model.matrix() 和 formula()简述

Linear combinations：线性组合简而言之就是假设y=f(x1)+f(x2)，那么y就是x1和x2的线性组合。假如两个变量包含一样的相同信心，那么model matrix将会产生一致的列

假设以老鼠饮食对应老鼠体重为例子, 模型公式为：

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon, i = 1, \ldots, N$$

这里的Y对应老鼠体重，x对应老师接受的饮食变量，通过n个不同的饮食变量实验，可得以上线性回归方程。然后使用矩阵乘法来表示以上公式：

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \text{ and } \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

as:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

or simply:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

这里的**X**就是我们design matrix。

design matrix的选择对于线性回归模型非常重要，因为它将指定哪些系数将会用来构成最终的回归模型，以下例子来自DESeq2。

DESeq2说明中matrix所包含变量：

```
> colData(dds)
DataFrame with 7 rows and 6 columns
           condition        type number.of.lanes total.number.of.reads
            <factor>    <factor>       <integer>              <factor>
treated1     treated single-read               5              35158667
treated2     treated  paired-end               2         12242535 (x2)
treated3     treated  paired-end               2         12443664 (x2)
untreated1 untreated single-read               2              17812866
untreated2 untreated single-read               6              34284521
untreated3 untreated  paired-end               2         10542625 (x2)
untreated4 untreated  paired-end               2         12214974 (x2)
           exon.counts        sizeFactor
             <integer>         <numeric>
treated1      15679615  1.63557509657607
treated2      15620018 0.761269768042316
treated3      12733865 0.832652635328833
untreated1    14924838  1.13826297659084
untreated2    20764558  1.79300035535039
untreated3    10283129 0.649547030603726
untreated4    11653031 0.751689223426488
```

- 单个变量的design matrix为：

```
> model.matrix(~condition,colData(dds))
           (Intercept) conditiontreated
treated1             1                 1
treated2             1                 1
treated3             1                 1
untreated1           1                 0
untreated2           1                 0
untreated3           1                 0
untreated4           1                 0
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$condition
[1] "contr.treatment"
```

这里我们使用线性模型来比较不同的condition，那么根据字母排列顺序(这里认为设定了)，untreated将会成为ref level。在design matrix中第一列的Intercept为1，第二列指定了哪些样本将会出现在第treated condition中。这样就有两个系数出现在线性模型中：the intercept表示untreated condition(first level, ref level)的均值；第二个系数，代表了treated condition和untreated condition的均值之间的差异。第二个系数就是我们感兴趣的，将会执行统计检测的稀释；通过统计检测，我们将知道2个condition间是否存在差异。

以上对应的回归模型公式为：

Y = β_0 + β_1(treated) +ϵ

colnames(model.matrix(..)): Intercept, conditiontreated

对应理解，Intercept表示ref状态均值；conditiontreated，表示conditiontreated时的coefficient，对应ref为conditionuntreated，可得二者比较，conditiontreated/conditionuntreated

`resultsNames(dds): "Intercept", "condition_treated_vs_untreated"`

- 当出现2个变量(~ condition + type)时，condition的ref为untreated，type的ref为paired-end，回归模型公式为：

$$Y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \varepsilon_i$$

此时的design matrix为：

```
> model.matrix(~condition+type,colData(dds))
           (Intercept) conditiontreated typesingle-read
treated1             1                1               1
treated2             1                1               0
treated3             1                1               0
untreated1           1                0               1
untreated2           1                0               1
untreated3           1                0               0
untreated4           1                0               0
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$condition
[1] "contr.treatment"

attr(,"contrasts")$type
[1] "contr.treatment"
```

这样回归模型公式：

Y = β_0 + β_1 * X_1(treated) + β_2 * X_2 (single-end) + ϵ

colnames(model.matrix(...)): Intercept, conditiontreated, typesingle-read

对应理解，Intercept为ref状态均值(untreated, paired-end)；conditiontreated，表示为不考虑type情况下，conditiontreated的coefficient，可得比较conditiontreated/conditionuntreated；typesingle-read，表示为可涉及condition时，typesingle-read的coefficient，可得比较typesingle-read/typepaired-end

`resultsNames(dds): "Intecept", "condition_treated_vs_untreated", "type_single.read_vs_paired.end"`

当出现交互项时，就是当前状态提供了额外限制条件，
model.matrix(~condition+type+condition:type, colData(dds))就等同于
model.matrix(~condition*type, colData(dds))

```
> model.matrix(~condition*type, colData(dds))
           (Intercept) conditiontreated typesingle-read
treated1             1                1               1
treated2             1                1               0
treated3             1                1               0
untreated1           1                0               1
untreated2           1                0               1
untreated3           1                0               0
untreated4           1                0               0
           conditiontreated:typesingle-read
treated1                                  1
treated2                                  0
treated3                                  0
untreated1                                0
untreated2                                0
untreated3                                0
untreated4                                0
attr(,"assign")
[1] 0 1 2 3
attr(,"contrasts")
attr(,"contrasts")$condition
[1] "contr.treatment"

attr(,"contrasts")$type
[1] "contr.treatment"
```

尚未理解！！！

1. Y = β_0 + β_1 * X_1(treated) + β_2 * X_2 (single-end) + β_3 * X_3(treated:single-end) + ϵ

colnames(model.matrix(...))：Intercept， conditiontreated，typesingle-read，conditiontreated:typesingle-read，最后一个就出现了typesingle-read给conditiontreated添加了type限定条件。理解为：conditiontreated:typesingle-read/conditionuntreated

`resultsNames(dds): "Intercept", "condition_treated_vs_untreated", "type_single.read_vs_paired.end", "conditiontreated.typesingle.read"`

- 连续变量

I()：In function ‘formula’. There it is used to inhibit the interpretation of operators such as “'+'”, “'-'”, “'*'” and “'^'” as formula operators, so they are used as arithmetical operators.

```
tt <- seq(0,3.4,len=4)
model.matrix(~ tt + I(tt^2))
```

```
##   (Intercept)       tt   I(tt^2)
## 1           1 0.000000  0.000000
## 2           1 1.133333  1.284444
## 3           1 2.266667  5.137778
## 4           1 3.400000 11.560000
## attr(,"assign")
## [1] 0 1 2
```