Paper SM02

Dynamic document creation framework for streamlining cSDRG

Umayal Annamalai, Algorics, Bengaluru, India
Madhu Annamalai, Algorics, San Francisco, US

ABSTRACT

cSDRG provides guidance and assistance to the clinical reviewers to review and evaluate the clinical trial data. This critical document for regulatory submission is created manually based on templates and guidelines. This leaves it prone to errors and lacks efficiency in the process. There is a clear need for the adoption of automated and reproducible approaches in regulatory documentation to ensure reduced error rates, transparency, improved document consistency, time savings, and regulatory compliance. Hence, we are proposing a novel approach using R Markdown to streamline the cSDRG creation process by pre-populating the cSDRG based on a dynamic document generation framework. R Markdown allows the integration of metadata details, required data, and narrative text within a single document, enabling automated and consistent generation of cSDRGs across different studies and analysis scenarios. The insights from this research will optimize the cSDRG creation process, ultimately contributing to more efficient and transparent regulatory submissions.

cSDRG

The cSDRG is an essential document that contributes to the transparency and consistency of the regulatory review process, helping regulatory agencies assess the validity and reliability of the clinical trial data submitted by sponsors during the drug development lifecycle. The primary purpose of the SDRG is to assist regulatory reviewers in understanding the structure, content, and quality of the clinical trial data submitted in electronic format, such as the Clinical Data Interchange Standards Consortium (CDISC) standard. CDISC standards are widely used in the pharmaceutical industry to ensure consistency and interoperability in the presentation of clinical trial data. Key aspects typically covered in an SDRG include Data Structure and Organization, Variable Definitions, Data Validation and Quality Checks, Statistical Analysis Procedures, Review Procedures, Data Cleaning and Query Resolution, Traceability and Audit Trails, Compliance with Standards.

Moving Beyond Manual SDRG Creation through Automation and Collaboration

Creating a Study Data Reviewer's Guide (SDRG) in real-time involves a systematic and collaborative process. The standard templates for SDRG and guidelines for preparing it are published by CDISC. These serve as the starting point for creating this document. The document is further updated fetching information based on the study such as study design, study milestones, special consideration, standards followed, description of standardized data presented, CDISC compliance, etc.

Creating a Study Data Reviewer's Guide (SDRG) manually can be associated with several challenges and potential problems. Some of the issues that could arise when creating the document manually include human errors, time consumption, lack of standardization, lack of consistency between similar studies, difficulties in collaboration, version control issues, difficulty in traceability, risk of incomplete documentation, scalability, difficulty in data extraction and handling large datasets, and less flexibility.

By incorporating real-time collaboration, leveraging automated tools where possible, and maintaining a structured and reproducible approach, the SDRG creation process can be streamlined, efficient, and responsive to the evolving needs of the study.

WHY R MARKDOWN?

To establish a streamlined, dynamic process, there must be a powerful tool which can offer variety of operations such as data processing and analysis, community support, reproducible and scalable operations, multiple output formats, combining narrative text with processed data, customization while being easy to learn. So, the choice of tool was R Markdown. Automating the creation of a Study Data Reviewer's Guide (SDRG) using R Markdown involves using the R programming language and the R Markdown framework to dynamically generate the document. R Markdown allows to blend code, text, and results into a single document, making it an excellent choice for automating the creation of documents based on data analysis.

DYNAMIC DOCUMENT CREATION FRAMEWORK

Dynamic document creation is of four steps understanding the source documents, r data extraction process, embed the R functions in r mark down and creation of cSDRG following this dynamic process. Figure 1 represents the Dynamic SDRG creation concept.

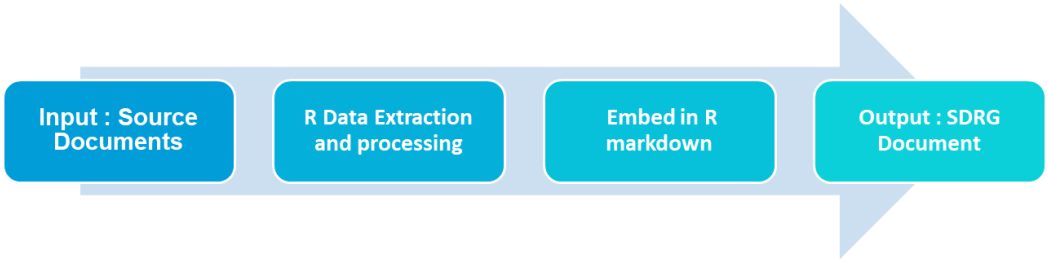

Figure 1: SDRG concept

1. **Source documents:** The data flow for the source documents helps in understanding what documents and how the information flows to the SDRG as per Figure 2.

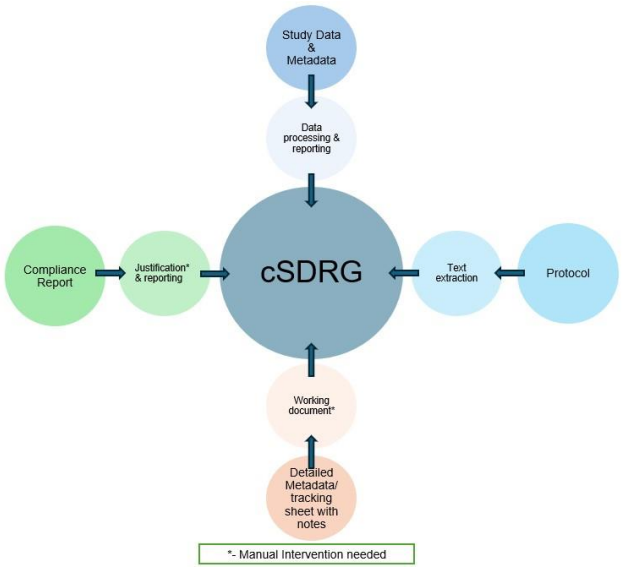

Figure 2: SDRG Dynamic Document data flow concept

2. **R data extraction process:** The prerequisite for dynamic document creation framework will be the standardization of source documents, combining the powerful tools like R & R Markdown to process and extract the source documents, and automation process and methods pertaining to the source documents as listed in Figure 2.

3. **Embed in R Markdown:** Several types of dynamic approaches on data extraction and data processing can be implemented on source documents and the dynamic codes can be stored as R functions for convenience. Several R functions based on R packages listed in Table 1 can be created based on the standardized source documents so that we can dynamically get the details needed to be presented in SDRG. A brief explanation of the application of dynamic document framework on each section of cSDRG is described in Table 1. These can be implemented and handled by R Markdown.

Section	Subsection	Source	Method	R package needed
General	Font/alignment	external word file	Reference in YAML Header	knitr, fancyhdr, rmarkdown
Title page		Protocol	Reference in YAML Header	knitr, rmarkdown
Table for Contents		Autogenerated	R Markdown options	knitr, rmarkdown
1. Introduction	1.1. Purpose	Standard text (Templates for study submission type)	R user defined functions created for different submission type	Base R Package
	1.2. Acronyms	Latest SDRG file generated before the current run	R user defined functions reading last created SDRG - Customizable library	Base R Package
	1.3 Study Data Standards and Dictionary Inventory	P21 report	Data extraction & processing	readxl, tidyverse, dplyr
2. Protocol Description	2.1 Protocol Number and Title 2.2 Protocol Design	Protocol	Data extraction & processing	pdftools
	2.3 Trial Design Datasets	Details from Tracking sheet/ Comments/Notes column in Metadata	Data extraction & processing	readxl, tidyverse, dplyr
3. Subject data Description	3.1 Study data overview & Additional Content of Interest	Details from Tracking sheet/ Comments/Notes column in Metadata	Data extraction & processing	readxl, tidyverse, dplyr
	3.2 Traceability Flow Diagram	Standard flowcharts	R user defined functions created for different submission type	readxl, tidyverse, dplyr
	3.3 Annotated CRF	Details from Tracking sheet/ Comments/Notes column in Metadata	Data extraction & processing	readxl, tidyverse, dplyr
	3.4 SDTM Subject Domains	Metadata/Details from Tracking sheet/ Comments	Data extraction & processing	readxl, tidyverse, dplyr
4. Data conformance summary	4.1 Conformance Inputs	P21 report	Data extraction & processing	readxl, tidyverse, dplyr
	4.2 Issues Summary	P21 report - P21 Explanations should be added to the P21 document	Data extraction & processing	readxl, tidyverse, dplyr
	4.3 Additional Conformance details	P21 report - additional explanations should be added to the P21 document	Data extraction & processing	readxl, tidyverse, dplyr
Appendix	Appendix 1: Inclusion/Exclusion criteria	Metadata	Data extraction & processing	readxl, tidyverse, dplyr

Table 1: R markdown approach for SDRG Sections

4. The Final SDRG document: Based on step1, 2 and 3, the source documents are standardized, and the contents of the source documents are extracted based on R functions. The blank framework for SDRG can be created in R Markdown based on CDISC SDRG completion guidelines and study requirements. R markdown can accommodate factors like the title page, titles, footnotes, fonts, and display properties. Once the blank framework is in place, incorporating the R functions in the appropriate sections in the program and knitting the R code can generate the SDRG document. Preparing the source documents, performing data extraction (saved as standard R functions), preparing the R Markdown framework, and embedding the R functions in the R markdown framework will be the key steps in SDRG creation.

The list of R packages required for handling the dynamic document framework is listed in Table 2 with their respective descriptions.

R package used	Description
markdown, rmarkdown, officer, officedown	To set basics for the document
knitr	General purpose package for generating dynamic reports in R Markdown
pdftools	To read the contents of Protocol
readxl	To import the metadata, Enhanced tracking sheet
tidyverse,dplyr	For basic data processing and manipulation techniques
fancyhdr	To populate header and footer

Table 2: R Packages for dynamic document framework

Sample Codes:

1. YAML Header: The YAML header should be provisioned enough based on several options to cover the title page, format, font, and styling options based on reference document. Sample header codes are referenced in Figure 3.

```
---
title: "<span style='font-size:2em;font-weight:bold;'>Clinical Study Data Reviewer's Guide</span>"
subtitle: |
            <span style='font-size:1.5em;font-weight:bold;'>SDRG, Inc</span> <br>
            <span style='font-size:1.2em;font-weight:normal;'>Study SDRG001A </span> <br>
            <span style='font-size:1.0em;font-weight:normal;'>cSDRG Template Version 2018 11 01 </span>

output:
  officedown::rdocx_document
  reference_docx: styles.docx
header-includes:
  - \usepackage{fancyhdr}
  - \usepackage{lastpage}
  - \pagestyle{fancy}
  - \fancyhf{}
  - \fancyhead[L]{'SDRG001A'}
  - \fancyhead[R]{"Clinical Study Data Reviewer's Guide"}
  - \fancyfoot[L]{'This document is confidential'}
  - \fancyfoot[C]{\thepage\ of \pageref{LastPage}}
---
```

Figure 3: Sample R markdown YAML Header

2. R table of contents and preparation for each section: This can be done in the initial few R chunks where most codes for each section are covered. R functions and custom codes can be created and repurposed for further studies. While repurposing these codes, efforts to standardize the functions should be made to cover multiple scenarios that could come up, making the user defined functions and custom codes robust enough.

```
```{r, results='asis', echo=FALSE}
cat("\\newpage")
```

Clinical Study Data Reviewer's Guide

Contents
<!---BLOCK_TOC--->

```{r include=FALSE}
##R library preparation, functions reference & documents pre-processing codes here
cat("\\newpage")
```
```

Figure 4: Title of contents and User defined R functions

3. Setting up the sections: Framework of the document can be prepared based on simple R markdown options as per Figure 5.

```
# *Introduction*

## Purpose
```{r eval=FALSE, include=FALSE}
##Section 1.1 function calls here
```

Acronyms
```{r eval=FALSE, include=FALSE}
##Section 1.2 function calls here
```

Study Data Standards and Dictionary Inventory
```{r eval=FALSE, include=FALSE}
##Section 1.3 function calls here
```
```

Figure 5: Document Framework

4. Provision the sections: The custom codes and user defined R functions can be referenced inside the R markdown framework as like Figure 6. This step utilizes the framework from Figure 5 to embed and include the R functions created in Figure 4. If table display options are required, those can be referenced inside the R chunks.

```
# *Introduction*

## Purpose
```{r echo=FALSE, results='asis}
#default standard text for purpose to be loaded into section1_1 function, if there is any special study needs to be mentioned,
this has to be defined inside the function with key word
section1_1 (type1)
##section1_1(type2)
```

Acronyms
```{r echo=FALSE}
##Section 1.2 function calls here
```

Study Data Standards and Dictionary Inventory
```{r echo=FALSE}
#alignment of the table p21_tab1 which contains details on data standards and dictionary details brought up from the P21 report &
metadata
ft <- flextable(p21_tab1) %>%
      align(align = "left", part = "all")%>%
      set_table_properties(width = .8, layout = "autofit") %>%
      set_table_font_size(size = 10) %>%
      bold(part = "header")
ft

```
```

Figure 6: Sample SDRG section 1

5. Generation of SDRG docx file: To generate the SDRG word document, one can directly 'knit' the .rmd file or can run the script as per Figure 7 to generate the docx file. Once the changes to the R markdown codes are final (SDRG is also reviewed & finalized), pdf file can be generated instead of docx file with few changes.

```
R -e "rmarkdown::render('csdrg.Rmd',output_file='csdrg.docx')"
```

Figure 7: Generate cSDRG document

CHALLENGES & DISCUSSION

While automating the creation of a Study Data Reviewer's Guide (SDRG) can offer many benefits, it also comes with its set of challenges. Here are some usual challenges associated with streamlining the SDRG creation with dynamic document framework listed as in Table 3. Addressing these challenges requires a combination of technological solutions, ongoing training, and a proactive approach to adapt the automation process to changing requirements. Regular feedback from users and continuous improvement are essential for successful SDRG automation.

Challenges	Solution
Data complexity and data disparity	While the data must be fetched from multiple documents and disparate sources of data, to reduce the number of updates to the R codes made, developing flexible R functions to accommodate and process different data structures and formats should be helpful.
Regulatory compliance and changes in regulations	With the changes to regulatory requirements, the automation framework needs to be revised to align with the latest regulatory requirements
Data quality checks	Developing robust algorithms for data quality checks and incorporate feedback from data reviewers to continually refine and improve the checks.
Collaborative editing and communication	Providing adequate training for team members on collaborative tools and establish clear communication channels. Ensure that manual input and feedback are easily integrated into the automated process.
Template customization	Creating flexible templates to accommodate the specific requirements of different studies and regulatory agencies can be challenging. However, designing templates with customization options and parameters that can be easily adjusted based on study-specific needs is possible.
Version control	Implementing robust version control systems (e.g., Git) is easy but to establish clear guidelines for team members regarding the proper use of version control features is challenging. With right training and collaboration, version control related problems can be reduced.
Integration with existing systems & Tool integration	A variety of tools and processes are followed within each organization especially for tracking, data and metadata. Always tools/any code modules should be developed with a focus on compatibility and integration. Consider dynamic coding techniques that can have seamless integration with other systems.
Data security and privacy	Implementing encryption, access controls, and other security measures to safeguard sensitive data. Ensure that the automated process adheres to data privacy regulations within the organizations.
Skill set and training	Providing comprehensive training programs and resources to ensure that team members are proficient in the tools and languages used for Dynamic data framework is needed for effective SDRG automation.

Table 3: Typical challenges and solutions

CONCLUSION

The adoption of R Markdown as a dynamic document generation framework for SDRGs not only modernizes the process but also brings about a paradigm shift in how clinical data analysis and reporting are approached. The combination of reproducibility, transparency, and collaboration makes R Markdown a powerful tool in the creation of SDRGs, contributing to improved data quality, regulatory compliance, and overall efficiency in the pharmaceutical and clinical research industry. However, it is important to acknowledge and address potential challenges associated with this approach. Addressing these challenges demonstrates a realistic understanding of the implementation and helps stakeholders prepare for potential hurdles. Concluding this paper with a more comprehensive and balanced discussion provides readers with insights into both the benefits and potential hurdles associated with implementing R Markdown for the creation of SDRGs. It also sets the stage for further research, collaboration, and development to overcome these challenges and optimize the adoption of this dynamic document generation framework.

REFERENCES

1. U.S. Department of Health and Human Services Food and Drug Administration, Center for Drug Evaluation and Research (CDER) and Center for Biologics Evaluation and Research (CBER). “Study Data Technical Conformance Guide”. FDA website. March 2022.
   Available at: https://www.fda.gov/industry/fda-data-standards-advisory-board/study-data-standardshttps://www.fda.gov/industry/fda-data-standards-advisory-board/study-data-standards-resourcesresources

2. Clinical Study Data Reviewer’s Guide Completion Guidelines, 2018, v1.4.
   Available at:
   https://advance.phuse.global/display/WEL/Clinical+Study+Data+Reviewer%27s+Guide+%28cSDRG%29+Package

3. Assembling Reviewer-friendly eSubmission Packages, Shefalica Chand, PharmaSUG 2021 - Paper SS-110
   Available at: https://www.pharmasug.org/proceedings/2021/SS/PharmaSUG-2021-SS-110.pdf

4. SAS Report Writing Interface vs R Markdown for Patient Profile Generation, Qin Meng, dMed, PharmaSUG China 2019 - Paper CC-022
   Available at: https://www.lexjansen.com/pharmasug-cn/2019/CC/Pharmasug-China-2019-CC22.pdf

5. Using R Markdown to Generate Clinical Trials Summary Reports, Radhika Etikala, Xuehan Zhang (Emily), PharmaSUG 2020 - Paper DV-164
   Available at: https://www.pharmasug.org/proceedings/2020/DV/PharmaSUG-2020-DV-164.pdf

6. A Guidance Document for the use of affiliated R packages in Regulated Clinical Trial Environments, RStudio PBC, September 2020
   Available at: https://www.rstudio.com/assets/img/validation-shiny-rmd.pdf

ACKNOWLEDGMENTS

We thank Sangavi Somasundaram at Algorics for continuous support towards this paper.

We thank Niveda Ramkumar at Algorics for continuous support in reviewing this manuscript.

CONTACT INFORMATION

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

Author Name: Umayal Annamalai
Company: Algorics
Email: umayal.annamalai@algorics.com
Website: www.algorics.com

Author Name: Madhu Annamalai
Company: Algorics
Email: madhu.annamalai@algorics.com
Website: www.algorics.com