On preparing a list of random treatment assignments

N.S.S. Magalhães, H.M. de Oliveira, A.J. Alves
E-mail: {nssm,hmo}@ufpr.br
Laboratório de Tecnologia Químico-Farmacêutica - LTQF
Departamento de Farmácia, Universidade Federal de Pernambuco UFPE,
Av. Prof. Arthur Sá, s/n, 50.740-520 Recife-PE BRASIL.

ABSTRACT

This short note presents the foundations of a computer-oriented approach for preparing a list of random treatment assignments to be adopted in randomized controlled trials. Software is presented which can be applied in the earliest stage of bioavailability or bioequivalence assays. This allocation of patients to treatment in clinical trials ensures exactly equal treatment numbers. The check of the randomness properties of an assignment leads to the concept of a "strong randomized list". The new approach introduced in this note is based on thresholds and produces a strong randomized list of treatment assignments.

1. INTRODUCTION

The Pharmaceutical industry has had a huge interest in organizing trials to resolve relevant therapeutic aspects. Clinical trials play a very important role in the evaluation of new issues for drug therapy. It is well recognized that uncontrolled trial potentially provide distorted outcome of the therapy. In contrast, clinical trials with a properly randomized control groups avoid bias and provide a basis for statistical tests. Simple randomization, replacement randomization, biased coin method and random permuted blocks are some approaches normally adopted to generate a list of assignments [1-5]. We present here a computer-oriented approach for preparing a random list of treatment assignments.

2. THE RANDOMIZATION LIST GENERATION

The use of software to produce lists based on a computer-generated sequence of pseudo-random numbers [6] is a fully convenient tool for groups extensively involved in clinical trials. Properly randomized control groups avoid the drawbacks of the simple randomization and the replacement randomization [1-4]. It is therefore helpful to design randomization to ensure similar treatment numbers throughout a trial.

The computer generation of a random assignment introduced in this note is based on an internal (0,1)-uniform distributed random number generator [6]. The list can be used one at a time as patients are registered into the trial. Some care should be taken to avoid repeating the same output list by assuring different lists of assignments in each program run. We can therefore use a clock-dependent seed. The procedure is flexible allowing an experiment design for any number of patients and *any number of groups* with an equal number of patients.

Let N be the number of patients registered into the trial and let g denote the number of patient groups. The uniformed-distributed random variable is denoted by U.
The approach is founded on a choice of a threshold set $\{T_1,T_2,T_3...,T_N\}$.

The key point is that the random sampling is made without reposition. Two lists (vectors) are then considered:

i) A list $\vec{L}$ of assignments $\vec{L}=\{L_1,L_2,L_3,...,L_N\}$ and

ii) A list $\vec{S}$ of survivors $\vec{S}=\{S_1,S_2,S_3,...,S_N\}$.

After the random generation of the list $\vec{L}$, it is divided into g groups namely $G_1=\{L_1,L_2,...,L_{N/g}\}$, $G_2=\{L_{N/g+1},L_{N/g+2},...,L_{2N/g}\}$, ..., $G_g=\{L_{N-N/g+1},...,L_{N-1},L_N\}$ each of them with cardinality $\|G_i\|=N/g$.

We begin with an empty list of assignments, $\vec{L}=\{\ \}$, and a full list of survivors, $\vec{S}=\{1,2,3,...,N\}$. Pick up then the first random number U_1. Setting the thresholds as $T_k=k/N$, a patient number k is chosen if and only if $T_{k-1}\leq U_1<T_k$. We therefore include k into the list $\vec{L}$, i.e., we let $L_1\leftarrow k$. Therefore, the probability of selecting a *particular* patient number k is uniformed distributed, $P(k)=1/N$ $(\forall k)$. Since the random sample must be taken with no reposition, we delete k in the survivor list $\vec{S}$. This circumvents to consider k in the next step. The likelihood of selecting an arbitrary patient number (except k) must be up-dated using conditional probabilities since that the sample space had changed [7]. We reset the thresholds to be now $T_k=k/(N-1)$, k=1,2,...,N-1. A new choice of the list is done by a second random guess U_2, performing a comparison with the up-dated set of thresholds. The procedure goes on in a similar way each new step and for the I^{th} step the thresholds are $T_k=k/\|\vec{S}\|$. The probability of selecting a *particular* patient number k is uniformed distributed and $P(k|\vec{S})=P(T_{k-1}\leq U_I<T_k\mid\vec{S})=1/\|\vec{S}\|$ so that we have the following algorithm:

3. THE THRESHOLD-BASED ALGORITHM

```
BEGIN
    P1. Read N and g. If g | N continue else reenter data.
    P2. Set initial conditions: I=0, $L_i=0$, $S_i=i$ ($\forall$i=1,N), i.e.
            $\vec{L}=\{0,0,0,...,0\}$ and $\vec{S}=\{1,2,3,...,N\}$.
    P3. WHILE I<N DO
            BEGIN
            P3.1. Threshold up-dating: $T_k=\frac{k}{N-I}$, k=0,1,2,...,N-I.
            (there are exactly N-I thresholds)
            P3.2. Next step: I $\leftarrow$I+1.
            P3.3. Pick the $I^{th}$ uniform random number U.
            P3.4. If $T_{k-1}\leq U_I<T_k$ (k=1,2,...,N-I) the k must be included
                    in the random list $\vec{L}$, that is, $L_I$=k.
            P3.5. Up-date (shrink) the survivor list.
                (*Since the random list should have no repetition,
                  k must be deleted from the survivor list.*)
             END;
    P4.     FOR m equal 1 to g:
            Print the group number and their corresponding elements;
END.
```

An illustrative example is presented in the sequel showing how to design a randomized trial with 12 patients and 2 treatments.

random number	list of assignments $\vec{L}$	survivor list $\vec{S}$
U1=0.168502561...	L={3}	S={1,2,4,5,6,7,8,9,10,11,12}
U2=0.658033330...	L={3,9}	S={1,2,4,5,6,7,8,10,11,12}
U3=0.093729293...	L={3,9,1}	S={2,4,5,6,7,8,10,11,12}
U4=0.756609143...	L={3,9,1,10}	S={2,4,5,6,7,8,11,12}
U5=0.463955829...	L={3,9,1,10,6}	S={2,4,5,7,8,11,12}
U6=0.070162761...	L={3,9,1,10,6,5}	S={2,4,7,8,11,12}
U7=0.222246588...	L={3,9,1,10,6,5,11}	S={2,4,7,8,12}
U8=0.706319757...	L={3,9,1,10,6,5,11,8}	S={2,4,7,12}
U9=0.586996776...	L={3,9,1,10,6,5,11,8,12}	S={2,4,7}
U10=0.752142819...	L={3,9,1,10,6,5,11,8,12,7}	S={2,4}
U11=0.937703174...	L={3,9,1,10,6,5,11,8,12,7,4}	S={2}
U12=0.669782608...	L={3,9,1,10,6,5,11,8,12,7,4,2}	S={ }

threshold calculations:

T[1]	T[2]	T[3]	T[4]	T[5]	T[6]	T[7]	T[8]	T[9]	T[10]	T[11]	T[12]
$0.08\underline{33}$	$0.01\underline{66}$	0.25	$0.\underline{33}$	$0.41\underline{66}$	0.5	$0.58\underline{33}$	$0.\underline{66}$	0.75	$0.8\underline{33}$	$0.91\underline{66}$	1.00
$0.0\underline{90}$	$0.\underline{18}$	$0.\underline{27}$	$0.\underline{36}$	$0.\underline{45}$	$0.\underline{54}$	$0.\underline{63}$	$0.\underline{72}$	$0.\underline{81}$	$0.\underline{90}$	1.00	
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.00		
$0.\underline{11}$	$0.\underline{22}$	$0.\underline{33}$	$0.\underline{44}$	$0.\underline{55}$	$0.\underline{66}$	$0.\underline{77}$	$0.\underline{88}$	1.00			
0.125	0.25	0.375	0.5	0.625	0.75	0.875	1.00				
...											
...											
0.5	1.00										
1.00											

The Ith set of threshold is given by T[k]=k/(12-I), k=1,2,...,12-I and the underline denotes an infinite repetition, e.g. $0.08\underline{33}$=0.08333333333...

For instance, the 4th patient assignment depends only on U4=0.756609143... The survivor list prior this step is S={2,4,5,6,7,8,10,11,12}.

Since that T[6]=0.66≤U4<T[7]=0.77 we pick the 7th element of the survivor list: Insert number 10 in the list of assignments and up-date the survivor list by erasing such a patient number: S={2,4,5,6,7,8,11,12}.

The treatment (A or B) is decided after obtaining the complete list of assignments:

L={3,9,1,10,6,5,11,8,12,7,4,2}

Treatment A: The first N/g=6 patients; Treatment B: The second N/g=6 patients.

4. EXAMINING THE RANDOMNESS OF A LIST OF ASSIGNMENTS

The following propositions corroborate that the threshold-based approach achieves properly randomized groups. The symbols $\cup$ and $\cap$ denote the union and intersection of events, respectively.

Proposition 1. An arbitrary patient (say number k) has exactly the same probability to be in any position of the list of assignments, that is,

$$P(L_i = k) = \frac{1}{N} \quad (\forall k = 1,2,\dots,N), \quad (\forall i = 1,2,\dots N).$$

Proof. Since that U is an uniform random variable, $P(L_1=k)=P(0<U_1<1/N)=\frac{1}{N}$. We remark then that $P(L_2=k)=P(L_2=k\cap L_1\neq k)$ so that

$$P(L_2=k)=P(L_1\neq k)P(L_2=k|L_1\neq k) = \frac{N-1}{N}\frac{1}{N-1}=\frac{1}{N}.$$

Following, we evaluate

$P(L_3=k)=P(L_3=k\cap L_1\neq k\cap L_2\neq k)$. Therefore,

$$P(L_3=k)=P(L_1\neq k)P(L_2\neq k|L_1\neq k)P(L_3=k|L_1\neq k,L_2\neq k)=$$

$$\frac{N-1}{N}\frac{N-2}{N-1}\frac{1}{N-2}=\frac{1}{N}.$$

By using a similar reasoning, we prove that

$$P(L_4=k)=\frac{N-1}{N}\frac{N-2}{N-1}\frac{N-3}{N-2}\frac{1}{N-3}=\frac{1}{N}$$ and so on Q.E.D.

<u>Proposition 2</u>. An arbitrary patient has exactly the same probability to be in a treatment, no matter which the treatment is.

Proof. Let us consider an arbitrary treatment, say m. The probability of the kth-patient be assigned to such a treatments is given by

$P(k\in G_m)=P(\bigcup_{i=(m-1)\frac{N}{g}+1}^{m.\frac{N}{g}} L_i=k)$. Since the events are mutually exclusive, we have then

$P(k\in G_m)=\sum_{i=(m-1)\frac{N}{g}+1}^{m.\frac{N}{g}} \mathrm{P}(L_i=k)$. Now using proposition 1, it follows that

$$P(k\in G_m)=\sum_{i=(m-1)\frac{N}{g}+1}^{m.\frac{N}{g}} \frac{1}{\mathrm{N}}=\|G_m\|\frac{1}{N}=\frac{N}{g}\frac{1}{N}=\frac{1}{g}.$$ Q.E.D.

A preparing of a random list in such a way that both Propositions 1 and 2 hold is defined as a strong randomization assignment.

5. CONCLUSIONS

Clinical trials are one of the most adopted strategies in the evaluation of new issues for drug therapy. A computer-oriented approach for preparing a list of random treatment assignments is offered as a tool for groups extensively involved in such trials. This way of patients' selection ensures *exactly* equal treatment numbers. The technique, referred as to the "threshold approach", is general and can be applied for a randomized trial with *any* number of treatments. It can also easily be extended to stratified randomization. The randomness of the patient's allocation to treatment is examined showing that a "strong randomization" is achieved.

ACKNOWLEDGEMENTS

This research had a partial financial support by the Pharmaceutical Laboratory of the State of Pernambuco - LAFEPE, Brazil.

REFERENCES

[1] S.J. Podok (1979), Allocation of patients to treatment in clinical trials, *Biometrics*, **33**, pp.183-197.

[2] British Medical Journal (1977), Randomized clinical trials (leader), *Br. Medical J.*, i, pp.1238-1239.

[3] P.W. Armstrong (1979), A new design for randomized trials, *N. Engl. J. Med.*, **300**, p.786.

[4] C.B. Begg and B. Iglewicz (1980), A treatment allocation procedure for clinical trials, *Biometrics*, **36**, pp.81-90.

[5] D.L. Sackett and M. Gent (1979), Controversy in counting and attributing events in clinical trials, *N. Engl. J. Med.*, **301**, pp.1410-1412.

[6] R.Y. Rubinstein (1981), Simulation and the Monte Carlo Method, New York: Wiley.

[7] W.B. Davenport Jr (1970), Probability and Random Processes, McGraw-Hill Kogakusha.

APPENDIX (Turbo Pascal Program) **freeware**

Available on request at hmo@ufpe.br (http://www2.ee.ufpe.br/codec/~trial)

```
PROGRAM Randomized_trial(input,output); USES DOS, CRT;
VAR    I,index,k,N,g,m: INTEGER;
VAR    hh,mm,ss,ms: WORD;
(* N = number of patients *)
(* g = number of groups *)
(* I = step number *)
VAR   remain,U: REAL;
VAR   error: BOOLEAN;
TYPE  vector= ARRAY [1..100] OF INTEGER;
(* Defining two lists:
  Assignment list and Survivor list *)
VAR   L: vector;
        S: vector;
(* Defining Thresholds *)
VAR   T: ARRAY [0..100] OF REAL;
BEGIN
(* input data: N and g *)
CLRSCR;
WRITELN;
error:=FALSE;
gettime(hh,mm,ss,ms);
WRITELN(hh,':',mm,':',ss);
RANDOMIZE;
WRITE('Enter the number of patients N=');
READLN(N);
IF (N>100) THEN
        BEGIN
        WRITELN('N must be less than 100');
        error:=TRUE;
        WRITELN('RUN AGAIN PLEASE ...');
        END;
WRITELN;
WRITE('Enter the number of groups  g=');
READLN(g);
(* checking if g | N then ok *)
remain:= N MOD g;
IF (remain<>0) THEN
```

```
          BEGIN
          WRITELN('The condition g | N does not hold!');
          error:=TRUE;
          WRITELN('TRY AGAIN, PLEASE);
          END;
WRITELN;
(* Setting initial conditions *)
FOR k:= 1 TO N DO
          BEGIN
          L[k]:=0; S[k]:=k;
          END;
I:=0;
WHILE (I<N) AND (error=FALSE) DO
        BEGIN
        (* theshold calculations *)
        WRITELN;
        WRITELN('Establishing the thesholds...');
        FOR k:=0 TO (N-I) DO T[k]:= k/(N-I);
        FOR k:=0 TO (N-I) DO WRITELN('T[',k,']=',T[k]);
        READLN;
        (* randomizer *)
        U:= RANDOM;
        WRITELN('Random Variable U=',U);
        READLN;
        (* finding the patient number ...*)
        K:=1;
        WHILE (T[k]<U) DO k:=k+1;
        (* Up-dating the list *)
        L[I+1]:=S[k];
        WRITE('STEP number',' ',I+1,': ');
        WRITE('Entra o paciente numero ', L[I+1]);
        READLN;
        (* up-dating survivors *)
        FOR index:= 1 TO (N-I) DO
                 IF (index>=k) THEN S[index]:=S[index+1];
        (* test; delete *)
        WRITELN;
        WRITELN('Actual survivor list');
        FOR index:= 1 TO (N-I-1) DO WRITE(S[index],' ');
        WRITELN;
        (* end test; delete *)
        I:=I+1;
        END;
(* writting down the list *)
WRITELN;
WRITELN('COMPLETE RANDOM LIST ');
WRITE('L=');
FOR index:= 1 TO N DO WRITE(L[index],' ');
READLN;
WRITELN;
FOR index:= 1 To g DO
            BEGIN
            WRITELN;
            WRITELN('Group number ',index,' ');
            FOR I:= 1 TO (N DIV g) DO
            WRITELN(L[I+(index-1)*(N DIV g)]);
            READLN;
            END;
READLN;
END.
```