CONFIDENTIAL

Testing protocol: Data

Submitted to:	Author:
Jeff Hane, PhD Metrum Research Group LLC	Daniel G. Polhamus, PhD Senior Scientist Metrum Research Group LLC 2 Tunxis Road, Suite 112 Tariffville, CT Phone: 860-372-7988 Fax: 860-760-6014 Email: danp@metrumrg.com
Initiator Submitted to QA / Date (Sign and print name)	
QA Approval to Proceed / Date (Sign and print name)	

Definitions

Purpose

To validate Data requirement of the Pharmacometrics TFL Generator app.

NONMEM run data (tab and partab files) can be read, displayed, and summarized

The application creates summary objects based on NONMEM output and source data. This step ensures that NONMEM output can be read into the app and displayed. Additionally, a quick summary of the loaded data object assists the scientist in examining the data; that functionality is tested here as well. NONMEM output (0069.tab) is read in and recorded for this step.

Run data can be manipulated using the code parser

A code parser is included so that the scientist can run R code against loaded data objects. The code parser can be run against the NONMEM output data. This step tests creating a factor and subsetting using the application against the NONMEM output data.

Source data can be read, displayed, and summarized

This step ensures that source data can be read into the app and displayed. Additionally, a quick summary of the loaded data object assists the scientist in examining the data; that functionality is tested here as well. A fake source file was created to test the merge with the 0069.tab output data. This data (source.csv) has a longitudinal covariate added to it (WGT) that is missing for a single study. The merge is intended to be a full merge, so no data loss should occur because of the absence of the variable for a subset of the patients.

This step verifies that source.csv can be read, displayed, and summarized in the application.

Source data can be manipulated using the code parser

The code parser specific for the source data is tested here, and may be used for data wrangling to enable the merge between the NONMEM output and source data. The same test is run here as in the NONMEM output data step. Data is subset down to just one subject, and that is verified in the data summary.

Analysis data can be created by merging run data and source data

Once the scientist is satisfied with the NONMEM output data and source data, a merge occurs between the two to create the analysis data. The analysis data is carried forward and used with all subsequent steps in the app. This step firsts reverts back to the full datasets for each of the NONMEM output and source data, then merges the data. Study 183 (missing the WGT variable) is then checked for presence in the merged dataset. If present, the merge is considered successful.

Analysis data can be manipulated using the code parser

Oftentimes it is easier for the analyst to manipulate the merged data (e.g., creating factors, new variables that are transformations of others). This is tested similarly to the other parsers, but a new variable for patient gender is created.

Analysis data can be viewed and summarized

After the merge, the analysis data should be viewable as should its summary. Both are shown here as screen captures.

Subject level exclusions can be specified and viewed

The application allows the user to specify subjects to be excluded from the analysis dataset, as well as including exclusion reasons. The tester defines a variable here that marks all patients with missing race information for exclusion at the subject level. Those patients are viewed and presented as verification.

Observation level exclusions can be specified and viewed

The application allows the user to specify observations to be excluded from the analysis dataset, as well as including exclusion reasons. The tester defines a variable here that marks all observations with concentration below 0.05 as BQL exclusion at the subject level. Those patients are viewed and presented as verification.

Data cache can be cleared from the app

Data caching is used to provide a more pleasant user experience, but the downside is that the cached data must be manually reset if the user wishes to load a different dataset. In this step, the user clears the cache and verfies no data is present via the data viewing tabs.

Testing procedures

Testing procedures are outlined in the attached testing document.

References and supporting documents

- Requirements document and overview: `tflgenerator_Requirements_R2.pdf`

Testing log

RID	Topic	Test ID	Step Description	Expected Result	Qualification Note(s)	Pass/Fail
3	NONMEM run data (tab and partab files) can be read, displayed, and summarized	1	Upload run data 0069 into /data via Rstudio	Upload successful		
		2	Load run data into the application	Screenshot of Data Input -> Model info and Data Input -> Change E-R SSAP Defaults		
		3	View the data	Screenshot of Data -> Run Data, showing data contents		
		4	View the data summary	Screenshot of Data -> Run Data Summary, showing data summary		
4	Run data can be manipulated using the code parser	1	Input parsing code: enter the following into Data Input -> Modify Data -> Table data manipulation code: ROUTF <- factor(ROUT, c(1,2), c("IV","SC")) subset($DATA, ID == 1, select=c(ID,TIME,EVID,STUD,ROUTF))	Input		
		2	Screenshot of Run Data view	Screenshot shows the selected subset of patients and variables, with the renamed Route factor		
5	Source data can be read, displayed, and summarized	1	Uplaod source data 0069/source.csv into /data	Upload successful		
		2	Load source data into the application	Screenshot of Data Input -> Model info and Data Input -> Change E-R SSAP Defaults		
		3	View the data	Screenshot of Data -> Source Data, showing data contents		
		4	View the data summary	Screenshot of Data -> Source Data Summary, showing data summary		

RID	Topic	Test ID	Step Description	Expected Result	Qualification Note(s)	Pass/Fail
6	Source data can be manipulated using the code parser	1	Input parsing code: enter the following into Data Input -> Modify Data -> Source data manipulation code: ROUTF <- factor(ROUT, c(1,2), c("IV","SC")) subset($DATA, ID == 1, select=c(ID,TIME,EVID,STUD,ROUTF))	Input		
		2	Screenshot of Source Data view	Screenshot shows the selected subset of patients and variables, with the renamed Route factor		
7	Analysis data can be created by merging run data and source data	1	Remove data parsing subsets, but leave ROUTEF. Take screenshots of data summaries for run and source data	Screencaps show that other patients beside subject 1 have been added back in to the datasets		
		2	Merge the datasets by selecting Data -> Analysis Data	Screencap of analysis data shows merged data		
		3	Confirm merge is a full merge by subsetting to study 183 and verifying that all values of WGT are missing	Screencap of summary shows all studies are 183 and no values for WGT		
8	Analysis data be manipulated using the code parser	1	Enter the following in Data Input -> Modify Data -> Analysis data manipulation code: SEXF <- factor(SEX, c(0,1), c("Female","Male"))	Screencap of analysis data summary shows SEXF factor with Male and Female		
9	Analysis data can be viewed and summarized	1	View the analysis data	Screencap of analysis data view shows data		

RID	Topic	Test ID	Step Description	Expected Result	Qualification Note(s)	Pass/Fail
10	Subject level exclusions can be specified and viewed	1	Create subject and observation level exceptions indicator column. In Data Input -> Modify Data -> Analysis data manipulation code enter: SUBJEXC = "Keep" SUBJEXC[RACE==88] = "Missing race" OBSEXC = "Keep" OBSEXC[EVID==0 & DV<0.05] = "BQL"	Code is input, new column is created in analysis data		
		2	From Data Exclusions -> Subject exclusions -> Subject exclusion specification enter: Keep:: Missing race::No race information for subject Press "Generate subject exclusions" button	Input allowed, no errors		
		3	View subject exclusions: Data Exclusions -> Subject exclusion specification -> Subject Exclusion Data	Screencap of data showing missing race for all patients		
		4	Verify that exclusions are no longer in analysis data	Screencap of analysis data summary shows that no patients with missing race are present		
11	Observation level exclusions can be specified and viewed	1	From Data Exclusions -> Observation exclusions -> Observatoin exclusion specification enter: BQL::Concentration BQL Keep:: Press "Generate observation exclusions" button	Input allowed, no errors		
		2	View observation exclusions: Data Exclusions -> Observation exclusion specification -> Observation Exclusion Data	Screencap of data showing BQL for all observations		
		3	Verify that exclusions are no longer in analysis data	Screencap of analysis data summary shows that no patients with missing race are present		

RID	Topic	Test ID	Step Description	Expected Result	Qualification Note(s)	Pass/Fail
12	Data cache can be cleared from the app	1	Clear cache: Data input -> Model info -> Clear cached data Reset Model Input filenames to point to nothing	Screencap shows no observation, source, or analysis data		