

**PharmaSUG 2026 - Paper PK-310**  
**Enhancing ADaM PK Datasets to Automate PK TFLs Generation**  
Jianli Ping, Karthik Sankepelli, Gilead Sciences Inc

## ABSTRACT

Automated tools for programming output have been increasingly applied to generate safety tables, figures, and listings, which demonstrates improved programming efficiency. Implementing this programming methodology for PK data poses specific challenges, primarily due to differences in how treatment effects are defined when generating PK TFLs. Specifically, the key input variables for PK output automation differs from the conventional treatment variables used in most other ADaM datasets. This discrepancy arises because PK analysis requires treatment effects to be further stratified based on reference dosing days, food effect and other study-specifics. Currently, these PK specific treatment variables are typically derived and managed at the TFL programming level. In this paper, we propose PK group variables (TRTAGxN) and their descriptions that can fit the needs for PK TFLs programming by adding them into ADPC and ADPP. These group treatment variables could be considered as key variables for automation and ensure consistency of PK treatment variables among ADPC, ADPP and TFLs. The treatment formats can also be updated dynamically by taking values from variable description. These new PK treatment variables can be used to automate the PK TFLs programs, for study designs such as SAD/MAD, food effects, and drug-drug interaction to generate PK outputs. This can improve PK programming timing, efficiency and accuracy. Additionally, this paper will also illustrate how updated PK ADaM datasets can be applied together with macro parameter adjustment and dynamic creation of different formats to automate PK TFLs generation.

## INTRODUCTION

The Standard TFL Analysis Reporting System (STARS) is a SAS-based, metadata-driven tool developed at Gilead Sciences to automate the production of safety tables, figures, and listings (TFLs), using standardized shells to ensure consistent and efficient reporting. By leveraging metadata files and predefined shells, STARS streamlines programming, reduces manual errors, and facilitates validation, enhancing both efficiency and compliance in clinical trial data analysis. Its implementation has significantly improved deliverable timelines and quality, establishing STARS as a foundational system for automated safety TFL generation and paving the way for further automation advancements, such as those targeting PK TFLs discussed in this paper.

The application of tools commonly used for safety outputs to pharmacokinetic (PK) Tables, Figures, and Listings (TFLs) presents several distinct challenges. These challenges are primarily related to the definition and formatting of treatment variables, differences in requirements for table versus figures, and the additional programming and validation efforts required, as detailed below:

### TREATMENT VARIABLE DEFINITION

In BDS (TRTP/A), the treatment values are sourced from ADSL.TRTP/A. For safety outputs, summaries are typically generated based on either cohort or treatment. In contrast, PK outputs require summarization by treatment or by a combination of treatment, fasting status, and study days. This complexity adds layers to the programming and data preparation process.

### FORMATTING DIFFERENCES BETWEEN PK TABLES AND FIGURES

There is a notable difference in formatting requests for PK tables versus figures. In PK tables, the treatment description needs to be detailed, including cohort, treatment, day, and fasting status. However, for PK plots, the wording must be condensed to accommodate the limited space available on the plots. This requires careful consideration to ensure clarity without compromising the necessary information.

### INCREASED PROGRAMMING EFFORT

The need to define treatment and formatting at both the PK table and plots leads to extra programming work. This involves creating and managing formats that satisfy the requirements for both tables and figures.

### POTENTIAL FOR INCONSISTENCY

Programming is often performed separately at the table and figure levels, which can lead to inconsistencies. Since data preparation is repeated for each output, there is an increased risk of discrepancies arising between tables and figures.

## COMPLEXITY IN FIGURE LEGENDS

Formatting the legends for figures is a tedious process, especially when displaying treatment levels and BLQ/non-BLQ counts within the constraints of limited space. Ensuring that all necessary information is clearly presented in the figure legends requires meticulous attention to detail.

## ADDITIONAL VALIDATION REQUIREMENTS

Verifying the accuracy of texts in PK outputs is a manual process, increasing the workload and the likelihood of errors being overlooked. The extra validation step is necessary to ensure consistency and accuracy in all PK output documents.

## ENHANCING ADAM PK DATASETS WITH TREATMENT GROUP AND DISPLAYED TIME POINTS VARIABLES

Although TRTP, TRTPN, TRTA, and TRTAN are well defined in BDS datasets, share same computation rules, work well for generating safety TFLs in most cases, these variables alone could not be used directly for generating PK TFLs with multiple dose treatment cohorts. Additionally, the values in TRTP and TRTA are typically carried forwarded from SDTM.DM or ADSL and are generally not feasible for treatment labels for PK tables and figures. As discussed in the previous section, stratified or subgroup treatments are normally required for many Phase I study to characterize the single- and multiple-dose PK and/or food effect of a study drug based on PK concentration and parameters as determined by different collection days on participants with same treatment or across different periods. Currently PK summary tables and figures at our department are generated using 4 global macros with treatments and their display formats determined at each table or figure programming level. Additionally, time points of PK collection are defined either at TFs programing level or study level format named taskfmt file.

To enable automation of PK TFLs in a similar manner as currently used for safety TFLs generation at our department, TRTAG1N, TRTAG1, TRTAG2N, and TRTAG2 have been added into APPC, and TRTAG1N and TRTAG1 have been added into ADPP. The values in TRTAG1 are used to dynamically define format for treatments to be displayed in the tables, while the values in TRTAG2 are used to dynamically define format for treatments to be displayed in the figures, as shown in Table 1. These values can be used for all PK concentration tables, PK parameter tables, and PK parameter comparison tables. In addition, three variables for time point displays are also added into ADPC, as DTIMEN (Displayed Time Point (N)), DTIMET (Displayed Time Point for Table), and DTIMEF (Displayed Time Point for Figure).

TRTAN	TRTA	TRTAG1N	TRTAG1	TRTAG2N	TRTAG2	AVISITN	AVISIT
1   1   xx1 mg GS-xxxx SD fasted		1   Cohort 1, xx1 mg GS-xxxx SD fasted, Day 1		1   Cohort 1, xx1 mg GS-xxxx SD fasted, Day 1\$Cohort 1 Day 1		1   Day 1	
2   2   xx2 mg GS-xxxx SD fasted		2   Cohort 2, xx2 mg GS-xxxx SD fasted, Day 1		2   Cohort 2, xx2 mg GS-xxxx SD fasted, Day 1\$Cohort 2 Day 1		1   Day 1	
3   3   xx2 mg GS-xxxx SD high-fat meal		3   Cohort 2, xx2 mg GS-xxxx SD high-fat meal, Day 5		3   Cohort 2, xx2 mg GS-xxxx SD high-fat meal, Day 5\$Cohort 2 Day 5		5   Day 5	
4   4   xx3 mg GS-xxxx SD fasted		4   Cohort 3, xx3 mg GS-xxxx SD fasted, Day 1		4   Cohort 3, xx3 mg GS-xxxx SD fasted, Day 1\$Cohort 3 Day 1		1   Day 1	
5   5   xx3 mg GS-xxxx SD high-fat meal		5   Cohort 3, xx3 mg GS-xxxx SD high-fat meal, Day 5		5   Cohort 3, xx3 mg GS-xxxx SD high-fat meal, Day 5\$Cohort 3 Day 5		5   Day 5	
6   7   xx1 mg GS-xxxx QD fasted		6   Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1		6   Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1\$Cohort 5 Day 1		1   Day 1	
7   7   xx1 mg GS-xxxx QD fasted		7   Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10		7   Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10\$Cohort 5 Day 10		10   Day 10	
8   8   xx2 mg GS-xxxx QD fasted		8   Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1		8   Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1\$Cohort 6 Day 1		1   Day 1	
9   8   xx2 mg GS-xxxx QD fasted		9   Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10		9   Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10\$Cohort 6 Day 10		10   Day 10	
10   9   xx3 mg GS-xxxx QD fasted		10   Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1		10   Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1\$Cohort 7 Day 1		1   Day 1	
11   9   xx3 mg GS-xxxx QD fasted		11   Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10		11   Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10\$Cohort 7 Day 10		10   Day 10	

**Table 1. Treatment levels and their descriptions among TRTA, TRTAG1N/TRTAG1 and TRTAG2N/TRTAG2**

Variables TRTPGy, TRTPGyN, TRTAGy, and TRTAGyN are listed as record-level treatment variables for BDS datasets in Table 3.3.2.1 of ‘Analysis Data Model Implementation Guide’, Version 1.3 (CDISC, 2021). Users can have these group variables included to facilitate generation of tables, figures, and listings, when commonly used treatment variables TRTP, TRTPN, TRTA, and TRTAN cannot fit their programming needs. There are no warnings from Pinnacle 21 reports with adding such TRTAGyN/TRTAGy and displayed time points variables.

The values in TRTAG2 have a delimiter ‘\$’, which allows 2 formats to be dynamically defined by a macro program to generate PK concentration figures with one macro call. For mean and median PK concentration figures, there are 2 legends used, due to limited spaces, as shown in Figure 1. The legend on the top right corner indicates full treatment labels, while the legend on the bottom left corner indicates short treatment labels for studies with longer description for the treatment groups.



```

%doit(tdsn=adpc, outid=%str(t-pkpc-sad), pkflg=%str(pk1fl='Y'), cond=%str(paramcd='GSxxxx' and
cohortn <=6 and pcspec='PLASMA' and atptn<8000 and avisitn ne -999), trtgrp=trtagln,
trtgrp=trtag1, trtfmt=tfmt, tdata=tdata, hrfmt=);

*** macro call parameters setting with user defined time points format in taskfmt.sas for SAD
part;
%doit(tdsn=adpc, outid=%str(t-pkpc-sad-user), pkflg=%str(pk1fl='Y'), cond=%str(paramcd='GSxxxx'
and cohortn <=6 and pcspec='PLASMA' and atptn<8000 and avisitn ne -999), trtgrp=trtagln,
trtgrp=trtag1, trtfmt=tfmt, tdata=tdata, hrfmt=hrfmt_pk_t);

*** macro call parameters setting without user defined time points format for MAD part;
%doit(tdsn=adpc, outid=%str(t-pkpc-mad), pkflg=%str(pk1fl='Y'), cond=%str(paramcd='GSxxxx' and
cohortn >6 and pcspec='PLASMA' and atptn<8000 and avisitn ne -999), trtgrp=trtagln,
trtgrp=trtag1, trtfmt=tfmt, tdata=tdata, hrfmt=);

```

Program 1. Macro Parameters Setting for Generating 3 PK Concentration Summary Tables

The main body of output file from first call is displayed in Figure 2 below as an example.

Table 15.10.1.1.1: Individual Data and Summary Statistics of Plasma Concentration (ng/mL) at Protocol-Specified Sampling Times

PK Analysis Set  
SAD Cohorts

Treatment header generated using TRTAG1/N from ADPC

Time points generated using DTIMEN and DTIMET from ADPC

Cohort, Treatment and Visit: Cohort 1, xx1 mg GS-xxxx SD fasted, Day 1

Participant ID	Predose	Sampling Time										
		0.25 h	0.5 h	1.0 h	2.0 h	3.0 h	4.0 h	6.0 h	8.0 h	12.0 h	24.0 h	36.0 h
12663-90001	BLQ	297	738	1280	1170	700	528	231	118	39.6	5.02	1.48
12663-90002	BLQ	619	1390	1460	838	542	341	128	69.9	18.8	3.42	1.16
25631-90043	BLQ	352	1190	1330	1300	1520	1320	517	289	99.1	12.2	4.43
25631-90044	BLQ	BLQ	104	748	1730	1290	877	392	233	83.5	11.3	2.73
25631-90045	BLQ	1.17	37.4	517	1230	829	637	280	154	48.9	6.42	2.44
25631-90046	BLQ	67.2	350	791	1620	1040	701	334	229	75.3	13.7	3.81
25631-90047	BLQ	98.3	943	1820	1370	1200	693	299	157	52.8	8.23	1.85
25631-90048	BLQ	527	1090	993	1190	1830	2040	547	229	54.3	6.25	1.99
N	8	8	8	8	8	8	8	8	8	8	8	8
Mean	0	245	730	1120	1310	1120	892	341	185	59.0	8.32	2.49
SD		241	512	431	278	431	545	141	72.2	25.7	3.70	1.13
% CV		98.2	70.2	38.6	21.3	38.5	61.1	41.3	39.0	43.6	44.5	45.6
Median	0	198	841	1140	1270	1120	697	317	193	53.6	7.33	2.22
Min	0	BLQ	37.4	517	838	542	341	128	69.9	18.8	3.42	1.16
Max	0	619	1390	1820	1730	1830	2040	547	289	99.1	13.7	4.43
Q1	0	34.2	227	770	1180	765	583	256	136	44.3	5.64	1.67
Q3	0	440	1140	1400	1500	1410	1100	455	231	79.4	11.8	3.27
N (LN-scale)	0	7	8	8	8	8	8	8	8	8	8	8
Geom. Mean		115	451	1040	1280	1040	776	313	170	53.3	7.56	2.27
% GCV		1090	217	43.4	22.5	42.5	59.4	49.3	48.8	55.6	50.9	47.9
# BLQ	8	1	0	0	0	0	0	0	0	0	0	0
>1/3 BLQ	Yes	No	No	No	No	No	No	No	No	No	No	No
% BLQ	100	12.5	0	0	0	0	0	0	0	0	0	0

Figure 2. Outputs of Summary PK Concentration Tables for SAD (left) and MAD (right) Parts

The macro m\_tpcdn populates input datasets based on defined filter parameters, generates treatments formats, checks if there are user defined time points format, generates time points format if not defined by user (Program 2). The treatment formats and time points formats will be generated and updated dynamically with each call based on selected datasets fitting the file name and data inclusion conditions.

```

%macro m_tpcdn(pkflg=, cond=, tdsn=, tdata=, trtfmt=, trtgrp=, trtgrp=, hrfmt=);

*** Get input dataset;
%fetch(library = adamdata,
data = adsl,
dataopt = where=(&pkflg),
keep = usubjid,
sortby = usubjid);

%fetch(library = adamdata,
data = &tdsn,
dataopt = where=(&pkflg and &cond),
sortby = usubjid);

```

```

*** Manipulate dataset for table generation;
data &tdata;
merge &tdsn(in=ina) adsl(in=inb);
by usubjid;
if ina and inb;
pkyn=1;
analyte=paramcd;
hour=atptn;
proc sort;
by analyte &trtgrp atptn;
run;

*** format for treatment labels;
data &trtfmt;
set &tdata;
retain fmtname "&trtfmt" type 'n';
start=&trtgrp;
end=&trtgrp;
label=&trtgrp;
keep &trtgrp &trtgrp fmtname type start end label;
proc sort nodupkey;
by start;
run;

proc format library=work cntlin=&trtfmt;
run;

*** format if user does not define x-axis label for PK concentration table;
%if %length(&hrfmt)=0 and %upcase(&tdsn)=ADPC %then %do;
%let ahrfmt=ahrfmt;

proc sort data=&tdata out=&ahrfmt(keep=dtimen dtimet) nodupkey; by dtimen; where dtimen>. ; run;

data &ahrfmt;
set &ahrfmt;
retain fmtname "&ahrfmt" type 'n';
start=dtimen;
end=dtimen;
label=dtimet;
keep dtimen dtimet fmtname type start end label;
proc sort nodupkey;
by start;
run;

proc format library=work cntlin=&ahrfmt;
run;
%end;

%mend m_tpcdsn;

```

**Program 2. Macro Populating Input Source Datasets, Generating Treatment and Time Points Formats for PK Concentration Tables**

The treatment formats for the MAD PK concentration are shown in Table 2 with 6 levels, using the values in the new variable TRTAG1 added into ADPC that was derived based on TRTA and collection visits at ADaM programming.

▲	TRTAG1	⑫	TRTAG1N	▲	FMTNAME	▲	TYPE	⑫	START	⑫	END	▲	LABEL
	Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1		6		tfmt		n		6		6		Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1
	Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10		7		tfmt		n		7		7		Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10
	Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1		8		tfmt		n		8		8		Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1
	Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10		9		tfmt		n		9		9		Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10
	Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1		10		tfmt		n		10		10		Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1
	Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10		11		tfmt		n		11		11		Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10

Table 2. Formats Generated for MAD PK Concentration Table

**PK parameter tables:**

With similar programming procedures, 2 PK parameter tables can be generated by below 2 calls with their macro parameters shown in Program 3 below.

```
%doit(tdsn=adpp, outid=t-pkpp-sad, pkflg=%str(pk1fl='Y'), cond=%str(ppcat='GS-xxxx' and cohortn
in (1,2,3,4,5,6) and ppspec='PLASMA'), parmnam=%str(AUCLST 14.6 CMAX 14.6 CLST 14.6 TMAX 14.6),
gparmnam=%str(AUCLST CMAX CLST TMAX), trtgrp=trtagln, trtgrpc=trtagl, trtfmt=tfmt, tdata=tdata);

%doit(tdsn=adpp, outid=t-pkpp-mad, pkflg=%str(pk1fl='Y'), cond = %str(cohortn > 6 and
ppspec='PLASMA' and ppcat='GS-xxxx'), parmnam=%str(AUCLST 14.6 CMAX 14.6 CLST 14.6 TMAX 14.6),
gparmnam=%str(AUCLST CMAX CLST TMAX), trtgrp=trtagln, trtgrpc=trtagl, trtfmt=tfmt, tdata=tdata);
```

Program 3. Macro Parameters Setting to Generate Three PK Parameters Summary Tables

## PK parameter comparison tables:

The PK parameter comparison tables can also be generated by setting macro parameters for SAD, MAD and food effect parts once TRTAGIN is included in ADPP and comparison header is defined with a macro with macro parameter settings. Two delimiter '|' and '#' are used to assign macro parameters to generate comparison listing outputs and intermediate dataset for comparison format and table output, with SAS codes partially showed in Program 4. One format to be displayed as comparison header is illustrated with Table 3 and associated output is shown in Figure 3. The PK parameter comparison header will be updated with each call by taking the values from user's inputs with the macro parameter setting. Since TRTAGIN represents all the treatment groups, the comparison tables for different parts can be generated with different calls that set the macro parameters accordingly.

\*\*\* listing output;

```
%let key=1-&outid.;

/* Macro used to generate PK comparison listing output */
%printset(orient=landscape, titlekey=&key., outname=&key.);

%do i=1 %to &totgrp;
/* totgrp indicates how many comparison pairs */
  %let id=&i;
/* if same group variable applies to all comparison */
  %if %index(&grp,|)=0 %then %do; %let grp=&grp; %end;
/* otherwise, applies different group variable separated by '|' */
  %else %do; %let grp=%scan(&grp, &i, |); %end;

/* '#' is used for getting test treatment value */
  %if %index(&test_refn,|)=0 %then %do; %let test=%scan(&test_refn, 1, #); %end;
  %else %do; %let test=%scan(%scan(&test_refn, &i, |), 1, #); %end;

  %if %index(&test_refn,|)=0 %then %do; %let ref=%scan(&test_refn, 2, #); %end;
  %else %do; %let ref=%scan(%scan(&test_refn, &i, |), 2, #); %end;

/* %qscan() should be used to keep original comparison header */
  %if %index(&labeln,|)=0 %then %do; %let label=&labeln; %end;
  %else %do; %let label=%qscan(&labeln, &i, |); %end;

  %if %index(&condn,|)=0 %then %do; %let cond=&condn; %end;
  %else %do; %let cond=%scan(&condn, &i, |); %end;

  %if %index(&parmn,|)=0 %then %do; %let vars=&parmn; %end;
  %else %do; %let vars=%scan(&parmn, &i, |); %end;

  %if %index(&estn,|)=0 %then %do; %let est=&estn; %end;
  %else %do; %let est=%scan(&estn, &i, |); %end;

  %if %index(&classn,|)=0 %then %do; %let class=&classn; %end;
  %else %do; %let class=%scan(&classn, &i, |); %end;

  %if %index(&modeln,|)=0 %then %do; %let model=&modeln; %end;
  %else %do; %let model=%scan(&modeln, &i, |); %end;

  %mod(id=&id, grp=&grp, test=&test, ref=&ref, label=&label, est=&est, cond=&cond, vars=&vars,
class=&class, model=&model);

%end;
```

```

%pageset(outname=&key.);

*** comparison paramters for MAD;
%comptbl(outid=pkratio-mad, pkflg=%str(pk1fl='Y'), totgrp=4,
  parmn=auclst aucifo cmax|
        auclst aucifo cmax|
        auclst auctau cmax|
        auclst auctau cmax,
  test_refn=8#6| 10#6| 9#7| 11#7,
  labeln=%nrstr(GS-xxxx, Cohort 6 xx mg QD vs Cohort 5 xx mg QD, Day 1)|
        %nrstr(GS-xxxx, Cohort 7 xx mg QD vs Cohort 5 xx mg QD, Day 1)|
        %nrstr(GS-xxxx, Cohort 6 xx mg QD vs Cohort 5 xx mg QD, Day 10)|
        %nrstr(GS-xxxx, Cohort 7 xx mg QD vs Cohort 5 xx mg QD, Day 10),
  estn=-1 1| -1 1| -1 1| -1 1,
  condn=trtagln in (6 8)| trtagln in (6 10)| trtagln in (7 9)| trtagln in (7 11),
  classn=subjid trtagln, modeln=trtagln, grpn=trtagln, random=);

```

Program 4. SAS Program to Read Macro Parameters for Comparison Listings Outputs and Intermediate Dataset for Table Output (top) and Macro Parameter Setting to Generate PK Parameter Comparison Table for SAD Cohorts (bottom)

LABEL	ANALYTE	FMTNAME	TYPE	START	END
GS-xxxx, Cohort 6 xx mg QD vs Cohort 5 xx mg QD, Day 1	01	\$analyte	C	01	01
GS-xxxx, Cohort 7 xx mg QD vs Cohort 5 xx mg QD, Day 1	02	\$analyte	C	02	02
GS-xxxx, Cohort 6 xx mg QD vs Cohort 5 xx mg QD, Day 10	03	\$analyte	C	03	03
GS-xxxx, Cohort 7 xx mg QD vs Cohort 5 xx mg QD, Day 10	04	\$analyte	C	04	04

Table 3. Formats Generated for MAD Cohorts Used for comparison Table

Parameters	Treatment				Statistical Comparison		
	Test		Reference		GLSM Ratio (%)	90% CI	Model rMSE
	n	GLSM	n	GLSM			
-----							
Analyte: GS-xxxx, Cohort 6 xx mg QD vs Cohort 5 xx mg QD, Day 1							
AUClast (h*ng/mL)	8	12500	8	5580	223	(136, 367)	0.564
AUCinf (h*ng/mL)	8	12500	8	5640	222	(135, 365)	0.564
Cmax (ng/mL)	8	2720	8	1210	226	(135, 379)	0.588
-----							
Analyte: GS-xxxx, Cohort 7 xx mg QD vs Cohort 5 xx mg QD, Day 1							
AUClast (h*ng/mL)	8	26900	8	5580	482	(293, 792)	0.565
AUCinf (h*ng/mL)	8	27100	8	5640	481	(292, 791)	0.566
Cmax (ng/mL)	8	5260	8	1210	436	(261, 729)	0.584
-----							
Analyte: GS-xxxx, Cohort 6 xx mg QD vs Cohort 5 xx mg QD, Day 10							
AUClast (h*ng/mL)	8	13000	8	7060	185	(133, 255)	0.369
AUCtau (h*ng/mL)	8	12800	8	6870	186	(134, 259)	0.374
Cmax (ng/mL)	8	2770	8	1550	179	(135, 238)	0.322
-----							
Analyte: GS-xxxx, Cohort 7 xx mg QD vs Cohort 5 xx mg QD, Day 10							
AUClast (h*ng/mL)	8	27700	8	7060	392	(281, 547)	0.379
AUCtau (h*ng/mL)	8	27200	8	6870	395	(282, 555)	0.385
Cmax (ng/mL)	8	5450	8	1550	352	(261, 474)	0.339

Figure 3. Output of PK Parameter Comparison Table for MAD Cohorts

## APPLYING NEW GROUPING VARIABLES TO STANDARDIZE AND PROMOTE AUTOMATION FOR PK FIGURE GENERATION

There are more challenges to generate PK concentration figures compared to PK tables programming due to treatment value to be re-adjustments from each call, 2 sets of treatments formats and exclusion of time points with more diverse treatment groups to be included with the global macro requirements. To make format generation for treatment grouping variables more flexible and dynamic, TRTAG2/N variables are added into ADPC, with delimiter '\$' being added into TRTAG2 to differentiate longer and short treatment labels. If TRTAG2/N cannot fit all requirements, TRTAG3/N and TRTAG4/N could also be added accordingly. With the test study, TRTAG2/N can fit the needs on the treatment formats for PK concentration figures. Due to consideration of comparison sequence, the global PK figure macro requires to reset treatment numeric variable starting from 1 for each call. While all collected time points are shown in the figure, there may not be always enough room to display every collected time point on the x-axis. As shown in the Figure 1, time points 0.25, 0.5, 1 and 3 hours postdose have been excluded from x-axis and the counts of total and nonBLQ counts, since there are no enough spaces between 0-2 and 2-4 hour intervals.

To facilitate all the requirements on PK concentration programming, a macro `m_fdsn` is generated and used to populate input datasets, check and define 2 sets of treatment labels if in need, reset treatment numeric variables starting from 1, check and define displayed points that should be included (Program 5).

```
%macro m_fdsn(pkflg=, cond=, fdsn=, fdata=, trtfmt=, annofmt=, trtgrpn=, trtgrpc=, xfmt=,
excltptn=);

*** Get input dataset;
%fetch(library = adamdata,
        data = adsl,
        dataopt = where=(&pkflg),
        keep = usubjid,
        sortby = usubjid);

%fetch(library = adamdata,
        data = &fdsn,
        dataopt = where=(&pkflg and &cond),
        sortby = usubjid);

*** Manipulate dataset for figure generation;
%if %upcase(&fdsn)=ADPC %then %do;
data &fdata;
merge &fdsn(in=ina) adsl(in=inb);
by usubjid;
if ina and inb;
pkyn=1;
analyte=paramcd;
hour=atptn;
subj=substr(subjid,7);
subject=strip(subjid);
if &trtgrpn>.;
proc sort;
by analyte &trtgrpn atptn;
run;
%end;

*** Get unique treatment group variables;
proc sort data=&fdata out=pktrtn(keep=&trtgrpn &trtgrpc) nodupkey; by &trtgrpn; run;

data pktrtn;
set pktrtn;
pktrtn=_n_;
run;

data chklabel;
set pktrtn;
by &trtgrpn;
if _n_=1;
run;

*** check if there are 2nd set of label defined in the source dataset;
data _null_;
```

```

    set chklabel;
    if index(&trtgrpc, '$') then call symput('chklabel', 'Y');
run;

%put chklabel=&chklabel;

*** treatment format for 2 sets of labels;
%if &chklabel=Y %then %do;
data &trtfmt;
    set pktrtn;
    retain fmtname "&trtfmt" type 'n';
    start=_n_;
    end=_n_;
    label=strip(scan(&trtgrpc, 1, '$'));
    keep &trtgrpn &trtgrpc fmtname type start end label;
proc sort nodupkey;
    by start;
run;
proc format library=work cntlin=&trtfmt;
run;

data &annofmt;
    set pktrtn;
    retain fmtname "&annofmt" type 'n';
    start=_n_;
    end=_n_;
    label=strip(scan(&trtgrpc, 2, '$'));
    keep &trtgrpn &trtgrpc fmtname type start end label;
proc sort nodupkey;
    by start;
run;
proc format library=work cntlin=&annofmt;
run;
%end;

/* treatment format for one set of label*/
%else %do;
data &trtfmt;
    set pktrtn;
    retain fmtname "&trtfmt" type 'n';
    start=_n_;
    end=_n_;
    label=&trtgrpc;
    keep &trtgrpn &trtgrpc fmtname type start end label;
proc sort nodupkey;
    by start;
run;

proc format library=work cntlin=&trtfmt;
run;

%end;

proc sql;
    create table &fdata as
    select a.*, b.pktrtn
    from &fdata a left join pktrtn b
    on a.&trtgrpn=b.&trtgrpn;
quit;

*** format if user does not define x-axis label;
%if %length(&xfmt)=0 %then %do;
%let axfmt=axfmt;

%if %length(&excltptn)=0 %then %do;
    proc sort data=&fdata out=&axfmt(keep=dtimef) nodupkey; by dtimef; where dtimef>.; run;
%end;

%if %length(&excltptn)>0 %then %do;
    proc sort data=&fdata out=&axfmt(keep=dtimef) nodupkey; by dtimef; where dtimef not in
    (&excltptn) and dtimef>.; run;

```

```

%end;

data &axfmt;
  set &axfmt;
  retain fmtname "&axfmt" type 'n';
  start=dtimen;
  end=dtimen;
  label=dtimelf;
  keep dtimen dtimelf fmtname type start end label;
proc sort nodupkey;
  by start;
run;
proc format library=work cntlin=&axfmt;
run;
%end;

%mend m_fdsn;

```

**Program 5. Macro Populating Input Source Datasets, Generating Treatment Labels and Time Points Formats for PK Concentration Figures**

Three sets of formats can be defined and refreshed with each call if there are no user defined formats for the displayed time points (Table 4). In the case there are no delimiters in TRTAG2, short treatment labels will not be defined and same treatment labels will be applied on the top and bottom treatment legends.

TRTAG2	TRTAG2N	FMTNAME	TYPE	START	END	LABEL
Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1\$Cohort 5 Day 1	6	ffmt	n	1	1	Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1
Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10\$Cohort 5 Day 10	7	ffmt	n	2	2	Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10
Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1\$Cohort 6 Day 1	8	ffmt	n	3	3	Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1
Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10\$Cohort 6 Day 10	9	ffmt	n	4	4	Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10
Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1\$Cohort 7 Day 1	10	ffmt	n	5	5	Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1
Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10\$Cohort 7 Day 10	11	ffmt	n	6	6	Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10

TRTAG2	TRTAG2N	FMTNAME	TYPE	START	END	LABEL
Cohort 5, xx1 mg GS-xxxx QD fasted, Day 1\$Cohort 5 Day 1	6	f2fmt	n	1	1	Cohort 5 Day 1
Cohort 5, xx1 mg GS-xxxx QD fasted, Day 10\$Cohort 5 Day 10	7	f2fmt	n	2	2	Cohort 5 Day 10
Cohort 6, xx2 mg GS-xxxx QD fasted, Day 1\$Cohort 6 Day 1	8	f2fmt	n	3	3	Cohort 6 Day 1
Cohort 6, xx2 mg GS-xxxx QD fasted, Day 10\$Cohort 6 Day 10	9	f2fmt	n	4	4	Cohort 6 Day 10
Cohort 7, xx3 mg GS-xxxx QD fasted, Day 1\$Cohort 7 Day 1	10	f2fmt	n	5	5	Cohort 7 Day 1
Cohort 7, xx3 mg GS-xxxx QD fasted, Day 10\$Cohort 7 Day 10	11	f2fmt	n	6	6	Cohort 7 Day 10

DTIMEF	DTIMEN	FMTNAME	TYPE	START	END	LABEL
0	0	axfmt	n	0	0	0
2	2	axfmt	n	2	2	2
4	4	axfmt	n	4	4	4
6	6	axfmt	n	6	6	6
8	8	axfmt	n	8	8	8
12	12	axfmt	n	12	12	12
24	24	axfmt	n	24	24	24
36	36	axfmt	n	36	36	36
48	48	axfmt	n	48	48	48
72	72	axfmt	n	72	72	72

**Table 4. Formats generated for MAD PK concentration figure with full treatment labels (top), short treatment labels (middle), and displayed time points (bottom)**

Once treatments and displayed time points formats are defined, global PK concentration figure macro can be called with determined macro parameters, which allows to generate different type of PK concentration figures in the linear and semi-log scales, including mean, median, and individual figures. Program 5 shows the macro setting for 4 PK concentration figures. Figure 4 is the output with third call taking the formats as shown in Table 4.



## CONCLUSION

By adding treatment group and displayed time points variables into ADPC and/or ADPP, the formats for treatment and time points can be defined dynamically for PK TFLs with macro calls that are adjusted with corresponding macro parameter settings. This can eliminate the needs of defining those formats at TFLs programming level and allows validation to be performed earlier at ADaM programming stage as well. This approach can simplify the programming procedures and have PK TFLs generated by modifying macro parameters across different studies and ensure consistency of PK treatment variables and timepoints across ADPC, ADPP and PK TFLs. Those macro parameters can also be entered with Excel to automate both safety and PK TFLs generated with batch run and increase programming efficiency and accuracy.

## REFERENCES

Clinical Data Interchange Standards Consortium (CDISC). November 2021. Analysis Data Model Implementation Guide Version 1.3 (Final). Available at <https://www.cdisc.org/standards/foundational/adam/adamig-v1-3>

## ACKNOWLEDGMENTS

We would like to thank Krishna Sivakumar, Sr. Director, Statistical Programming, for her encouragement and guidance and providing constructive suggestions throughout the paper development. We would also like to thank Macro Programming Team, STARS Program Team, Phase 1 Programming Team and Statistical Programming Department for their support.