PharmaSUG China 2024 - Paper AP-10002

One-stop solution for QA - Automation Tool by Python

Yangxue Ma, Jiangsu Hengrui Pharmaceuticals Co. Ltd.

ABSTRACT

Quality Assurance (QA) plays a crucial role in the pharmaceutical industry, especially in the clinical data analysis process. It involves a huge investment in human resources to ensure the reliability and accuracy of the clinical data as well as compliance with regulations. However, quality assurance could still be challenging since human judgment is subjective, time-consuming, and repetitive. In addition, even experienced employees have difficulty spotting minor errors. Therefore, developing an integrated auto-QA tool is imperative to overcome these limitations.

Our tool, Compare Plus (CP) could provide a one-stop solution for QA. By simply dragging target documents to the window and clicking the 'Run' button, you are able to obtain all of the QA results summarized in a single Excel file. The comparison report is easy to read and it is also traceable toward specific intermediate files with built-in hyperlinks. Furthermore, CP has various options on the user interface, allowing personalized customization for different clinical studies. We are proud to introduce you to our Compare Plus and hope to explore more of the potential of QA automation with you.

INTRODUCTION

In the pharmaceutical industry, SDTM and ADAM are two CDISC standard data models used to organize and standardize clinical data. As for tables, listings, and figures (TLFs), people use specific standards (e.g. ICH E3 guidelines, FDA/EMA guidelines, etc.) to present clinical data consistently and comprehensively. By using these standards during new drug development, SAS programmers are able to ensure regulatory compliance, improve data quality, facilitate data sharing, and ultimately speed up the drug development process. In this case, quality assurance (QA) is a key process for managing SDTM and ADAM data, as well as TFLs. Without proper quality assurance, the reliability and acceptability of data may be compromised, leading to potential delays or other issues in drug development and approval procedures. While people play an integral role in QA, there are certain limitations in being time-consuming, inconsistent, and error-prone. Given these restrictions, it is essential to use automated QA tools.

On this basis, Compare Plus (CP) allows you to automatically compare data sets and tables from developing programmers and QC programmers, generating individual comparison results as well as a summary report.

WORKFLOW AND PROCESS

First, let's define what kind of projects are able to pass the inspection:

1. In a qualified project, each production-side data set (SDTM/ADaM) or TFL must have its corresponding QC output.
2. Additionally, the corresponding output on the QC side must be consistent with the production side. The definition of "consistency" can be further broken down into the following aspects:
    - Consistency in the number of variables and their names for identical data sets.
    - Consistency in the observations and values for identical variables.
    - Consistency in the length, format, and labels for identical variables.
3. Intermediate files such as comparison codes and results should be properly archived so that the entire quality inspection process is traceable.

Based on these three principles, our tool automates the QA process in 3 steps:

1. PREPARE THE DATA SETS

In a standard clinical data-generating process, we put the production-side data sets and the QC-generated data sets in the same project folder. As for pre-check, our tool simply counts the number of target documents in this quality check and tries to find the corresponding files in the QC folder. If QC files cannot be found, CP marks them as 'MISSING', indicating that these data sets violate the 1st principle and fail QA. For TFLs, the tool programmatically converts RTF files into SAS data sets (SAS7BDAT), and the subsequent processes are the same as SDTM/ADaM.

2. COMPARE THE DATA SETS

For target data sets with corresponding QC files, we ensure that the production-side data sets are reproducible, which means the data sets on both sides are identical (2nd principle). We implement this step with the SAS built-in macro, PROC COMPARE. Additionally, we store the comparison results of each pair into an LST file for traceability (3rd principle).

3. SUMMARIZE AND INTERPRET COMPARISON RESULTS

CP searches and captures keywords in LST files and then automatically generates an Excel file showing summarized comparison results. This Excel report contains the following information: whether the target data sets pass the quality check, hyperlinks to LST files generated by PROC COMPARE, the interpretation of the LST files, etc. In short, the QA workflow chart is shown in Figure 1.

Figure 1 is the QA workflow chart.

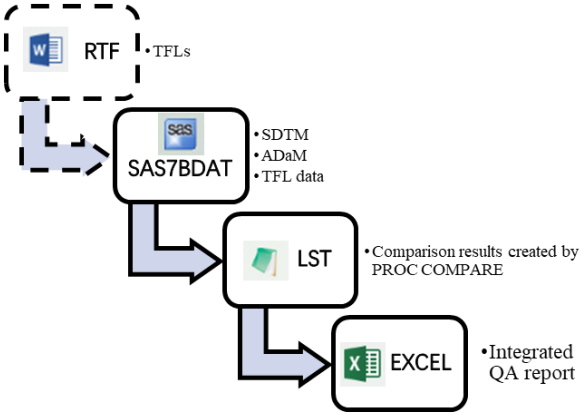


Figure 1. QA Work Flow

USER INTERFACE

In order to achieve efficient QA, unlike the complex programs behind the tool, our user interface is very simple and user-friendly (Figure 2 left). You only need to drag the target files (SAS7BDAT or RTF) into the CP window and click “Run” in the lower right corner to generate the QA report. Furthermore, you can personalize settings on the main interface according to different clinical study designs. For example, whether to ignore attribution differences between data sets, or whether to upload intermediate files and the comparison report to the project folder. In addition, you can select the encoding of the target files. Currently, our tool is compatible with UTF-8 and GKB-encoded data sets. Finally, you can fill in the precision of the comparison in the “Criteria” box to meet the QA requirements of different projects. After the program runs, a prompt interface will appear to guide you to locate the path of the report and intermediate files (Figure 2 right). A folder named CP-Output is created on your desktop; the QA report and all intermediate files are stored in a time-stamped subfolder. This time-related naming method makes every QA result unique and traceable and prevents historical records from being overwritten. Additionally, as we mentioned above, if you check the “Upload” option for this QA process, this time-stamped subfolder will be uploaded to the project folder and archived as part of the official QA documentation.

Figure 2 shows the UI of the tool.

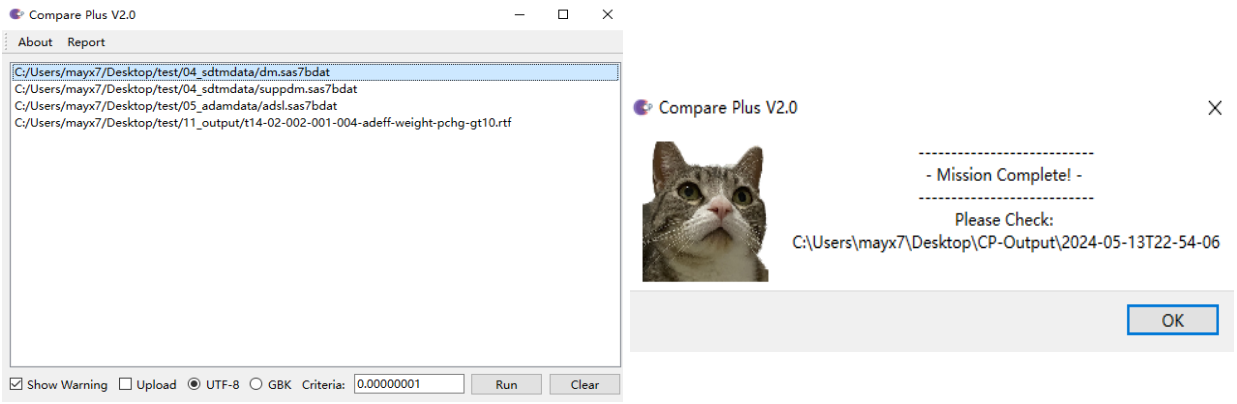


Figure 2. Main Interface (left) and QA Completion Interface (right)

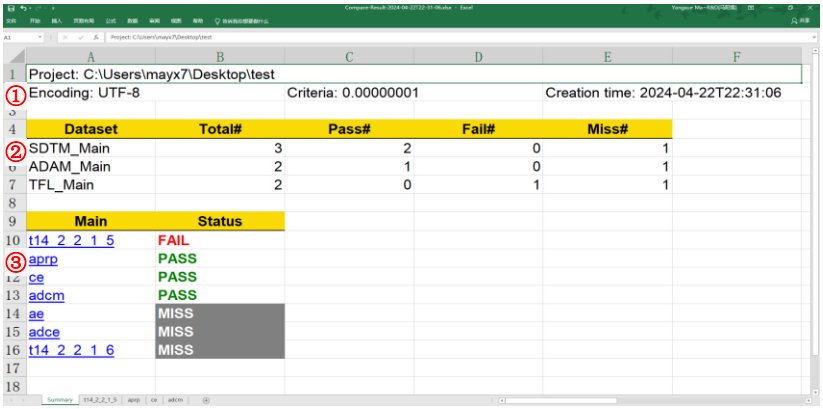
QA SUMMARY REPORT

The QA report generated by CP is comprehensive and easy to read. We divide the report into two parts: a summary sheet and multiple detail sheets for target files.

SUMMARY SHEET

The summary sheet is shown in Figure 3. The first part of the summary sheet shows the root path of the project file, input encoding, precision, and the creation time of this report, helping you review the basic information of this inspection at a glance (① in Figure 3). The second part mainly displays the numbers of input files and their corresponding QC files; these numbers are further classified as “Pass”, “Fail”, or “Miss”. This section gives you a preliminary understanding of the overall comparison results (② in Figure 3). The third part is a list showing target files and corresponding check results. All file names are hyperlinks, and you can jump directly to the corresponding detail sheet with one click (③ in Figure 3).

Figure 3 is the Summary sheet of the QA report.



	A	B	C	D	E	F
1	Project: C:\Users\mayx7\Desktop\test					
①	Encoding: UTF-8		Criteria: 0.00000001		Creation time: 2024-04-22T22:31:06	
4	**Dataset**	**Total#**	**Pass#**	**Fail#**	**Miss#**	
②	SDTM_Main	3	2	0	1	
6	ADAM_Main	2	1	0	1	
7	TFL_Main	2	0	1	1	
9	**Main**	**Status**				
10	t14_2_2_1_5	FAIL				
③	aprp	PASS				
12	ce	PASS				
13	adcm	PASS				
14	ae	MISS				
15	adce	MISS				
16	t14_2_2_1_6	MISS				

Figure 3. Summary Report

DETAIL SHEET

Let's take the QA results of one of the TFLs, t14_2_2_1_5, as an example. On the summary sheet, you can see that its status is FAIL, meaning that it failed QA. Clicking on the filename in section three (③ in Figure 3) takes you to its detail sheet showing the specific comparison results (Figure 4). The information in each detail sheet can also be divided into three parts. The first part contains the name of the target file and the results of its inspection. This filename is also a clickable hyperlink that points directly to the original LST file generated by PROC COMPARE (① in Figure 4).

The second part displays general statistics of the comparison results. As mentioned earlier, CP compares data sets from three aspects: variables, observations, and attributions. Only if all three aspects of the data sets are identical, the QA status will show PASS. However, if you check the “Show Warning” box on the main interface, the conditions for PASS will change: attribution consistency will no longer be a necessary condition for QA pass. Even if the attribution fails the quality inspection, the final status is still PASS, and CP will present a WARNING to remind you to pay attention to the attribution differences.

According to our experience, the most common reason for FAIL cases is the difference in the number of variables, so we have designed a simple variable number count on the production side and the QC side respectively to achieve the purpose of rapid quality inspection. The variable counting results are shown in row 4, column C&D (② in Figure 4).

The third part displays more details of the results. As shown in Figure 4, there are corresponding data sets of T14_2_2_1_5 on both sides. In this pair of data sets, each data set has 7 columns from C1 to C7, and all columns are stored as Characters, so the variable comparison status is PASS (③ in Figure 4). For the Observation part, we can clearly see that there are inconsistencies in the first, second, third, fifth, sixth, and seventh rows among C3 to C7 on the production side and the QC side. The corresponding values on the production side are displayed in the left column (main), and the values on the QC side are displayed in the right column (QC) (④ in Figure 4). For the sake of simplicity, the report only shows the first 5 distinct rows for each variable; to see the complete comparison result, you can click on the filename in the upper left corner to open the corresponding LST file. The last part of this report is the Attribution. The Attribution results are further divided into three categories: Length, Format, and Label. Figure 5 shows that the variable lengths on both sides of C1-C7 are different. The variable length on the production side is 1000, while the corresponding length on the QC side is 200 or 500. At the same time, in this example, the Format and the Label of variables on both sides are the same. Our tool does not list all comparison results, but only the different parts (⑤ in Figure 4).

In summary, on the one hand, due to the differences in multiple observations between multiple pairs of variables, the Observation part shows ERROR, and the overall QA status of t14_2_2_1_5 is FAIL, meaning it does not pass the QA. On the other hand, although the same variable has different lengths, instead of showing ERROR, the Attribution part displays WARNING since the “Show Warning” option has been checked on the main interface during this QA.

Figure 4 shows the Detail Sheet for T14_2_2_1_5.

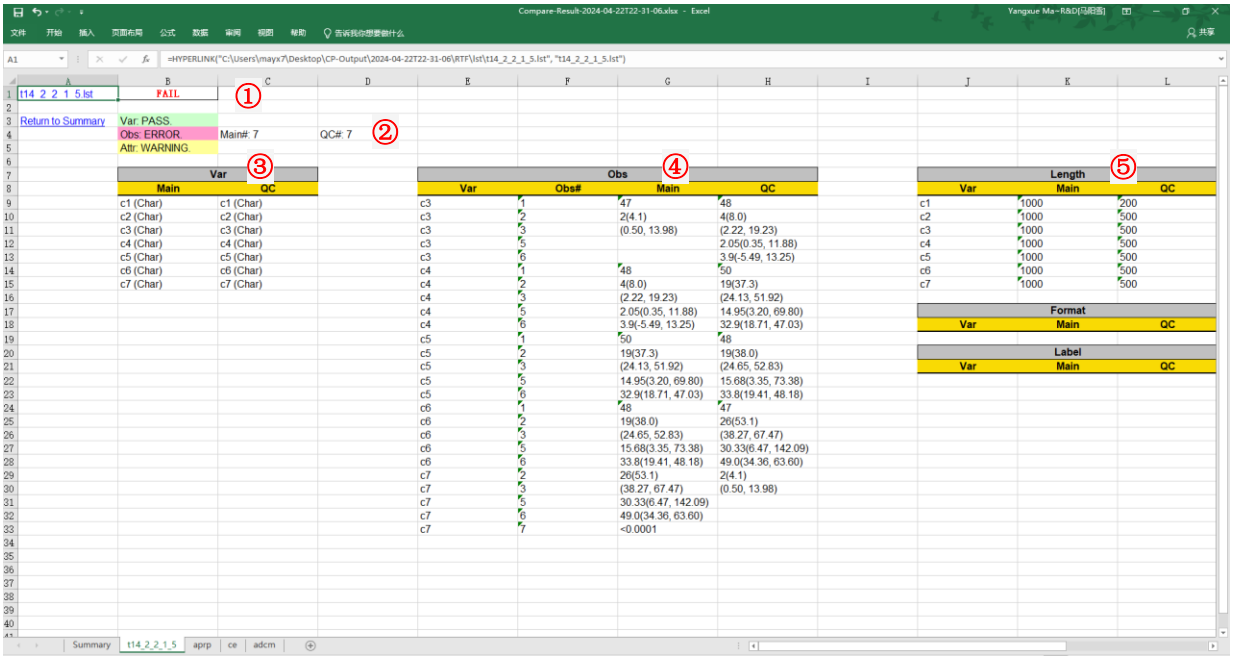


Figure 4. Example of the Detail Sheet for T14_2_2_1_5

CONCLUSION

Overall, in the pharmaceutical field, QA is becoming more and more important. However, time cost and manpower limitations bring great challenges to programmers. Fortunately, CP can significantly raise efficiency and improve the reliability of quality inspections. We always adhere to the principle of the user first in the design of Compare Plus; simple operation, traceable process documents, detailed information, and easy-to-read inspection reports are the goals we are pursuing. In the future, we will keep exploring faster running speed, stronger compatibility, and more personalized functions of our QA tool to achieve the pursuit of a one-stop QA solution.

REFERENCES

"Pyside6" PyPI. Accessed July 31, 2023. https://pypi.org/project/PySide6/.

Fitzpatrick, Martin. "Pyside6 Tutorial 2023, Create Python Guis with Qt." Python GUIs, March 16, 2023. https://www.pythonguis.com/pyside6-tutorial/.

CONTACT INFORMATION

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

Yangxue Ma
Jiangsu Hengrui Pharmaceuticals Company Ltd.
Yangxue.ma.ym4@hengrui.com