
**Paper DV13**

# Best Practice for Interactive Analysis for Decision-making: An update from PHUSE Data Visualization Project

Xiangyun Wang, Genentech, South San Francisco, USA
Zachary Skrivanek, Eli Lilly, Indianapolis, USA

## ABSTRACT

The goal of the best practice for interactive analysis PhUSE project is to evaluate how interactive visualization tools can be used as part of regulatory review process. How to balance the best of self-exploratory analysis with the rigor of regulatory requirements? What is the feasible framework to share interactive tools with the FDA? Is there a need to provide an industry-wide common user interface for regulatory reviewers to easily digest data from different organizations? How to address those and many other challenges facing interactive analyses in a regulatory setting is what we aim to explore.

We will provide an update on regulatory and business considerations for different approaches and share our learning from an html interactive display pilot and Docker R-Shiny pilot with FDA. Next steps will be covered. We invite everyone to collectively explore the best way to leverage emerging technologies to advance how we consume and share data.

## INTRODUCTION

Drug development costs are a growing problem in the pharmaceutical industry and interactive analyses can be a way to mitigate those costs by decreasing review time and possibly improving decision-making. Interactive analysis refers to analysis and visualization results generated using an interactive tool, whether it is using a proprietary software such as JReview, Spotfire and Tableau, or an open-source programming language with web production capabilities like R/Shiny and Python/Dash. The main advantage of interactive analysis is its capability to allow end-users to stay in the workflow and answer questions in a fluid manner. The visualization-based self-analysis enables quick data insights and thus informed decisions without back-and-forth waiting for communication or changes. It also provides an effective and transparent communication platform that all parties including pharmaceutical companies, regulators, payers, and even medical practitioners can look at the data using the same sets of interactive displays.

Can sponsors provide interactive visualization tools as part of the submission to move away from the “digitized paper”[1]? What is the best practice for interactive analysis in order to ensure compliance in a regulatory setting? Formed in 2017, the FDA/PHUSE collaboration project aimed to explore and assess the benefits and technical feasibility of providing interactive analysis as part of a submission or regulatory interaction. The project team is “Best Practices for Interactive Analyses for Decision Making & Submissions” and resides within the Emerging Trends and Technologies Working Group. Members of the project team consist of PHUSE membership from the pharmaceutical industry, vendors, consultants, and FDA. The PHUSE steering group provides oversight and review of this project.

In this paper, we will share our key learning based on two proof of concept (POC) projects and a pilot with real submission on how to provide interactive analyses as a reviewing aid to the FDA.

## BEST PRACTICE FOR INTERACTIVE ANALYSIS

The unique feature of interactive analyses for submissions is that it is flexible; the user can determine how they want to interrogate the data within the limits of the application. This flexibility can lead to novel insights, but it also poses some challenges such as “How can we ensure that end-users understand their responsibilities in using these tools properly, which sometimes could require deep understanding of the data and analysis?”

In addition, interactive analyses require a platform to host the interactive analyses. Both the platform and the application need to meet regulatory requirements to ensure quality. For instance, how can we validate the interactive

visualization tools with the large number of filter permutation? Interactive analyses pose unique challenges when it comes to submission that we have not faced with traditional methods where the output is static.

Currently each organization has followed its own practice for interactive analyses by defining the scope of use and providing guidance on validation and proper documentation. The PHUSE project team intended to provide recommendations based on learning from sponsors' practices; however, it is out of scope for this paper.

Below are the principles of quality assurance, which we all follow with any submission.

- Reproducibility / Archiving: ability to reproduce analysis results based on requirements independent of software/tools
- Validation: the process of checking that a software system meets specifications and that it fulfills its intended purpose.
- Trace-ability: ability to identify the source of the data for a presentation of analysis results, starting from the data collection through all intermediate data transformations that led to the final presentation of analysis results.

How we implement the above principles of quality assurance for interactive analysis depends on the approach we take to share it with the regulators.

## APPROACHES CONSIDERED

Unlike static outputs, sharing interactive visualization tools with external partners especially with regulators is quite challenging. Many factors need to be considered such as type of software used (commercial versus open-source), how to protect data integrity and confidentiality for both sponsors and regulators, and infrastructure and process to enable data sharing and access while ensuring quality and security of the information being shared. How these factors are addressed depends on how the interactive analyses are shared.

Below summarizes the five approaches we have explored thus far with pros and cons highlighted for each approach. The first four approaches require a platform to host the application. Either the FDA hosts it, the sponsor hosts it or a third party hosts it. These pose unique challenges, each having their advantages and disadvantages. The fourth approach is a variation of the first one, but the FDA could host the interactive analyses on a virtual machine instead of a dedicated server. All four approaches require support from Information Technology to implement. The last approach, "HTML file in submission (eCTD)", just requires a modern web browser, HTML 5 compliant, in order to run the application.

| Platform | Pros | Cons |
|---|---|---|
| 1. FDA hosted. | All data and analyses stay behind FDA firewall. | Lack of infrastructure. No Spotfire server for mass consumption. No R/Shiny server. FDA would need to add capabilities to handle this. Has cost implication for FDA. Need IT support. |
| 2. Sponsor hosted. | All data and analyses stay behind sponsors' firewall. | Regulators would get access behind sponsors' firewalls. FDA concerned about tracking. Sponsors concerned about confidentiality. Sponsor needs to host indefinitely and show integrity maintained. Need IT support. |
| 3. Third party hosted. (This could be a stand-alone server or a cloud-based solution.) Both sponsor and regulatory agency have access to the same interactive analysis. | Mitigates against concerns by FDA of tracking. Mitigates against sponsors' concerns around confidentiality. | Sponsor needs to host platform indefinitely and show integrity maintained. Need IT support. |

| | | |
|---|---|---|
| 4. Containerization. Submit entire app using containerization within the FDA submission process (eCTD). | All data and analyses stay behind FDA firewall. Addresses the reproducibility issue. | Requires the FDA to support the app on a virtual machine. Need IT support. |
| 5. HTML file. Submit entire app as stand-alone web files (HTML, jason, js, etc…) in submission (eCTD) | All data and analyses stay behind FDA firewall. Ease of use. | Need to use modern browser. No statistical engine. The amount of data that can be analyzed is limited on the client's side. Limit to programming languages that are browser compatible. |

Among the above five approaches, the team focused on the last two: 4. Containerization; 5. HTML file. Before the team engaged with a study team to pilot these approaches, we first did a technical proof of concept for each approach using a Sample Submission application number[1] to emulate a real submission with fake data. We created a fake CSR, cover letter, data sets and applied an interactive analysis to the data sets. We then used the sample submission to emulate a real submission and then we met with the FDA to verify if it was successful. Once the POC was considered a technical success, then the team would follow up with piloting the approach with a real submission.

The rest of the paper will elaborate in detail about each POC project including how we followed the eCTD process to include the interactive visualizations as part of standard submission package, how we addressed validation and documentation, feedback from FDA and summary.

## HTML SUBMISSION POC PROJECT

The motivation behind an HTML submission is that the reviewer would not need any specific software or system other than a modern (HTML5 compliant) browser. This addresses the resources issue raised with the 'FDA hosted' approach in which the FDA would have to host a system, such as RStudio professional or Spotfire web server, in order to provide them an interactive data visualization. The FDA would not need to install any system or software in order to be able to interact with the data visualization. All they need is a modern browser like Chrome or FireFox, which is freely available. Browsers are built with the capability to interpret Javascript, so there is no additional software needed. By providing the HTML file along with the Javascript file and all of its dependencies, the reviewer does not even need access to the internet to interact with the data visualization.

Another goal of the HTML submission is to address the issues of the learning curve associated with using new software tools, such as Spotfire. The interactive data visualization is designed in such a way that it is intuitive to use, not requiring lengthy instructions. To distinguish this from a full-blown software tool we refer to this interactive data visualization as an interactive display.

### TECHNICAL PROOF OF CONCEPT

As part of the PHUSE working group on "Interactive Analysis for Decision Making & Submission", we explored the feasibility of providing an interactive analysis embedded in a stand-alone web-page, using HTML and Javascript to create the data visualization. A technical proof of concept was conducted on February 18, 2019. The team utilized the sample submission process[2] to test the system with this technology. In addition to the interactive data visualization, we also included an animation of data, in MP4 format, to verify that reviewers would be able to access this as well.

The team used publicly available data to create a Volcano Plot[3] in HTML and Javascript. See Description of Volcano Plot Used in 'HTML Submission' for details about the Volcano Plot. The dependencies were packaged into the project at development time, so you did not need them at run time. This added ~250 kB of additional code that would not be needed if we were serving files independently.

[1] https://www.accessdata.fda.gov/cder/eCTD/sample/sample/fd_07_01_0050.htm
[2] https://www.fda.gov/drugs/electronic-regulatory-submission-and-review/submit-ectd-or-standardized-data-sample-fda
[3] O'Connell M, Knudsen S. Statistical graphics and reporting in drug development. Paper presented at: PHUSE 2006; October 9-11, 2006; Dublin, Ireland. Paper TS04


They provided a cover letter in addition to a fictional CSR with a hyperlink embedded in the document that references the Volcano Plot. In the cover letter and fictional CSR, it was noted that the reviewers must use a modern browser (i.e. HTML 5 compliant) in order to be able to view the Volcano Plot.

The team considered five different ways to populate the backbone of the electronic Common Technical Document, eCTD. The five different ways are listed below, indicated by the section (5.3.1.x) where that method was utilized:

1. 5.3.1.1
    a. CSR, mp4, html, Javascript, and icon files all listed in the backbone in 5.3.1.1 and referenced within the STF with file tags of "Study Report Body"
    b. Folder location (xlink values) place all files in 5311-ba-stud-rep section under a study subfolder.
    c. Risk – Potential for long paths; maximum path name: 180 characters (some agencies it's 150 characters, others 230 characters)
    d. Positive - Follows most closely with guidance, but not as organized
2. 5.3.1.2
    a. CSR, mp4, html, Javascript, and icon files all listed in the backbone in 5.3.1.1 and referenced within the STF with file tags of "Study Report Body"
    b. Files related to the html (html, js, and ico files) have a file location (xlink) of m5/html/study/, similar to how datasets and CRFs are handled
    c. Positive - All HTMLs together in html folder - allows us to reduce the path
3. 5.3.1.3
    a. CSR, mp4, and html files are listed in the backbone and STF. JS and ICO files are provided in the submission but not referenced in the index.xml file.
    b. Risk – Does not follow the existing specifications, which states that unreferenced files should not be submitted. Though this does not cause rejection at the US FDA, could cause rejection for other agencies.
    c. Positive - Single html file for each Volcano Plot reduces clutter in the viewing tool. Easier for the reviewer to go directly into the Volcano Plot.
    d. Requirement: Need to be able to send un-referenced files - come across as warnings to the agency
4. 5.3.1.4
    a. CSR is the only file referenced in the backbone and STF.
    b. Positive – Backbone displays fewer files. Control the access to the files through the narrative within the CSR.
    c. Risk - None of data visualizations would be available directly from the viewing tool, reviewer must access via links from the CSR. Similar to 5.3.1.3, unreferenced files are sent in the submission.
5. 5.3.1.5 – (Method discussed but not attempted)
    a. Place all files in 5.3.1.5, apply new file tag to Volcano Plot and supporting files.
    b. Requirement – add additional values for available file tags (such as Data Visualization), apply the new tags to assist in the organization of the files. Would require industry group to influence specifications.
    c. This would be specific to US and Canada, as other countries do not utilize STFs for organizing files within a study.

The team decided to test the first four approaches in this sample submission. The fourth approach was not desirable because the reviewer would not be able to see the data visualization directly (with a simple 'click' of a hyperlink), which is not desirable from a user experience point of view. The fifth approach would require a change to the structure of the eCTD which would need to be coordinated across the industry and FDA. The team did not have the resources to make this happen, but this could be considered for the future if we want to make these types of submissions routine.

The sample submission was received successfully, and we met with the reviewer and a subset of the team on April 30, 2019. The reviewer was able to access the Volcano Plot and the MP4. He did commit the error (in spite of a warning in both the CSR and the cover letter) of opening the plot in an antiquated browser, Microsoft Explorer. We had to tell him to open it in a modern browser. We reviewed three Volcano Plots from the three different forms of

submission and determined that 5.3.1.2 was the best. It was easier to access than 5.3.1.1, the first method, and was more compliant than the third method, 5.3.1.3.

**PILOT STUDY**

After the successful proof of concept, we piloted the Volcano Plot in a real submission by Eli Lilly & Co. We created a Volcano Plot based on real data, similar to what was done for ~~shown in Exhibit A: Materials for FDA~~ Technical Proof of Concept with the same dependencies as described in the Technical Proof of Concept section. The Volcano Plot was submitted along with the drug submission in the third quarter of 2019. Since this was a pilot, we submitted the data contained in the Volcano Plot using the traditional table approach. That would mitigate any negative consequences on the review if something went wrong with the Volcano Plot in the submission.

In the technical proof of concept conducted in the first quarter of 2019, we provided a Volcano Plot that supported information within CSR. In the actual submission, the Volcano Plot was representing a table in the Summary of Clinical Safety, which is a document in Module 2 that does not utilize Study Tagging files. As a result, the Volcano Plot was added into the backbone in 2.7.4 along with the Summary of Clinical Safety (SCS).

**VALIDATION**

The study team derived and validated the Analysis Results Data Sets (ARDS) that contained the inferential results for the 78 pages of tables that were the basis of the Volcano Plot. The data set was validated using independent replication. This data was converted to a json file and then was used in the HTML code. We validated the plot using independent replication with a different software than that used in the submission. We used R and we leveraged the package plotly to create the replicate program. The validation approach relied on the fact that both data visualizations were based on the same validated ARDS. The study team cross-checked the visualizations, comparing the HTML based Volcano Plot to the one created by plotly, the replicate program, for different combinations of the values in the two drop down menus. Not all values of all combinations were assessed. The study team also checked the values of the statistics associated with the analyses of some of the Preferred Terms in the Volcano Plot against the validated ARDS.

**TRACE-ABILITY**

The production of the Volcano Plot can be traced back to the json file, which can be traced back to ARDS, which can be traced back to ADAM, which can be traced back to SDTM, which can be traced back to source data. All programs and data sets were stored in a validated system (CFR part 11 compliant).

**COMPUTATIONAL REPRODUCIBILITY / ARCHIVING**

The author and validation author (who created the replicated program) based their programs on a requirements document. They stored the requirements document as well as their programs in a validated system (CFR part 11 compliant) along with the ARDS and ADAM and stats programs used to create the ARDS and ADAM. They used GitHub and forked a clone of the version that was validated into the repository. The versions of the softwares were recorded and the package packrat was used to save the dependencies in the replicate program that used R. The requirement documents and the programs were e-signed following the company's SOPs.

**DISCUSSION**

There were no technical issues with the pilot submission. The FDA received the submission with the Volcano Plot successfully.

Many other features could have been added to this Volcano Plot. For instance, we could have included the risk difference as an alternative to odds-ratios using a drop down menu. We could also have included a feature where the reviewer could drill down to see patient listings of relevant information for all of the patients suffering from a particular adverse event. Given the technical success of the pilot we are confident that there are no technical impediments to adding these features in future submissions. The point of the pilot was to carve a technical path to be able to share interactive data visualizations as part of submissions and this was achieved. The only limit to what type of interactive features you can include are the limits inherent in HTML and Javascript. The key to making this interactive visualization broadly available is that it relies on a modern web browser to be able to 'run' the interactive data visualization.


There is no analytical engine inside browsers, hence we could not compute analytics dynamically. All analytics had to be pre-calculated and contained within Analysis Results Data Sets. These were used to create the Volcano Plot.

Analysis Results Data Sets (ARDS) play a key role in this HTML based approach to providing an interactive data visualization as part of a submission. All inferential results need to be computed ahead of time since the browser has no analytical engine. To make this approach routine it is important to standardize ARDS, just like the industry has standardized SDTM and ADaM. This is the natural next step for CDISC: develop standards for ARDS.

In fact, this is just a better way to approach statistical deliverables in general. Currently, stats departments typically derive Tables, Figures and Listings (TFLs) directly from ADaM (and sometimes from SDTM and rarely from data upstream, though this happens occasionally). If they do produce ARDS, it is often a by-product of their TFL production. This process merges the workflow of the analytics with the presentation of the analytics.

We believe it would be wiser to separate these two steps. It is wiser from a programming point of view. The programming to produce ARDS does not get confounded with the programming to present the ARDS. In fact, these two work flows might better be executed by people with different skill sets. The first group focuses just on the analytics and producing ARDS. The second group just focuses on the presentation of the results, whether that be a TFL or an interactive data visualization.

It is wiser from a reproducibility point of view. Study teams need to represent the same results in different venues with different formats. This is often done by a medical writer and CRP transcribing the results from a source stats table into another TFL. This introduces the possibility of transcription error. To mitigate against transcription error, a peer review typically conducted. But this is no guarantee to avoid all transcription errors. We believe that a programmatic approach to re-deriving TFLs based on ARDS is a much better guarantee that you will have consistent results. And the only evidence that a regulator has that a peer review was conducted is documentation. In contrast, if the new TFL were derived programmatically based on ARDS, it could be validated using the gold standard of replicate programming and this could be easily verified by the regulators using traditional methods like logs and e-signatures.

It is reasonable to conjecture that one could submit an HTML visualization to any regulatory agency that accepts electronic submissions since the main requirement is that they have access to a modern browser which are freely available and widely used. It is still important to verify these assumptions. It is also reasonable to assume that other interactive features can be included such as patient drill downs since this can easily be done in HTML + Javascript. But it is always wise to test these assumptions in case the assumptions are wrong or more likely wrong in certain circumstances. For instance, the data for patient drill downs can get too large for the browser due to limits on the client's RAM and/or CPU. It would be useful to understand these limits.

**SUMMARY**

The HTML approach is a viable way to provide interactive visual analytics as part of submissions. We had a successful technical proof of concept with the FDA using a real submission by Eli Lilly & Co in July. It is reasonable to assume that additional interactive features would work and that other regulatory agencies who receive electronica submissions would be able to review the data in the interactive tools, but it is important to do additional pilot tests to truly understand the boundaries of this approach and the technical limits with other regulatory agencies.

This HTML approach is promising. It is very flexible and there are low barriers to accessing this technology. The main limitation is that analytics cannot be done interactively. Instead, the analytics need to be computed ahead of time and the interactive tool is simply a reporting tool. Consequently, it is important that we have standards for Analysis Results Data Sets if this approach is to become routine.

As noted, in the pilot we provided the Volcano Plot as supplemental information, all of the data contained in the Volcano Plot was also contained in two 39 page tables that were part of the submission. In the future, we hope that we can replace these tables completely, and just rely on interactive data visualizations. They contain all of the information contained in the tables; they are just simply a better way of conveying complex information.

## CONTAINERIZATION SUBMISSION POC PROJECT

The PHUSE Interactive Analysis Working Group conducted a second technical proof of concept (POC) project to explore how to submit an open-source based interactive data visualization tool with the FDA via containerization approach. Genentech served as the industry sponsor on this POC project. The POC team decided to deploy a simple

**Commented [WX{SF1]:** ARDS already follow CDISC ADaM standard for BDS in terms of metadata. Are you referring to contents? If so, I'm not sure whether it is possible to fully standardize ARDS as it would be likely indication specific.

**Commented [WX{SF2]:** I do not agree with this paragraph. ARDS is ADaM dataset, and thus the base for CSR TLG reporting. Please consider to revise or remove.


Shiny application in R to visualize dummy demographic data, using Docker as the container to package the interactive tool. The test submission was sent to the FDA via CD-ROM (as advised by FDA Test eSub team) on December 20, 2019.

**WHY CONTAINERS?**

The tools for interactive visualization are expanding beyond off-the-shelf visualization suites, encompassing web-hosted visualization frameworks and tools with extensive cohorts of installation dependencies. These tools often provide more flexibility and customizability that make them appealing for clinical analysis, but the challenges with installation and reproducibility of an analytic environment pose new challenges in delivering interactive data visualizations. Containers are a modern solution to this problem and are used extensively to host self-contained web services. This same technology can be extended to provide thoroughly reproducible analytic environments and delivery of interactive visualizations with complete articulation of software dependencies and virtualization of hardware dependencies. Docker is one of several popular flavors of this technology, and was chosen due to its thorough documentation, large user base and good cross-platform support.

**WHY R SHINY?**

R-Shiny is a web application framework for R. It can be easily used to turn analyses into interactive applications with intuitive and yet powerful user-interface. As an open source programming language, R is free, and Shiny can be accessible simply with a web browser. Most importantly, R offers an ever-growing toolkit with its vibrant and prolific user community. With its rich capabilities and ease of use, Shiny provides a promising interactive data visualization tool to support exploratory analyses for clinical data in the biopharmaceutical industry.

**TECHNICAL PROOF OF CONCEPT**

In this test submission, the team has provided a cover letter and all files including scripts which can be used by the Docker system executable to build and run a set of related docker containers. These containers serve both a Shiny server instance as well as an RStudio instance for interacting with the underlying app code. Notably, we did this all while conforming to existing eCTD guidelines. Below depicts the eCTD file structure under \123456\0001\m5\datasets\study02\analysis\adam, where

- \datasets subfolder contains adsl.xpt and reviewersguide.pdf;
- \programs subfolder has
- \apps: contains a demo R shiny tool
- \docker: include all docker files
- \r.library: contains a readme.md file. All R packages from docker container will go here.
- asl.r is an example R script file. All R script files used for a submission package will go to this folder.

```
123456
└── 0001
        ├── m1 ...
        ├── m2 ...
        ├── m3 ...
        ├── m4 ...
        ├── m5
        │   ├── 53-clin-stud-rep ...
        │   ├── 54-lit-ref
        │   └── datasets
        │       ├── bimo ...
        │       ├── ise ...
        │       ├── iss ...
        │       ├── study01 ...
        │       ├── study02
        │       │   ├── analysis
        │       │   │   └── adam
        │       │   │       ├── datasets
        │       │   │       │   ├── adsl.xpt . . . . . .  AdAM SAS XPT Dataset
        │       │   │       │   └── reviewersguide.pdf
        │       │   │       └── programs
        │       │   │           ├── apps
        │       │   │           │   └── demoApp
```

```
        │       │   │           │       └── app.r . . . . . . . Shiny web app script
        │       │   │           ├── asl.r . . . . . . . . . .  Example analysis script
        │       │   │           ├── docker
        │       │   │           │   ├── docker-compose.yml . . . . . . Docker script
        │       │   │           │   ├── install.r . . . . . . . . R package installer
        │       │   │           │   ├── rlib-installer.dockerfile . .  Docker script
        │       │   │           │   ├── rlib-rstudio.dockerfile . . .  Docker script
        │       │   │           │   └── rlib-shiny.dockerfile . . . .  Docker script
        │       │   │           └── r.library
        │       │   │               └── readme.md
        │       │   └── tabulations ...
        │       └── uspi ...
        └── util ...
```

Both README.md and reviewersguide.pdf provide users instructions on how to install and launch a docker container and how to start the demo R-Shiny tool. The reviewers guide has more detailed instructions including troubleshooting tips.

The delivered content includes three containers with shared dependencies. In addition to rlib_installer that captures all required R packages with dependencies for this analysis delivery, the R studio container provides reviewers flexibility for further modification or additional exploratory analyses, and the Shiny container allows users to review analysis and visualization results as well as to execute interactive features, such as a subgroup analyses via pre-specified filters. Below provides technical details for each container.

1. rlib_installer

Upon bootup, runs an install.R script which checks for the availability of two packages, “haven” and “cowsay,” installing these two packages and any dependencies if needed. Any packages installed during this process are stored within the container at /usr/local/lib/R/reg-library/. During execution, this path is mapped to the local path ../r.library.

2. rstudio

This container is a very slight modification of the rocker/tidyverse:3.5.2 base image, with the only modification being the addition of the /usr/local/lib/R/reg-library/ library path to the R_LIBS environment variable specified by the Renviron startup file.

3. shiny

Like the “rstudio” container, this applies the same environment variable modifications to the rocker/shiny:3.5.2 base image. One important design consideration concerns how the package library is installed. The team chose to install R packages on the host machine such that they can easily be reused across containers. This method is easy to debug and iterate upon, but does pose some risk to reproducibility as it allows for mutability after installation. A more robust alternative may include a package installation layer as a base image for all subsequent services and using established tooling like the renv package to validate a cohort of installed packages against a more robust measure like a SHA hash of the installation contents. However, that process would require additional development effort that seemed unnecessary in this early stage. This level of robustness was deemed out of scope for this pilot, but should we confirm the viability of delivering a container, this development would be a clear next step.

**VALIDATION**

The team took two steps to ensure the quality of this test submission package. First, the team validated the Shiny application by cross-checking visualization results against quick counts from the analysis data set. The Shiny application contains a very simple histogram of patient age distribution, and thus a quick cross-check was considered sufficient. Next, the team checked the visualization results and functionalities of Shiny tool between the versions from the docker image and original application residing inside the company server. With only minimal knowledge of the R language, one can run the app.R file used to build the Shiny app independently from any R session on the host machine. In doing so, one can confirm the app results assuming no breaking changes were introduced to any of the system R package dependencies since the versions used within the containers. Given the limited package footprint of this particular pilot, breaking changes would be unlikely, but are exactly the challenges the container approach hopes to alleviate.


**TRACE-ABILITY**

The source data used for the Shiny application is the same adsl.xpt file under \123456\0001\m5\datasets\study02\analysis\adam\datasets. By using the same submitted ADaM xpt datasets for the interactive tools, the team sees many benefits:

- The analyses can be easily traced back to the same submitted datasets to avoid potential discrepancies between the submitted static PDF TLG outputs and interactive visualization results from the Shiny tool, and therefore relieve any concerns from reviewers.
- It follows the eCTD process without adding extra files.
- The native eCTD structure is mapped to volumes within and between the containers such that minimal processing is required.

Furthermore, the code used to implement a shiny application is easily accessible and thus makes it much easier and convenient to follow the code that is used in the app through the applied analytic methods and back to the source datasets. Compared to visualization dashboarding suites which may require familiarity with a set of tools or even a local instance of deployment software, applications built in modern interactive frameworks like R / Shiny or Python / Dash are constructed within the same programming languages already familiar to analysts. Likewise, the containerized environment allows for clear articulation of software dependencies to enable traceability of the installation of the system.

**COMPUTATIONAL REPRODUCIBILITY**

The container technology contains not only the software, but also the dependencies including libraries, binaries and configuration files, and all together, they are migrated as a unit like the shipping container in the shipping industry. Containers also facilitate the deployment of software and its dependencies to a server. This enables the replication of the analyses and its analytical environment without running into any potential incompatibilities and crashes due to differences in the underlying operating systems.

In this test submission, the team used Docker as the container technology to specify a related group of contained services and delivered R studio as the analytic development portal and Shiny interactive application for access via web browser. The reviewers' guide provides step-by-step instructions on how to run the docker script files to install the Shiny tool and R studio on a host machine. The team ran testing successfully with the exact same analyses replicated as if one is accessing the original Shiny application. If the POC works out well, the container framework seems to offer a solution to the reproducibility considerations for clinical data analysis including interactive analysis.

**eCTD CONSIDERATIONS**

Although we strived to conform to existing standards for submission, the eCTD format does pose some challenges for delivery. Many conventions idiomatic to the R language need to be subverted for delivery of this content. For example, it is considered best practice to use a capital .R extension for R scripts. The typeface casing imposed by the eCTD guidelines requires that we instead use the .r extension. Although functional, this poses a barrier of separation between our work and what is broadly considered best practice.

Looking further downstream, we will likely want to eventually submit R packages that have been developed in-house. R package source code has a rigidly defined directory structure that is required in order to use the wealth of build tools provided with the R language. Where case sensitivity is inconvenient, but functional with a .r file extension, it renders an R package directory structure unuseable as it requires a source code directory labelled "R", and strictly cased "DESCRIPTION" and "NAMESPACE" files. Although a workaround using a script can restructure these malformed packages into a functional codebase, a preferred solution would be to modify eCTD standards.

**POC FEEDBACK FROM THE FDA**

The FDA staff was able to receive all files except the ReviewersGuide, which we need to further clarify whether it was a process related issue. Furthermore, during installation, FDA staff encountered a technical problem related to Docker implementation issues. By design, the initial pilot was made to install an R package library on the host machine outside of the container instance. This simplified development, but deviates from container design best practices. This created a security exception with FDA firewalls, and consequently created a roadblock for further progress of the pilot. We look forward to putting forth a second iteration of the pilot which will take into account these security issues. We plan to re-send the test submission within next few months by addressing the security concern as well as restructuring to support a more efficient workflow such as a shared R package library across multiple

interfaces including an analytical and visualization portal.

**SUMMARY**

Containerization is one of the emerging technologies that is being considered for regulatory environments. Containers provide a mechanism for reproducing a computing software stack from operating system to system binaries, installed software, analytic tools and user interfaces. This completeness makes containers an appealing technology for reproducible analyses and visualizations. Additionally, containers have the potential to facilitate the sharing of interactive data visualizations with health authorities. Despite these benefits, technical and process challenges may be anticipated when Docker technology is utilized to share data in regulatory environments.

Our goal is to facilitate and simplify workload for regulatory review. Container technology is used primarily by system administrators and technical developers with some technical expertise. Technical challenges may arise when attempting to deploy containers on a Windows operating system. We acknowledge that the environment may pose some challenges and issues for smooth deployment.

What process changes are needed to deploy Docker technologies in regulatory environments? This pilot offers a first step to help identify potential hurdles and the need for infrastructure changes to enable sharing of interactive data visualizations and results of analyses with health authorities. If the viability of delivering results in a container shows promise, one could envision supporting infrastructure that eases these process and technical barriers.

In addition, we would like to share what we have learned from our proof of concept project thus far.

1. Initial development was relatively uncomplicated in a Linux based environment. The required functionality is common and well documented.
2. Close collaboration with FDA staff is required for FDA deployment.
    a. Docker is sensitive to which Windows version and build is used.
    b. FDA firewall restrictions highlighted challenges with deploying containers.

For the sake of POC, the team decided to go with a very simple interactive application as the main purpose of this pilot is to assess the technical feasibility of Docker solution. When it comes to the real submission pilot, we need to consider the benefit of leveraging interactive tools for a comprehensive analysis.

## DISCUSSION

To move beyond the industry approach of sharing static reports with regulators and take advantage of interactive analysis requires a collaboration across multiple functions including IT and regulators. Many of the approaches also require a significant amount of investment. This is not merely a technological challenge. There are many nuances introduced by working within a regulatory environment and being constrained by a submission process that was designed for 'digitized paper'. It requires a concerted effort from the industry and regulators to modernize how we share information between sponsors and regulators.

There are other challenges facing interactive analysis in a regulatory setting, such as whether there is a need for common user interface for interactive tools to minimize the learning curve for reviewers, what changes are needed from both infrastructure and process perspectives in order to effectively facilitate data sharing between sponsors and regulators, etc..

## CONCLUSION

We believe that interactive analysis offers a potential to improve the efficiency and effectiveness of data sharing and data review between sponsors and health agencies. The POC projects just embark on a journey of our effort to explore the best way to leverage interactive visualizations as part of the regulatory review process. We need a regulatory environment that can take advantage of advances in technology to improve the drug development process.

A cloud based solution opens many possibilities for sharing interactive analyses. But this would require significant investment and coordination across the industry and regulatory bodies and changes to our current submission process. Think big and start small. We can put together a guidance to streamline the use of interactive analysis from exploratory to decision-making in submissions within our current regulatory environment. But we will not be able to leverage new technologies or even current technologies fully given the current constraints by a submission process that is designed to share 'digitized paper'. Therefore, we would like to call for more participation from industry and regulatory bodies to modernize the submission process, including the standards around eCTD. We welcome


collaboration opportunities to enable us to achieve our mission of modernizing the submission process to facilitate interactive analyses for decision making. We hope that one day a real-time review of clinical data submission enabled by interactive visualization tools will become a reality.

## REFERENCES

1. Sue Sutter, Reliance on Digitized Paper is Slowing Drug Development – US FDA's Woodcock https://pink.pharmaintelligence.informa.com/PS124255/Reliance-On-Digitized-Paper-Is-Slowing-Drug-Development--US-FDAs-Woodcock
2. PHUSE original proposal: https://PHUSE.teamworkpm.net/#files/2189064

## ACKNOWLEDGMENTS

We would like to thank to all team members under the PHUSE project "Best Practice for Interactive Analysis for Decision-Making and Submission" for their active contributions. Special thanks go to

- FDA representatives: Paul Schuette and Alan Shapiro
- Eli Lilly POC team: Mary Nilsson, Zachary Skrivanek
- Genentech POC team: Nilesh Narayan, Doug Kelkhoff, Xiangyun Wang
- Industry representatives: Raj kiran Boddu (Takeda), Phil Bowsher (Rstudio), Karl Brand (Bayer), Maria Dalton (GSK), Josephine Fong (Genentech), Eric S. Herbel (I-Review), James Kim (Pfizer), Mary E Nilsson (Eli Lilly), Catherine Xia (Gilead)

## CONTACT INFORMATION

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

Xiangyun (Sharon) Wang
Genentech, a member of Roche Group
South San Francisco, CA, USA
wang.xiangyun@gene.com

Zachary Skrivanek
Eli Lilly, Indianapolis, USA
skrivanek_zachary@lilly.com