PharmaSUG 2012 - Paper PO16

Scrambling of Un-Blinded Data without 'Scrambling Data Integrity'!

Jaya Baviskar, Pharmanet/i3, Mumbai, India

ABSTRACT

Scrambling of data is widely used and successfully implemented across several functional sectors and the pharmaceutical domain is one such area to effectively implement this technique. It is an efficient way to work with data while retaining data integrity as this is critically important while working on sensitive data seen across the pharmaceutical domain.

The 'scrambling of data' is more in demand for 'Blinded' studies with a short life-span that have to execute all processes in relatively smaller timelines. Hence the idea behind scrambling data is to facilitate 'Programmers', 'Biostatisticians', 'Data Managers' and other team members a glimpse of study data without compromising on data integrity. This helps to pre-empt the process of working on 'Analysis / Derived Data sets' or assess and design study-specific programs; thereby providing comparatively longer time-frame than usual. The scrambling can happen across Phase I, II and III studies and the decision to scramble is usually initiated by the Biostatistician or the Project Manager.

Keeping this concept in mind the paper will elaborate further on ways to scramble data, types of scrambling utilized, the type of data considered for scrambling, SAS® readily available functions and any associated general information.

INTRODUCTION

The concept of scrambling is applicable to 'Blinded' studies during a clinical trial.

Usually any RAW data file received is uploaded and the resultant RAW SAS Data set is made available to the study team to process the information as per their respective functional areas. This process is modified during a 'Blinded study' so as to retain data integrity. Since the received data is in its 'Un-blinded' form (actual information of subject collected as per study requirement); it has to be ensured that such sensitive data gets 'Blinded' so as to respect the objective of the study. Thus to achieve this objective; a technique called as 'Scrambling' is implemented.

Scrambling is a data manipulation technique whereby actual 'Un-blinded' data pertaining to subject(s) is randomly swapped so that the subject(s) information remains 'Blinded' to the intended group(s) for the duration of the study. Thus the project team comprises of a 'Blinded' and 'Un-blinded' team members of 'Biostatistician(s)', 'Statistical Programmers', 'Clinical Data Programmers' and 'Data Managers'.

When any 'Un-blinded' data is to be received for such 'Blinded studies' then the 'Un-blinded' team members are provided restricted access to this 'Un-blinded' data. All other study team members besides the 'Un-blinded team' receive the scrambled 'Blinded data'. This external data usually contains laboratory values and related information that are valuable in determining the efficacy and safety of the drug. The scrambling is usually done by a programmer using three approaches and is usually initiated by the 'Biostatistician' or a 'Statistical Programmer' on the project team.

PURPOSE:

The scrambling of data is usually seen across blinded studies with a short duration. Hence in order to adhere to timelines or just with the intention to keep the programming in place; the un-blinded data is scrambled so that the 'Blinded' Statistical Programmers and Biostatisticians can start to work on the project. This way the process of working on Analysis / Derived Data sets is pre-empted rather than waiting until the study database has been locked and frozen.

APPROACH:

Depending on the requirements of the study the Biostatistician would indicate the way the scrambled Data sets are to be displayed. Hence, the un-blinded data file received is uploaded and scrambled utilizing either of the approaches discussed further.

APPROACH 1:

The RAW data set that has been received is placed in a ‘Un-blinded’ study folder with access only for the selected team who are able to see the actual data. The data is then scrambled using SAS functions to randomly assign a unique number to each record. These data sets are then uploaded and are made accessible to the entire team; who as per their functional area process the available information.

APPROACH 2:

The received ‘un-blinded’ data file is uploaded and the resultant SAS RAW data set is placed in a directory with restricted access. Every successive transfer is uploaded and always placed in the restricted location until the database is locked and frozen. Once this has occurred; the scrambled data is provided to the project team.

APPROACH 3:

The data file is uploaded however the values from some / all of the ‘Blinded fields’ are replaced with ‘blanks’. So the RAW SAS data set will retain its database structure but will have blanks in the blinded fields. This way the un-blinded data can be made available to the team.

METHOD:

Usually a Biostatistician decides the intensity of scrambling to be applied. So data may go under a ‘complete’ or ‘partial’ scramble depending on the requirement that has been put forth. Keeping this in mind the methods are elaborated in detail.

The scrambling can be obtained by using -

-SAS available functions like **RANUNI**, **RANPERM**
-By using procedures like **PROC FCMP**, **PROC IML**
-Or by just applying ‘Simple Logic’

In the following course of the paper we will see an Input **‘UN-BLINDED DATA SET’ -‘EGG’** scrambled to obtain a **‘BLINDED DATA SET’ – ‘SCRAMBLED_EGG’**.

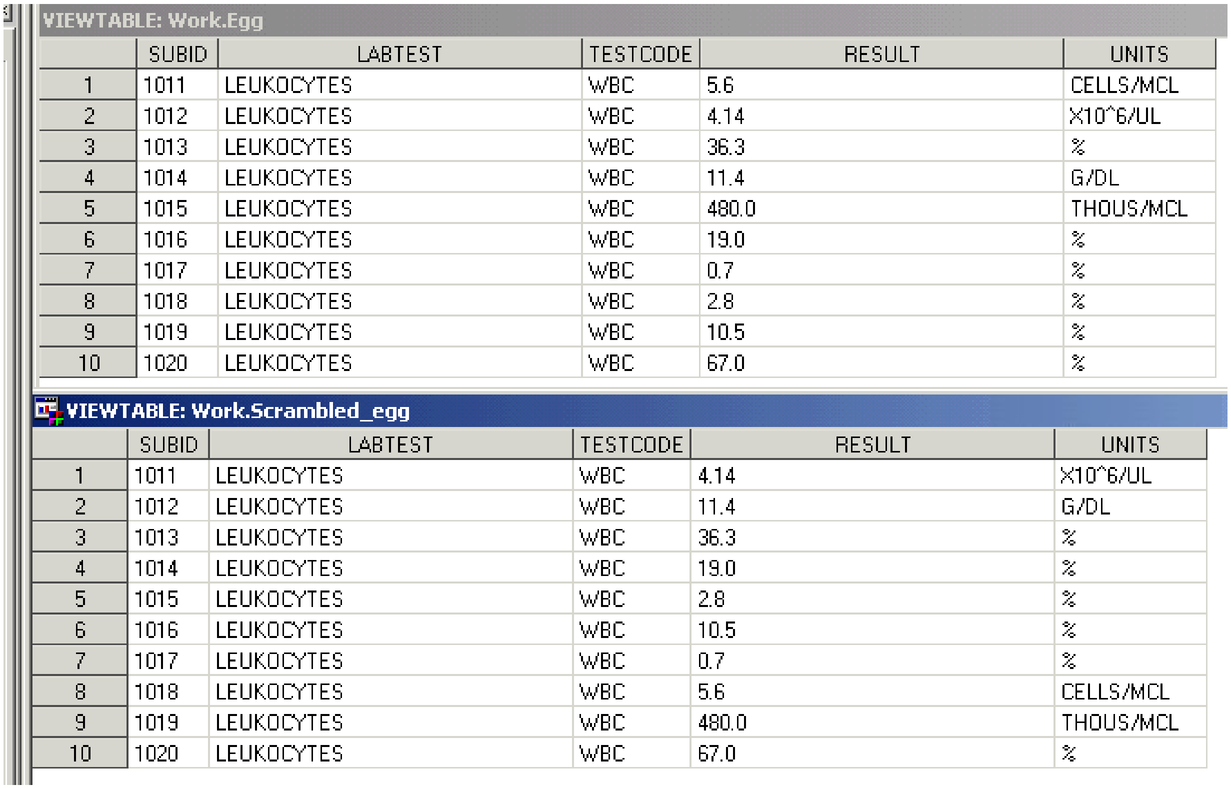
```
DATA EGG;
LENGTH SUBID $5 LABTEST $40 LABTESTCODE $8 RESULT $200 UNITS $10;
INPUT SUBID LABTEST LABTESTCODE RESULT UNITS;
DATALINES;
1011 LEUKOCYTES WBC 5.6 CELLS/MCL
1012 LEUKOCYTES WBC 4.14 X10^6/UL
1013 LEUKOCYTES WBC 36.3 %
1014 LEUKOCYTES WBC 11.4 G/DL
1015 LEUKOCYTES WBC 480.0  THOUS/MCL
1016 LEUKOCYTES WBC 19.0 %
1017 LEUKOCYTES WBC 0.7 %
1018 LEUKOCYTES WBC 2.8 %
1019 LEUKOCYTES WBC 10.5 %
1020 LEUKOCYTES WBC 67.0 %
; RUN;
```

METHOD 1: COMPLETE SCRAMBLE

In this type of scramble; the subject numbers and the results are swapped such that the results no longer belong to the actual subject. Each row is assigned a 'unique identifier' using readily available functions in SAS. This way the subject information is masked and made available for the team. **(Refer: Output 1)**

```
**************************************************************************;
/***** Identify the key variables to be scrambled           *****/
**************************************************************************;

*      Scramble unique identifier like a Subject Number;

DATA BLINDED (KEEP=SUBID RAND_NUM1);
       SET EGG;
       RAND_NUM1 = RANUNI(0); * Use the RANUNI function and creating a seed
                                     to randomize. Default seed is '0' ;
RUN;

PROC SORT DATA = BLINDED; BY RAND_NUM1 ; RUN;

DATA BLINDED2 (DROP=RAND_NUM1);
       SET BLINDED;
              RETAIN COUNT;
              COUNT+1;     * Assign an incrementing counter ;
RUN;

PROC SORT DATA =BLINDED2; BY COUNT; RUN;

*      Scramble the key variables like results or any subject specific information. ;

DATA SCRAMBLE;
       SET EGG;
              RAND_NUM2 = RANUNI(052604); * The seed can be any number that can be used
to randomize.;
RUN;

PROC SORT DATA=SCRAMBLE; BY RAND_NUM2; RUN;

DATA SCRAMBLE2 (DROP=RAND_NUM2 SUBID);
       SET SCRAMBLE;
              RETAIN COUNT;
              COUNT+1;
RUN;

PROC SORT DATA =SCRAMBLE2; BY COUNT; RUN;

**************************************************************************;
/***** Combine Data sets to obtain a completely scrambled data set *****/
**************************************************************************;

DATA SCRAMBLED_EGG (DROP=COUNT);
       MERGE BLINDED2 SCRAMBLE2;
       BY COUNT;
RUN;

PROC SORT DATA =SCRAMBLED_EGG; BY SUBID; RUN;
```

OUTPUT 1:



VIEWTABLE: Work.Egg

	SUBID	LABTEST	TESTCODE	RESULT	UNITS
1	1011	LEUKOCYTES	WBC	5.6	CELLS/MCL
2	1012	LEUKOCYTES	WBC	4.14	X10^6/UL
3	1013	LEUKOCYTES	WBC	36.3	%
4	1014	LEUKOCYTES	WBC	11.4	G/DL
5	1015	LEUKOCYTES	WBC	480.0	THOUS/MCL
6	1016	LEUKOCYTES	WBC	19.0	%
7	1017	LEUKOCYTES	WBC	0.7	%
8	1018	LEUKOCYTES	WBC	2.8	%
9	1019	LEUKOCYTES	WBC	10.5	%
10	1020	LEUKOCYTES	WBC	67.0	%

VIEWTABLE: Work.Scrambled_egg

	SUBID	LABTEST	TESTCODE	RESULT	UNITS
1	1011	LEUKOCYTES	WBC	4.14	X10^6/UL
2	1012	LEUKOCYTES	WBC	11.4	G/DL
3	1013	LEUKOCYTES	WBC	36.3	%
4	1014	LEUKOCYTES	WBC	19.0	%
5	1015	LEUKOCYTES	WBC	2.8	%
6	1016	LEUKOCYTES	WBC	10.5	%
7	1017	LEUKOCYTES	WBC	0.7	%
8	1018	LEUKOCYTES	WBC	5.6	CELLS/MCL
9	1019	LEUKOCYTES	WBC	480.0	THOUS/MCL
10	1020	LEUKOCYTES	WBC	67.0	%

OUTPUT 1: The displayed output indicates the 'Un-blinded Data set -EGG' and the obtained 'Scrambled Data set – SCRAMBLED_EGG'.

METHOD 2: PARTIAL SCRAMBLE

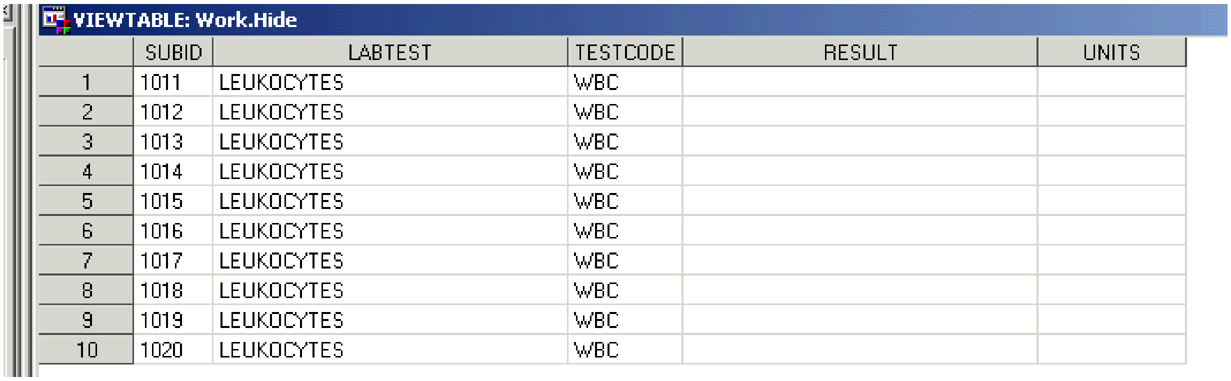
In this type; the records of subjects are partially scrambled so that only selective information gets scrambled. The intensity of scrambling depends on the variables that are considered for the scrambling.

1) FIRST METHOD:

A simple approach to 'un-blind' the information is by simply setting the fields in question to 'Missing / Blank'. This way the relevant information is still visible however the values associated to the 'Subject' is masked. **(Refer: Output 2)**

```
DATA HIDE;
        SET EGG;
        RESULT='';
        UNITS='';
RUN;
```

OUTPUT 2:



	SUBID	LABTEST	TESTCODE	RESULT	UNITS
1	1011	LEUKOCYTES	WBC		
2	1012	LEUKOCYTES	WBC		
3	1013	LEUKOCYTES	WBC		
4	1014	LEUKOCYTES	WBC		
5	1015	LEUKOCYTES	WBC		
6	1016	LEUKOCYTES	WBC		
7	1017	LEUKOCYTES	WBC		
8	1018	LEUKOCYTES	WBC		
9	1019	LEUKOCYTES	WBC		
10	1020	LEUKOCYTES	WBC		

OUTPUT 2: The displayed output indicates the ‘Scrambled Data set – HIDE’ obtained by masking the information with blanks.

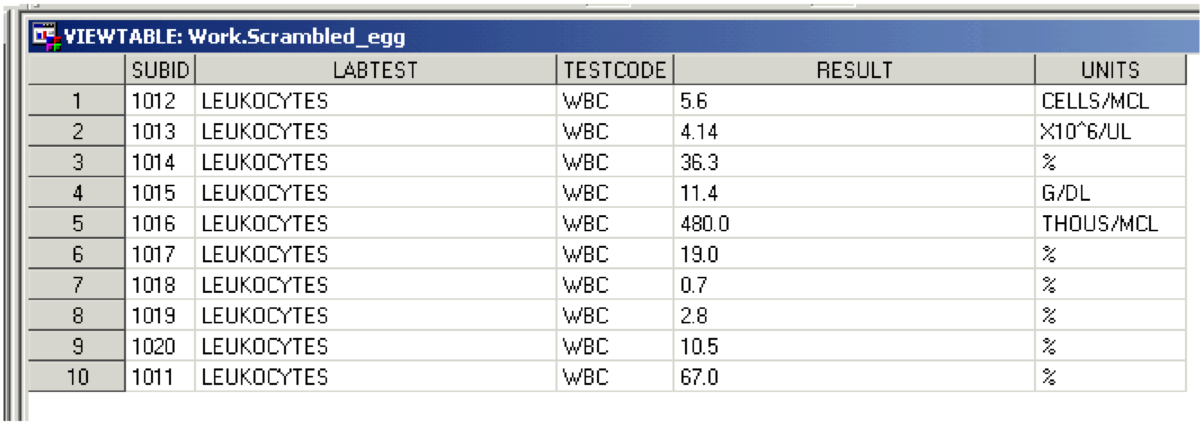
2) SECOND METHOD:

The partial scramble is implemented using a ‘Unique Identifier’ like ‘Subject Number’ and then processed to swap numbers so that the results can no longer be associated with the subjects as seen in the source data. **(Refer: Output 3)**

```
**************************************************************************;
/***** Identify the unique identifier to be scrambled  *****/
**************************************************************************;

DATA SCRAMBLE;
       LENGTH TEMPSUB NEWSUB $4 COUNT Y 8;
       SET EGG;

         %LET dataname=%SYSFUNC(OPEN(EGG)); * Use SAS default macros to
                                             open the source data set ;
         %LET num=%SYSFUNC(ATTRN(&dataname,NLOBS));   * Obtain the
                                                    Number of observations;
         %LET dataclose=%SYSFUNC(CLOSE(&dataname));   * Close the
                                    source data set after reading information;
              COUNT +1;
              X=&NUM;

              DO I=1 TO &NUM;     * Iterate until all unique identifiers have
                                    been assigned a new Subject Identifier;
                     NEWSUB=SUBID + 1;
              END;

              IF COUNT=1 THEN DO;
                     TEMPSUB=SUBID;
              END;

       Y=MAX(COUNT,&NUM);

       IF (COUNT=Y) THEN DO; *    Swap subject identifier of the first record with the
                                  last record   ;
              NEWSUB=TEMPSUB;
       END;
       RETAIN TEMPSUB;
       DROP X I Y;
RUN;
```

```
DATA SCRAMBLED_EGG (RENAME=(NEWSUB=SUBID));
       SET SCRAMBLE (DROP=SUBID COUNT TEMPSUB);
RUN;
```

OUTPUT 3:



	SUBID	LABTEST	TESTCODE	RESULT	UNITS
1	1012	LEUKOCYTES	WBC	5.6	CELLS/MCL
2	1013	LEUKOCYTES	WBC	4.14	X10^6/UL
3	1014	LEUKOCYTES	WBC	36.3	%
4	1015	LEUKOCYTES	WBC	11.4	G/DL
5	1016	LEUKOCYTES	WBC	480.0	THOUS/MCL
6	1017	LEUKOCYTES	WBC	19.0	%
7	1018	LEUKOCYTES	WBC	0.7	%
8	1019	LEUKOCYTES	WBC	2.8	%
9	1020	LEUKOCYTES	WBC	10.5	%
10	1011	LEUKOCYTES	WBC	67.0	%

OUTPUT 3: The displayed output indicates the 'Scrambled Data set –SCRAMBLED_EGG' obtained by partially scrambling the information. In this case the 'Subject Number' is scrambled whereas rest of the information is intact.

ALTERNATIVE METHODS:

1) SCRAMBLING USING THE PROC FCMP PROCEDURE

This function available in Version 9.2 allows creating 'User-Defined' functions which cannot be done using a simple DATA STEP. This incredible ability provides a free-rein to users to process information effectively and in fewer steps. The second advantage it offers is the reusability of the function; if made available in a 'Permanent library'. Thus having a permanent availability makes the function accessible to all programs where it is intended to be used.

The below program creates a user-defined function 'SCRAMNUM' using the 'PROC FCMP' procedure. It accepts an argument 'RANSUB' which is passed for processing to the function. The function then returns the processed value which is then collected by the 'NEWSUB' variable created in the DATA step 'SCRAMBLED'. For further information see reference 2.

```
PROC FCMP
OUTLIB=WORK.SCRAMBLING.TEST; * Storing functions. Recommended to store in a permanent
library for ready access for intended users. ;

LENGTH TEMPSUB NEWSUB1 $4 COUNT RANSUB Y 8;

 FUNCTION SCRAMNUM (RANSUB) $ ; * Declare a function and declare the type and
                                   number of arguments it will receive;

 * Alternatively keyword 'SUBROUTINE' can be used instead of 'FUNCTION' ;

       OUTARGS RANSUB;

* Obtain the number of observations from the input data set ;

  %LET dataname =%SYSFUNC(OPEN(EGG)); * Use SAS default macros to
                                         open the source data set ;
         %LET num=%SYSFUNC(ATTRN(&dataname,NLOBS));   * Obtain the
                                                Number of observations;
         %LET dataclose=%SYSFUNC(CLOSE(&dataname));   * Close the
                                        source data set after reading information;
```

```
             X=&NUM; *Assign number of observations to a variable;

 DO I=1 TO &NUM;
       NEWSUB1=PUT((RANSUB + 1), 4.);
     END;
             COUNT +1;

RETURN(NEWSUB1); * Return the processed value to the data set that invoked the
                     function call;
 ENDSUB;          * End the subroutine  ;

QUIT;

OPTIONS CMPLIB=WORK.SCRAMBLING; * Allows function to be available to the
                                        current library;

*      Created data set that will receive scrambled information    ;

DATA SCRAMBLED;
       LENGTH NEWSUB $4;
             NEWSUB = ' ';* Initialize the variable;

       SET EGG;
       OLDSUB = SUBID *1;
       NEWSUB = SCRAMNUM(OLDSUB);
       PUT NEWSUB=;
RUN;

DATA SCRAMBLED_EGG (RENAME=NEWSUB=SUBID);
       SET SCRAMBLED (DROP=SUBID);

       DROP OLDSUB;
RUN;
```

2) SRAMBLING USING THE PROC IML PROCEDURE

PROC IML can be used to obtain a more complex scrambling and when a value has to be scrambled at ‘Byte’ or ‘Character’ level. This can be used in a combination of functions like **RANPERM**, **RANUNI** to attain the desired level of complexity. For further information see references 1, 3 and 4.

Note: The code is subjected to modification based on the scope of the requirement.

POINTS TO CONSIDER:

1) While handling such Un-Blinded Data measures should be taken to understand the requirement thoroughly.

2) Determine unique identifiers and the information that requires scrambling. If the protocol indicates that trials have been conducted at multiple sites then the ‘Unique Identifier’ has to be a combination of ‘Study-Identifier’, ‘Site-Identifier’ and ‘Subject Identifier’.

3) Identify the approach and the method that is to be considered.

4) Consider the effects that a scrambling can have on the data. Hence it is always necessary to

cross-verify the output.

5) It is recommended to keep such data in a restricted access folder and any output from this folder has to be thoroughly checked before making it available to the intended end users.

CONCLUSION:

Scrambling is a very effective way to manipulate data in cases where the integrity of data has to be preserved without exception. We can achieve this by carefully understanding the data; the requirement and the purpose of it. By utilizing the ability to apply simple logic, use readily available SAS® functions or create user-defined functions; one can attain the expected results with consistency. The techniques can also be extensively applied in general to scramble any intended information.

REFERENCES

- **Useful techniques and detailed help available at SAS website.**

  SAS Sample Notes: Proc FCMP Procedure

  http://support.sas.com/notes/index.html
- **Functioning at an Advanced Level: PROC FCMP and PROC PROTO**
  Peter Eberhardt, Fernwood Consulting Group Inc.,
  Toronto ON Canada
- **Using SAS® Bitwise Functions to Scramble Data Fields with Key**
  Sheng Luo, Providian Corporation, Frazer, PA
  Xinsheng Lin, IMS America, Plymouth Meeting, PA
- **Statistical Programming with SAS/IML Software, Rick Wicklin**

ACKNOWLEDGMENTS

Special thanks to -- **Kelly Bussell,** Principal Clinical Data Programmer, Pharmanet-i3; for generously providing her expertise, meticulous details, insight and also appreciated for carefully reviewing the paper.

-- **Colleen Benjamin,** Senior Manager, Statistical Programming, Pharmanet-i3; for her invaluable inputs.

I am grateful to **Dr. Prashant Kirkire,** Country Manager, India, Pharmanet-i3; for his encouragement provided throughout and for the careful review of the paper. I thank **Sandeep Sawant, Naina Pandurangi, Neha Mohan and Sneha Sarmukadam** for guiding this work with comments and suggestions. **Debra Santolini** and **Ann-Marie Hess** are greatly appreciated for their invaluable feedback and suggestions at a very short notice.

CONTACT INFORMATION

Your comments and questions are greatly appreciated and encouraged. Contact the author at:

Name: **Jaya Baviskar**

Enterprise: **Pharmanet/i3 (Inventiv Health Clinical),**

Address: 7th Floor, Corporate Center,
Opposite VITS Hotel,
Andheri-Kurla Road,
Andheri (E)

City, State ZIP: **Mumbai, 400059,**

Country : **India**

Work Phone: **+91-22-30554052**

E- mail: JBaviskar@Pharmanet-i3.com ; JBaviskar@yahoo.com