

1.2.1

Rの基本的動作はユーザとの会話的なものであり、入力した式(含コマンド)に対して結果を返し、

これを繰り返すものである。

また、必要に応じて図を表示する。

Rは可能な限りエラーを起こさない形で動作し、式に対する結果を出してくる。よって結果が出ない、ということはないが、Rによる動作にユーザの意図との齟齬がないか、注意を要する。

この式は「オブジェクト」として機能する。オブジェクトとは通常のプログラミングでいう「変数」

の意。数学的な変数と区別するため、Rではオブジェクトと称し、場合に応じた可変の値を割り当てて使うことができる。ただしRで用いることのできるオブジェクトの種類は、「数値」「複素数」「文字列」「論理値」である。

1.2.2

ほとんどの関数ではデフォルトの引数があり、指定しなくても一般的な条件で処理される。そのため、必要に応じてユーザは自分で処理のセッティングを行う。プロットのマーク、ラインの幅、項目名などが指定できる。

デフォルト

```
> plot(height, weight)
```

軸指定

```
> plot(y=height, x=weight)
```

```
> plot(x=height, y=weight)
```

プロットマーク変更

```
> plot(height, weight, pch=2)
```

デフォルト値の確認 (NULL 値は設定なし)

```
> args(plot.default)
```

```
function (x, y = NULL, type = "p", xlim = NULL, ylim = NULL,
```

```
log = "", main = NULL, sub = NULL, xlab = NULL,
```

```
ylab = NULL,
```

```
ann = par("ann"), axes = TRUE, frame.plot = axes, panel.first = NULL,
```

```
panel.last = NULL, col = par("col"), bg = NA, pch = par("pch"),
```

```
cex = 1, lty = par("lty"), lab = par("lab"), lwd = par("lwd"),
```

```
asp = NA, ...)
```

```
NULL
```

1.2.3

ベクトル: 論理数、実数、文字列などのデータの系列。コマンド'c'の概念は、いくつかの値を結合しひとつの固まりとして扱うことで、ベクトルにその値を当てはめベクトルを作る。またベクトルとベクトルを結合する。

文字列データ(Character vector)はダブルクォート、もしくはシングルクォートで囲む。

```
> c('Huey', 'Dewey', 'Louie')
```

```
[1] "Huey" "Dewey" "Louie"
```

論理数(Logical)はTRUEとFALSE、それぞれT、Fと略することができる。

```
> c(T, T, F, T)
```

```
[1] TRUE TRUE FALSE TRUE
```

論理演算

```
> bmi > 25
```

```
[1] FALSE FALSE FALSE FALSE TRUE FALSE
```

1.2.4

不定データ: 欠損値、桁溢れ、0による除算などの結果は'NA'と表記される。

NAに対する演算結果はすべてNA

```
> NA
```

```
[1] NA
```

1.2.5

ベクターを作る関数：c

実数のベクターを通常どおり作る (c)

```
> c(42, 57, 12, 39, 1, 3, 4)
[1] 42 57 12 39 1 3 4
```

実数範囲指定 (seq)

```
> seq(4, 9)
[1] 4 5 6 7 8 9
> seq(2, 12)
[1] 2 3 4 5 6 7 8 9 10 11 12
```

もしくは

```
> 2:12
[1] 2 3 4 5 6 7 8 9 10 11 12
```

3つめの引数は「いくつおきの」を指定する (偶数、奇数が指定可能)。

```
> seq(2, 50, 4)
[1] 2 6 10 14 18 22 26 30 34 38 42 46 50
> seq(4, 10, 2)
[1] 4 6 8 10
```

繰り返し (rep)

```
> oops<-c(7, 9, 13)
> oops
[1] 7 9 13
> rep(oops, 3)
[1] 7 9 13 7 9 13 7 9 13
> rep(oops, 4)
[1] 7 9 13 7 9 13 7 9 13 7 9 13
```

複数の要素に対する繰り返し回数

```
> rep(1:2, c(10, 5))
[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
> rep(1:2, c(2, 10))
[1] 1 1 2 2 2 2 2 2 2 2 2
> rep(1:3, c(2, 10, 2))
[1] 1 1 2 2 2 2 2 2 2 2 2 3 3
```

1.2.6

行列、配列：ベクトルを列行の指定をして二次元の表として表示

ベクトル x に次元の属性を与え、三行四列の配列として表示。

```
> x<-1:12
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12
> dim(x)<-c(3, 4)
> x
      [1] [2] [3] [4]
[1,]  1  4  7 10
[2,]  2  5  8 11
[3,]  3  6  9 12
```

行数を指定して表を作る。引数 byrow は、行幅指定

```
> matrix(1:12, nrow=3, byrow=T)
      [1] [2] [3] [4]
[1,]  1  2  3  4
[2,]  5  6  7  8
[3,]  9 10 11 12
```

x にマトリクス (行列) を変数として指定

```
> x<-matrix(1:12, nrow=4)
> x
      [1] [2] [3]
[1,]  1  5  9
[2,]  2  6 10
[3,]  3  7 11
[4,]  4  8 12
```

1.2.7

factor (型) : 数値データとして処理されているベクトルの数値に「この値はなにを意味するものか」という情報 (class 属性) を付加する。分類変数、名義尺

度などに使用。

痛みの程度を表すデータの例

```
> pain<-c(0, 3, 2, 2, 1)
> pain
[1] 0 3 2 2 1
```

ベクトル pain の数値幅を「fpain」として設定

```
> fpain<-factor(pain, levels=0:3)
```

fpain の数値の名義を、四種の程度に設定

```
>
levels(fpain)<-c("none", "mild", "medium", "severe")
```

```
> fpain
```

```
[1] none severe medium medium mild
Levels: none mild medium severe
```

fpain を数値データとして表示

```
> as.numeric(fpain)
[1] 1 4 3 3 2
```

1.2.8

リスト：オブジェクトをまとめたオブジェクト、値のまとめかたのひとつとして多く利用される。

2つのベクトルがある場合

```
> intake.pre<-c(5260, 5470, 5640, 6180, 6390, +
6515, 6805, 7515, 7515, 8230, 8770)
```

```
> intake.pre
```

```
[1] 5260 5470 5640 6180 6390 6515 6805 7515 7515
8230 8770
```

```
>
```

```
intake.post<-c(3910, 4220, 3885, 5160, 5645, 4680, 5265, 5975, 6790, 6900, 7335)
```

```
> intake.post
```

```
[1] 3910 4220 3885 5160 5645 4680 5265 5975 6790
6900 7335
```

```
>
```

```
mylist<-list(before=intake.pre, after=intake.post)
```

```
> mylist
```

```
$before
```

```
[1] 5260 5470 5640 6180 6390 6515 6805 7515 7515
8230 8770
```

```
$after
```

```
[1] 3910 4220 3885 5160 5645 4680 5265 5975 6790
6900 7335
```

表示する時の行の名義は設定自由

```
> mylist<-list(A=intake.pre, B=intake.post)
```

```
> mylist
```

```
$A
```

```
[1] 5260 5470 5640 6180 6390 6515 6805 7515 7515
8230 8770
```

```
$B
```

```
[1] 3910 4220 3885 5160 5645 4680 5265 5975 6790
6900 7335
```

リストから1行だけ取り出す場合

```
> mylist$B
```

```
[1] 3910 4220 3885 5160 5645 4680 5265 5975 6790
6900 7335
```

1.2.9

データフレーム：様々な種類の値の用いて多次元配列とする。

```
> d<-data.frame(intake.pre, intake.post)
```

```
> d
```

```
intake.pre intake.post
1      5260      3910
2      5470      4220
3      5640      3885
4      6180      5160
```

```

5      6390      5645
6      6515      4680
7      6805      5265
8      7515      5975
9      7515      6790
10     8230      6900
11     8770      7335

```

#データフレームから一部の要素を取り出す

```

> d$intake.pre
[1] 5260 5470 5640 6180 6390 6515 6805 7515 7515
8230 8770
> d$intake.pre[5]
[1] 6390

```

1.2.10

行列データの索引。たとえばこのデータからの索引

```

> intake.pre
[1] 5260 5470 5640 6180 6390 6515 6805 7515 7515
8230 8770

```

```

> intake.pre[5]
[1] 6390
> intake.pre[c(3, 5, 7)]
[1] 5640 6390 6805

```

索引の条件をベクトルとして設定

```

> v<-c(3, 5, 7)
> intake.pre[v]
[1] 5640 6390 6805

```

「～を除く」指定

```

> intake.pre[-1]
[1] 5470 5640 6180 6390 6515 6805 7515 7515 8230
8770

```

1.2.11

条件選択：データの固まりからデータ呼び出すための条件として、論理式などの設定ができる

データフレームを例とした論理式条件

```

> d
      intake.pre intake.post
1         5260         3910
2         5470         4220
3         5640         3885
4         6180         5160
5         6390         5645
6         6515         4680
7         6805         5265
8         7515         5975
9         7515         6790
10        8230         6900
11        8770         7335

```

```

> intake.post[intake.pre>7000]
[1] 5975 6790 6900 7335

```

用いることのできる条件

<, > (大なり小なり)

<=, >=

== (イコール)

!= (not イコール)

& (且つ)

| (または)

! (not)

#例

```

> intake.pre[intake.pre>6000 & intake.pre<8000]
[1] 6180 6390 6515 6805 7515 7515

```

この条件は、論理式条件としてもOK

```

> intake.pre>6000 & intake.pre<8000
[1] FALSE FALSE FALSE TRUE TRUE TRUE TRUE
TRUE TRUE FALSE FALSE

```

1.2.12

データフレームからの索引法

#以下のデータフレームを例とする

```
> d
  intake.pre intake.post
1      5260      3910
2      5470      4220
3      5640      3885
4      6180      5160
5      6390      5645
6      6515      4680
7      6805      5265
8      7515      5975
9      7515      6790
10     8230      6900
11     8770      7335
```

5行一列

```
> d[5, 1]
[1] 6390
```

5行目

```
> d[5, ]
  intake.pre intake.post
5      6390      5645
```

2列目

```
> d[, 2]
```

1列目で7000以上の値の行

```
> d[d$intake.pre>7000, ]
  intake.pre intake.post
8      7515      5975
9      7515      6790
10     8230      6900
11     8770      7335
```

条件を変数に設定可能

```
> sel<-d$intake.pre>7000
> d[sel, ]
  intake.pre intake.post
8      7515      5975
```

```
9      7515      6790
10     8230      6900
11     8770      7335
```

1.2.13

データの変形、変更方法。

#既存のデータ thuesen を開く

```
> data(thuesen)
```

このデータを変形

```
> thuesen
  blood.glucose short.velocity
1          15.3          1.76
2          10.8          1.34
3           8.1          1.27
4          19.5          1.47
5           7.2          1.27
6           5.3          1.49
7           9.3          1.31
8          11.1          1.09
9           7.5          1.18
10         12.2          1.22
11          6.7          1.25
12          5.2          1.19
13         19.0          1.95
14         15.1          1.28
15          6.7          1.52
16          8.6           NA
17          4.2          1.12
18         10.3          1.37
19         12.5          1.19
20         16.1          1.05
21         13.3          1.32
22          4.9          1.03
23          8.8          1.12
24          9.5          1.70
```

```
> thue2<-subset(thuesen, blood.glucose<7)
```

```
> thue2
```

```

blood. glucose short. velocity
6          5.3          1.49
11         6.7          1.25
12         5.2          1.19
15         6.7          1.52
17         4.2          1.12
22         4.9          1.03

```

元データに新たに処理データを加え変形する

```

>
hue3<-transform(thuesen, log. gluc=log (blood. gluc
ose))
> thue3
  blood. glucose short. velocity log. gluc
1          15.3          1.76 2.727853
2          10.8          1.34 2.379546
3           8.1          1.27 2.091864
4          19.5          1.47 2.970414
5           7.2          1.27 1.974081
6           5.3          1.49 1.667707
7           9.3          1.31 2.230014
8          11.1          1.09 2.406945
9           7.5          1.18 2.014903
10         12.2          1.22 2.501436
11          6.7          1.25 1.902108
12          5.2          1.19 1.648659
13         19.0          1.95 2.944439
14         15.1          1.28 2.714695
15          6.7          1.52 1.902108
16          8.6           NA 2.151762
17          4.2          1.12 1.435085
18         10.3          1.37 2.332144
19         12.5          1.19 2.525729
20         16.1          1.05 2.778819
21         13.3          1.32 2.587764
22          4.9          1.03 1.589235
23          8.8          1.12 2.174752
24          9.5          1.70 2.251292

```

1. 2. 14

データの塊とデータフレーム

```

> data (energy)
> energy
      expend stature
1    9.21  obese
2    7.53  lean
3    7.48  lean
4    8.08  lean
5    8.09  lean
6   10.15  lean
7    8.40  lean
8   10.88  lean
9    6.13  lean
10   7.90  lean
11  11.51  obese
12  12.79  obese
13   7.05  lean
14  11.85  obese
15   9.97  obese
16   7.48  lean
17   8.79  obese
18   9.69  obese
19   9.68  obese
20   7.58  lean
21   9.19  obese
22   8.11  lean

```

このデータに「stature」ベクターによる条件付けを加える場合…

```

>
exp. lean<-energy$expend [energy$stature=="lean"]
>
exp. obese<-energy$expend [energy$stature=="obese"]
> exp. lean
 [1]  7.53  7.48  8.08  8.09 10.15  8.40 10.88
    6.13  7.90  7.05  7.48  7.58  8.11

```

```
> exp.obese
[1] 9.21 11.51 12.79 11.85 9.97 8.79 9.69
9.68 9.19
```

関数 split を使うことで

```
> l <-split (energy$expend, energy$stature)
> l
$lean
[1] 7.53 7.48 8.08 8.09 10.15 8.40 10.88
6.13 7.90 7.05 7.48 7.58 8.11
```

```
$obese
[1] 9.21 11.51 12.79 11.85 9.97 8.79 9.69
9.68 9.19
```

1.2.15

ソート (並べ替え)

単純な昇順ソート

```
> intake.post
[1] 3910 4220 3885 5160 5645 4680 5265 5975 6790
6900 7335
> sort (intake.post)
[1] 3885 3910 4220 4680 5160 5265 5645 5975 6790
6900 7335
```

データ数値の順位

```
> order (intake.post)
[1] 3 1 2 6 4 7 5 8 9 10 11
```

1.2.16

決まった形のくり返し処理 (apply)

複数の行、列を持つオブジェクトに対し、「それぞれ各行毎に」コマンドを与える。

各列の平均値を求める。ここでは thuesen を 2次元 (列) で平均

```
> thuesen
```

```
blood.glucose short.velocity
1 15.3 1.76
2 10.8 1.34
3 8.1 1.27
4 19.5 1.47
5 7.2 1.27
6 5.3 1.49
7 9.3 1.31
```

```
> apply (thuesen, 2, mean)
blood.glucose short.velocity
10.3 NA
```

NA 値が入っているのでそれを除く場合

```
> apply (thuesen, 2, mean, na.rm=T)
blood.glucose short.velocity
10.300000 1.325652
```

コマンド「lapply」は「list」に表記し、「sapply」の場合は「simple」に結果を表示する。この場合「二次元」指定は必要ない

```
> sapply (thuesen, mean, na.rm=T)
blood.glucose short.velocity
10.300000 1.325652
```

```
> lapply (thuesen, mean, na.rm=T)
$blood.glucose
[1] 10.3
```

```
$short.velocity
[1] 1.325652
```

IV. グラフの書き方

1. 相関図

全部のペアを見る

```
> pairs (iris)
```

```
<img src=soukan_all.jpeg>
```

特定のペアを見る

```
> plot (x, y)
```


2. ヒストグラム

hist (オブジェクト名) で作成

> hist (height)

V. 補足概念

1. 値の取り扱い方

R では値を以下の型に分けている。

- ・ 数値
- ・ 複素数
- ・ 文字データ
- ・ 論理値

この型はベクトルや配列を構成した時にも反映される。

2. データフレームの特徴

値の固まりをつくるベクトルやリストは、値の種類が異なるものを混在させることはできない。

データフレームではこの規制がなく、様々なかたちのデータを含めて多次元配列とすることができる。

3. ベクトル

もっとも基本的なデータオブジェクトの形。最小単位のオブジェクトであり、データフレーム等を多次元的と考えるならば、ベクトルは一次元的ということが出来る。

4. 配列

ベクトルに次元属性をプラスしたオブジェクト。数値ベクトルがそのままであっても、10個の数値は配列とした場合「1行10列」の多次元属性を持つ。

5. リスト

複数のデータオブジェクトを、並べて結合させたもの。よって結合するそれぞれのオブジェクトの値の形式

に違いがあっても、リスト化することは可能である。多様なデータを1つの塊にする、もっとも基本的な方法。

VI. genetics Package

パッケージ”genetics” : ゲノムデータを解析処理する上で、必要なユーティリティ集。

主要な機能としては、遺伝子型のクラス、また haplotype クラスを設定することで、各対立遺伝子に対しアクセス

が簡便で、おのおのを変数として扱えるようになっている。

1. 遺伝子型オブジェクト

遺伝子型をオブジェクトとして扱う場合は、「genotype」を用いる。以下のどの方法を用いても同じ要素として認識される。

#スラッシュ区切り、空白区切りで認識

```
> g1<-genotype(c('A/A','A/C','C/C','C/A'))
```

```
> g1
```

```
[1] "A/A" "C/A" "C/C" "C/A"
```

```
Alleles: C A
```

```
> g3<-genotype(c('A A','A C','C C','C A'), sep=' ', remove.spaces=F)
```

```
> g3
```

```
[1] "A/A" "C/A" "C/C" "C/A"
```

```
Alleles: C A
```

#区切りのない場合

```
> g2<-genotype(c('AA','AC','CC','CA'), sep=1)
```

```
> g2
```

```
[1] "A/A" "C/A" "C/C" "C/A"
```

```
Alleles: C A
```

#二つのベクトルにデータがまたがる場合

```
>
```

```
g4<-genotype(c('A','A','C','C'),c('A','C','C','A'))
```



```
> g4
[1] "A/A" "C/A" "C/C" "C/A"
Alleles: C A
```

#データフレーム等にあるデータ

```
>
gm<-cbind(c("A", "A", "C", "C"), c("A", "C", "C", "A"))
)
```

```
> gm
      [1] [2]
```

```
[1.] "A" "A"
[2.] "A" "C"
[3.] "C" "C"
[4.] "C" "A"
```

```
> g5<-genotype(gm)
```

```
> g5
[1] "A/A" "C/A" "C/C" "C/A"
Alleles: C A
```

2. 主な機能

HWE.test :

ハーディ・ワインバーグ平衡

(Hardy-Weinbergequilibrium) に基づく検定を行う。

homozygote と heterozygote :

homo と heter の組み合わせがそれぞれ対立遺伝子上にあるかどうかを示す論理的なベクトルを作成。

```
> heterozygote(X2)
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
TRUE FALSE FALSE FALSE FALSE
[13] FALSE TRUE FALSE TRUE TRUE TRUE FALSE
FALSE FALSE TRUE TRUE FALSE
[25] TRUE FALSE TRUE TRUE FALSE TRUE FALSE
FALSE FALSE TRUE TRUE FALSE
```

```
> homozygote(X2)
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE
FALSE TRUE TRUE TRUE TRUE
[13] TRUE FALSE TRUE FALSE FALSE FALSE TRUE
TRUE TRUE FALSE FALSE TRUE
```

```
[25] FALSE TRUE FALSE FALSE TRUE FALSE TRUE
TRUE TRUE FALSE FALSE TRUE
```

carrier :

指定の遺伝子がベクトル中に存在するかどうかの論理的ベクトルを返す。

```
> carrier(X2, "T", na.rm=TRUE)
```

```
[1] FALSE FALSE FALSE TRUE FALSE FALSE FALSE
TRUE FALSE FALSE FALSE FALSE
[13] FALSE TRUE FALSE TRUE TRUE TRUE FALSE
TRUE FALSE TRUE TRUE TRUE
[25] TRUE FALSE TRUE TRUE TRUE TRUE FALSE
TRUE FALSE TRUE TRUE FALSE
```

allele :

対立遺伝子を個別に読み出す。列を指定すれば取り出せる。

```
> allele(X2)
```

```
      [1] [2]
```

```
[1.] "C" "C"
[2.] "C" "C"
[3.] "C" "C"
[4.] "T" "T"
[5.] "C" "C"
```

(略)

```
attr(,"which")
```

```
[1] 1 2
```

```
attr(,"allele.names")
```

```
[1] "C" "T"
```

```
> allele(X2, 2)
```

```
[1] "C" "C" "C" "T" "C" "C" "C" "T" "C" "C" "C"
"C" "C" "T" "C" "T" "T" "T" "C"
[20] "T" "C" "T" "T" "T" "T" "C" "T" "T" "T" "T"
"C" "T" "C" "T" "T" "C"
```

```
attr(,"which")
```

```
[1] 2
```

```
attr(,"allele.names")
```

```

[1] "C" "T"
>
allele.count :
指定した対立遺伝子の数を数える

> allele.count (X2, "C")
[1] 2 2 2 0 2 2 2 1 2 2 2 2 2 1 2 1 1 1 2 0 2 1
1 0 1 2 1 1 0 1 2 0 2 1 1 2
attr(,"allele")
[1] "C"

```

3. Sample

```

#csv ファイルを読み込む
> tpmt<-read.csv("tpmt.csv")
> tpmt
  X  X1  X2  X3  X4  X6  X7  X8  X9 X10 X11 X12
1  2 C/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
2  3 C/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
3  4 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
4  5 T/T T/T C/T G/G A/T A/G A/C C/T A/G A/T T/T
5  6 C/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
6  8 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
7  11 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
8  12 C/T C/T C/T T/G A/T A/G A/C C/T A/G A/T T/T
9  13 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
10 14 C/T C/C C/T G/G A/A G/G C/C T/T A/A T/T T/T
11 15 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
12 16 T/T C/C C/C T/T A/T G/G C/C T/T A/A T/T T/T
13 18 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
14 19 C/T C/T C/T T/G A/T G/G C/C T/T A/A T/T T/T
15 20 T/T C/C C/C T/T A/A A/G A/C C/T A/G A/T T/T
16 21 C/T C/T C/T T/G A/T G/G C/C T/T A/A T/T T/T
17 22 C/T C/T C/T T/G A/T G/G C/C T/T A/A T/T T/T
18 23 C/T C/T C/T T/G A/T G/G C/C T/T A/A T/T T/T
19 24 T/T C/C C/C T/G A/A A/G C/C C/T A/G T/T T/T
20 25 T/T T/T T/T G/G T/T G/G C/C T/T A/A T/T T/T
21 26 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T
22 27 C/T C/T C/C T/G A/T A/G A/C C/C G/G A/T T/T
23 29 C/T C/T C/T T/G A/T A/G A/C C/T A/G A/T T/T
24 30 T/T T/T T/T G/G T/T A/G A/C C/T A/G A/T T/T

```

```

25 31 C/T C/T C/C T/G A/A A/G A/C C/T A/A A/A T/T
26 32 T/T C/C C/C T/T A/A G/G C/C T/T A/G T/T T/T
27 34 C/T C/T C/T G/G T/T G/G C/C C/T A/G T/T T/T
28 35 C/T C/T T/C T/G A/T G/G C/C T/T A/A T/T T/T
29 36 T/T T/T C/T G/G A/T A/G A/C C/T A/G T/T T/T
30 37 T/T C/T T/T G/G T/T G/G C/C T/T A/A T/T T/T
31 38 T/T C/C C/C T/G A/A G/G C/C T/T A/A T/T T/T
32 42 T/T T/T T/T G/G T/T A/G A/C C/T A/G A/T T/T
33 44 T/T C/C C/C T/T A/A G/G C/C T/T A/A A/T T/T
34 45 C/T C/T C/T G/G A/T A/G C/C C/T A/G A/T C/T
35 47 C/T C/T C/T T/G A/T A/G A/C C/T A/G A/T T/T
36 48 T/T C/C C/C T/T A/A G/G C/C T/T A/A T/T T/T

```

#X2 の列の対立遺伝子群を、X2 の変数に入れる

```

> X2<-genotype(tpmt$X2)
> X2
[1] "C/C" "C/C" "C/C" "T/T" "C/C" "C/C" "C/C"
"C/T" "C/C" "C/C" "C/C" "C/C"
[13] "C/C" "C/T" "C/C" "C/T" "C/T" "C/T" "C/C"
"T/T" "C/C" "C/T" "C/T" "T/T"
#[25] "C/T" "C/C" "C/T" "C/T" "T/T" "C/T" "C/C"
"T/T" "C/C" "C/T" "C/T" "C/C"
Alleles: C T

```

この列に対して Hardy-Weinberg 均衡による検定を行う。

```
> HWE.test(X2)
```

```

-----
Test for Hardy-Weinberg-Equilibrium
-----

```

Call:

```
HWE.test.genotype(x = X2)
```

Raw Disequilibrium for each allele pair (D)

	C	T
C		-0.03684414
T	-0.03684414	

Scaled Disequilibrium for each allele pair (D')

			Overall	D	-0.03684414	(-0.11111111,
					0.03722994)	0 YES
	C	T	Overall	D'	-0.16947649	(-0.50000000, 0.43750000)
C	-0.1694765				0	YES
T	-0.1694765		Overall	r	0.16947649	(-0.16425532,
					0.50000000)	0 YES
Correlation coefficient for each allele pair (r)						
	C	T				
	C	1.0000000	0.1694765			
	T	0.1694765	1.0000000			
Overall Values						
Value						
D	-0.03684414					
D'	-0.16947649					
r	0.16947649					

Significance Test:
 Pearson's Chi-squared test with simulated p-value
 (based on 10000
 replicates)

data: X2
 X-squared = 1.034, df = NA, p-value = 0.2597

Confidence intervals computed via bootstrap using
 1000 samples

Observed	95% CI	NA's
Contains Zero?		

(資料)

厚生労働科学研究費補助金

(小児疾患臨床研究:小児疾患臨床研究分野 若手医師・協力者活用等に要する研究チーム)

総合研究報告書

薬理遺伝学検査の運用

山岸敬幸 (慶應義塾大学医学部小児科 専任講師)

研究要旨 本研究班では、抗けいれん薬クロバザムについて「CYP2C19 の遺伝子多型による薬剤反応性の予測に関する研究」および抗凝固薬ワーファリンについて「CYP2C9 の遺伝子多型による薬剤反応性の予測に関する研究」を行った。インフォームドコンセントを実施、検体の匿名化、検体管理の方法、検査等における問題点を検討した。その上で、日本人に認められる TPMT*3C 多型の患者情報を臨床で利用可能 (6MP やアザチオプリンの投与前に TPMT 多型解析を利用して副作用を事前回避する試み) にする為、迅速な TPMT*3C 多型スクリーニングシステムを確立した。これまで本研究で構築してきた薬理遺伝学検査を運営するまでの基盤整備により、慶應義塾大学病院外からの依頼にも対応している。迅速で安価な薬理遺伝学検査を実施しうるシステムを構築し、治療前遺伝子検査を実施しうる体制を整備した。

研究協力者

山岸千尋	慶應義塾大学医学部小児科	医師
田村和代	慶應義塾大学医学部小児科	医師
鮫島葉月	慶應義塾大学医学部小児科	研究員
高橋大輔	慶應義塾大学医学部小児科	研究員

T活性を低下させることが知られる遺伝子多型である。小児リンパ芽球性白血病 (ALL) 治療薬、6MP (6メルカプトプリン) を投与する際、TPMT*3C 多型を持つ患者においては、6MP の抗腫瘍効果は増大し、通常の投与量によっても強い骨髄毒性を来す。既に欧米においては、6MP の投与前に TPMT の遺伝子多型の検査が行われ、遺伝子多型に応じて、初期投与量が調節されている。

A. 研究目的

本研究班では抗けいれん薬クロバザムについて「CYP2C19 の遺伝子多型による薬剤反応性の予測に関する研究」、抗凝固薬ワーファリンについて「CYP2C9 の遺伝子多型による薬剤反応性の予測に関する研究」を行ってきた。インフォームドコンセントを実施、検体の匿名化、検体管理の方法、検査等における問題点を明らかにした。

これら院内の薬理遺伝学検査における基盤整備をもとに、院外からの受諾検査として、TPMT*3C ジェノタイプ検査を行った。

TPMT*3C ジェノタイプは、日本人において TPM

この TPMT*3C 検査は、院外からの受諾検査も含めた実際の臨床に応用可能なシステムの構築を目的としており、特に*3C ジェノタイプの検査結果を迅速に検査依頼医へ報告し治療に貢献することを目的とした。そのための簡便で安価な検査方法を DHPLC 法により確立し、現場での検査の運用を行った。

B. 研究方法

1. 対象

CYP2C19—慶應義塾大学病院小児神経外来に受診しクロバザム内服中の患者

CYP2C9—慶應義塾大学病院小児心臓外来に受診しワーファリンを内服中または過去にワーファリンの投与を受けた患者

TPMT*3C—2004年から2005年まで、獨協医科大学より送付された11検体が対象。過去に維持療法で6MPを投与、あるいは今後投与する患者。

2. インフォームドコンセントの実施

(院内受諾検査)

診療時間内に主治医が研究の概要についての5分間ほど説明を行った。その後、遮音された別室の中で、研究者（主治医と異なる者）より『「CYP2C19の遺伝子多型による薬剤反応性の予測に関する研究」の説明』（添付資料）に沿って、患児の親または患児に詳細な説明を行った。患児が幼少である場合は、同室内のベッドまたは、両親のいずれかの膝の上で遊んでいた。説明は平易な言葉を用い、特にメリットとデメリットについて理解できるまで説明を行った。研究者からの説明を行った後、両親からの質問を受けた。質疑応答を含めて20～30分であった。質疑応答後、同意書の説明を行った。父親または母親の同意が得られた場合に署名を頂いた。母親のみで来院している場合、一旦帰宅し父親と話し合い、後日サインした承諾書を持って来院する場合もあった。未成年の患者の場合は、法的な保護者からも同意書を得た（小児心臓外来では長期的なフォローアップを要する者が多く、患者が成人である場合も多かった）。本年度は、未成年の患者のみでの来院された方はいなかった。診療目的で血中濃度測定やINR値を測定する日にあわせて採血研究のための採血を行った。本研究のみを目的とした採血は行わないように配慮した。

(院外受諾検査)

依頼時は各医療機関にて、「チオプリンメリル転移酵素 (TPMT) の遺伝子多型による薬剤反応性の予測に関する研究」の試行について説明の上、本人および家族の同意を得た。所定の書式1～3（添付資料1～3）および同意書、患者血液を送付することで検査依頼を受諾した。

3. 検体管理と個人情報保護

(院内受諾検査)

小児科外来で採血を行った。採血後、情報管理者が連結可能匿名化を行い、個人情報の保護に努めた。この匿名化の詳細については、分担研究報告書「薬理遺伝学的検査の実施における個人情報匿名化に関する研究」を参照のこと。①患者の個人情報、カルテの情報等は担当の臨床医のもとでのみ管理される。②検査のため提供された患者血液は、情報管理者の手により採血スピッツの名前ラベルを剥離する。③匿名化専用ソフトウェアを用い、検査用のプロジェクト名と乱数からなる識別番号を付与する。④研究担当者および検査技師は、情報管理者から識別番号を付与された血液を受け取り、検査を実施する。⑤研究担当者は検査報告書を作成する際に、実名を記載せず識別番号を利用する。⑥検査報告書を臨床担当医へ進展として届ける。①から⑥のプロセスを踏むことにより本検査から検査結果を得るまでの間、検査結果と患者本人の個人情報と分離された状態にあった。

(院外受諾検査)

依頼元の各医療機関では、検体、および同意書に関し匿名化をすることが検査要件である。検査検体は報告終了時まで一貫して匿名コードのまま扱われ、患者を特定する個人情報とは結びつけられることはない。

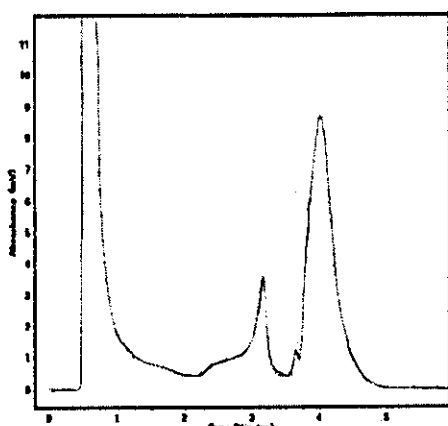
4. 遺伝子多型検査方法

患者から提供された血液からDNAを抽出(50ng/ml)、これを鋳型として以下の目的SNPを含む範囲をPCR法で増幅。DHPLC法および

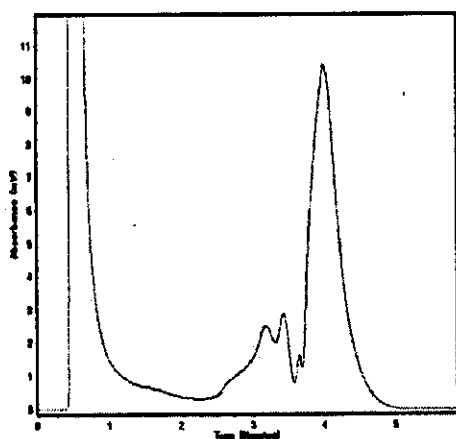
ダイレクトシーケンス法によって、ジェノタイプピ
ングを行った（詳細は平成 14 年度報告書参照）。
ダイレクトシーケンス法に比べ DHPLC 法は、コ
スト・時間の面から有利であり、DHPLC 法のみ
による検査体制を確立する計画である。本研究で
はバリデーションの目的で、両方の方法を利用し
た。

DHPLC 法 (Wave system) による
クロマトグラム

a) CYP2C19

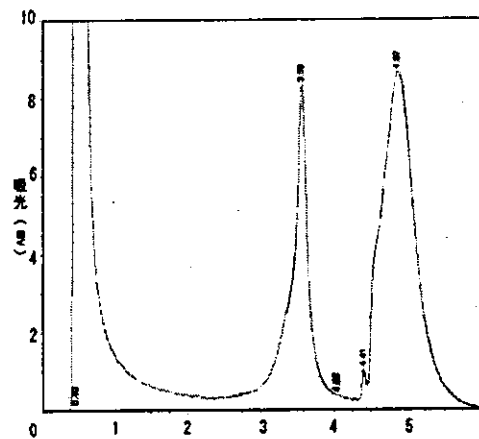


CYP2C19 健常型(*1/*1)

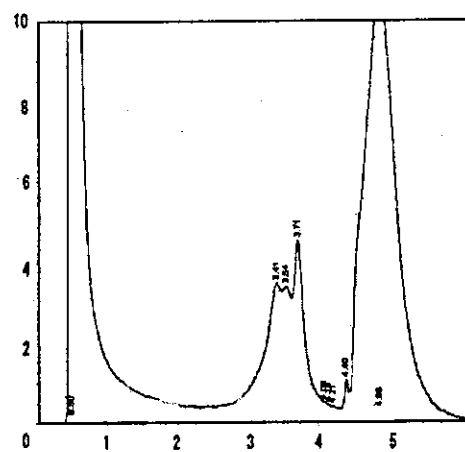


CYP2C19 多型(*1/*2)

b) CYP2C9

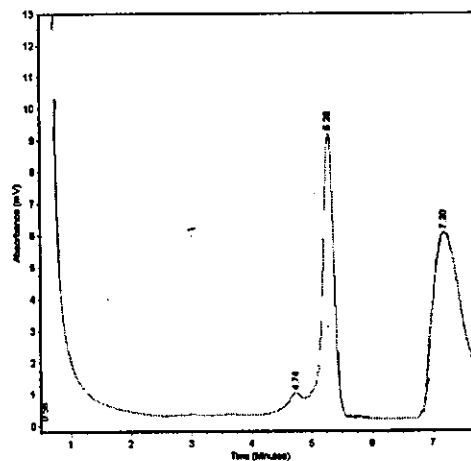


CYP2C9 健常型(*1/*1)

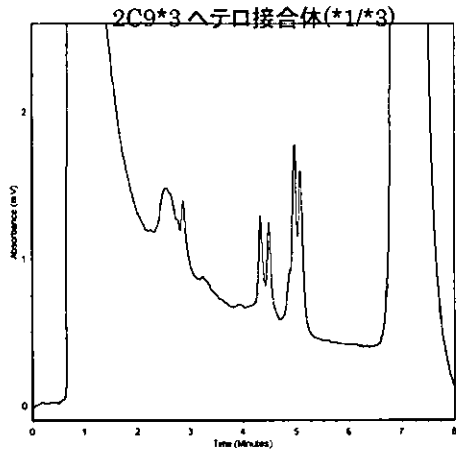


CYP2C9 多型(*1/*3)

c) TPMT*3C



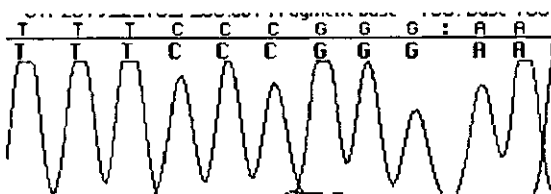
TPMT 健常型(*1/*1)



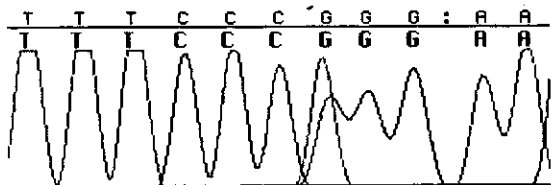
TPMT*3C 多型(*1/*3)

直接シーケンシング法による
クロマトグラム

a) CYP2C19

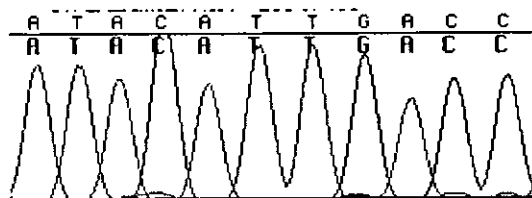


CYP2C19 健常型(*1/*1)

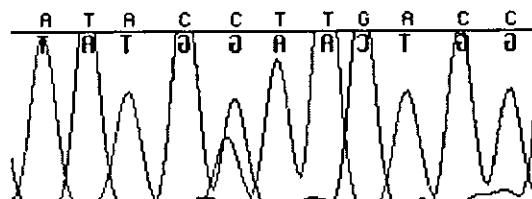


CYP2C19*2 ヘテロ接合体(*2/*1)

b) CYP2C9

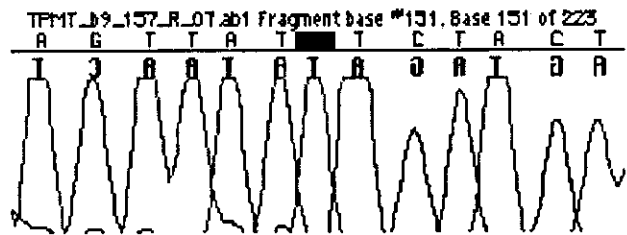


CYP2C9 健常型(*1/*1)

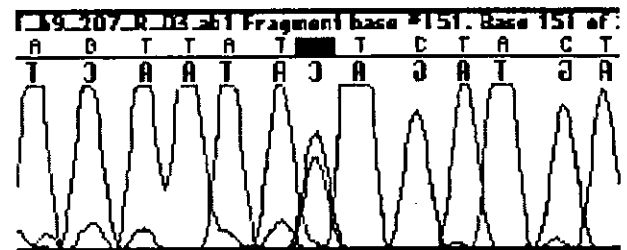


CYP2C9*3 ヘテロ接合体(*3/*1)

c) TPMT*3C



TPMT 健常型(*1/*1)



TPMT*3C ヘテロ接合体(*1/*3)

5. 遺伝子検査結果報告書

(院内受諾検査)

本検査の報告書は結果の開示を求めた患者に対して、臨床医の説明のもと患者本人に手渡されるものであり、患者本人および家族における結果のよりよい理解のために、平易な文章を以て作成された(添付文書)。検査前の説明と同様に、個室の中で結果の説明を行った。変異アレルを有した患者および両親に対しては、異常という言葉を使用しないように留意し、あくまで体質の一つであるという点を強調して説明を行った。患児の検査の結果以外に、今回の研究全体の成績について興味深く質問する両親が多数いた。

(院外受諾検査)

検査依頼時、患者が今後 6MP を投与する事が明記されている場合、検査によって今後の治療に支障が出ないように至急の対応をし、結果返却まで 1 週間~10 日の日数で行った(表 1)。

C. 研究結果

1. インフォームドコンセント

<同意率>

クロバザム研究では、23名中、21名より同意を得た。2名からは同意を得られなかった。同意を得られなかった2名のうち1名は、患児の母が遺伝子及びテーラーメイド医療についての理解を深めた後、再度説明を受けたいとの希望があった。もう1名は、遺伝子という言葉に対する嫌悪感を感じたと本人（成人）からの説明があった。同意した両親のうちの約5名からは、テーラーメイド医療に関する今後の研究、また、今回の研究と関連がない他の研究にも協力したいという申し出があった。また、今後の研究に対する興味、期待を持った両親が多数見られた。91%という高い同意を得られた理由として1) 患児が幼少である場合、今後の患児の治療に対する期待及び興味が大きいと思われた。2) 患児が精神運動発達遅延である場合、同様の患者に対する親近感、責任を両親が感じていることが多いと感じられた。3) クロバザムの内服により症状が軽快している場合、今後安心して内服できることを期待していると思われた。4) 検査がルーチンの検査と一緒にされる為、負担が少ない。5) 大学病院に受診している以上、研究に協力することを当然と考えているという意見もあった。

ワーファリン研究では、36名中、33名から同意を得た（92%）であった。比較的高い回収率を得られた理由として、1) 毎日かつ長期的に飲まなくてはならない薬剤であり、薬効が生命予後に関わる薬であるため、薬の効き方に対して関心が高い。2) 定期的な採血検査と同時に検体の採取を行った。3) 原病が重症であり、遺伝子検査など最新の医療に関心が高い、などの意見があった。同意書が得られなかった3名のうち、1名からは、研究に協力したくないという意思表示があった。残りの2名は、同意書を自宅に持ち帰り、その後返答がなかった。

2. 残存検体の長期保存について

同意書における同意項目における「遺伝子解析研究終了後の残存検体の長期保存・および他の医

学研究への使用」について説明し、長期保存を望まなかった患者および両親はクロバザム研究では1名、ワーファリン研究では3名であった。同意されている検体に関してのみ、匿名化の方針のもとに保管した。これ以外の患者検体は、検査終了後直ちに破棄した。

3. 結果の開示

（院内受諾検査）

クロバザム研究では、遺伝子検査を受けた21名中全員が結果の開示を求めた。ワーファリン研究では、結果の開示を求めた患者は36名中33名であった。ヘテロの変異について、臨床的な意義が確立していない。

（院外受諾検査）

受諾した検査は、以下の期間で依頼元へ返却が可能であった。

表1・検査依頼から検査終了までの日数

	受諾日	返却日	至急返却要	返却日数
D-1	2004 10/13	11/4		22
D-2	10/13	11/4		22
D-3	10/27	11/2	○	6
D-4	10/28	11/2		5
D-5	11/4	11/10		6
D-6	11/4	11/10		6
D-7	11/10	11/17	○	7
D-8	11/17	11/27	○	10
D-9	11/17	11/27		10
D-10	12/1	12/7	○	6
D-11	2005 2/16	3/3		16

検査結果返却日数 平均 10.5日

D. 考察

院内受諾検査においては、インフォームドコンセント時、全体として研究の参加を呼びかけた患

者の 90%以上から同意を得た。大部分の患者が individual feedback を求めている。これらの傾向は、一般小児科外来を受診する患者を対象としたアンケート調査と一致した。結果の開示にあたり、臨床的な意義が確立していない場合に、注意を要した。例えば、ワーファリン内服患者を対象とした CYP2C9 多型と服用量の関係についてのメタ解析では、ヘテロ接合体では、維持投与量が野生ヘテロ接合体の半量程度となる。しかしながら、この成績をもとに個人の至適予測投与量を予測することは現時点では、困難である。

クロバザム研究の場合は、1年間の研究により、ホモ接合体においてデスメチル・クロバザムの血中濃度が上昇する傾向を証明したが、このような研究プロジェクト全体の成果を患者に戻すための説明資料の作成が必要と考えられた。今後、CYP2C9やCYP2C19の多型を有する患者で強い副作用を来す薬剤が同定された際に、これらの患者に recontact してほしいかどうか、事前に両親・患者の希望を確認することが望ましい。

院外受諾検査については、検査主目的である「治療方針に有効な検査結果を、迅速に医療機関に提供する」点で貢献可能であった。

現在、平均日数 10 日で遺伝子検査結果が臨床応用可能であった。。

E. 結論

薬理遺伝学検査における、インフォームド・コンセント、検体の匿名化、検体の管理、検体検査、結果のフィードバックを一貫して行いうるシステムを確立した。迅速で安価な薬理遺伝学検査を実施しうるシステムを構築し、治療前遺伝子検査を実施しうる体制を整備した。

F. 健康危険情報

特記すべきことなし。

G. 研究発表

<論文発表>

Kosaki K, Tamura K, Sato R, Samejima H, Tanigawara Y, Takahashi T. A major influence of CYP2C19 genotype on the steady-state concentration of N-desmethyloclobazam. *Brain and Development*, in press.

Shimada H, Mori T, Shimasaki N, Shimizu K, Takahashi T, Kosaki K. Somatic PTPN11 mutation with a heterogeneous clonal origin in children with juvenile myelomonocytic leukemia. *Leukemia*, in press.

<学会発表>

Kosaki K, Utaka T, Samejima H, Fujita H, Yahagi N, Takahashi T. The COPPER plate system: Effective use of DHPLC for clinical genetics and pharmacogenetics. *American College of Medical Genetics Annual Clinical Genetics Meeting*, March 2004 Orlando, FL, USA

小崎健次郎、前山克宏、菅谷明則、百々秀心、山岸敬幸、高橋孝雄

CYP2C19*3 ヘテロ接合体患者におけるワーファリン投与量のメタ解析
第 107 回日本小児科学会学術集会、平成 16 年 4 月、岡山

H. 参考文献

Itoh K, Inoue K, Nakao H, Yanagiwara S, Tada H, Suzuki T. Polymerase chain-reaction-single-strand conformation polymorphism based determination of two major genetic defects

responsible for a phenotypic polymorphism of cytochrome P450 (CYP) 2C19 in the Japanese population.

Anal Biochem. 2000 Aug 15;284(1):160-2.

Fukuda T, Nishida Y, Zhou Q, Yamamoto I, Kondo S, Azuma J.

The impact of the CYP2D6 and CYP2C19 genotypes on venlafaxine pharmacokinetics in a Japanese population.

Eur J Clin Pharmacol. 2000 May;56(2):175-80.

Gaikovitch EA, Cascorbi I, Mrozikiewicz PM, Brockmoller J, Frotschl R, Kopke K, Gerloff T, Chernov JN, Roots I.

Polymorphisms of drug-metabolizing enzymes CYP2C9, CYP2C19, CYP2D6, CYP1A1, NAT2 and of P-glycoprotein in a Russian population.

Eur J Clin Pharmacol. 2003 Aug;59(4):303-12.
Epub 2003 Jul 15.

Scordo MG, Pengo V, Spina E, Dahl ML, Gusella M, Padrini R.

Influence of CYP2C9 and CYP2C19 genetic polymorphisms on warfarin maintenance dose and metabolic clearance.

Clin Pharmacol Ther. 2002 Dec;72(6):702-10.

Borlak J, Thum T.

Identification of major CYP2C9 and CYP2C19 polymorphisms by fluorescence resonance energy transfer analysis.

Clin Chem. 2002 Sep;48(9):1592-4.

厚生労働科学研究費補助金

(小児疾患臨床研究:小児疾患臨床研究分野 若手医師・協力者活用等に要する研究チーム)

総合研究報告書

ジアゾキシドの母集団薬物動態試験法の検討

谷川原祐介 (慶應義塾大学医学部教授 薬剤部長)

研究要旨 ジアゾキシドは、高インスリン血性低血糖症の治療薬として使用されている薬剤であるが、我が国においては未承認である。本薬剤の適正使用を図るためには、血中薬物濃度を測定し薬物動態を検討することが必要である。しかし、小児患者が対象でしかも多くは新生児や乳児であるため、多数回の採血を必要とされる標準的な薬物動態試験の実施は困難である。そこで母集団薬物動態 (PPK) 解析が有用と考えられる。本研究では、実施可能な例数で PPK 解析が行えるかどうか、シミュレーションの手法を用いて検討した。Multiple trough screen を想定し、32 人の患者から一人 2 点の血中濃度を測定できると仮定した。1-コンパートメントモデルにあてはめ、NONMEM にて擬似的に血中濃度を発生させ、その擬似データに基づいて解析するというステップを 200 回繰り返して、得られたパラメータの正確さと推定精度を確認した。その結果、投与間隔内で一人 2 点の血中濃度データが得られれば、採血時点を特定しなくても、同様の精度で CL が推定できた。服薬記録の誤差による影響は、誤差の大きさに伴い大きくなることから、採血前の服薬時間は正確に記録することが望ましい。可能であれば一人 3 点以上血中濃度データが得られれば、さらに推定精度が高まると期待できる。

研究協力者

田村和代 慶應義塾大学医学部小児科 医師
佐藤玲子 慶應義塾大学医学部薬剤部 助手

試験デザインの中で採血時間を厳密に決めなくても、採血時刻と投薬時刻の正確な記録さえあれば解析できる点でも有用である。また、医薬品情報として最終的に必要なのは、患者一人一人のパラメータではなく、その薬剤が適用される集団における特性値、すなわち母集団パラメータである。

A. 研究目的

適切な薬物治療のためには、その薬剤の用法・用量を明確にすることが重要である。薬物の多くは薬物動態 (PK) と薬効との間に密接な関連を示すので、血中薬物濃度を測定し薬物動態を検討することは、用法・用量の合理的な設定や臨床試験の計画を行う上で不可欠である¹⁾。しかし、小児からの採血は量的にも回数的にも制限がある上、外来治療を受けている患者を対象とした場合、採血時間帯も限られる。そこで、一人あたりの採血回数を減らす最も有用なアプローチは、母集団薬物動態 (PPK) 解析である。その上 PPK 解析では、

対象薬物は、高インスリン血性低血糖症の第一選択薬として使用されているジアゾキシドとした。ジアゾキシドは米国、欧州などで血圧降下薬および小児の低血糖症に対する適応があり、日本では少なくとも 71 人の患者で使用されている²⁾。しかし、本薬剤は我が国で未承認のため、用法・用量の設定根拠としての安全で有効な薬用量について科学的エビデンスがほとんどない状況にある。

そこで本研究では、ジアゾキシドの PPK 解析

を行うための、試験デザインおよび解析方法について検討することを目的とした。

B. 研究方法

想定される症例数および採血時点から得られた血中濃度データから、目的の解析結果がどの程度の精度で得られるか、コンピューター・シミュレーションの手法を用いて検討した。本薬剤は半減期に比べ投与間隔が非常に短く、しかも内服投与であることから、投与後のピーク血中濃度とトラフ血中濃度の差が捉えにくいことが予想される。さらに投与期間が長期にわたるので、ほとんどの患者で定常状態に達していると考えられるため、薬物動態スクリーニング法(図1)として multiple trough screen つまり、一人あたり2点以上のトラフ血中濃度データを得てクリアランスを求めるというデザインを想定した。シミュレーションは、薬物動態モデルとPKパラメータを用いてNONMEM³⁾ ver.Vにて擬似的に血中濃度

データを発生させ、その擬似データに基づいて解析を行うというステップを200回繰り返して行った。PKパラメータは、単回急速静注における成人高血圧患者から一人あたり15点以上採血し2-コンパートメントモデルで解析して薬物動態パラメータを求めた報告⁴⁾と、内服における小児の低血糖症患者のありそうな血中濃度⁵⁾を仮定したデータに基づき予備解析を行った結果を参考に決定した。実施可能な例数を32例と仮定し、解析が行えるかどうか、また解析できた場合、パラメータの推定精度をより高くするためにどのような条件が必要か、以下の項目について検討した。

1. モデルの検討
2. 採血時点の検討
3. 投与間隔の違いが薬物動態パラメータの予測に影響するかどうかの検討
4. 服薬記録の誤りが薬物動態パラメータの予測にどのように影響するかの検討
5. 一人あたりの採血点数の検討

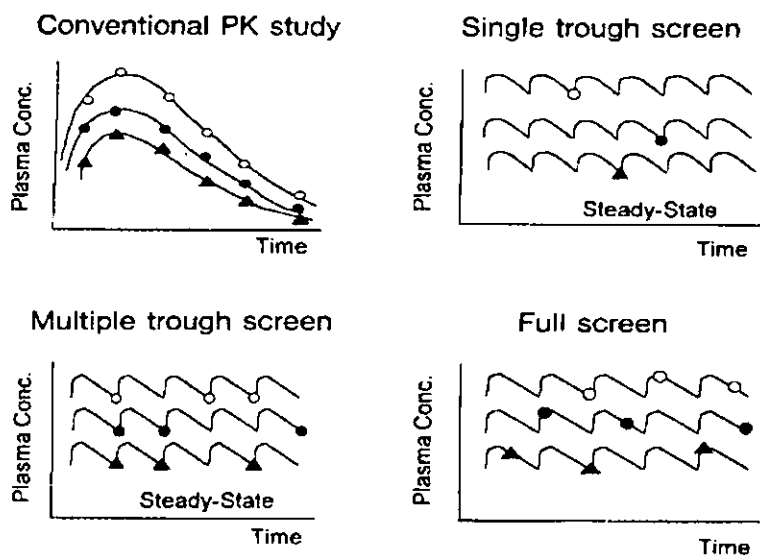


図1. 薬物動態スクリーニング方の概念