは 45MB ほどであり、インストール後は 300MB ほどのサイズになる。インストール完了後、RStudio のアイコンをダブルクリックして RStudio を起動する。RStudio 起動の際には、R を事前に起動しておく必要は無い。RStudio が正常に起動できたら、RStudio のインストー

ルは完了である。今後はRアイコンを選択してRを起動すること無く、いきなりRStudioのアイコンを選択してRStudioを起動すれば、RStudio内で自動的にRが起動される。

R
RStudio
fastcluster (R パッケージ)
ggplot2 (R パッケージ)
cummeRbund **(R パッケージ)**
のインストールがまだな方は、インストールをお願いします。
トラブっている場合は手を上げて下さい

[2] RStudio を用いたパッケージのインストール

R はオープンソースであるため、標準でインストールされている関数群に加えて、世界中の開発者が開発して配布している様々な関数をパッケージとして組み込むことが可能である。ここでは、RNA-Seq の実習で用いる 3 パッケージをインストールする。
右下の Packages タブを選択し、Install を選択する。
Install from が Repository (CRAN)になっていることを確認し、Packages 欄に
fastcluster
と入力して、Install を選択。R コンソール上にインストール結果が表示されれば成功。
同様に、
ggplot2
もインストールする。
最後に、biocLite パッケージ中の cummeRbund 関数をインストールする。
biocLite については、CRAN Repository には存在しないため、R コンソール上でソースコードのある場所を以下のように指定する必要がある。
source("http://bioconductor.org/biocLite.R")
その後、
biocLite("cummeRbund")

で cummeRbund パッケージと、さらに依存関係にあるパッケージをインストールする。その際に、boot 等のいくつかのパッケージが古く、R コンソール上で

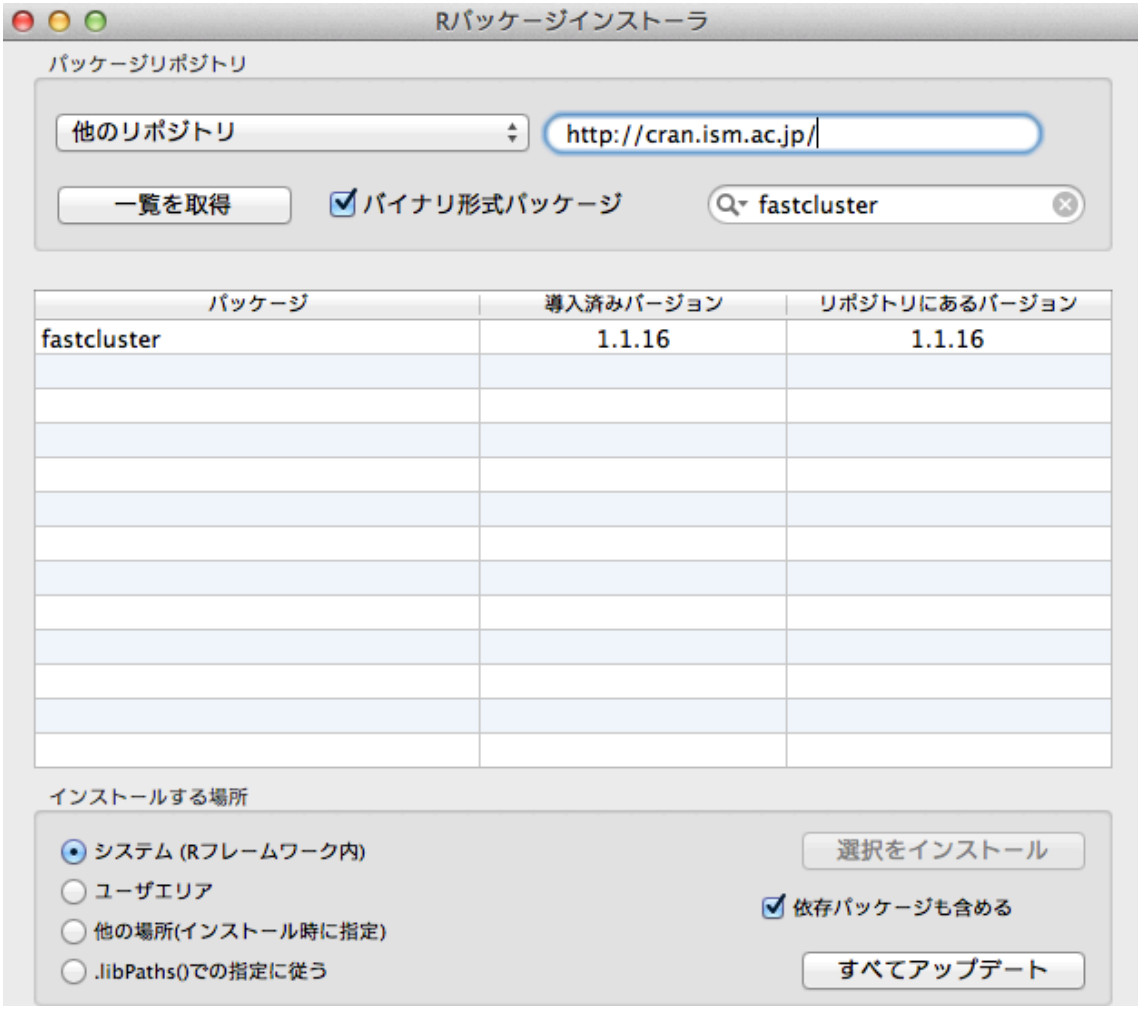
Update all/some/none?

と聴かれたら、

a

と入力してアップデートしておく。

RStudio でのパッケージインストールがうまくいかない場合は、良くあるパターンとしては、自分のコンピュータのユーザ名にスペースが入っていたり、日本語が入っていたりする場合などがある。
そのような場合は、対処が困難なので、RStudio を使わず、R そのものを使うことになる。
R 本体のみを使う場合は、パッケージのインストールは、
メニューのパッケージとデータ → パッケージインストーラ で下記の R パッケージインストーラを立ち上げ、他のリポジトリを選択し、http://cran.ism.ac.jp/ を入力し、例えば fastcluster をインストールしたい場合はパッケージ名を下記のように指定して「一覧を取得」ボタンを選択することでパッケージを選択し、依存パッケージも含めるチェックボックスを選択して、「選択をインストール」を選択すれば個々のパッケージを R 本体にインストールできる。



[3] RStudio を用いた基本的な統計量の算出およびグラフ描画

(4) RStudio でのプロジェクト作成

RStudio 上では、プロジェクトを作って R コンソール上に入力した R のコマンドを保存することが出来る。まずは、右上の Project:(None)と表示されている部分を選択して、New Project を選択する。

すると、New Project window が表示されるので、New Directory を選択して新しいディレクトリを作成する。Project Type を選ぶ必要があるので、Empty Project を選択してディレクトリの名前を入力する。ディレクトリの名前は、コマンドラインからも使うことを考えると、スペース等の特殊文字を含めない英

数字で記述すべきである。
ディレクトリを作る場所も、Browse ボタンを選択して指定する。
それらの記述および指定が終わったら、Create Project を選択して、Project を作成する。
これで、Project の作成が出来た。
Project を作ることで、R のコマンド履歴等がこのプロジェクト内に保存されるようになる。

(5) RStudio での行列データの入力

それでは、実際に RStudio 上でデータを入力して簡単な統計量の計算を行う。左側の R コンソールは R のコマンドを入力する Window なので、ここに 5 行 2 列の行列データを、例えば以下のように入力する。

```
x <- matrix(c(12, 2, 5, 2, 3, 11, 31, 44, 12, 34), nrow=5, ncol=2, byrow=T)
```

Enter を押すと、問題無ければ右上の Environment Window 中に、変数 x の中身が表示される。右端の表アイコンを選択すると、直感的に分かりやすい形で表データが表示される。
データを R コンソール上で入力しなくても、テキストファイル等を Environment window の Import Dataset から読み込むことも可能である。
1 行目のみ出力したい場合は、

```
x[1,]
```

1 列目のみ出力したい場合は、

```
x[,1]
```

1 列目の 1 行目から 3 行目までを出力したい場合は、

```
x[1:3,1]
```

(6) 平均・中央値などの基本的な統計量の計算とグラフ描画

先ほど入力した行列 x について、平均値や中央値などの基本的な統計量を知りたい場合には、

```
summary(x)
```

で最小値・中央値・平均値・最大値等を知ることが出来る。

基本的な統計量はわかったが、データの分布を視覚的に見てみたい。
まずは散布図を作成する。

```
plot(x)
```

次にヒストグラムを作成する。これは、2 列を分けて描く必要がある。
まず 1 列目(V1 列)

```
hist(x[,1])
```

次に 2 列目(V2 列)

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

最後に、Boxplot を作成する。

```
boxplot(x)
```

Boxplot の場合、太い線が中央値、箱の両端の線が第 1/第 3 四分位数、箱の上下の横線が基本的には最大値と最小値、白丸が外れ値となっている。
R の箱ひげ図の詳しい説明は、http://bio-info.biz/tips/r_boxplot.html
等が参考になる。
これらの図は、右の Plots Window の Export の Save as...で PNG や PDF 等の形式で保存できる。

[4] 分割表データを用いた統計検定

次に、これら 2 群のデータセットの間に統計的な有意差があるか否かを検定する。今回のデータを、例えばコントロールと処理区の 2 群で得られた各遺伝子の発現量だと仮定する。

両者の間でどの遺伝子が発現量に統計的に有意な差があるかを見たい場合には、遺伝子ごと、つまりは行ごとに比較する必要がある。
その場合の帰無仮説は、
2 群間で遺伝子の発現量に差が無い
となる。

例えば 1 行目の遺伝子について検定をしたい場合には、その遺伝子の発現量と、他の全ての遺伝子の発現量の和の間の比率がどの程度 2 群間で極端であるかを検定する。
2 行目から 5 行目までの列ごとの和(y,z)は以下のコマンドで求まる。

```
y <- sum(x[2:5,1])
```

z <- sum(x[2:5,2])

求めた上記の値を用いて、検定用の 2 行 2 列の行列データを作成する。

x2 <- matrix(c(x[1,1],x[1,2],y,z), nrow=2, ncol=2, byrow=T)

x2 の値を確認する。

x2

これで統計検定を行う準備が整った。

今回のような 2×2 の分割表のデータで有意差検定を行う際には、Fisher‘s Exact test (Fisher の正確確率検定)が一般的に用いられている。Fisher‘s Exact test では、分割表において、1 行目、2 行目、V1 列目、V2 列目のそれぞれの合計値を変えずに分割表の 4 つの値を変更する全パターンの中で、観察データよりもさらに偏りの激しいデータが得られる確率の大きさを問題にする（下表)。その確率が十分小さければ、2 群は偶然とは考えにくいほど異なっているとして、有意差ありとする。

	Sample 1	Sample 2	合計
遺伝子 1	12	2	14
他の遺伝子合計	51	91	142
合計	63	93	312

さらに極端な可能性としては、

	Sample 1	Sample 2	合計
遺伝子 1	13	1	14
他の遺伝子合計	50	92	142
合計	63	93	312

などが存在。

RStudio において Fisher‘s Exact test を行うコマンドは以下。

fisher.test(x2)

一般的には p-value が 0.05 よりも小さければ、有意差ありと判定する。
ここで、p-value とは、日本語では有意水準と言い、どの程度偶然でその差が起こりうるかを表す確率であり、その検定のエラー率であるとも言える。つまり、通常はエラー率 5%を閾値として、群間の有意差の有無を判定する。

分割表の検定については、Fisher's Exact test 以外にも、χ^2 独立性の検定などがある。χ^2 独立性の検定は以下のコマンドで実行可能

```
chisq.test(x2)
```

χ^2 独立性の検定は分割表のどれかの値に 0 があったり値が極端に小さかったりする場合には（具体的には各項目の期待値が 5 未満）、検定結果が不正確になるので、その場合は Fisher's Exact test を用いるべきである。

[5] 多重性の問題

RNA-Seq などの大規模解析の場合、非常にたくさんの遺伝子について発現量の統計検定を行う必要がある。その際にも上記の Fisher's exact test による有意差検定は広く用いられているが、数千・数万個もの遺伝子に対して上記の検定を行う場合には、多重性の問題が発生する。

先ほど p-value は単一の検定のエラー率であると述べたが、複数回検定を繰り返した場合には、個別の検定は 5%に有意水準を設定してエラー率を管理出来ているように見えても、どれか一回の検定で間違う確率は 5%よりもかなり大きくなる。この、多数回検定を繰り返す場合に、検定全体のどれかで間違う確率(familywise error rate)が個別の検定で設定した有意水準よりも大きくなってしまう問題を多重性の問題と呼ぶ。

(7) Bonferroni 補正

多重性の問題に対応する単純な方法は、0.05 等の個々の検定の有意水準の値を、検定回数で割って得られた値を有意性の判定に用いる有意水準とする方法である。これは、Bonferroni 法と呼ばれ、familywise error 調整済み有意水準を簡単に計算できるため、広く用いられている。

例えば、個別の検定の有意水準を 0.05 にしていた場合は、検定を 10 回繰り

返す場合には調整済み有意水準は 0.005 になり、個々の検定で p-value が 0.005 よりも小さいか否かで、有意差があるか否かの判定が行われる。

(8) False Discovery Rate

Bonferroni 補正でも多重性の問題には対応できるが、検定回数で元々の個々の検定の有意水準の値を割る手法はあまりにも厳しすぎ、数千・数万個もの遺伝子に対して検定を繰り返す場合には、本当は差があるのに検定回数が多すぎるために有意差は無いという問題が頻繁に起こりうる。

そこで、familywise error rate の代わりに、有意差ありと判定された結果の中に本当は差が無いものを含む確率 (False Discovery Rate)を一定の水準以下にする、False Discovery Rate (FDR)を p-value の代わりに用いる手法が広く用いられている。

FDR の計算方法にはいくつか種類があるが、一番基本的な Benjamini Hochberg 法では、各検定で得られた p-value を小さい順に並べて、N/順位を p-value にかけることで個別の検定における FDR を求める。
多くの場合、FDR は p-value と区別するために q-value と呼ばれる。
以下が、2 回の検定における q-value を求める方法である。
まずは、2 つめの分割表を x のデータから作成する

```
y2 <- sum(x[1:4,1])
z2 <- sum(x[1:4,2])
x3 <- matrix(c(x[5,1],x[5,2],y2,z2), nrow=2, ncol=2, byrow=T)
```

x2 と x3 それぞれ Fisher's exact test で p-value を計算し、x2p および x3p という変数に代入する。

```
a <- fisher.test(x2)
x2p <- a$p.value
b <- fisher.test(x3)
x3p <- b$p.value
```

p-value リストを基にそれぞれの検定の q-value を求める。

```
q <- p.adjust(c(x2p,x3p), method="fdr", 2)
```

小さい方の p-value が q-value になると 2 倍された値になっていることがわかる。

[6] 多変量解析

RNA-Seq の配列データを配列解析することで得られた大規模な遺伝子発現の組成データは、2 サンプルの比較であれば上記の統計検定が有効であるが、多サンプルになると、サンプル間の全体的な組成の違いを単純なグラフ等では要約することが難しい。そこで、多数の変数からなるデータを要約したり、パターンを抽出する統計手法である、多変量解析の様々な手法を用いることが必須となる。

ここでは、多数の遺伝子の発現量を少数の要約値にまとめることにも使用可能な、主成分分析を用いてデータを要約する。主成分分析における情報の圧縮は、概念的には、次ページの図のように説明出来る。

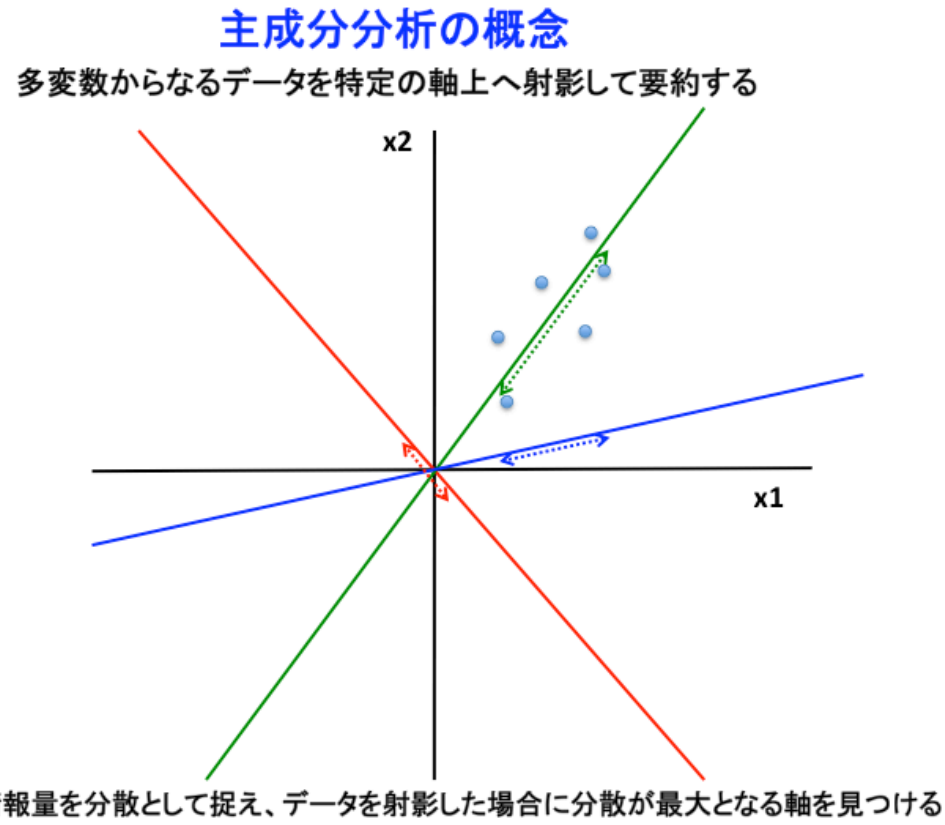


図. 主成分分析における次元圧縮の概念. 図の場合では、2 次元のデータを 1 次

元に圧縮する場合を表している。

それでは、実際に主成分分析を R で行ってみる。
テストデータとして、下記のようなデータを 4 サンプル 5 遺伝子の発現データと仮定して R に入力する。

```
a <- matrix(c(12, 2, 5, 2, 3, 11, 31, 44, 12, 34, 12, 1, 3, 1, 2, 1, 2, 3, 4, 5), nrow=5, ncol=4, byrow=F)
```

R では、主成分分析はいくつかのコマンドで実行出来るが、本実習では、prcomp 関数を用いて行う。

```
a.pca <- prcomp(t(a))
```

t()は行列の行と列を入れ替える（転置する）関数である。
これで、主成分分析の計算は出来た。
では、各主成分の寄与率を見てみる。
寄与率とは、次ページのように、その主成分で説明出来る、元のデータが持つ情報量のことである。

次元圧縮による情報の損出の定量

例: (50, 10), (20, 40)の2データを y = x1 + x2 軸上へ射影すると、どちらも60になる

主成分分析では、データが持つ情報量を分散で捉えている。

全主成分の分散の和 = 元の多変数のデータが持つ分散の和

主成分の分散は、多変数のデータの分散共分散行列の固有値

例: 第1主成分の分散 λ_1 が全体の分散のどの程度を説明出来ているか？

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \ldots + \lambda_n}$$

寄与率 = ある主成分の分散がデータ全体の分散のどの程度を説明出来ているか

寄与率を用いることで、主成分分析で何個かの主成分に次元を圧縮した場合に、
どの程度元のデータが持つ情報が損失しているかは、定量可能
基本的に、変数間の相関が高いほど、第1主成分の寄与率は高くなる

R における各主成分の寄与率は、あるデータについての様々な統計量を計算してくれる、summary 関数で表示することが出来る。

```
summary(a.pca)
```

Proportion of Variance が各主成分（PC）の寄与率である。

各主成分の、サンプルごとの値を見る

```
a.pca$x
```

各サンプルの第一主成分の値と第二主成分の値をグラフ化してみる。

```
plot(a.pca$x, type="n")
text(a.pca$x,labels=c("1","2","3","4"))
```

各主成分と各遺伝子の発現量間の相関を計算し、各主成分の意味を推定。

```
cor(a.pca$x, t(a))
```

このように、R を用いれば、RNA-Seq の塩基配列データを遺伝子ごとの発現量データに変換した後の様々な統計解析を、コマンド入力で行うことができ、非常に便利である。

参考文献

・内田治、西澤英子　R による統計的検定と推定　オーム社 2012

R で統計検定を行う上で必要なコマンドが一通り紹介されている。

・市原清志　バイオサイエンスの統計学　南江堂 1990

各統計検定手法が非常にわかりやすく説明されている。