PharmaSUG China2015 - Paper 73

A customized tool for systematic input/output dependency diagnose

Chunlan Xing, Fountain Medical Development, Inc., Nanjing, China
Chao Wang, Fountain Medical Development, Inc., Nanjing, China

ABSTRACT

After the final database lock in a clinical study, when facing changes in one of few raw datasets through database re-open/re-lock, study team will need to make a strategic decision based on the impact of these changes to the analysis datasets and related outputs. In the case that the team decides to re-run only those analysis datasets and output that are related to these raw data changes, the dependency between the changed raw datasets and the analysis datasets and outputs will need to be diagnosed. A systematic diagnose of such dependency will ensure all related programs be re-run. In this paper, the SAS I/O functions and SAS Dictionary techniques will be used to create a customized report for dependency relationship checking.

INTRODUCTION

In pharmaceutical industry, the raw datasets maybe changed after database lock due to some unexpected issues. In this situation, related outputs (datasets, tables, listings and figures) must be updated with the newest data.

Two methods are always used to handle this task. First one, batch-run all the SDTM, ADaM and TLFs programming, it is easy to do so when the database is smaller and the elapsed time of batch-run is limited. If the database is large and there are amount of datasets and TLFs for the study, especially the phase III studies, it will cost a lot of time beyond expectation, not only run the program, but also check the outputs. It would be better to only run the datasets and TLFs programs that related to the changed raw datasets.

The paper intent to present a strategic decision based on the impact of the change of raw datasets to the analysis datasets and TLF outputs. In the case that the team decides to re-run only those analysis datasets and output that are related to these raw data changes, the dependency between the changed raw datasets and the analysis datasets and output will need to be diagnosed. A systematic diagnose of such dependency will ensure all related programs be re-run.

GENERAL PROCEDURE

In this section, several flowcharts will be used to explain the procedure that how to handle the process. There are two assumptions: 1. From SDTM and ADaM specifications, the dataset dependency between Raw-SDTM and SDTM-ADaM could be got. 2. From TLF tracking log, the ADaM-TLF dependency could be got.

Figure 1 is the general programming process from raw datasets to TLF outputs.

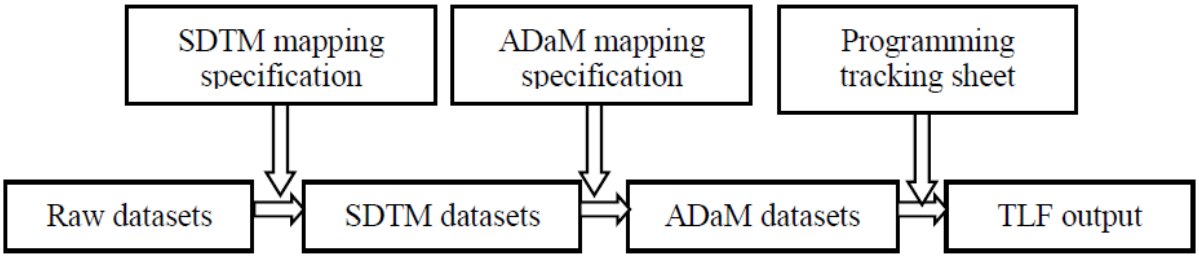

Figure 1 Work Flowchart of the Clinical Programming

There are four steps (Figure 2) to re-run SDTM datasets, ADaM datasets and TLF outputs which are related to the changed raw datasets.

- Firstly, get the dependency chain from Raw Data to ADaM datasets.
- Secondly, get the dependency chain between ADaM datasets and TLFs.
- Thirdly, connect these two chains to get the dependency from Raw Datasets to TLF outputs.
- Finally, get the creation time of all the elements (Raw Datasets, SDTM Datasets, ADaM Datasets and TLFs). Based on the relationship map, check the compliance by comparing the time. The general rule is the time of low level should be great than the high level. Such as, time of Raw datasets <time of SDTM

datasets< time of ADaM datasets < time of output TLFs. And the dependency between the SDTM datasets also need to check. Such as the time of ADSL dataset should be lower than any other ADaM.

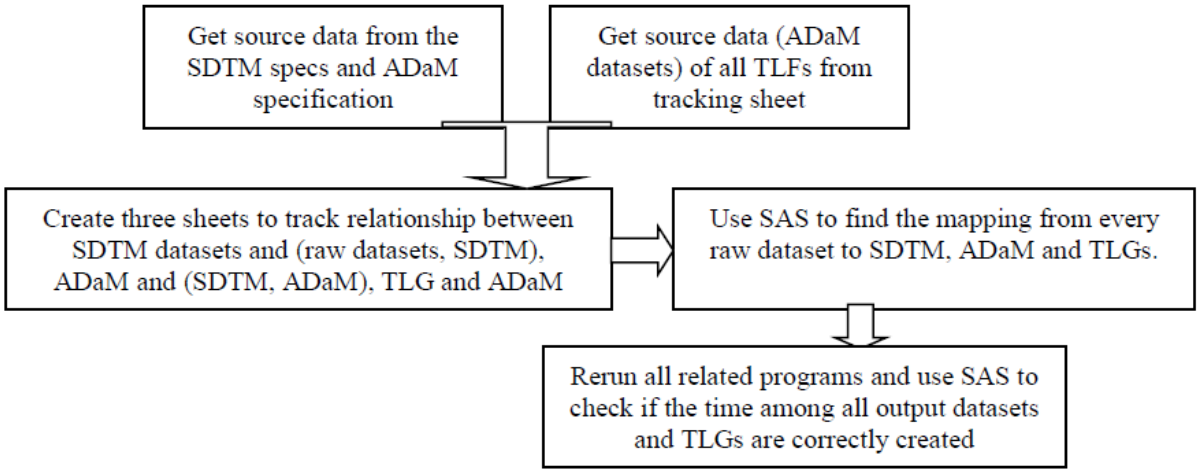

Figure 2 The steps of local updates for some related datasets and output TLGs

DEVELOPMENT

Here we will display how to accomplish this procedure by SAS, only consider the dependency between Raw datasets and SDTM datasets as an example. The mapping rules for ADaM and output TLGs are the same.

Firstly, get the date and time for each datasets in the folders: Raw datasets folder and SDTM folder. The macro blew is used to accomplish this requirement.

Read all the datasets in aimed folder and display all of them in a new SAS dataset. The macro variable ***postfix*** is help to just keep the necessary document type like ".sas7bat". Another macro variable ***dir*** will provide the location of input file and macro variable ***path*** can specified the aimed folder.

```
Filename dirlist pipe "&dir.&path.\*.* /b";
Data dirlist ;
length filename $1024;
infile dirlist length=reclen ;
input filename $varying1024.reclen;
memname=scan(filename,1,".");
if scan(filename,2,".")="&postfix";
run;
```

After get all datasets in the folder, then read each datasets name "***memname***" into a macro variable ***dataname*** and use macro variable ***count*** to represent the total number of datasets. After that, we get all the datasets prepared and another important part is getting the last modification time for each datasets, related code is listed as follow:

```
%do i=1%to&count;
Filename fileref"&path.\&&dataname&i...&postfix";
Data &&dataname&i;
length data $20;
infile fileref truncover;
fid=fopen('fileref');
moddate=finfo(fid,'Last Modified');
data=upcase("&dir")||"."||upcase("&&dataname&i");
keep data moddate;
run;
%end;
```

Now the datasets macro variable(&&dataname&i) has contain all the time information about each dataset, then we set all of them together to create a summary dataset for &dir which contain the name and the last modified time for each datasets.

Secondly, create the dependency map between source datasets for each SDTM dataset. We can get this dataset through the SDTM mapping specification. As shown in Table1, the dataset REL1 display the datasets need to be checked and their source data. Use the EG domain as the example to show the detail process. The source datasets listed now are raw.EG1, raw.EG2, SDTM.DM, SDTM.SV.

Dataset name	Source data
DM	FSHI, IC, SV, ENOT, AE, DM, RAND
SUPPDM	FSHI, IC, SV, ENOT, AE, DM, RAND
EG	EG1, EG2,SDTM.DM, SDTM.SV
SUPPEG	EG1, EG2,SDTM.DM, SDTM.SV
SV	SV, SDTM.DM

Table 1 The source datasets of each SDTM domain

Thirdly, as shown in Table 2, in source_data2 column, the SDTM.DM in source data column will be replaced by FSHI, IC, SV, ENOT, AE, DM, RAND and the SDTM.SV will be replaced by SV, and SDTM.DM. There is a SDTM source dataset still in source data2 and it will replace by its source data in RAW. As displayed in source_data3 column, the SDTM.DM in source_data2 column has replaced by FSHI, IC, SV, ENOT, AE, DM, RAND, at last all the duplicated source datasets will also be removed.

Dataset Name	Source Data	Source Data2	Source Data3
DM	FSHI, IC, SV, ENOT, AE, DM, RAND		
SUPPDM	FSHI, IC, SV, ENOT, AE, DM, RAND		
EG	EG1, EG2,SDTM.DM, SDTM.SV	FSHI, IC, SV, ENOT, AE, DM, RAND, SV, SDTM.DM	FSHI, IC, SV, ENOT, AE, DM, RAND
SUPPEG	EG1, EG2,SDTM.DM, SDTM.SV	FSHI, IC, SV, ENOT, AE, DM, RAND, SV, SDTM.DM	FSHI, IC, SV, ENOT, AE, DM, RAND
SV	SV, SDTM.DM	FSHI, IC, SV, ENOT, AE, DM, RAND	

Table 2 The source datasets of each SDTM domain

Fourthly, get the creation time of all the source datasets of SDTM.EG (connect source data, source data2, and source data3). If the time of any source datasets is after the time of SDTM.EG, the message will be outputted to the SAS dataset. As shown in Table 3, the date/time of SDTM.DM and SDTM.SV are after the date/time of SDTM.EG, the SDTM.EG dataset should be rerun after the updates of SDTM.DM and SDTM.SV.

Dataset Name	SDTM Data Modification Date	Source Data	Source Data Modification Date
SDTM.EG	28Jul2015:14:43:39	RAW.AE	27Jul2015:23:53:22
SDTM.EG	28Jul2015:14:43:39	RAW.DM	28Jul2015:23:46:29
SDTM.EG	28Jul2015:14:43:39	RAW.EG1	27Jul2015:23:47:43
SDTM.EG	28Jul2015:14:43:39	RAW.EG2	27Jul2015:23:53:07
SDTM.EG	28Jul2015:14:43:39	RAW.ENOT	27Jul2015:23:52:26
SDTM.EG	28Jul2015:14:43:39	RAW.FSHI	27Jul2015:23:53:27
SDTM.EG	28Jul2015:14:43:39	RAW.IC	27Jul2015:23:46:27
SDTM.EG	28Jul2015:14:43:39	RAW.RAND	27Jul2015:23:50:24
SDTM.EG	28Jul2015:14:43:39	RAW.SV	27Jul2015:23:46:25
SDTM.EG	28Jul2015:14:43:39	SDTM.DM	29Jul2015:10:39:55
SDTM.EG	28Jul2015:14:43:39	SDTM.SV	28Jul2015:17:31:24

Table 3 All the source datasets of SDTM.EG domain

The macro below is used to generate the relationship form of each dataset and output the SDTM dataset which are not in time order and need to re-run.

Firstly, reading the information in REL1 to macro variables ***data_name*** and ***sorce_data***, and it is better to added the prefix 'SDTM' which can help to identify the dataset.

```
%macro rel;
data _null_;
set rel1 end=eof;
call symputx(compress("data_name"||put(_n_,best.)),"SDTM."||dataset_name);
callsymputx(compress("source_data"||put(_n_,best.)),source_data1);
if eof then call symputx("count",_n_);
run;
```

After get the draft source information about each dataset, we will resolve the SDTM source to their original RAW data source until all the SDTM datasets have been replaced by RAW dataset, which is aimed to check the entire source data related.

```
%let j=0;
%let obs=1;

%do %until(&obs=0);
%let j=%eval(&j+1);
%put &j;

data rel1;
length source_data%eval(&j+1) $200;
set rel1;
source_data%eval(&j+1)="";
if find(source_data&j,"SDTM.") then do;
%do i=1%to&count;
if find(source_data&j,"&&data_name&i") then source_data%eval(&j+1)=catx(", ",
strip(source_data%eval(&j+1)),"&&source_data&i");
%end;
end;
run;
```

Aimed to ensure no data have been ignored, we will check whether has SDTM data still in source_data&i.

```
data check;
set rel1;
if find(source_data%eval(&j+1),"SDTM.");
run;

%let obs=0;
data _null_;
set check end=eof;
if eof then call symput("obs",trim(left(put(_n_,8.))));
run;

%put &obs;
%end;

%put &j;
```

After all the source datasets no matter RAW or SDTM data have been prepared, we join them together and delete the duplicated datasets to create a one to one paired dataset which just contain two variables - check data and their source data. The code below can help to handle this.

```
data rel2;
length source_data1 source_data $200;
set rel1;
format _all_;
informat _all_;

%do i=1%to&j;
if strip(source_data%eval(&i+1))^="" then source_data1=strip(source_data1)||",
"||strip(source_data%eval(&i+1));
```

```
%end;
count=count(source_data1,",")+1;
do i=1 to count;
sdata=upcase(strip(scan(source_data1,i,",")));
output;
end;
keep dataset_namesdata;
run;

proc sort data=rel2 nodupkey;
by dataset_namesdata;
run;
```

At last, we just need to merge the time information for each datasets as shown in Table 3, it's quite easy to compare the time between SDTM Data Modification Date and Source Data Modification Date. Any time for source data is earlier than the SDTM data will be outputted, which implies the SDTM dataset should be updated after those source data modified. And it will save lots of time through only re-run the specified items.

CONCLUSION

This article discussed the dependency between the RAW datasets, SDTM datasets, ADaM datasets and output TLFs. According to the logical map and time checking, we could only update the must-be-updated part when meet the re-lock issue during the clinical programming. In the future, a macro can be created to batch-run all the identified programs based on checking.

CONTACT INFORMATION

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

Name: Chunlan Xing
Enterprise: Fountain Medical Development, Inc.
Address: Room 402, Building 43, No.70, Headquarter Base, Phoenix Road, Jiangning District City,Nanjing, China
Work Phone: 025-86155036
E-mail:chunlan.xing@fountain-med.com