PharmaSUG 2018 - Paper DS-08

ADaM data structure layout: common issues and ways to avoid

Yurong Dai, Jianfei Jiang, Eli Lilly and Company

ABSTRACT

From reviewing many studies' ADaM validation reports, which are the outputs from running Pinnacle 21 Enterprise (P21E), some common issues draw my attentions. The major problem was that some variable's values were not extended correctly or sufficiently due to complex study design – a design with multiple treatment periods, the consequence was getting over 10,000 warnings or errors from one data set only. Although the issues could be explained in ADRG, but they still exist, data quality and ADaM compliance were compromised. It is desirable to get the cleanest conformance check reports, and we should bear this idea in mind during ADaM specification development. Here I list a few points for good ADaM layout practice that may get not only cleaner conformance check report, but also better data traceability and less burden on explaining the issues in ADRG.

INTRODUCTION

In some therapeutic area, multiple treatment periods may exist within one clinical trial study, and multiple derived endpoints per period, for example, last observed value (LOV), Maximum (Max) and Minimum (Min) are required. Some ADaM data sets (for example, ADLB, ADVS) may need multiple types of Baseline (for example, last observed value, Maximum and Minimum), and different periods use different timepoint's baseline records. This resulted in ADaM data sets' structure complexity. In this case, variables TRTxxP/TRTxxA, AVISIT/AVISITN, BASETYPE have to be carefully utilized and extended with appropriate values to meet both study's requirement and ADaM conformance.

A few key points had been applied in my study's ADaM data and got clean P21E reports. The practice includes good use of APERIOD or APHASE, AVISIT value's extension, BASETYPE values' extension, AVALCATx etc.

1. APERIOD HAS TO MATCH XX IN ADSL.TRTXXP, AND USE APERIOD AND APHASE

Per the ADaM IG, the non-missing values of APERIOD must match the xx values in the ADSL.TRTxxP variable. However per analysis requirement, more analysis periods than the number of treatment periods are required. Ignoring this restriction has been observed in reality. In order to avoid this issue, APHASE is adopted to contain all analysis periods or combined study drug treated periods. For example, a study has 3 study drug treatment periods (named as Period 1 – Period 3), a screening period, a follow-up period, and a combined analysis period (period 1 and period 2 combined), we may define APERIOD and APHASE as in the following table.

Period name in protocol	Variable names and values in a ADaM data set			
	APERIOD	**APERIODC**	**APHASE**	**TRTxxP**
Screening	Null	Null	Screening	
Period 1	1	Period 1	Treatment	TRT01P
Period 2	2	Period 2	Treatment	TRT02P
Period 3	3	Period 3	Treatment	TRT03P
Follow-up	Null	Null	Follow-up	
Period 1 and Period 2 combined	Null	Null	Period 1 and Period 2 combined	

Table 1. APERIOD and APHASE's layout, and APERIOD values' association with TRTxxP.

APERIOD/APHASE's derivation may follow study protocol or are based on analysis need, i.e. each APERIOD/AHASE's start date/time may be consistent with the study's protocol.

If a "Null" cell in table 1 is assigned a value for APERIOD or APERIODC in a ADaM data set, it would incur error messages from P21E report, such as "BDS.APERIOD xx does not have a corresponding ADSL.TRTxxP variable".

2. AVISIT AND AVISITN HAVE TO 1-1 MATCHED - AVISIT VALUES' EXTENSION PER APERIOD/APHASE FOR ENDPOINT RECORDS

Multiple endpoints (Maximum, Minimum, Last of observed) analyses per analysis period, especially for lab and vital sign are often required. In this case, additional endpoint records have to be derived to the ADaM data set, and AVISIT/AVISITN need assigned values. I saw cases where one AVISIT value matched to multiple AVISITN values, for example one AVISIT = "Post Baseline LOV" is assigned to multiple records in multiple analysis period, and this can be illustrated in the table below.

AVIST	AVISITN	APERIOD/APHASEN
Post Baseline LOV	511	APERIOD = 1
Post Baseline MIN	512	APERIOD = 1
Post Baseline MAX	513	APERIOD = 1
Post Baseline LOV	611	APERIOD = 2
Post Baseline MIN	612	APERIOD = 2
Post Baseline MAX	613	APERIOD = 2
Post Baseline LOV	711	APERIOD = 3
Post Baseline MIN	712	APERIOD = 3
Post Baseline MAX	713	APERIOD = 3
Post Baseline LOV	811	APHSEN=9
Post Baseline MIN	812	APHSEN=9
Post Baseline MAX	813	APHSEN=9
Post Baseline LOV	1011	APHSEN=99
Post Baseline MIN	1012	APHSEN=99
Post Baseline MAX	1013	APHSEN=99

Table 2. Example for extended AVISIT and AVSITN values, one to many matched.

Apparently, this practice may cause error or warning message from conformance check. A good practice is extending further more to meet AVISIT and AVISITN's one-to-one match requirement. In my studies that have multiple Period/APHASE, I defined AVISIT/AVISITN values for the derived post baseline endpoint records as illustrated in table 3.

AVIST	AVISITN	APERIOD/ or APHASEN
Post Baseline LOV (period 1)	511	APERIOD = 1
Post Baseline MIN (period 1)	512	APERIOD = 1
Post Baseline MAX (period 1)	513	APERIOD = 1
Post Baseline LOV (period 2)	611	APERIOD = 2
Post Baseline MIN (period 2)	612	APERIOD = 2
Post Baseline MAX (period 2)	613	APERIOD = 2
Post Baseline LOV (period 3)	711	APERIOD = 3
Post Baseline MIN (period 3)	712	APERIOD = 3
Post Baseline MAX (period 3)	713	APERIOD = 3
Post Baseline LOV (periods 1 & 2 combined)	811	APHSEN=9
Post Baseline MIN (period 1 & 2 combined)	812	APHSEN=9
Post Baseline MAX (period 1 & 2 combined)	813	APHSEN=9
Post Baseline LOV (FU)	1011	APHSEN=99
Post Baseline MIN (FU)	1012	APHSEN=99
Post Baseline MAX (FU)	1013	APHSEN=99

Table 3. Example for extending AVISIT and AVSITN values, 1 to 1 matched

Note for Table 3: all AVISIT and AVISITN are newly assigned values for the derived endpoint records in addition to the ones from SDTM data. The general rule in my practice for the AVISIT values for the derived records is endpoint name + APERIOD/APHASE short name. The records are clearer on what they are exactly for than the ones showed in table 2.

3. BASETYPE SHOULD BE TRANSPARENT AND DESCRIPTIVE - BASELINE RECORDS AND BASETYPE VALUES' EXTENSION PER TIMEPOINT

When multiple baselines are required, variable BASETYPE is required and BASETYPE's value should be populated to both baseline and post baseline records.

LOV, MAXIMUM and MINIIMUM are most frequently used as BASETYPE's values. More complexity arises when a study needs multiple baseline types and multiple baseline records per different time points. For this case, the extended BASETYPE values should meet record level's traceability and baseline records' uniqueness, i.e. when we look at a record, its baseline record can be easily identified by BASETYPE value and ABLFL= "Y".
For example, in one of my study, the baseline timepoints were outlined as follows:

1. At time point before first dose of study treatment, 3 baselines are required at this time point - BASELINE LOV, BASELINE MIN and BASELINE MAX, which are the last non-missing records or the MAXIMUM or MAXIMUM record before first treatment period's first dose date and time. These baselines were used for treatment period 1's analyses.
2. Before each treatment period's 1st dose/or Phase's start date and time, one baseline is required - BASELINE LOV, which is the last observed value before the analysis period, is required for each analysis period and follow-up's analyses.

To meet this study's need, additional baseline records have to be added since one record can be both a baseline, and can be >=2 types of baseline, and a post baseline too. For these added/derived records, we assigned BASETYPE's value in the format of "Base Type + timepoint name", and AVISIT values in "Baseline + Base Type + timepoint name", where "Base Type" is "LOV", or "MINIMUM" or "MAXIMUM", and this can be illustrated in the table below.

AVISIT value for the Baseline records	BASETYPE values for the Baseline records	ABLFL
Baseline LOV (Prior to first dose of Period 1)	LOV (Prior to first dose of Period 1)	Y
Baseline MINIMUM (Prior to first dose of Period 1)	MINIMUM (Prior to first dose of Period 1)	Y
Baseline MAXIMUM (Prior to first dose of Period 1)	MAXIMUM (Prior to first dose of Period 1)	Y
Baseline LOV (Prior to first dose of Period 2)	LOV (Prior to first dose of Period 2)	Y
Baseline LOV (Prior to first dose of Period 3)	LOV (Prior to first dose of Period 3)	Y
Baseline LOV (Prior to FU)	LOV (Prior to FU)	Y

Table 4. Example of AVISIT values for baseline records and their BASETYPE values – BASETYPE value is unique per patient per test

From the above table, even each patient's lab/vital test has multiple records which were flagged as baseline records, each of them is unique. For each post baseline record, regardless it is copied from SDTM or derived for endpoint analysis per period/or phase, BASE and BASETYPE values would be populated together from the same baseline record's AVAL and BASETYPE, so that record level's baseline value can be easily traced back through the BASETYPE value.

If each patient's lab/vital test has multiple baseline records that have the same BASETYPE value, for example, if BASETYPE in table 4 is extended as in the below table, Error message "Multiple baseline records exist for a unique USUBJID, PARAMCD, BASETYPE" would be generated from a conformance check.

AVISIT value for the Baseline records	BASETYPE values for the Baseline records	ABLFL
Baseline LOV (Prior to first dose of Period 1)	LOV	Y
Baseline MINIMUM (Prior to first dose of Period 1)	MAXIMUM	Y
Baseline MAXIMUM (Prior to first dose of Period 1)	MINIMUM	Y
Baseline LOV (Prior to first dose of Period 2)	LOV	Y
Baseline LOV (Prior to first dose of Period 3)	LOV	Y
Baseline LOV (Prior to FU)	LOV	Y

Table 5. Example of AVISIT values for baseline records and their BASETYPE values, which may cause conformance check issues

4. ONE RECORD'S AVALCATy/CRITy, SHOULD BE DERIVED WITHIN THE RECORD

Per the ADaM IG, one record's variables AVALCATy, CRITy should be derived within the record. AVALCATy should be based on AVAL/AVALC only. In addition to AVAL/AVALC, CRITy's derivation can be based on other variables within the record, for example, BASE, CHG, PCHG etc. However, in practice, they are often derived across parameters or records.

For example, AVALCATy was used to categorize anti-drug antibody's (ADA) results and treatment emergent anti-drug antibody (TE-ADA)'s results in immunogenicity analysis data, where the ADA results depend on both test results ("DETECTED" or "NOT DETECTED") and the study drug's concentration in human body. Due to the derivation's logic, one test value can result in 2 or more ADA results. For

example, for a “NOT DETECTED” test result, when drug concentration is below the cut point, the ADA result is negative, otherwise the ADA result is inconclusive (we were not sure ADA is truly negative or the drug concentration was too high which can affect the ADA test result, so ADA could not be detected). This practice resulted in error message “Inconsistent value for AVALCAT1” or “Inconsistent value for AVALCAT2” from a P21E report.

For the above case, new parameter(s) and records should be added to hold the derived values that is based on multiple parameters, and use AVAL/AVALC rather than AVALCATy.

CONCLUSION

In summary, even for a complex study design with multiple analysis periods, a clean P21E report is achievable when laying out ADaM BDS and extending APERIOD/APERIODN, APHASE/APHASEN, AVISIT/AVISITN, BASETYPE’s values appropriately and in line with the IG. Special attention should be paid when using AVALCATy and CRITy related variables.

ACKNOWLEDGMENTS

Thank Larry Lixiang Liu for reviewing this paper

CONTACT INFORMATION

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

Yurong Dai
Eli Lilly and Company
myydai@yahoo.com

Jianfei Jiang
Eli Lilly and Company
Jiang_jianfei@lilly.com