·学术交流·

流行病学研究中数字型 ID 编码的校验

徐涛

在流行病学研究中存在大量个体样本，每个样本含有不同类型的研究内容，如样本的流行病学信息、实验室检测信息和用药信息等，这些信息存在于同一样本的不同调查表和不同的研究现场。为了确保个体样本信息的惟一性、关联性和易管理性，研究者大多采用十进制数字作为编码元素编制统一的 ID 编码，然后分配给每个研究对象作为其惟一的身份标识。但在实际工作中，处理这些十进制数字时常出现识别、书写和录入错误。这些错误主要表现为：①单个数字识别错误，如把 3 误认为 8；②相邻数字之间顺序颠倒错误，如 23 误认为 32，223 误认为 233；③数字跳跃错误，如 235 误认为 532，525 误认为 252；④遗漏一位或增加一位数字错误等。这些错误一般出现在调查表的填写和录入阶段。如果在流行病学研究中出现 ID 编码错误，会造成个体样本信息无法识别，信息无法利用。特别是在多中心流行病学研究中，同一个研究对象的数据来自不同的现场、实验室和医院等多个研究中心，当某个 ID 编码出现问题时，相同研究对象的数据无法连接，信息难以交换和利用。

为了尽量避免数据输入阶段出现 ID 编码错误，通常在程序设计时，对 ID 编码字段设置最小值和最大值的逻辑检查方式。这种方式只能够解决上述④中的错误，对符合逻辑检查范围合法值的其他类“错误”难以发现。例如原编码为 53230201 被误认为是 53320201 或 53230501 等类似错误。以下介绍一种十进制编码差错校验方法，以解决实际应用中存在的这类问题。

一、基本原理

1. Verhoeff's两面体群 D_5 校验算法：数学家为了解决十进制数字编码的校验问题，先后提出了多种编码算法。其中，以Verhoeff's的十进制两面体群 D_5 码表校验算法应用较为广泛，该算法能够发现 100% 的单个数字错误和相邻数字之间顺序颠倒错误，95% 的双数字错误，94% 的数字跳跃错误和 100% 的数字缺失和增加数字错误。

Verhoeff's以两面体群 D_5 码表（表 1）和Verhoeff's校验码表（表 2）为基础，提出了以校验等式为零的校验算法。校验等式为：

$$f_0(a_0) \# f_1(a_1) \# f_2(a_2) \# \cdots \# f_{n-1}(a_{n-1}) = 0$$

式中的操作符“#”不是通常意义上的数字运算符，而是组群运算，f_i 表示固定顺序数列的第 i 次迭代。

表1 Verhoeff's 两面体群 D_5 码表

*	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

表2 Verhoeff's 校验码表

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	7	6	2	8	3	0	9	4
2	5	8	0	3	7	9	6	1	4	2
3	8	9	1	6	0	4	3	5	2	7
4	9	4	5	3	1	2	6	8	7	0
5	4	2	8	6	5	7	3	9	0	1
6	2	7	9	3	8	0	6	4	1	5
7	7	0	4	6	9	1	3	2	5	8

2. ID 编码规则：可校验的 ID 编码格式为“ID 编码 = 类标识位 + 顺序号 + 1 个十进制数字校验码”。该编码与普通编码的区别仅在于在编码最右侧增加 1 个十进制数字校验码。该格式编码各部分含义：①类标识位：类标识位表示样本个体的类别，研究者可以根据自己研究需要确定类标识位的长度和类别数，类标识中可以加入“-”以区别子类。例如某项艾滋病流行病学研究中要涉及两个现场和四类人群，其类标识符为：现场标识位 + 人群标识位。②顺序号：是 1，…，n、增量为 1 的正整数，如 0001，0002，…，n，其中 n 为研究者最小子类中所需的最大样本数。③校验码：以“类标识位 + 顺序号”为基础，根据Verhoeff's校验算法由程序计算得到的校验码。

3. ID 编码生成及校验程序：根据Verhoeff's校验法，ID 编码的校验软件包含两部分内容：生成可校验的 ID 编码程序和编码校验程序。

（1）生成可校验的 ID 编码：本程序生成 ID 编码的最右侧的 1 位十进制数字校验码：

```
check_digit {
#定义局部变量
local(@string) = split(//, $_[1]);
local( $status, $i, $j, $k, $m, $rows, $cols);
local(@ip_rows, @ij_rows, %ip_cols, %ij_cols);
#定义两个二维数组码表
```

作者单位：100050 北京，中国疾病预防控制中心公共卫生监测与信息服务中心

```
%ip_cols=定义表2的2维数组
%ij_cols =定义表1的2维数组
for ( $cols=1; $cols <=8; $cols++ ) {
@ip_rows=split(/,/, $ip_cols{$cols});
for( $rows=1; $rows <=10; $rows++ ) {
$ip{$rows, $cols} = $ip_rows[$rows];
}}
for( $cols=1; $cols <=10; $cols++ ) {
@ij_rows=split(/,/, $ij_cols{$cols});
for( $rows=1; $rows <=10; $rows++ ) {
$ij{$rows, $cols} = $ij_rows[$rows];
}}
$k=0;
$m=0;
if( $k == 0) {$status="1,";}
else{$status="0,";}
# 生成校验码
for( $i=0; $i <=9; $i++ ) {
if( $ij{$k+1, $ip{$i+1, ($m%8)+1}+1} == 0)
{$status .= $i;
return( $status);
}}}
```

生成 ID 编码：

```
for( $i= $first; $i <= $last; $i++ );#first是第1个样本的号;last是最后样本的号
$id=sprintf("%0${id_length}d", $i);
$num= $site. $id; #site是ID编码的类标识符,num=ID编码的类标识编号+顺序号
($check, $digit)=split(/,/, &check_digit($num));
print ID "$site-$id-$digit\n";
}
```

(2)ID 编码校验程序：编码校验程序由 D_5 码表二维数组、Verhoeff's 校验码表二维数组和编码校验三部分组成，两个二维数组定义与 ID 编码生成程序相同，此不再重复。以下是 ID 编码校验程序的主要部分：

```
while( $j<length( $cptid)); #cptid是调查表中个案的实际填报的ID编码
$digit=substr( $cptid, $j,1);
if( $digit>=0 & $digit<=9) {
$k=@ij[$k*10+@ip[$digit*8+($m%8)]];
$m++;}
$j++;
}
```

二、实例分析

以中国国际综合性艾滋病研究项目(CIPRA)的一个研究项目为例来说明可校验 ID 编码的使用过程。该研究是以 8 个地区的两类人群为研究对象，估计总样本量为 5500，研究目的是估计两类人群的 HIV-1 新发感染率和性病感染率。该研究由国家性病艾滋病中心等 5 个研究单位组成。各疾病预防控制中心(CDC)承担流行病学现场调查、相关疾病的实验室初筛检测、确认检测和实验室质量控制检测等。每个 CDC 将调查(检测)后得到的数据以调查表的方式通过传真或 Email 传到 CIPRA 项目的数据中心，不同中心同一研究对象的数据是以 ID 编码进行关联。

1. 可校验 ID 编码的处理过程：①生成 ID 编码：CIPRA 数据核心根据研究设计所需样本量，利用 ID 编码生成程序生成各类地区的可校验 ID 编码。例如第 4 类地区所需样本量为 1500，编码生成程序将产生 1500 个 ID 编码。例如 4-0988-9是一个实际使用的编码，编码中 4 表示第四类地区，0988 是第四类地区中第 988 个研究对象，最后的数字 9 是该编码的校验码。②ID 编码使用：数据核心将产生的可校验的 ID 编码提供给该研究项目负责人，项目数据管理员将 ID 编码提供给现场调查人员，现场调查员将编码分配给每个研究对象，需要注意的是每个调查对象只能分配给一个 ID 编码，即 ID 编码是惟一的。③ID 编码的自动校验：当填有可校验 ID 编码的调查表发送到数据核心时，数据核心的数据管理员对收到的调查表进行数据录入，当录入到 ID 编码字段时，编码校验程序自动捕获 ID 编码错误并通知数据管理员，数据管理员根据正确的 ID 编码进行纠正，不能纠正的错误将通知发生错误的研究中心(现场)，并将纠正后的调查表重新传送给数据中心。ID 编码校验程序调用非常简单，当前流行的数据库管理软件都支持这种外部调用。

2. CIPRA 项目使用此类 ID 编码的效果：CIPRA 项目在中国开展了 5 个课题的研究，在这些课题中的研究对象 ID 编码全部采用本文所述的编码方式，在 CIPRA 的 5 年工作中，共处理 ID 编码241 836次，及时发现和纠正现场 ID 编码填写错误 200 多起，数据管理人员错误录入 ID 编码现象更是无法计算。在如此大的错误发生率情况下，未出现因未捕获 ID 编码错误而造成数据无法归档和关联情况，使用效果令人满意。

三、讨论

当前对十进制数字编码的校验还没有一个无差错的编码解决方案，Verhoeff 提出的十进制两面体群 D_5 码表校验算法虽然应用广泛，但也不能发现所有的十进制数字编码错误，所以使用时应注意这一点。但从 CIPRA 项目对 24 万次的 ID 编码校验效果来看，该编码算法还是一个值得信赖和实用的编码方法。

编码错误发现的及时性在流行病学资料处理时很重要，若错误的编码不能及时被发现，当资料处于分析阶段时再被发现，将给纠正错误带来几何级数的工作量，有时甚至无法纠正，这样会给后续的资料分析带来不必要的麻烦。本编码方案能做到“即录即纠”的效果，也就是说，当数据管理员第一次将原始资料录入到数据库时，就能及时捕捉到 ID 编码的错误，在第一时间与现场联系纠正错误编码。

本文所述方法只能在数据录入阶段实现，而不能在现场手工填写时使用，这也是此方法的一个缺憾。另外本文所涉及程序是在 PERL 语言基础上开发的，该程序只能在具有 PERL 语言编码解释器的计算机中运行。PERL 语言编码解释器可以从大部分的软件下载网站下载。基于 Windows 操作系统的 PERL 语言编码解释器是 Active PERL，目前最新版本为 ActivePerl 5.8.8.822。

(收稿日期：2008-01-14)
(本文编辑：张林东)