Exploring Phase 1 ADaM Deriving System from SDTM
Yiwen Li, Gilead Sciences, Foster City, CA

ABSTRACT

ADaM programming is an essential and challenging part in biometrics delivery, especially for Phase 1 studies with trend of more complex design and tighter timeline. The goal here is to standardize programming deliverables to gain maximum efficiency and consistency.

The paper describes automated ADaM Programming Deriving System compliant with CDISC standards to meet the goal. It contains six macros to streamline the process all together: %fetchadam, %atrtpanel, %paramcd, %newvisid, %selfmul, %chg_bl, %atotoxicity.

With better efficiency, the tool provides user more freedom to check on data instead of manual coding per study, enables efficient variable deriving/labeling process, and make debugging and maintenance easier. What's more, it allows initiate analysis data watching from study early stage since first patient first visit (FPFV). As user starts to implement with the tool, user is going to have clearer mind of ADaM data structure and scientific variable deriving process.

It has been successfully implemented to HIV Phase 1 study and significantly increased the efficiency of ADaM Deriving by at least 50%.

BACKGROUND

The deriving project for optimization started when Gilead initiated Phase 1 team August 2013. Back that time, we have restricted resources but need to handle growing number of studies of all phases including Phase 1 studies. As I understand eventually all Phase 1 studies would shift to Phase 1 team from each Therapeutic Area (TA), my intention was to create a template-embedded macro deriving system to help optimize the process from Gilead legacy studies/standards. The project was developed within HIV TA, and could be applied to other TA with one-time modification, as long as ADaM data structure is Subject-Level ADSL/BDS/ODS.

INTRODUCTION & IMPLEMENTATION

As CDISC based SDTM and ADaM data structure is well regulated, the key to optimization is to figure out the most reasonable order in the deriving process. Below is the overview flow chart for what I figured out.

[Diagram showing flowchart with steps such as Auto-import/merge from Domain and SUPP Domain, Auto-derive Analysis Date Time/Period/Treat variables, Derive unique Parameter for BDS structured ADaM, Auto-derive Analysis Visit ID/Window variables for each BDS structured ADaM, Auto-select unique Visit in the event of Multiple Records, Auto-derive Analysis Baseline Visit variables for each BDS structured ADaM, Auto-derive Analysis Baseline Toxicity variables for ADLB, etc., with macros for each step: %fetchadam, %atrtpanel, %paramcd, %newvisid, %selfmul, %chg_bl, %atotoxicity.]
1. %fetchadam

1.1. It is generic fetching process for ADaM, eg. ADSL, ADAE, ADVS, ADLB, ADEG, ADPC, ADPP.
1.2. It reads SDTM domain and automatically combine with SUPP domain if available.
1.3. It merges with ADSL or other domains in need.
   1.3.1. variables need to be carried over to other ADaMs
   1.3.2. domain=DM \rightarrow DS/SUPPDS, EX/SUPPEX, LB, VS, SV, PC, PP, FA
   1.3.3. domain=VS \rightarrow VS baseline weight or height
   1.3.4. domain=PC \rightarrow PC EXDOSE or EXDOSU
1.4. It works for both production and validation side

2. %atrtpanel

2.1. It is generic version for deriving treatment related/analysis date/time related/analysis period
related---several analysis flags:
ADTC, ADTM, ADT (ASTDT, AENDT), ATM,
APERIOD, APERIODC,
TRTP, TRTN, TRTEMFL, TRTA, TRTAN,
BASETYPE, BASETYPEN,
ADY (ASTDY, AENDY), SAFRFL, SAFRFN, CRIT1, CRIT1FL

3. %paramcd

3.1. It corrects 1-1 mapping for PARAMCD, PARAM, LBCAT related variables for ADLB.
3.2. It is more template-like for each specific TA.

4. %newvisid

4.1. It creates analysis visit variables based on the date relative to study day 1:
   ANLDY, ANLID, SCHED, _DIST_, _SIGN_, _METHOD_, ANLVISF, ANL01FL, ANLVIS,
   ANLFISC, AVISITN, AVISIT, AWTDIFF, AWTARGET.
4.2. It needs pre-defined study level formats:
   value _viscp_
   112 = '10'
   113 = '33';
   value _viscpvital_
   111 = '0,3,9,11'
   112 = '0,3,8,12'
   113 = '0,3,13';
   value _viscf_
   111 = '14'
   112 = '14'
   113 = '14';
   value _visc_
   0  = 'Predose/Baseline'
   1  = 'Postdose Day 1'
   3  = 'Postdose Day 3'
   6  = 'Postdose Day 6'
   8  = 'Postdose Day 8'
   9  = 'Postdose Day 9'
   9000 = 'Early Termination Visit'
   9999 = 'Follow-up Visit'
   other = 'N/A';
4.3. For visit not in scheduled analysis window, show as "Day xxx" in ADaM output for data
checking at study early stage
4.4. It imputes AVISITN, AVISIT, AWTDIFF, AWTARGET for Unscheduled visit records not
selected as analysis flagged
4.5. It adjusts APERIOD, APERIODC per AVISITN
data viscplst;
  length viscplst $20;
  viscplst=111;
  output;
  viscplst=112;
  output;
run;

data viscplst;
  set viscplst;
  %if &outdata. = vital %then %do;
    viscplst=put(viscp, _viscpvital_);
  %end;
  %if &outdata. = lab %then %do;
    viscplst=put(viscp, _viscplab_);
  %end;
  %if &outdata. = ecg %then %do;
    viscplst=put(viscp, _viscpecg_);
  %end;
run;

data _null_; 
  set viscplst end=eof;
  call symput('viscp'||strip(viscp),strip(viscplst));
  if eof then call symput('viscp',strip(viscplst));
run;

%do i=1 %to &partcnt.;
  %do j=1 %to &cohortcnt.;
  %do k=1 %to &seqpcnt.;
    if part=&i. and cohortn=&j and basetypen=&k. then do;
      if basetypen=1 then do;
        if .<adtm< tr0&k.sdtm then anlid  = 0;
        else                                          anlid  = floor(visitnum);
        if anlid not in (&viscp&i.&j.&k.) then anlid  = -99;
      end;
      else do;
        if .<adtm< tr0&k.sdtm then anlid  = 0;
        else                                          anlid  = floor(visitnum)-input(put(&i.&j.&k.,_viscp_.),best.);
        if anlid not in (&viscp&i.&j.&k.) then anlid  = -99;
      end;
    end;
  %end;
%end;
%end;

5. %selfmul
5.1. It selects unique data record in the event of multiple records on the same date&time for predose visits and on the same date for postdose visits
5.2. Data subsets to be three panels of data including categorical list data, categorical order data and numerical data before selection

6. %chg_bl
6.1. It derives several analysis baseline visit related variables for each analysis period
6.2. It needs pre-defined study level format shiftnecg
6.3. Change-from-baseline related analysis variables are calculated for all postdose analysis flagged safety visit records including postdose unscheduled visits

7. %atoxicity
7.1. It derives Analysis (Treatment-Emergent) (Maximum) (Hyper, Hypo) Lab Toxicity Grade related variables for ADLB
8. \%varexist (utility)

8.1. It checks whether certain variable exists in certain dataset, and would create one if not exist.

CONCLUSION & STRENGTH

1. More Freedom to Check on Data instead of Manual Coding per Study
2. Comply with Gilead Legacy Study Coding Logic Convention, Naming Convention
4. Initiate Analysis Data Watching from study early stage since Fist Patient Fist Visit (FPFV)
5. Clearer Mind of ADaM Data Structure
6. Clearer Mind of ADaM Variable Deriving Process
7. Fit complex study design
   • Part, Cohort, Fixed/Random Sequence
   • Replicate Cross-Over with Washout(s)
   • Treatment Package within a Sequence

Complex Study Design Example:

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>A8</td>
<td>1 2-4 5 6-10 11-14 15 16-18 19 20-21</td>
</tr>
</tbody>
</table>

REFERENCES

CDISC Metadata Submission Guidelines, www.cdisc.org/sdtm
Analysis Data Model (ADaM) v2.1 and Implementation Guide v1.0, www.cdisc.org/adam
SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies

CONTACT INFORMATION

Any comments and questions are highly valued and encouraged. Contact the author at:
Yiwen Li
Statistical Programming
Gilead Sciences, Inc.
300 Lakeside Drive, Foster City, CA 94404
Office: 650-522-5780
Email: Yiwen.Li@Gilead.com