Paper PRE_ST08

Innovate Clinical Trial Reporting with Open Source

Soma Sekhar Sriadibhatla, AstraZeneca, USA
Francis Kendall, AstraZeneca, USA

ABSTRACT

A Statistical Programming division of clinical trials is the powerhouse supporting drug development and trials data reporting, helping millions of patients worldwide. A robust department, reporting patient data as eCRT packages to regulatory authorities. They are created and submitted following CDISC standards using Licensed closed software, SAS® on GxP compliant platforms as per eCTD specifications and Technical Conformance guides.

However, like any highly engaged division, it needs innovation to increase efficiency and streamline processes. So we have introduced Open source programming e.g. R following Reverse Engineering of eCRT package to identify programming components on a sample study. This will further enable us to separate coding from algorithms, efficiently document dataset specifications, analysis results metadata and validate R programs.

The paper will further provide information on challenges and solutions to submit eCRT packages created using open source programming, meeting FDA and PMDA guidance.

Keywords: eCRT, Reverse Engineering, Algorithms, Validation, Open Source

INTRODUCTION

The Clinical trial data and analysis reporting are created and submitted as eCRT package to regulatory authorities as per eCTD specifications. These packages are created following CDISC standards on GxP compliant platforms like Unix or entimICE® (Entimo Integrated Clinical Environment - within AstraZeneca®), using software programs following Technical Conformance Guide of regulatory authorities[1]. These software programs are created and validated by clinical and statistical programmers to analyze and report collected data following specific instructions from Statistical Analysis Plans, Dataset Specifications, Data Presentation Plans for TLFs etc.

Although FDA has clarified that it does not require use of any specific software[2], we are inclined in using Statistical Analysis Software (SAS®) programming to create eCRT packages because:

a. A licensed closed Software which has published Statistical procedures and functions to follow validation and verification process
b. The version of SAS® software can be controlled and selected to create eCRT components as per Statistical procedure and Graphs requirements in Statistical Analysis Plan
c. Every programmer is a SAS® user and cannot modify any of the SAS® procedures or functions, which eliminates bias from analysis
d. The regulatory authorities can reproduce analysis results irrespective of Operating Systems

Due these points and preferences, our processes and documentations like dataset specifications and SAPs are designed to wrangle data and generate reports using SAS® software. This might pose challenge while introducing multiple programming languages . To investigate this further we have identified a published study with CSR results, reverse engineer to deconstruct eCRT and recreate it with an open source language, like R.

We propose innovation of clinical trial reporting by streamlining process documents and procedures to handle multiple programming languages for regulatory submissions. Thus, increasing flexibility and efficiency of Statistical programming division. This leads to a discussion to shift towards algorithm based.

The machine readable, reusable, programming language neutral, validated algorithms (logical sequential steps), which might generate same results as SAS® or any other programming language will further strengthen our programming documentation, in compliance with regulatory guidelines reach to a point where we can plug-in-code of interest to execute. Shifting to Algorithm based department will also install trust among programmers and statisticians to use Opensource languages, mitigate risk as discussed in below sections.

REVERSE ENGINEERING

The concept of reverse engineering involves several steps, including disassembling the product, analyzing its components and functions, documenting the findings, and reassembling the product or creating a new design based on the information obtained[3]. In our case the product is a submitted eCRT package, published late phase clinical trial study.

SCHEMA:

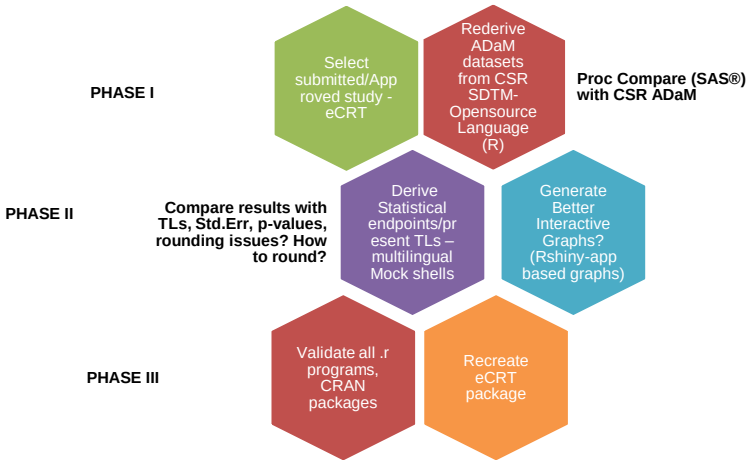

Figure.1 Explaining Reverse Engineering of SAS® generated eCRT package and plan to reconstruct it with Open Source language,R

Phase I of Reverse Engineering was rederivation of ADaM datasets from SDTM using R packages. We could derive safety ADaM datasets and validate them against SAS® generated CSR ADaM. There were challenges to derive efficacy variables due to rounding issues and finding analogous functions to SAS®, as dataset specifications and SAP assumed these will be taken care by software programs. The reliability on SAS® to analyze clinical trials was observed and documented in this step.

Currently we are planning to reach to **Phase II**. However this step, can be quite challenging. One of the user experiences being discrepancy of statistical results like rounding numbers, including standard errors, p-values, across programming languages. Should we round away from zero or to even? This issue is not addressed that often, as the QC process involves SAS® too. If two independent programmers use same language to generate results with same outcome, we consider the results to be validated! Our process documents or derivations were never written to address discrepancies between programming languages. Further how will a programmer decide which results are correct? If we want shift to multilingual programming department we should first write our derivations and specifications in a logical sequence and programming code neutral.

Once we pass this Phase, we will be able to move to **Phase III** that involves validation of open source – R packages and recreate entire eCRT package. In this process we will examine new approaches in solving the same problem, strength and streamline our process documents, specifications to ease programming, probably able to just plug in code to achieve analysis results.

INNOVATION AND ALGORITHMS

Algorithms are a set of instructions or rules that are followed to accomplish a task[4]. Statistical programming department is already based on algorithms. For example: *Statistical analysis algorithms:* to calculate summary statistics such as mean, standard deviation, test of hypothesis between treatment groups, *Bayesian Algorithms:* to adjust probability that a treatment is effective based on the data collected during the trial, *Data wrangling algorithms:* tabulation of data, formatting into domains etc.

These algorithms are a part of Statistical Analysis Plans, Dataset Specifications, presentation plans etc. developed in accordance to regulatory and CDISC standards. However, these *programming instructions* have to be translated to code by programmers which is *implementation step* – to generate datasets and analysis reports.
Then comes the last step, *regulatory submission*. These algorithms will be submitted as a part of define files, complex algorithms, analysis results metadata etc. supporting data and analysis reports to regulatory agencies. The process of eCRT generation thus, need to depend on a particular programming language, rather can be diverse.

If the algorithms are developed as simpler logical sequences or steps in natural language, free of programming language or code programmers might be able to rely on them to program in multiple languages. This leads to another step to develop machine readable algorithms, more of define.xml style, to achieve following: *accuracy, efficiency, reproducibility, Interoperability and scalability*. So not only we would be generating multi lingual eCRT packages, but also improving efficiency, thereby reducing cost.

Writing algorithms in define.xml style would also streamline processes. However, not completely possible without answering: How do we validate these machine readable Algorithms?
Validation would be critical to ensure effectiveness and reliability of algorithms. The validation of algorithms would also reduce risk and increase transparency as preferred by Statistical programmers and regulatory agencies.

Validation can be achieved through two steps:

a. *Read in algorithms with input and output variables or tasks to regenerate results generated by SAS® programming*
b. *Generate same outcomes in multiple programming languages through algorithms*

If we shift to a complete algorithms based department then we will be able to reach to a point to Plug-in code and create eCRT packages based on multi-lingual programming languages.

SCHEMA:

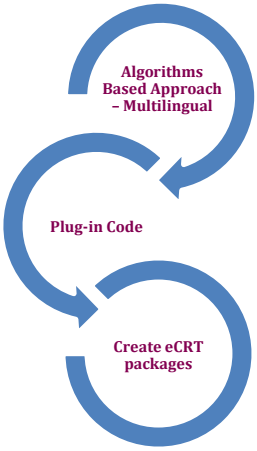

Figure.2 Algorithms based approach that support multi-lingual programming languages to generation of eCRT packages for regulatory submissions

FUTURE

The future for this approach looks promising with revolution in Artificial Intelligence. The derivations from dataset specifications, SAP etc. are extracted into define.xml. In similar manner, we can prepare XML files with algorithms (machine readable) . The AI tools can be trained on a dataset of XML files and corresponding programming code in different languages. Once the AI tool is trained, can be used to generate accurate code from XML file.

Currently ChatGPT® (OpenAI) has capability to translate algorithms to desired programming code. With this advancement soon we will be in an era, creating and validating Algorithms.XML that can readily be translated to multiple programming languages, powered by AI to create eCRT packages.

CONCLUSION

Our article can be summarized as following : This research is not to compare Opensource programming with SAS® rather to streamline processes, increase efficiency and flexibility for statistical programming department, diverse and multilingual. Shift to algorithm based department, test and validate algorithms to mitigate risk in using Open Source programming to generate eCRT packages for regulatory agencies. Lay a path for future so that these machine readable algorithms can be used to train AI tools to generate multi-lingual code (Open Source code) that can be used to generate complete eCRT packages with results consistent across programming languages, even better validated packages!

REFERENCES

1. US Food & Drug Administration Study Data Technical Conformance Guide: March 2018 www.fda.gov/media/88173/download

2. US Food & Drug Administration Statistical Software Clarifying Statement: May 2015 www.fda.gov/media/161196/download

3. Overview of reverse engineering and reuse research: April 1992
www.sciencedirect.com/science/article/abs/pii/0950584992900809

4. On the inference of Turing Machines from Sample computations 1972
https://www.sciencedirect.com/science/article/abs/pii/0004370272900483

ACKNOWLEDGMENTS

AstraZeneca® – Oncology Biometrics

AstraZeneca® – R in Biometrics (Innovative Use of R) Workstream

AstraZeneca® - China Biometrics Team

AstraZeneca®- Information Science Team

PHUSE® – Data Visualisation & OpenSource Technology - Clinical Statistical Reporting in Multilingual World Working Groups

R Consortium®

Powered by ChatGPT®

CONTACT INFORMATION

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

Author Name : Soma Sekhar Sriadibhatla
Company: AstraZeneca
Address: Gaithersburg, MD
Email:somasekhar.sriadibhatla@astrazeneca.com