PharmaSUG 2020 - Paper SI 173

PROC Future Proof;

Amy Gillespie, Susan Kramlik, Suhas Sanjee, Merck & Co., Inc.

ABSTRACT

Clinical trial programmers are critical contributors to regulatory submissions, manuscripts, and statistical analyses. They operationalize analysis plans to create high-quality, innovative, compliant deliverables according to stakeholders' needs. Clinical trial programmers author programming code and leverage programming standards to produce deliverables in a validated, efficient, and reproducible manner. However, the function itself has stayed relatively constant for more than 20 years, and it is natural to question whether there are opportunities and a need to transform the clinical trial programming role for continued success. This paper evaluates recent advances in technology and the clinical trial programming skillset to identify opportunities for improved programming efficiencies and compliance to regulatory requirements while ultimately optimizing the programming function. Use cases leveraging natural language processing (NLP) and linked data will be explored to evaluate whether digital solutions are applicable within clinical trial programming processes. The use of different software tools and methods will also be evaluated. We expect this paper to be the first of a series of publications on this topic.

KEY WORDS

Clinical trial programming; natural language processing; linked data; digital; software; data science; python

INTRODUCTION

The clinical trial programmer has an important role in the drug and vaccine development process. A clinical trial programmer operationalizes analysis plans to create high quality, innovative, compliant deliverables according to stakeholders' needs in a validated, efficient, and reproducible manner. All data collected for a clinical trial are processed through the computer programs written by a clinical trial programmer. The clinical trial programmer utilizes pieces of disparate information such as analysis plans and data and synthesizes them by authoring programming code. This code generates important knowledge in the form of analysis results, new data, tables, and figures which inform critical decision making for both new and marketed drug and vaccine products.

A clinical trial programmer's domain knowledge differentiates him or her from other data scientists. The domain knowledge consists of a comprehensive understanding of clinical trial process, clinical trial data and its complexities, the ability to identify, interpret and apply relevant information from the study protocol and analysis plan, and knowledge of industry and regulatory standards. Clinical trial programmers use their professional programming abilities plus their domain knowledge to complete their job efficiently, with high quality and according to compliance requirements.

The contributions of a clinical trial programmer are numerous, and while the overall job process has remained unchanged for many years, incremental advancements for efficiency, quality and compliance have been implemented mostly through the development of programming standards such as macro libraries, template code, and automation.

But has it been enough? Has the clinical trial programming role evolved to fundamentally improve the way work is completed? Have new processes and tools been transformative and have technology advances been appropriately leveraged? And finally, have we anticipated future business and stakeholder requirements and developed methods to minimize impact? In other words, is the clinical trial programming role future proof?

FUTURE PROOFING

Companies in all sectors have been leveraging the power of digital technologies to transform their businesses. There is a compelling desire and need to innovate and modernize ways of working and one way to do that is to leverage technology. Technology however, moves fast and solutions may only stay relevant for a short time. This is particularly problematic for large companies in regulated industries which often require considerable time to adopt new technologies and innovative ways of working.

The question of staying relevant is necessary and while the future can't necessarily be predicted, clinical trial programmers can prepare and adapt to changing technologies and advancements. Clinical trial programmers have the scientific knowledge and analytical skillset to do so. Clinical trial programmers possess problem-solving skills, a natural curiosity, and a willingness and eagerness to learn and grow their capabilities.

In a recent market research report compiled by Accenture, trends and opportunities for the clinical trial programming skillset include process standardization, end to end process automation, data visualization, intelligent study plan development, and artificial intelligence and machine learning across datasets for different signal detections. The report also indicates an opportunity for programmers to expand their skillsets and expertise in analytical capabilities and the ability to handle large datasets[1]. These opportunities are certainly attainable.

The following strategies can be considered to address the above items and future proof the clinical trial programming role. These ideas have been adopted from an article discussing 8 proven strategies to future proof a business.[2]

DON'T LIMIT SUPPORT TO ONE BUSINESS AREA FOR COMPLETE SUCCESS

Clinical trial programmers add value wherever clinical data requires manipulation, analysis and reporting. Clinical trial programmers have traditionally supported the creation of tables, listings, and figures for clinical trials and regulatory submissions using SAS software and working closely with their statistical and clinical stakeholders. But there are many more opportunities to consider. For example, programmers can extend their support to different stakeholders, work with other data sources and leverage more software and analytical tools.

At our company several years ago, a group of traditional "clinical trial" programmers transitioned their programming expertise to the observational and real-world data space. In this role the programmers collaborate with epidemiologists and health economists, manipulate, analyze, and report data from electronic healthcare databases, apply different analytical methods and utilize different tools. Their foundational experience in the clinical trial environment supported their success in the new role.

Another way to extend support to different business areas is by looking for new opportunities within the same area. One example is using machine learning within traditional analysis and reporting activities. Imaging and statistical monitoring are two potential opportunities for machine learning due to the larger quantity of data available in these areas.

A third example is using linked data. Linked data has shown promise for providing gains in quality and efficiency in the clinical study report. For example, the PhUSE CS Semantic Technology Working Group has demonstrated a way to potentially automate traceability and consistency by outputting results into a Resource Description Framework (RDF) data cube, querying it to create a table within a report document, and then linking the data in the body of a report to the data in the table[3]. The authors from the working group concluded that while the feasibility was demonstrated, more work needs to be done to make the approach usable for production. The capability would greatly reduce or even remove the possibility of transcription or copy-paste errors, and it would provide evidence of clear data traceability and consistency.

IDENTIFY AND START MANAGING RISKS

Two potential risks for the clinical trial programming role are the popularity of the data scientist role and the use of open source software. According to SAS Insights, "Data scientists are a new breed of analytical data expert who have the technical skills to solve complex problems – and the curiosity to explore what problems need to be solved."[4]

However, we suggest that expertise in clinical trial analysis and reporting and in the challenges faced in getting results to the customer are an advantage for the clinical trial programmer. Domain knowledge is also invaluable and is typically acquired through years of experience.

We shouldn't ignore that data science courses, certification programs and advanced degrees are new opportunities for clinical trial programmers to expand their skills. Expanding one's skills and knowledge is one element in identifying and managing risks. Staying current on upcoming near-term and longer-term changes in industry and regulatory requirements is another. Combining these with a problem-solving frame of mind to apply new learnings, technologies or techniques manages the risk of staying relevant. It also has the potential to develop transformative solutions.

LISTEN TO CUSTOMERS AND OBSERVE THEIR BEHAVIOR

Anticipating stakeholder needs is critical. Today, stakeholders want their analysis and reporting deliverables faster and more easily accessible. In today's world, people have most of their needs at their fingertips, through an app, or at the click of a button. Technology solutions are designed to meet the needs of different users and often includes opportunities for customization.

Clinical trial programmers must recognize that static tables, listings, and figures executed after a database lock may not be enough to meet all stakeholder needs. Opportunities to provide dynamic deliverables, custom visualizations, and on-demand results may be customer requirements for both the near term and future.

FOLLOW THE TRENDS

Artificial intelligence (AI) has become widely popular and extremely successful in many industries due to the availability of large volumes of data.

The transportation and retail industries are great examples of leveraging the power of AI. Data collected by Uber are used to predict customer needs, to analyze safety and to gain insight to optimize customer experience. Uber has expanded its business from delivering rides to also delivering food. Amazon has used large volumes of data to optimize its operations, and to gain insight and adapt to business needs[5].

Within biomedical research large amounts of data, such as from electronic health records, have become available. Processing and annotating the data and using it to train an artificial intelligence model has been where there are the most advances. The model is applied to evaluate new data and make decisions [6]. The ability to evaluate skin lesions images and diabetic retinopathy images are examples where advanced AI techniques show promise.

The use of AI in the clinical trial programming area requires additional thought and experimentation.

CREATE A CULTURE OF INNOVATION

Regulatory constraints, resource constraints, and expectations for exceptional quality and timeliness for voluminous and complex analyses are hallmarks and challenges of the clinical trial programming environment. Getting safe and effective medicine to patients is the ultimate need and focus. A culture of innovation improves the ability continually to achieve this end.

One way to foster a culture of innovation is to commission a clinical trial programming innovation team, with the expectation that the team will solve a complex problem or develop a new and impactful capability to enhance analysis and reporting quality and efficiency. In our company, this team exists as a core group of programmers representing multiple clinical trial programming functions. Membership is nominated by senior clinical trial programming leadership, and it rotates biennially. The two-year term allows the team to propose, research, and creatively solve a problem, presenting progress along the way. Innovation team members retain their primary clinical trial programming role, but they are also given license to flex creative muscle and talent researching and collaborating on how to employ new technologies to solve problems and enhance capabilities.

In addition to solving problems within a formal innovation team, clinical trial programmers are encouraged to experiment

and share as another component of an innovative culture. They are rewarded for thinking creatively, applying innovative solutions at the project level and for sharing these ideas across the organization. There are multiple forums to encourage sharing, including a forum where programmers share knowledge by rehearsing for external conference presentations, and they receive constructive feedback. These presentations are open to the entire statistical programming organization.

Engagement and knowledge sharing in external industry working groups and participation in conferences is a third component of building a culture of innovation, and it applies industry-wide. Collaboration in these external groups expands knowledge beyond our "four walls" from the formal presentations and posters, as well as the synergy from informal conversations and networking in working groups and at conferences.

A professional culture that encourages career growth via continued learning also contributes to a culture of innovation. The abundance of online courses and professional training in newer technologies make learning more convenient than ever. It is up to the imagination and ingenuity of the clinical trial programmer to think of ways to apply the learning to automate, improve efficiency and promote quality.

USE CASES

Recent advances in data sciences can be an opportunity to increase operational efficiency and improve compliance. To future proof the role of clinical trial programmers, we identified a few opportunities leveraging technology solutions, NLP and linked data, within the clinical trial analysis and reporting process. These opportunities were identified to increase efficiency and compliance. Applying NLP and linked data principles in the creation, reporting, and review of analysis results has potential to improve the current time and resource intensive process for both the sponsor and regulatory agencies[7].

NLP AND LINKED DATA

Wikipedia defines NLP as a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages and how to program computers to process and analyze large amounts of natural language data.

Linked data is structured data (metadata) that is interlinked with other data in a way that it becomes more useful through semantic queries. It builds upon standard Web technologies such as HTTP, RDF and URIs and extends them to share information in a way that can be read automatically by computers. The linked data can be queried using state-of-the-art query technologies for presentation in diverse, dynamic, and user-friendly ways.

USE CASE #1: DOCUMENT DRIVEN PROGRAMMING

Before any clinical trial programming begins, a careful review of the protocol, statistical analysis plan, mock table shells and analysis dataset specifications must be completed by the statistician and clinical trial programmer. Information from these documents are then used for developing programs to create analysis datasets, tables, listings and figures (TLFs). Information from these TLFs and study documents is then used by statisticians, clinicians, and medical writers to author CSRs, manuscripts, posters, and presentations. Multiple people in different functions must manually review many sources of information to complete their different tasks. Is there a better way?

One solution may be the application of document driven programming using NLP. NLP can potentially be used to extract required information from the study documents to create metadata files which can then be used to automatically generate some parts of the analysis and reporting programs. A second solution is to use linked data techniques to convert study documents into RDF which enables information sharing between documents and programs. NLP & linked data concepts can then be used to parse through the program specifications, study documents and associated programs to flag inconsistencies.

The use of NLP and linked data can eliminate some of the manual reviews and increase the effectiveness of different tasks. This results in efficiency gain since it reduces manual intervention by retrieving information automatically from the study documents. It also increases accuracy of analysis results by making sure that the information used is consistent across study documents and between documents and programs.

USE CASE #2: CONSISTENCY CROSS CHECKING

A consistency cross checking use case is to ensure information in the SDTM comments domain (CO) is consistent with other structured SDTM datasets. One example is reviewing comments to identify patient deaths and then cross checking them with the death details (DD) SDTM domain, ensuring all deaths are captured in the death CRF page. This is an important check because there are some instances where the investigators enter the comments about patient death but fail to enter the patient death information in the death CRF. These deaths will therefore be missed in the DD SDTM domain which is used for CSR reporting. We therefore propose to use NLP techniques to parse through the free text in comments to identify patient deaths. This can then be cross checked with the SDTM DD domain to flag any inconsistencies. This use case can further be extended to other SDTM domains for adverse events and concomitant medications. Identifying such discrepancies and proactively addressing them prior to a submission will facilitate a high-quality data delivery, streamlined regulatory review, and faster approval.

A second consistency cross checking use case is using NLP and automation to review and validate information across submission documents. Reviewer guides, define.XML, the analysis results metadata, protocol, and study reports may contain duplicative information. An important quality check is to ensure the information presented in these different documents is consistent and accurate. Often these consistency checks are completed manually, requiring significant time and resources. The use of automation in this use case is expected to greatly improve overall document quality and reduce the amount of time spent conducting manual reviews.

USE CASE #3: LINKING ANALYSIS RESULTS IN CSRS AND MANUSCRIPTS

In the world of linked data, every data point has a Uniform Resource Identifier (URI). URI can be a name, locator, or both for an online resource whereas a Uniform Resource Locator (URL) is just the locator. For example, your name could be a URI because it identifies you, but it couldn't be a URL because it doesn't help anyone find your location. On the other hand, your address is both a URI and a URL because it both identifies you and it provides your location.

There is no straightforward way in the current state to link the statistics in the study reports to the TLFs delivered by clinical trial programmers. Authors of study reports and manuscripts often review hundreds of TLFs to identify which statistics to include in reports and publications. When the TLFs are refreshed the author need to make sure the statistics used in the reports are still accurate and consistent with the updated TLFs. This process is manual, cumbersome and hence error prone. Linked data can potentially be used to provide better traceability between TLFs and study reports.

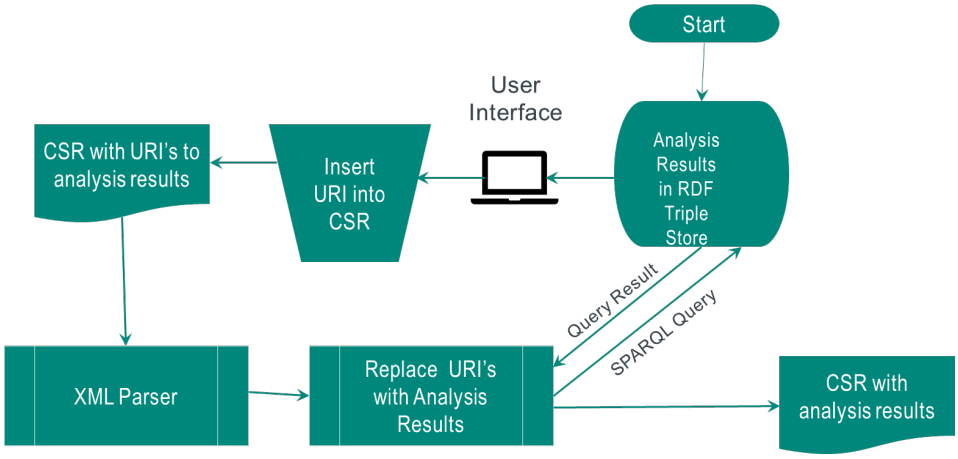

Figure 1. Illustration of Linking Analysis Results in CSR

To accomplish this, analysis results from TLFs can be stored as linked data (RDF) [3]. The URI of these results can be referenced in the study reports rather than copying the statistics from TLFs. This will make sure that the results used in study reports are consistent with those in the TLFs. This will also reduce manual effort required for performing quality checks on the study documents. The process of linking analysis results for clinical study report (CSR) is illustrated in Figure 1. This applies for any report such as manuscripts, conference presentations, investigator brochures where study results are reported.

A word document (.docx file) is a ZIP archive of XML files. The contents of study reports in .docx format can be read into Python using an XML parser. The content can then be parsed to identify URIs which can be replaced by the results after querying the RDF triple store using SPARQL. Analysis results in RDF format can be loaded into Python and queried using RDFLIB package. Creation of new DOCX file after replacing the URIs with the analysis results can be accomplished using the package PYTHON-DOCX.

Summary of Results:
Disposition:
A total of http://www.example.org/dc/tab1x01/ds/obs07 subjects were randomized and entered the double-blind treatment phase.
The number of subjects randomized to each treatment arm was:
http://www.example.org/dc/tab1x01/ds/obs01 to placebo,
http://www.example.org/dc/tab1x01/ds/obs03 to the
Xanomeline low dose treatment group and http://www.example.org/dc/tab1x01/ds/obs05 to the Xanomeline high dose treatment
group. Of the http://www.example.org/dc/tab1x01/ds/obs07 subjects randomized to treatment, http://www.example.org/dc/tab1x01/ds/obs47 completed the treatment phase (Week 24).

Figure 2a. CSR with URIs to analysis results

Summary of Results:
Disposition:

A total of 254 subjects were randomized and entered the double-blind treatment phase. The number of subjects randomized to each treatment arm was: 86 to placebo, 84 to the Xanomeline low dose treatment group and 84 to the Xanomeline high dose treatment group. Of the 254 subjects randomized to treatment, 118 completed the treatment phase (Week 24).

Figure 2b. CSR with URIs replaced with analysis results

Figure 2a. shows a section of CSR with URIs and Figure 2b. shows the same after the URIs are replaced with analysis results returned by SPARQL queries. Linking analysis results with study reports will allow authors to automatically refresh content resulting in accelerated timelines and higher quality. Less time can also be spent on tedious quality checks.

USE CASE #4: IDENTIFYING MORE STANDARDIZATION OPPORTUNITIES

The processes of identifying standard analysis and reporting deliverables is often manual and usually relies on a cross functional team of clinical trial programmers, statisticians and clinical scientists to identify the need for new standards. This particular use case proposes a more automated approach by developing an algorithm to compute a similarity score between two files. Files can be SAS programs, analysis datasets, tables, listings or figures.

One example of this application is automatically mine programs and outputs across protocols to identify potential candidates for standardization thereby reducing the need for individual protocol teams to create the same program and output separately. A second example is computing a similarity score between standard program templates available in the global library and those used at a study level. This score is an indicator of how much customization is being done by the protocol teams to the standard templates. A high score indicates a need for changes to be implemented to the standard template to minimize the amount of customization at the protocol level.

CONCLUSION

Technology provides opportunities for clinical trial programmers to broaden skillsets, contribute in new and different ways and excel in their role. NLP, linked data, and a greater use of automation are a few examples which can be leveraged to expand and evolve clinical trial programming processes for increased operational efficiency and compliance improvements. It is important to remember however that the domain knowledge of the clinical trial programmer and the soft skill “human” element of clinical trial programming are irreplaceable and are just as important as technological opportunities are in evolving the role. Humans possess soft skills of imagination, creativity, vision, ingenuity, ability to reason and think critically, empathy, communication and ability to question[8]. While AI can do routine things well, it cannot substitute humans in thinking and soft skills. If one automates too much, then the skeptical

view, creativity and ingenuity are removed. There is no specific procedure or "Proc Future Proof" to immediately evolve the clinical trial programming role. However, an investment in innovation, experimentation, and upskilling will ultimately result in long term success.

REFERENCES

1. R&D Clinical Operations Market Research Report, Accenture, March 2020

2. Katrel, Harshal. How to Future Proof Your Business? https://www.profitbooks.net/future-proof-business-8-proven-strategies/

3. Marc Andersen, Marcelina Hungria, S. Sanjee. Generating Analysis Results and Metadata. PhUSE EU Connect, 2016

4. SAS Insights. What Is a Data Scientist? Who they are, what they do, and why you want to be one. https://www.sas.com/en_us/insights/analytics/what-is-a-data-scientist.html

5. PhUSe Educating for the Future Working Group: Data Engineering Project "Use Cases in Other Industries" https://education.phuse.eu/eftf/data-engineering/other-industry-use-cases

6. Zhu, L., & Zheng, W. J. (2018). Informatics, Data Science, and Artificial Intelligence. JAMA, 320(11), 1103–1104. https://doi.org/10.1001/jama.2018.8211

7. PhUSE CS Semantic Technology Working Group, Analysis Results & Metadata Project. "Improving the Analysis Results Creation and Use Process: Modeling Analysis Results & Metadata as Linked Data". [Draft White paper, publication pending on PhUSE Wiki]

8. Marr, Bernard (2018) 7 Job Skills Of The Future (That AIs And Robots Can't Do Better Than Humans). Forbes August 6, 2018. https://www.forbes.com/sites/bernardmarr/2018/08/06/7-job-skills-of-the-future-that-ais-and-robots-cant-do-better-than-humans/#1cb3ea3f6c2e

ACKNOWLEDGMENTS

The authors thank their colleagues in the BARDS Statistical Programming organization for their collaboration, expertise and inspiration.

CONTACT INFORMATION

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

Amy Gillespie
Merck & Co., Inc
amy.gillespie@merck.com

Susan Kramlik
Merck & Co., Inc
susan.kramlik@merck.com

Suhas Sanjee
Merck & Co., Inc
suhas.sanjee@merck.com