Paper PP08

Quality Control and Validation – More than Just PROC COMPARE

Evgeny Starostin, Worldwide Clinical Trials, Nottingham, UK
Michael Rimler, GlaxoSmithKline, Cincinnati, Ohio, US

ABSTRACT

Quality, accuracy, and integrity of data and statistical analysis results are essential in clinical trials. Patients and regulators must be able to trust analyses of clinical data. To further this goal, the PhUSE Standard Analyses & Code Sharing Working Group is collaborating on a white paper to propose the best practices for robust quality control (QC) by health and life sciences organisations. The central part of the white paper discusses the QC process on analysis datasets and specifications, statistical tables, listings, and figures (TLFs), and eSubmissions.

During Independent QC, many new starters focus on obtaining a clean PROC COMPARE for datasets and TLFs, or standard tools to validate the define.xml file. While those are critical components, it is but one aspect of validation. This paper demonstrates how both manual and electronic checks can be applied for each category of deliverables, including those which are not directly programmable, that is analysis dataset specifications and eSubmission components.

INTRODUCTION

In clinical trials, QC of the deliverables created as results of the statistical analysis process traditionally relies on the principle of independent validation by others. In the typical notion of validation (that of datasets or TLFs), this concept manifests itself as independent programming of the output using the same inputs. The presumption is that an independently developed program, using the same inputs and requirement specifications, which yields the same outputs, serves as validation on the aspects of the deliverable that the independent programming replicates.

The notion of independence also applies for other levels of validation such as code review by a peer reviewer. But in both cases the focus is on ensuring that both parties' interpretation of the specification is the same and they apply same programming logic for data mapping, derivations, and processing data issues. This "getting match" approach is often relatively easy to automate, e.g. by using SAS® PROC COMPARE, which leads to individuals, and at times organisations, becoming too reliant on electronic checks and neglecting other elements of QC.

Within the PhUSE working group Standard Analyses and Code Sharing, a white paper provides an in-depth review of best practices for robust QC, including checks used by CRO and pharma sponsor organisations in clinical trials. This paper leverages the content of the white paper to summarise how both manual and electronic QC checks can be applied for each category of deliverables: analysis dataset specification, analysis datasets, TLFs, and eSubmission. We do not attempt to advocate any of the QC strategies of particular practices, nor do we cover the QC of SDTM datasets, although most QC principles and validation checks described below are applicable for SDTM.

QC BEFORE PROGRAMMING BEGINS

Quality control should accompany every step of the statistical analysis process. Before programming even commences to develop first draft outputs, it is necessary to ensure that the inputs are appropriate. Quality programming is dependent on quality inputs such as source data and output specifications. Table 1 summarises the items that should be checked before developing any programming code.

Table 1. Elements of QC Prior to Programming

Item	Check	Details
Source Data / Source Data Specifications	Fit for purpose	• Check if eCRF collects all data required for the planned analysis per the Protocol • For electronic data transfers (eDT) ensure the data is consistent with case report form (CRF) (format of subject identifiers, visit and timepoints structure) and includes all expected data required for the planned analysis • Check that SDTM collects all data from eCRF and electronic data transfers required for the analysis
Analysis Dataset Specifications	Clear traceability	• All dataset variables, derived or predecessor, can be traced back to the source data (e.g. SDTM or subject-level analysis dataset) • Source values are retained for traceability
Analysis Dataset Specifications	Analysis ready	• All planned analyses are supported by analysis datasets • All significant derivations are planned for the Analysis Dataset programming and not the TLF programming
ADaM Dataset Specifications	Conformance to ADaM standards	• ADaM Version and Implementation Guide (IG) version are checked against current regulatory agency requirements (e.g. FDA Data Standards Catalog) • Dataset names and labels are per standard • Variable names, labels and expected values are per standard • ADSL, BDS, OCCDS, OTHER assigned to data according to ADaM IG
TLF specifications (mock-up) shells	Alignment with SAP and study design	• Check against the Statistical Analysis Plan (SAP) if shells are fit-for-purpose to support the analyses • Check against the source data if the output design is supported by the data collected (e.g. the shell prescribes that the protocol deviations should be summarised as Major/Minor, are the Major/Minor categories assigned in raw database?) • Check against the protocol and raw data specifications if the shell structure corresponds to the study design (e.g. visit schedule, codelists for categorical data, etc.) • Ensure data listings can be generated using analysis datasets or SDTM datasets without significant additional derivations

Arguably, none of the checks detailed in Table 1 are possible using PROC COMPARE. Indeed, they are difficult (and in some cases, impossible) to automate, requiring manual review. Nevertheless, the principle of independence continues to apply here, e.g. review of the TLF or ADaM specification should be conducted by a programmer or/and statistician who was not involved in writing of the specification.

VALIDATION OF PROGRAMMING DELIVERABLES

The reporting of analysis results from a clinical trial generally involves numerous datasets and TLFs. These outputs, compiled together, can be referred to as the *analysis package*. In the typical statistical programming process, programmers or statisticians develop code to produce (or validate) individual components of the package. In addition to the validation of each individual component, the analysis package also usually undergoes a degree of validation including cross-consistency checks and checks against reporting standards. We look at these two types of QC in turn.

VALIDATION OF INDIVIDUAL OUTPUTS

Validation of the individual outputs often involves electronic comparison of the results of the produced output and the output generated by the QC programmer, for example obtaining a PROC COMPARE result clean of any reported differences. While this is an important step to electronically identify discrepancies and mismatches, a QC process which solely relies on this electronic comparison is not sufficient. There are many elements where PROC COMPARE fails to capture issues in the produced output. Table 2 outlines the elements which a validation programmer should engage in, both electronic and manual, including those where PROC COMPARE is not a useful mechanism for performing the check.

Table 2. Common Validation Checks for QC of Individual Outputs

Type	Check	Details
Electronic	Dataset Comparison	• SAS PROC COMPARE • Other tools to automatically compare analysis datasets, including database compare tools
	TLF Comparison	• SAS PROC COMPARE or other tools to comparing the dataset which is rendered to the TLF (e.g., via PROC REPORT or PROC GPLOT). • Specific tools to compare the dataset resulting from parsing the rendered output (e.g., RTF or PDF)
	Missing data	Check for complete data (i.e., no missing data) for variables for which it is required per standard. Examples include age, sex, race, and ethnicity, or population flags
	SAS Log Review	Errors, warning, and other unacceptable messages. A list of log messages to consider for review can be found in Good Programming Practices in Healthcare Creating Robust Programs (Tip #4)
Manual	Code Review	Check the code aligns with Good Programming Practice and company practices including the items below: • Code readability • Code format, logic, and content • Code is aligned with the specifications • Program header • Sufficient commenting (both in terms of volume and clarity) • Version control tracking
	Dataset Review	The list of checks may include, but is not limited to: • Alignment with specification/requirements • Character strings contain text which is factually correct and meaningful, is spelled correctly, is in the required case • Units of measure align with results • Numeric variables have equivalent character variables and they are aligned (match) • Data falls within defined value ranges • Rounding and truncation of numeric values is appropriate and per specifications • Precision of decimals is applied as specified
	TLF comparison	Manual comparison of the numbers on the outputs created by production and validation programmers, including, but not limited to: • Population Ns (e.g., in column headers of summary tables) • Ensure correct analysis population is used based on specifications • Calculations embedded in footnotes such as p-values or patient counts, e.g. “[1] The analysis excludes 3 patients in the placebo group due to protocol deviations.”
	TLF Review	The list of checks includes, but is not limited to: • Alignment with mock-up shells • Character strings contain text which is factually correct and meaningful, is spelled correctly, is in the required case • Text in column headers (such as treatment group labels) • Truncation of character results • Units of measure align with results • Rounding of numeric values is appropriate • Precision of decimals is applied as specified • Consistency of population counts across outputs, e.g. population counts (Ns) • Categorical grouping of continuous values (especially boundary values) • Data falls within defined value ranges of axes • Footnotes meaningful, sufficient, and consistent with mock shells • Ensure there are no unresolved macro variables appearing in the output • Sort order of rows

Type	Check	Details
		• Column order • Text wrapping • Page breaks, pagination, unexpected blank pages • Titles are rendered properly and consistent with specifications • Consistent indentation

ANALYSIS PACKAGE VALIDATION

Before every reporting milestone, be it an Interim Analysis, a Dry Run, or the final results, all analysis datasets and TLFs need to be validated as a complete package of interrelated and interdependent components supporting the SAP and protocol. This task is usually performed by an experienced programmer and/or statistician. Programmatically developed electronic checks, including PROC COMPARE, remain vital in this part of the process, but additional manual steps are essential as well. Table 3 summarises the common manual and automatic checks used by organisations at this stage. Many of the checks intentionally repeat the checks performed at the individual component level to ensure programs execute as expected and produce consistent results following changes to the programming infrastructure such as updated source data or installation of programs to the production environment.

Table 3. Common Validation Checks for QC of Analysis Package

Type	Check	Details
Electronic	Dataset Comparison	See Table 2
	TLF Comparison	See Table 2
	Missing data	See Table 2
	SAS Log Review	See Table 2
	Compliance with standards	E.g. Pinnacle 21® Community
Manual	Input checks	Correct inputs used: • Latest eCRF cutoff, eDT, and other raw data • Dummy or Final Randomisation and Kit list • If before database lock, is there any risk for the data to be unblinding
	Code Review	See Table 2
	Dataset Review	The list of checks may include but is not limited to: • See Table 2 • Relationships between datasets • Common variables have same values and properties in different datasets
	Date-Time Stamps	Ensuring the date/time stamps are sequential: • Raw data > SDTM > Analysis Dataset > TLFs • Order within analysis datasets (e.g. should ADSL be generated first?) • Specification > Program > Output • Production > Independent QC
	TLF comparison	• See Table 2 • Count checks across TLFs (e.g. listing vs table, figure vs table)
	TLF Review	The list of checks includes but is not limited to: • See Table 2 • Alignment with planned analysis • Consistency of representation (fonts, colors/symbols/lines in figures etc.)
	Completeness of the package	• No planned but missing outputs • No unplanned or legacy files included

Different organisations may use different combinations of validation activities and specific checks. The exact set of checks may also vary within an organisation for different types of deliveries. For example, date-time stamps checks may be a requirement for the final delivery and skipped for a Dry Run. Some organisations also adopt a risk-based approach when different combinations of checks are used depending on likelihood and severity of potential error. Regardless of the QC strategy and chosen set of validation activities, the principle of independent validation should

be adhered to and sufficient combination of manual checks should be used in addition to PROC COMPARE and other electronic tools.

It is also worth noting that electronic vs manual classification assigned in this paper is a subjective classification for some checks. For example, Date-Time stamps sequence is classified in Table 3 as manual, however an organisation may develop a mechanism to automate this check through system file properties. Another example might involve electronically validating the population Ns displayed in column headers of a summary table by parsing the output file into a dataset and using PROC COMPARE or by requiring the Ns be in the label attributes of the display's source dataset to be included in the validation programmer's PROC COMPARE.

QC OF ESUBMISSION PACKAGE

The study data submission package, formerly referred to as the Case Report Tabulation (CRT) package comprises of files of different types and properties, including the define.xml as a critical component. In this paper, we use eSubmission package as a synonym to an electronically generated study data submission package. Validation of the define.xml and underlying database focuses on checking adherence to CDISC standards using electronic solutions such as the Pinnacle 21® Community tool (formerly known as OpenCDISC Validator). PROC COMPARE is rarely used for eSubmission package validation. Similarly to a tendency to rely on PROC COMPARE results for validation of datasets and TLFs, eSubmission validation can become myopically reliant on Pinnacle 21 checks, which is equally insufficient.

Electronic tools, such as those from the Pinnacle 21 Community, include checks to validate the define.xml file against XML schema, perform content-related checks such as submission specific validation rules (e.g. FDA Study Data Validator Rules) and confirming correct implementation of CDISC controlled terminology [1]. Automated validation using Pinnacle 21 is powerful and efficient, but it does not check meaningfulness and comprehensiveness of the provided information, nor does it verify the correctness of the content of Analysis Data Reviewer's Guide (ADRG) and other documents included in the package. Therefore, manual review of additional aspects of all eSubmission components is critical to ensure the quality of the complete package.

The manual review is focused on three main areas: package content, define.xml, and ADRG, as well as the interconnection between them. Table 4 provides a list of common checks used when performing manual review of the eSubmission package. Although not exhaustive, it gives a flavor of the checks resulting from the PhUSE Standard Analyses and Code Sharing Working Group review. It is not feasible to provide the exhaustive list of checks since minimal requirements tend to vary from study to study.

Table 4. Common Manual Checks for eSubmission Packages

eSubmission package content
• All expected datasets (XPT files) are included • Only relevant files (datasets, XLS stylesheet, define.xml, ADRG and additional documents referred from define.xml) and nothing extra (e.g. automated validation report) are included • No empty (with zero records) datasets present • Datasets greater than 5 gigabytes in size are split into smaller datasets (based on current FDA guidance) • All file names in lower case
define.xml
• Spelling and grammar • Dataset order ADSL, then ADAE, then BDS alphabetically, then other domains alphabetically. This order is not required, but is reasonable and logical [2] • All derived variables have clear and sufficient comments and derivations. Use of programming code is not advised and should be avoided • Proper use of ORIGIN. There are occasions when the person creating define.xml sets the origin of derived variables to Assigned purely to save the time it takes to properly document all derivations [3] • Readability of comments. Define.xml does not support formatting like bullet points and numbering lists which makes large derivations with multiple conditions difficult to read. In this case it is suggested to refer to the Complex Algorithms document and provide hyperlink at the end of the derivation [3] • Codelists should use terminologies from NCI (National Cancer Institute) Enterprise Vocabulary Services for variables for which standard terms exists [4] • Separate UNIT codelist should be used for each unit variable, for example for exposure and laboratory data. Same rule should be applied for other variables like frequency, route of administration, but particularly important for UNIT codelist due to its volume

• Codelists should include all possible values assigned to it even if some options are not present in data. For example, if for race there are 2 options ASIAN and WHITE but study design allows ASIAN, WHITE and BLACK OR AFRICAN AMERICAN, all three terms should be included in the codelist. There is also no need to include the whole list of 7 terms the CDISC RACE codelist provides • Value Level Metadata should be applied to all AVAL or/and AVALC in BDS data structures and for other variables "when there is a need to describe differing metadata attributes for subsets of cells within a column" • Links to external files lead to the correct places within the documents
Analysis Data Reviewer's Guide - technical properties
• File name – adrg.pdf • File properties: PDF version 1.4 through 1.7 optimized for fast web view • Bookmarks: bookmarks panel should be displayed when the file opens, bookmarks are collapsed and inherit zoom applied • Hyperlinks lead to correct places inside and outside the document • Table of contents: all links work properly and there are no issues like "**Error! Bookmark not defined**"
Analysis Data Reviewer's Guide - content
• All required sections of the document are included as per the required template, e.g., PhUSE ADRG template, even if some of them are not applicable • Sections and subsections are numbered consequently without gaps • Correctness of data standards and dictionaries specified. This may require cross-checking against the clinical Study Data Reviewer's Guide (cSDRG) • Study Description is phrased in relation to data structure, specifically treatment variables (TRTxxP, TRTxxPN etc.) and Phase and Period variables (APHASE, APERIOD etc.). For the complex studies in which textual explanation is not sufficient, it is recommended to include a graphical representation of the study design with the variable annotations on the schematic • The list of Key variables should be present and checked against the data and define.xml • All datasets are listed • All special derivation methods such as visit windowing or processing of incomplete dates should be fully described and cross-checked vs the SAP and define.xml • Variable names should be checked against data (XPT files) and define.xml • The results of automated validation checks are fully and clearly explained in the data conformance summary section

CONCLUSION

Quality control of the statistical analysis results in clinical trials is a complex task which is explained by the large volume and variety of input data, complexity of derivations and analyses, and diversity of the outputs: datasets, TFLs and components of eSubmission. While there seems to be a general consensus within the industry that QC should be based on principle of independence, the exact QC strategy and specific checks adopted by organisations may vary over a wide range. We have made an attempt to summarise the most commonly used checks and organise them into two categories: electronic or manual. We illustrate that while electronic tools are an integral component to effective validation of statistical analysis results, they are not sufficient. A robust QC process should include a thoroughly selected combination of electronic and manual checks. Furthermore, the independent validator must not lose sight of the value of manual checks on the validation process.

REFERENCES

[1] Sergiy Sirichenko, Max Kanevsky. "Diagnostics of technical errors in define.xml file". PhUSE EU Connect, 2018. https://www.phusewiki.org/docs/Frankfut%20Connect%202018/SI/Papers/SI13-phuse-eu-2018-si07-sergiy-sirichenko-19308.pdf

[2] CDISC. "CDISC Define-XML Specification Version 2.0". 24 Apr 2014. https://www.cdisc.org/standards/foundational/define-xml/define-xml-v20

[3] David Roulstone. "Do's and Don'ts of Define.xml". PhUSE EU Connect, 2018. https://www.phusewiki.org/docs/Frankfut%20Connect%202018/SA/Papers/SA12-sa04-phuseeu2018-define-19291.pdf

[4] US Food and Drug Administration. "Study Data Technical Conformance Guide". March 2019. https://www.fda.gov/industry/fda-resources-data-standards/study-data-standards-resources

ACKNOWLEDGMENTS

Gayathri Kolandaivelu, Eli Miller, Sai Bhavaraju, Daryna Hololovich, Vamshi Gande, Jane Marrer, Maria Dalton, Dawn Edgerton, Mark Foxwell, Aadesh Shah

CONTACT INFORMATION

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

Evgeny Starostin
Worldwide Clinical Trials
1st Floor, Waterfront House, Beeston Business Park, Beeston, Nottingham, NG9 1LA, UK
Email: evgeny.starostin@worldwide.com

Michael Rimler
GlaxoSmithKline
1250 S. Collegeville Road
Collegeville, Pennsylvania, US, 19426-0989
Email: michael.s.rimler@gsk.com