PharmaSUG China 2024 - Paper AA-10089

MetaTFL - A Dynamic Configuration Application for a Centralized Clinical Analysis Platform

Cheng Zhang, BeiGene (Beijing) Co.,Ltd.

ABSTRACT

MetaTFL is a dynamic configuration application designed to support centralized clinical analysis platforms, enabling real-time updates to configuration files to expand ADaM dataset generation and diversify TFL (Tables, Figures, and Listings) presentations. MetaTFL allows for dynamic setting of ADaM datasets and TFL generation parameters, which are then differentially displayed on the centralized clinical analysis platform (LiveTFL).

The application reads traditional ADaM datasets and uses metaprogramming to preprocess and generate a series of analytical variables for functional modules. Each specific TFL is configured with module parameters and scripts, providing previews and real-time interactive parameter adjustments. User pages are differentiated through user permission file configurations.

This process greatly enhances the scalability of LiveTFL, enabling the rapid generation of TFLs for multiple trial designs and significantly reducing the time required to prepare TFLs for projects. The configuration file approach ensures that metadata updates are decoupled from the Shiny application itself, allowing for more timely responses. Additionally, the control settings logic makes it easy for non-programming users to update configuration files.

INTRODUCTION

The ADaM dataset supports statistical analyses performed in clinical studies. However, in order to generate TFLs for various purposes, we need to further process the ADaM dataset, including but not limited to deriving new variables, changing variable labels, and generating a range of marker variables. Similarly, for a specific TFL, take a demographic table for example, different studies may wish to analyze different variables, and if different default parameters can be assigned to the TFLs of different studies, it is possible to manage the TFLs of multiple studies on the same platform.

This paper will be divided into the following sections. The first part will give a background of the platform. The second part will describe how to use the platform. The first step will be how to manage user rights, as well as implementing the creation of TFL lists for different studies and analyses and validating the availability of the selected TFLs based on all the ADaMs that already exist. The second step describes how to implement ADaM dataset preprocessing using metaprogramming techniques in R, including generating intermediate variables for the ADaM dataset. The third step is to set default filter conditions for the data. The fourth step describes how to use parameterized forms to interact with modular functions in real time and dynamically preview TFL results. Then introduce how to perform configuration file downloads and view global database data. The third section will describe future development plans, including AI-assisted automated configuration, applicability to pooled studies, and dynamic updating of configurations based on changes in the ADaMs dataset or specification.

BACKGROUND

We have built a real-time centralized TFL platform (LiveTFL). The platform needs to be able to display differentiated analysis results for different analysis purposes for various studies. However, in order to produce standardized TFLs, some data preprocessing of ADaMs is required. In a way to make the platform accessible to programmers who are non-R or R beginners, we consider using metaprogramming to implement data preprocessing. Also, in order to diversity the results of TFLs, we can set up some parameters in the metadata form, and then transfer them to the developed module function to use.

LIVETFL

A centralized clinical analysis platform that support diverse TFLs output for over 20 studies.

Here's a demo of the demographic table. The dummy data is from scda.2022 r package.

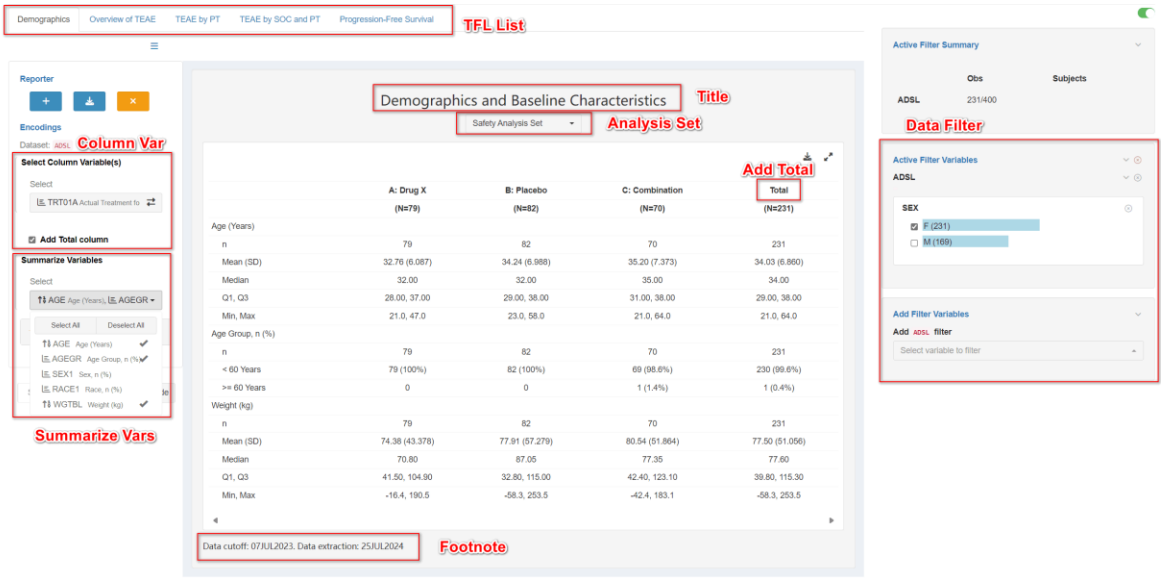


Figure 1. A Demo of the Demographic Table

METATFL

HOMEPAGE

The homepage is used to select specific study and analysis and then update/add the required TFLs.

1. Select compound and study. Through user access management, different users can only select a specific number of studies.

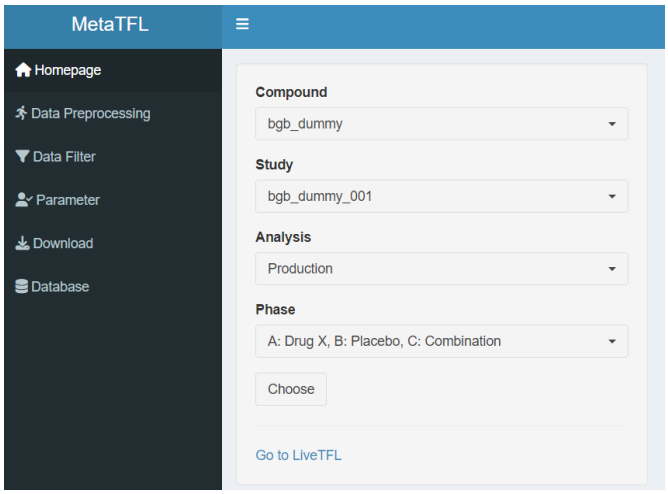


Figure 2. Homepage

2. Select analysis. For different analysis purposes, you can set up different numbers of TFLs.

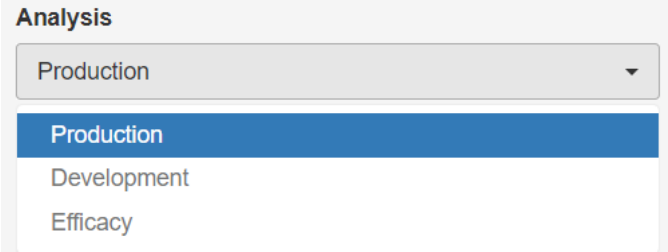


Figure 3. Select Analysis

3. Select phase. A variable in ADSL can be specified to take a subset of the data for subgroup analysis. Note that the filtering of data here cannot be undone in the subsequent process.

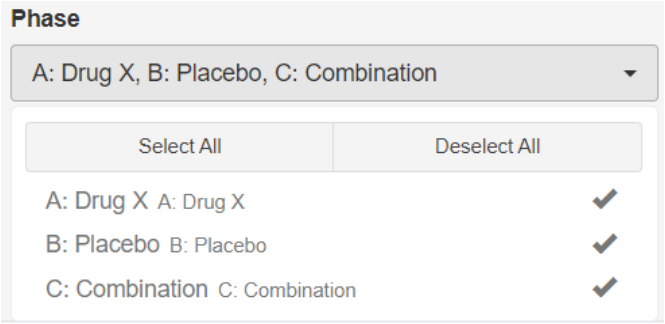


Figure 4. Select Phase

4. Clicking on the choose button will display the currently selected TFLs and allows you to add or delete the specified TFLs. Then the server will print out the corresponding data path of study, all available ADaMs under the current path, and validate the availability of the selected TFLs based on all existing ADaMs.

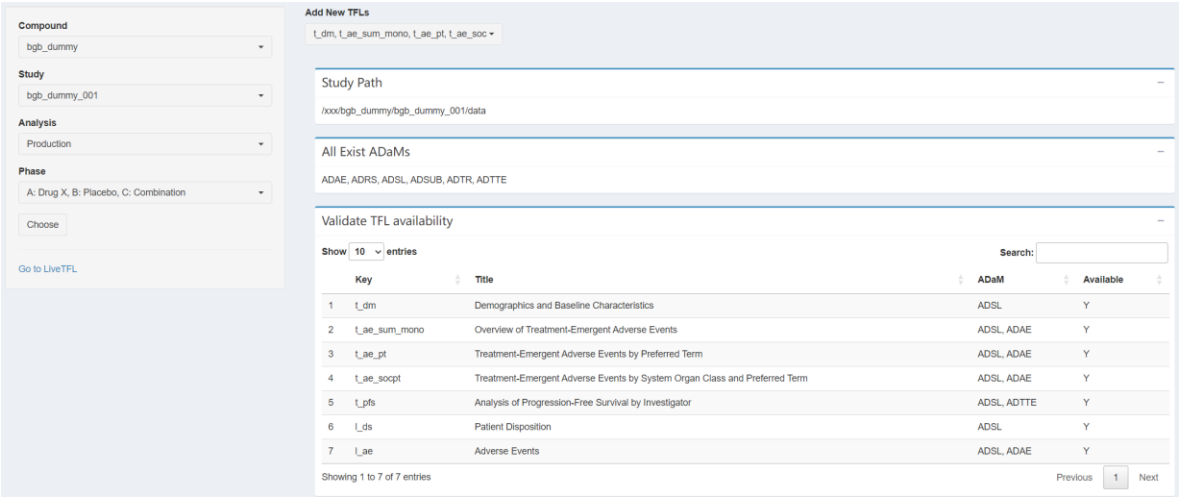
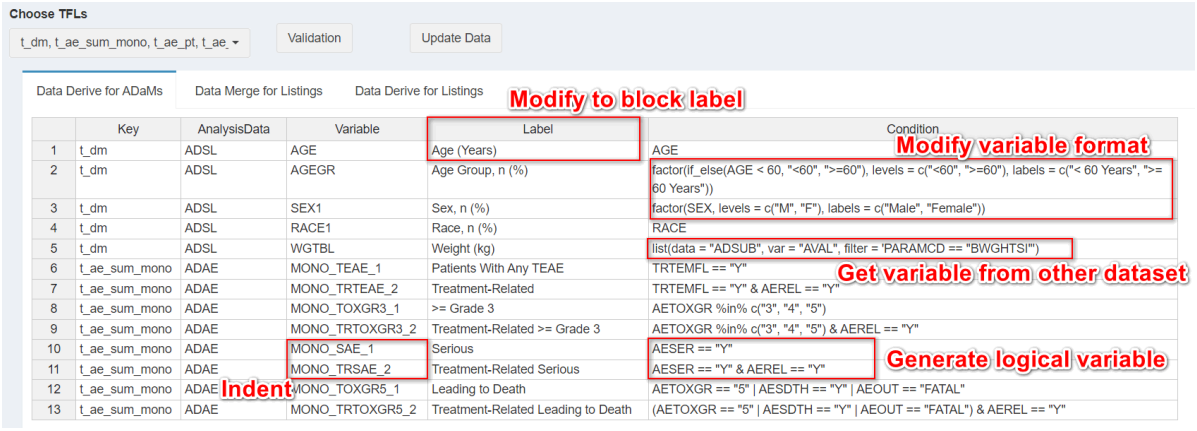


Figure 5. Click on the Choose Button

DATA PREPROCESSING

Preprocessing will generate variables and datasets for TFLs.

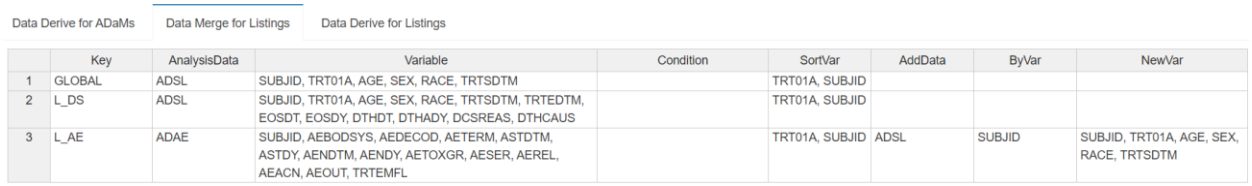
1. “Data Derive for ADaMs” Tab will generate intermediate variables for TFLs.



	Key	AnalysisData	Variable	Label	Condition
1	t_dm	ADSL	AGE	Age (Years)	AGE
2	t_dm	ADSL	AGEGR	Age Group, n (%)	factor(if_else(AGE < 60, "<60", ">=60"), levels = c("<60", ">=60"), labels = c("< 60 Years", ">= 60 Years"))
3	t_dm	ADSL	SEX1	Sex, n (%)	factor(SEX, levels = c("M", "F"), labels = c("Male", "Female"))
4	t_dm	ADSL	RACE1	Race, n (%)	RACE
5	t_dm	ADSL	WGTBL	Weight (kg)	list(data = "ADSUB", var = "AVAL", filter = 'PARAMCD == "BWGHTSI"')
6	t_ae_sum_mono	ADAE	MONO_TEAE_1	Patients With Any TEAE	TRTEMFL == "Y"
7	t_ae_sum_mono	ADAE	MONO_TRTEAE_2	Treatment-Related	TRTEMFL == "Y" & AEREL == "Y"
8	t_ae_sum_mono	ADAE	MONO_TOXGR3_1	>= Grade 3	AETOXGR %in% c("3", "4", "5")
9	t_ae_sum_mono	ADAE	MONO_TRTOXGR3_2	Treatment-Related >= Grade 3	AETOXGR %in% c("3", "4", "5") & AEREL == "Y"
10	t_ae_sum_mono	ADAE	MONO_SAE_1	Serious	AESER == "Y"
11	t_ae_sum_mono	ADAE	MONO_TRSAE_2	Treatment-Related Serious	AESER == "Y" & AEREL == "Y"
12	t_ae_sum_mono	ADAE	MONO_TOXGR5_1	Leading to Death	AETOXGR == "5" | AESDTH == "Y" | AEOUT == "FATAL"
13	t_ae_sum_mono	ADAE	MONO_TRTOXGR5_2	Treatment-Related Leading to Death	(AETOXGR == "5" | AESDTH == "Y" | AEOUT == "FATAL") & AEREL == "Y"

Figure 6. Data Derive for ADaMs

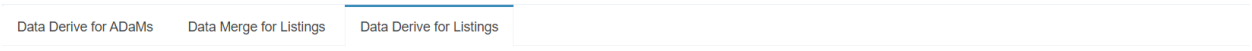
2. “Data Merge for Listings” Tab will generate a new dataset for each listing. It will be presented in the form of a data table, which will allow users to filter and review the listing in a better way. And we'll use a GLOBAL dataset (similar to ADSL) to respond to all listings.



	Key	AnalysisData	Variable	Condition	SortVar	AddData	ByVar	NewVar
1	GLOBAL	ADSL	SUBJID, TRT01A, AGE, SEX, RACE, TRTSDTM		TRT01A, SUBJID			
2	L_DS	ADSL	SUBJID, TRT01A, AGE, SEX, RACE, TRTSDTM, TRTEDTM, EOSDT, EOSDY, DTHDT, DTHADY, DCSREAS, DTHCAUS		TRT01A, SUBJID			
3	L_AE	ADAE	SUBJID, AEBODSYS, AEDECOD, AETERM, ASTDTM, ASTDY, AENDTM, AENDY, AETOXGR, AESER, AEREL, AEACN, AEOUT, TRTEMFL		TRT01A, SUBJID	ADSL	SUBJID	SUBJID, TRT01A, AGE, SEX, RACE, TRTSDTM

Figure 7. Data Merge for Listings

3. “Data Derive for Listings” tab will modify the variables for each list.



	Key	AnalysisData	Variable	Label	Condition
1	global	GLOBAL	SUBJID	Patient ID	SUBJID
2	global	GLOBAL	TRT01A	Dose Level	TRT01A
3	global	GLOBAL	AGE	Age (Years)	AGE
4	global	GLOBAL	SEX	Sex	SEX
5	global	GLOBAL	RACE	Race	RACE
6	global	GLOBAL	TRTSDTM	First Dose Date	as.Date(TRTSDTM)
7	l_ds	L_DS	SUBJID	Patient ID	SUBJID
8	l_ds	L_DS	TRT01A	Dose Level	TRT01A
9	l_ds	L_DS	AGE	Age (Years)	AGE
10	l_ds	L_DS	SEX	Sex	SEX
11	l_ds	L_DS	RACE	Race	RACE
12	l_ds	L_DS	TRTSDTM	First Dose Date	as.Date(TRTSDTM)
13	l_ds	L_DS	TRTEDTM	Last Dose Date	as.Date(TRTEDTM)
14	l_ds	L_DS	EOSDT	Study Discontinuation Date	EOSDT
15	l_ds	L_DS	EOSDY	Study Discontinuation Days	EOSDY
16	l_ds	L_DS	DTHDT	Death Date	DTHDT
17	l_ds	L_DS	DTHADY	Death Days	DTHADY
18	l_ds	L_DS	DCSREAS	Study Discontinuation Reason	DCSREAS
19	l_ds	L_DS	DTHCAUS	Death Reason	DTHCAUS

Figure 8. Data Derive for Listings

We use rhandsontable to edit cell contents and perform row manipulation. And will validate the condition before execution.

R code for meta-programming:

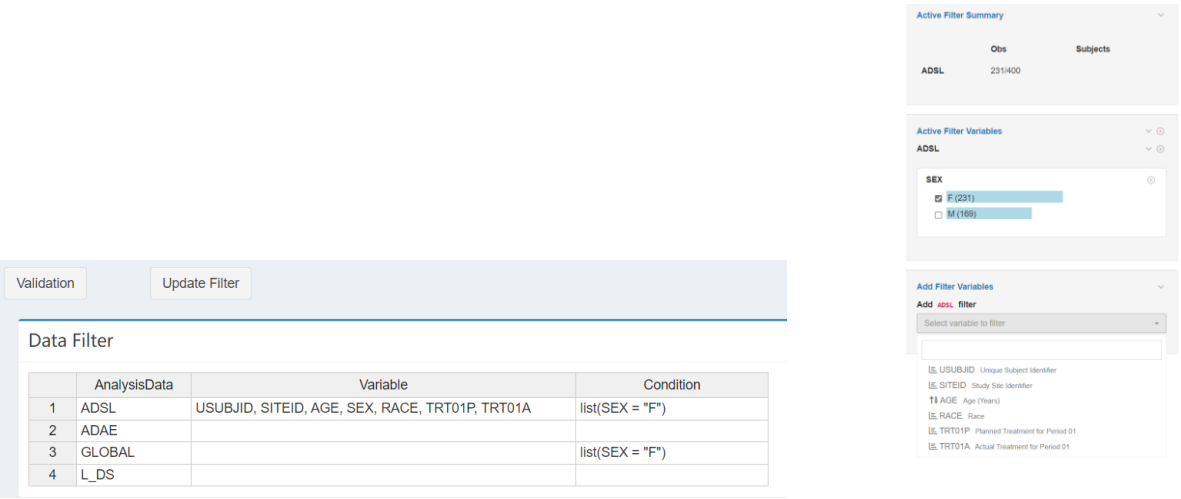
```
data_derive_expr <- function(AnalysisData, Variable, Label, Condition) {
  if (!str_detect(Condition, "^list\\(")) {
    expr <- str_glue(
```

```
      '{AnalysisData} <<- {AnalysisData} %>% mutate({Variable} =
with_label({Condition}, "{Label}"))'
    ) %>% rlang::parse_expr()
  } else if (!Variable %in% colnames(get(AnalysisData))) {
    cond <- str_squish(Condition) %>% rlang::parse_expr() %>% eval()
    if (!"data" %in% names(cond)) {cond$data = AnalysisData}
    if (!"filter" %in% names(cond)) {cond$filter = "1==1"}
    if (!"by" %in% names(cond)) {cond$by = "SUBJID"}
    expr <- str_glue(
      '{AnalysisData} <<- {AnalysisData} %>%
       admiral::derive_vars_merged(
         dataset_add = {cond$data},
         by_vars = exprs({cond$by}),
         new_vars = exprs({Variable} = with_label({cond$var}, "{Label}")),
         filter_add = {cond$filter}
       )'
    ) %>% rlang::parse_expr()
  } else if (Variable %in% colnames(get(AnalysisData))) {
    expr <- str_glue(
      '{AnalysisData} <<- {AnalysisData} %>% mutate({Variable} =
with_label({Variable}, "{Label}"))'
    ) %>% rlang::parse_expr()
  }
  # print(expr)
  eval(expr)
}

data_merge_expr <- function(Key, AnalysisData, Variable, Condition,
SortVar, AddData, ByVar, NewVar) {
  if (AddData == "") {
    expr <- str_glue(
      '{Key} <<- {AnalysisData} %>% filter({Condition}) %>%
select({Variable}) %>% arrange({SortVar})'
    ) %>% rlang::parse_expr()
  } else {
    expr <- str_glue(
      '{Key} <<- {AddData} %>% select({NewVar}) %>% inner_join(
         {AnalysisData} %>% filter({Condition}) %>% select({Variable}), by
= "{ByVar}", multiple = "all"
       ) %>% arrange({SortVar}) %>% var_relabel(SUBJID = "Patient ID")'
    ) %>% rlang::parse_expr()
  }
  # print(expr)
  eval(expr)
}
```

DATA FILTER

The data filter will set up the variable list that users can use for filtering (it will not actually drop the rest of the variables), and you can set up default conditions for filtering.



	AnalysisData	Variable	Condition
1	ADSL	USUBJID, SITEID, AGE, SEX, RACE, TRT01P, TRT01A	list(SEX = "F")
2	ADAE		
3	GLOBAL		list(SEX = "F")
4	L_DS		

Figure 9. Data Filter

PARAMETER

It will be used to set up parameters for each TFL based on the corresponding module function. And only value is allowed to be modified.

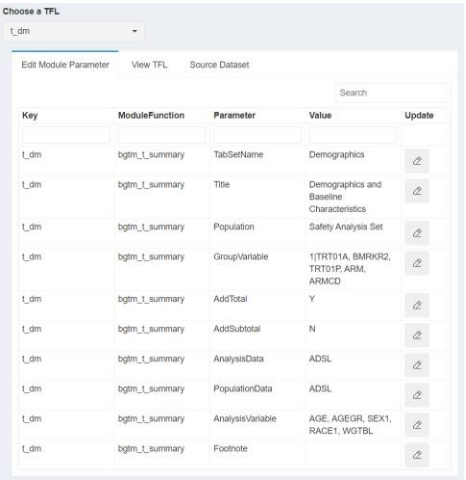


Figure 10. Edit Module Parameter

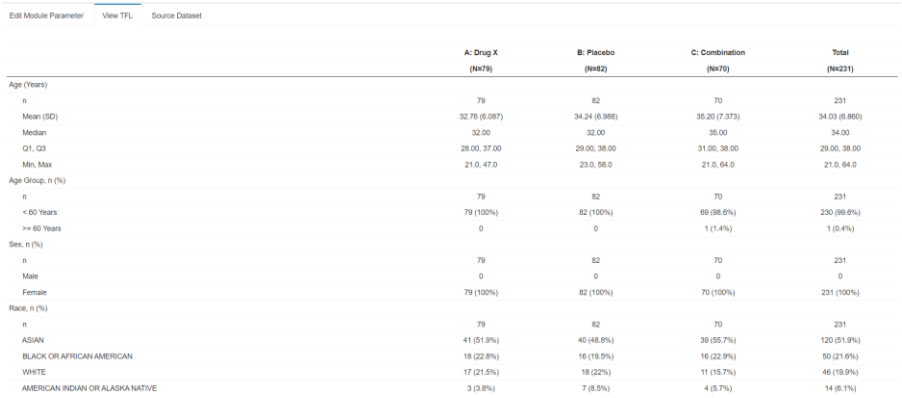
```r
# Generate all exist module parameters
parameter_list <- get_parameter(module = "t_dm")

dm_anl_spec <- data_extract_spec(
  dataname = "ADSL",
  select = select_spec(
    choices = variable_choices(ADSL, parameter_list$anl_vars),
    selected = parameter_list$anl_vars,
    multiple = TRUE,
    fixed = FALSE
  )
)

t_dm <- bgtm_t_summary(
  label          = parameter_list$label,
  dataname       = parameter_list$dataname,
  parentname     = parameter_list$parentname,
  arm_var        = parameter_list$arm_vars,
  add_total      = parameter_list$add_total,
  add_subtotal   = parameter_list$add_subtotal,
  summarize_vars = dm_anl_spec,
  useNA          = "no",
  pre_output     = div(align="center", h3(parameter_list$title)),
  post_output    = parameter_list$post_output,
  population     = parameter_list$population
)
```

Figure 11. R Script to Accept Parameter

Then you can view TFL result in real time.

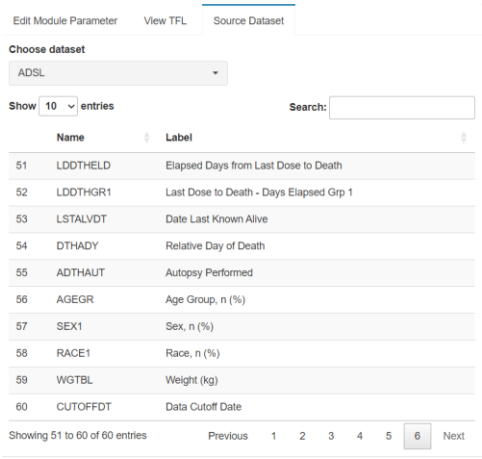


Edit Module Parameter | View TFL | Source Dataset

	A: Drug X (N=79)	B: Placebo (N=82)	C: Combination (N=70)	Total (N=231)
Age (Years)				
n	79	82	70	231
Mean (SD)	32.76 (6.087)	34.24 (6.988)	35.20 (7.373)	34.03 (6.860)
Median	32.00	32.00	35.00	34.00
Q1, Q3	28.00, 37.00	29.00, 38.00	31.00, 38.00	29.00, 38.00
Min, Max	21.0, 47.0	23.0, 58.0	21.0, 64.0	21.0, 64.0
Age Group, n (%)				
n	79	82	70	231
< 60 Years	79 (100%)	82 (100%)	69 (98.6%)	230 (99.6%)
>= 60 Years	0	0	1 (1.4%)	1 (0.4%)
Sex, n (%)				
n	79	82	70	231
Male	0	0	0	0
Female	79 (100%)	82 (100%)	70 (100%)	231 (100%)
Race, n (%)				
n	79	82	70	231
ASIAN	41 (51.9%)	40 (48.8%)	39 (55.7%)	120 (51.9%)
BLACK OR AFRICAN AMERICAN	18 (22.8%)	16 (19.5%)	16 (22.9%)	50 (21.6%)
WHITE	17 (21.5%)	18 (22%)	11 (15.7%)	46 (19.9%)
AMERICAN INDIAN OR ALASKA NATIVE	3 (3.8%)	7 (8.5%)	4 (5.7%)	14 (6.1%)

Figure 12. View TFL

The “Source Dataset” tab provides a helpful variable list for all available datasets. It is usually the name and label of the variables, and for BDS structured datasets it is PARAMCD and PARAM.



Edit Module Parameter | View TFL | Source Dataset

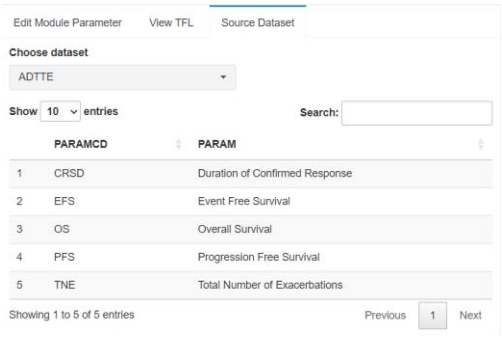
Choose dataset: ADSL

Show 10 entries Search:

	Name	Label
51	LDDTHELD	Elapsed Days from Last Dose to Death
52	LDDTHGR1	Last Dose to Death - Days Elapsed Grp 1
53	LSTALVDT	Date Last Known Alive
54	DTHADY	Relative Day of Death
55	ADTHAUT	Autopsy Performed
56	AGEGR	Age Group, n (%)
57	SEX1	Sex, n (%)
58	RACE1	Race, n (%)
59	WGTBL	Weight (kg)
60	CUTOFFDT	Data Cutoff Date

Showing 51 to 60 of 60 entries Previous 1 2 3 4 5 6 Next

Figure 13. Source Dataset for ADSL



Edit Module Parameter | View TFL | Source Dataset

Choose dataset: ADTTE

Show 10 entries Search:

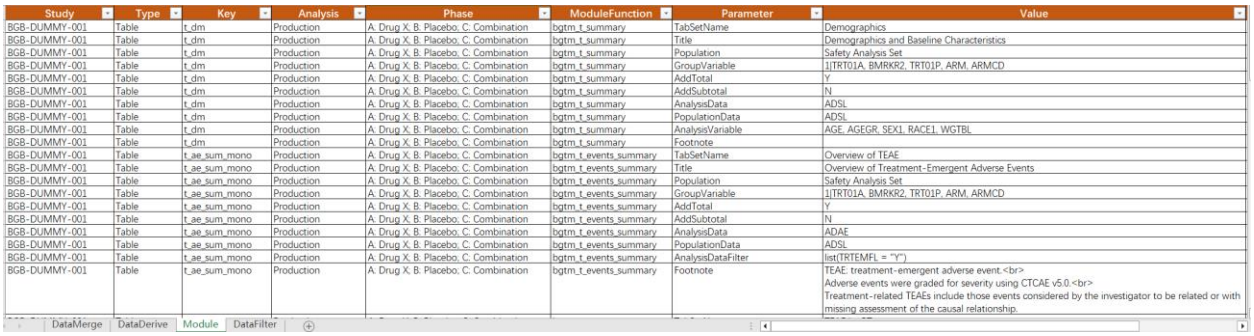
	PARAMCD	PARAM
1	CRSD	Duration of Confirmed Response
2	EFS	Event Free Survival
3	OS	Overall Survival
4	PFS	Progression Free Survival
5	TNE	Total Number of Exacerbations

Showing 1 to 5 of 5 entries Previous 1 Next

Figure 14. Source Dataset for ADTTE

DOWNLOAD

You can download configuration file in excel format to back up for review and reproduction.



Study	Type	Key	Analysis	Phase	ModuleFunction	Parameter	Value
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	TabSetName	Demographics
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	Title	Demographics and Baseline Characteristics
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	Population	Safety Analysis Set
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	GroupVariable	1[TRT01A, BMRKR2, TRT01P, ARM, ARMCD
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	AddTotal	Y
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	AddSubtotal	N
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	AnalysisData	ADSL
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	PopulationData	ADSL
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	AnalysisVariable	AGE, AGEGR, SEX1, RACE1, WGTBL
BGB-DUMMY-001	Table	t_dm	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_summary	Footnote	
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	TabSetName	Overview of TEAE
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	Title	Overview of Treatment-Emergent Adverse Events
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	Population	Safety Analysis Set
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	GroupVariable	1[TRT01A, BMRKR2, TRT01P, ARM, ARMCD
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	AddTotal	Y
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	AddSubtotal	N
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	AnalysisData	ADAE
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	PopulationData	ADSL
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	AnalysisDataFilter	list(TRTEMFL = "Y")
BGB-DUMMY-001	Table	t_ae_sum_mono	Production	A. Drug X, B. Placebo, C. Combination	bgtm_t_events_summary	Footnote	TEAE: treatment-emergent adverse event. Adverse events were graded for severity using CTCAE v5.0. Treatment-related TEAEs include those events considered by the investigator to be related or with missing assessment of the causal relationship.

Figure 15. Download File

DATABASE

We have series of data to support the whole application, and users have read-only permission.

- Authority file: it is used to control user access. Users can see all currently available study and analysis for themselves.
- Module Function: it gives all available TFL list and corresponding functions.
- Module Parameter: it gives parameter conversion and reference values.
- Analysis Set: centralized population set.
- Cancer Type: centralized cancer type and abbreviation.

FUTURE

We also have something we plan to do in the future, including and not limited to:

- AI-assisted automation configuration.
- Applicable for pooling study.
- Dynamically update the configuration according to changes in the ADaMs dataset or specification.

REFERENCES

Website teal: Interactive Exploratory Data Analysis with Shiny Web-Applications. https://github.com/insightsengineering/teal.

Website TLG Catalog. TLG Catalog - Stable (insightsengineering.github.io)

CONTACT INFORMATION

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

Cheng Zhang
BeiGene (Beijing) Co.,Ltd.
18120776522
cheng2.zhang@beigene.com