Paper OS13

Unleashing the Power of {teal}: New Features and Cross-Industry Collaboration

Nina Qi, Roche/Genentech, South San Francisco, USA
Dony Unardi, Roche/Genentech, South San Francisco, USA

ABSTRACT

Since we announced the release of {teal} on CRAN, the framework has garnered tremendous support and enthusiasm from the pharma community. We've seen a growing interest from both individuals and companies who are exploring {teal} to streamline and enhance their clinical data exploration. Following our previous presentations at PHUSE US Connect in the past 2 years, we are excited to present the latest advancements that further enhance the framework's usability and scalability, including, but not limited to, introduction to {teal_data_transform} modules for module data manipulation, {teal} output decoration to customize teal module's output, using {teal} as a module, our new UI/UX PoC, and many more. This year, we will also cover our take in leading the open-sourcing project, along with sharing our experiences in cross-industry collaboration.

INTRODUCTION

The need for innovative and efficient solutions for exploring clinical trial data has grown significantly in recent years. Traditional analytical methods, which often involve time-consuming and repetitive tasks, are becoming increasingly impractical for data scientists. {teal}, as an open source R-shiny based framework that enables interactive data analysis of clinical trial data, has been recognized by the pharma industry for its user-friendly interface and powerful data analysis capabilities. Since its official release on CRAN in February 2024, {teal} has received tremendous support and enthusiasm from data scientists in the pharma industry, and has continued to evolve with enhancements driven by user feedback and industry collaboration. In this paper, we will introduce the latest features of {teal}, share our experiences with open-source development, and highlight the cross-industry collaboration efforts that have shaped the framework's evolution and its future direction.

A RECAP OF THE `teal` FRAMEWORK

DESIGN

The teal framework offers a modular approach to R-shiny app development, making it highly flexible, datatype agnostic, and enabling collaboration and crowd-sourcing as an open-source software. For users, the teal workflow starts with gathering and preprocessing data into an analysis-ready shape. It automatically embeds key mechanisms like dynamic data filtering, reproducible code generation, and analysis results reporting engine.

ANATOMY

A typical teal app consists of distinct areas such as the App Header, which shows the title of the web app, and Teal Modules, which offer each statistical analysis, data visualization, and/or tabular summarization as separate components.

TEAL-VERSE PACKAGES

Teal is part of a broader ecosystem of packages known as the teal-verse, which enhances the capabilities of teal through various supporting packages. This allows developers to extend and customize their teal apps even further, fostering a richer and more versatile user experience.

Please refer to our previous presentation for a more comprehensive overview of teal and its functionalities.

NEW DEVELOPMENTS IN `teal`

Since our last presentation at PHUSE 2024, several significant updates have been introduced:

NEW CRAN RELEASE

{teal} v0.16.0 is on CRAN now! We've recently wrapped up the release of teal framework packages, together with module packages `teal.modules.clinical` and `teal.modules.general`. This will ensure a stable version of {teal} before future major updates, such as UI/UX improvements.

TEAL TRANSFORMATOR

`teal transformators` is a newly introduced feature that enhances data transformation capabilities before passing it to a module. It is implemented as a Shiny module, allowing seamless integration into `teal` applications. This feature enables users to modify datasets dynamically, ensuring they are processed appropriately before analysis. A new optional argument, `transformators`, has been introduced in `teal::module`, allowing users to pass a list of objects of the `teal_data_module` class. These objects, created using `teal_transform_module()`, facilitate efficient data transformations without modifying the core module functionality. By incorporating `teal_transform_module()`, users gain greater control over data preparation, ensuring adaptability to a variety of analytical needs while maintaining a streamlined workflow within the `teal` framework.

When transformators are supplied, `teal` applies transformations when a module is loaded and whenever the original dataset is updated. The transformed data is then used in subsequent analyses, ensuring that modifications occur in a controlled and reproducible manner. The `ui` elements of the transform module are automatically added to the filter panel, while the `server` function contains the logic for data manipulation.

For example, a simple transformation module can be created to allow users to filter the first *n* rows of the `iris` dataset, as illustrated in Figure 1. The UI provides a numeric input field where users specify the number of rows to retain, and the server function dynamically applies the subset. Multiple transformation modules can be combined within a single {teal} app, allowing various datasets to be processed in different ways before reaching analysis modules. This structured and interactive approach to data preprocessing improves flexibility and enhances the overall usability of {teal} applications.

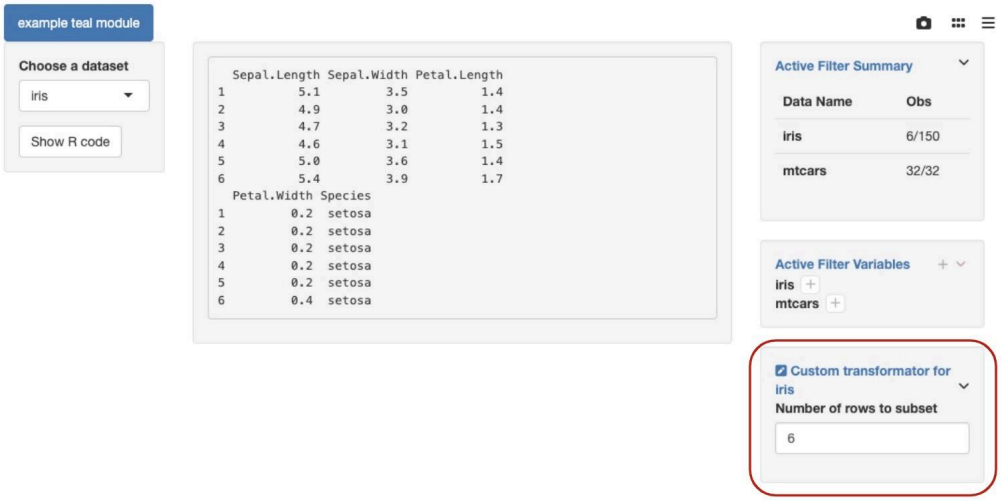

Figure 1 a teal transformator that performs data subsetting

TEAL DECORATOR

The Output Decorator feature enhances the flexibility of {teal} applications by allowing users to modify or customize module outputs without generating a new module from scratch. This feature is particularly useful for adapting existing visualization and analysis modules based on user-specific needs. App developers can choose between non-interactive decorators which apply predefined modifications (Figure 2.1, where user adds title and x-axis label to the original plot from a module), and interactive decorators which introduce user input elements for dynamic customization (Figure 2.2, where user can add x-axis label in the app).

The `teal_transform_module()` function (same function for teal data transformation) can be leveraged to create decorators seamlessly, allowing efficient integration within the {teal} framework. Additionally, the utility function

`make_teal_transform_server()` simplifies the process of creating decorators by automating key aspects of the server logic. This new feature enhances output customization, supporting cross-industry adoption and increasing the reusability of {teal} modules for interactive data exploration and analysis.

Figure 2.1 a teal decorator with no UI

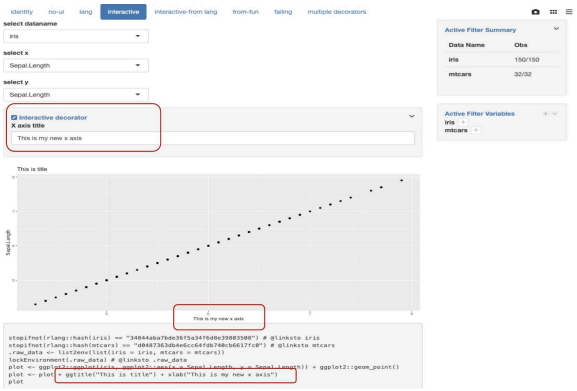

Figure 2.2 a teal decorator with an interactive UI

TEAL AS A MODULE

A new enhancement allows developers to embed {teal} as a Shiny module within other Shiny applications. By calling `ui_teal()` and `srv_teal()`directly, {teal} can be seamlessly integrated into custom Shiny dashboards, enabling greater flexibility. This approach differs from using `init()`, as it allows developers to embed multiple {teal} instances, dynamically generate components such as datasets and modules, and control session behavior more explicitly. Additionally, developers can manually include the session info footer with `ui_session_info()` and `srv_session_info()` if needed.

For example, as shown in Figure 3, a {teal} module can be embedded within a parent Shiny app, where users select datasets dynamically, and these selections are passed as reactive inputs to `srv_teal()`. This method enhances interoperability, making it easier to integrate {teal} into existing applications or combine multiple analysis workflows. By leveraging {teal} as a modular component, other organizations can also integrate teal with their clinical data analysis applications, adding enhanced scalability and customization.

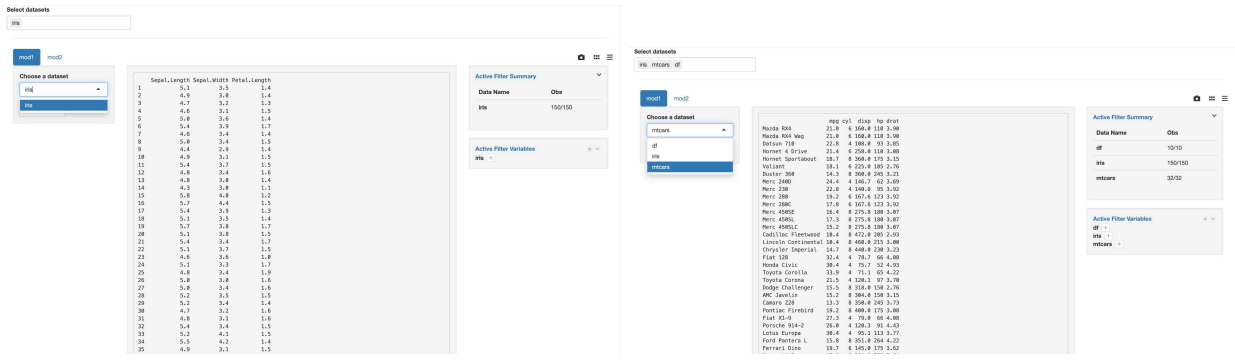

Figure 3 teal used as a Shiny module

IMPROVED DOCUMENTATION AND NEW VIDEO MATERIALS

To further support users and developers, we have significantly improved the documentation and released new video tutorials covering key {teal} features. New documentation has been added for teal transformator, teal decorator, and

using {teal} as a Shiny module, offering step-by-step guidance on these functionalities. Additionally, we have enhanced existing documentation on creating customized {teal} modules and integrating `teal_data` into applications. These improvements aim to make {teal} more accessible and easier to adopt across different user groups, facilitating a smoother learning curve and wider adoption within the industry.

A new video tutorial is available to help users get started with {teal} and explore its latest features. This video is structured to cater to three primary user groups: Teal Users, Teal App Developers, and Teal Module Developers (Figure 4). This video provides a practical walkthrough of the framework's functionalities, demonstrating how to leverage {teal} for efficient clinical data analysis for all user groups.

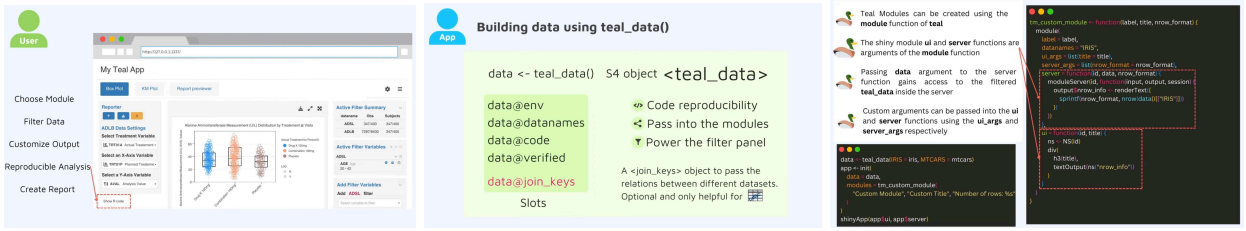

Figure 4 Video: A Complete Guide to Getting Started with teal

Proof Of Concept - NEW UI/UX UPDATES

With the addition of teal transformators, the filter panel may become overcrowded, prompting a review of the UI/UX design of {teal}. As part of our ongoing efforts to enhance usability, we have been developing a UI/UX proof of concept that offers a more modern and intuitive user experience (Figure 5). This redesign focuses on improving the logical structure of the application workflow, streamlining the process from data input to output visualization. One of the key aspects of this concept is relocating the sidebar to the left by default, reinforcing a structured left-to-right data flow—starting with data filtering and transformation, moving to module-specific configurations, and culminating in the final analysis outputs, with the aim to provide users with a clearer understanding of data progression within the application. App developers will still have the flexibility to make changes if this is not preferred. Additionally, we are consolidating the filter panel by merging "Active filter variables" and "Add filter variables" into a single interface where users can dynamically add filters via a "+" button.

In order to ensure usability and scalability, we have been actively gathering feedback on key UI preferences, including filter panel placement, sidebar behavior, and feedback on placement of global features like the data module and teal reporter, where these insights will guide the final design and implementation.

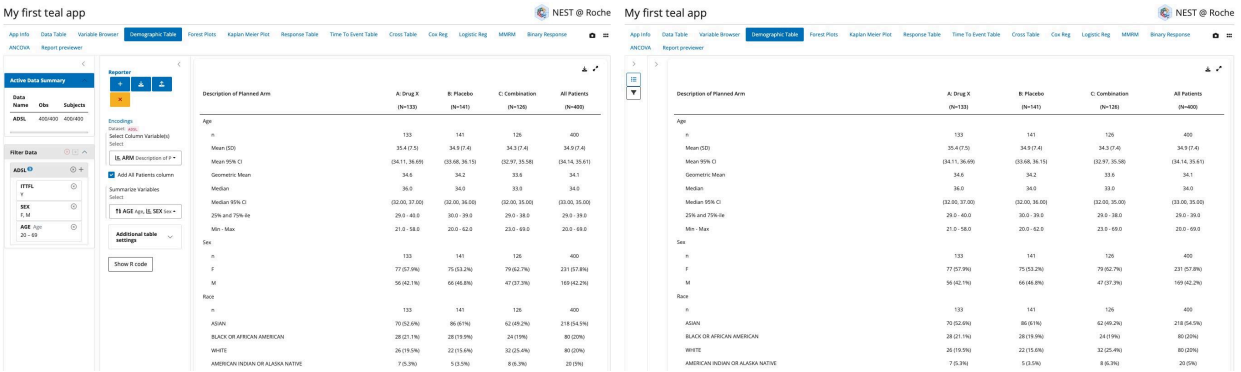

Figure 5 PoC for new teal UI/UX

POC - INTERACTIVE REVIEWS ON INDIVIDUAL DATA

In the medical data review process, patient-level data is often critically reviewed to ensure safety and efficacy in clinical trials. To address the need for a more flexible framework that allows for in-depth examination of such data, we have developed a Proof of Concept (PoC) utilizing the teal framework. This PoC aims to extend the capabilities of teal, making it a versatile tool for drilling down into detailed patient-level data. One of the key enhancements in this PoC is the incorporation of user interactivity components, such as Plotly, where app users will be able to select data points of interest and take a deep-dive of relevant safety and efficacy data points, which provided a more engaging and informative exploration experience for app users.

We began this initiative by developing specific modules targeting the early review phases of safety and efficacy data. These modules are designed to facilitate the review process by allowing precise and interactive data filtering and visualization at the patient level, thereby enabling a more granular and insightful analysis. By leveraging these enhancements, our goal is to empower data scientists and medical reviewers with more customizable and interactive tools, ultimately facilitating insights generation and decision-making in clinical research.

OPEN SOURCE EXPERIENCE

Open sourcing {teal} has been a transformative journey, providing both challenges and immense opportunities. Rather than viewing it as a hurdle, we recognize it as an opportunity to drive collaboration, transparency, and innovation across the industry. By making {teal} open source, we have created an environment that fosters cross-company collaboration, allowing contributors from different organizations to improve the framework collectively. This openness helps identify gaps for wider adoption, facilitates collaboration, and ultimately with the aim of reducing duplication of efforts by sharing resources and solutions. Additionally, open source development brings increased transparency, offering insights into product development progress, maintaining high code quality, and enabling continuous learning and improvement within the community.

The process of open sourcing {teal} required careful evaluation and preparation to ensure a smooth journey. This involved key considerations such as licensing compliance to meet legal requirements, enhancing documentation for both developers and users, and refining code quality and structure to align with industry best practices. A robust pipeline was also established to support continuous integration of new features and improvements. Beyond technical aspects, a major focus has been on community engagement and long-term maintenance, as fostering an active and sustainable contributor base is critical for the success of any open-source project. By embracing open source, {teal} is not just a tool but a collective effort, shaping the future of clinical data analysis through shared innovation and expertise.

CROSS-INDUSTRY COLLABORATION

teal is being used by a variety of organizations across the pharmaceutical industry, including pharmaceutical companies, CROs, as well being included as supplementary in pilot submission studies. The open-sourcing of {teal} has led to successful collaborations across the pharmaceutical industry, with multiple companies benefiting from and contributing to its development in various ways. These partnerships have driven innovation, improved usability, and expanded the framework's capabilities.

One of our notable collaborations has been with **Johnson & Johnson (JnJ)**. JnJ requested the feature for output decoration, and further contributed to the development of **teal decorator**, improving how users can modify and extend module outputs dynamically. Beside code collaboration on Github, their contributions have also helped tremendously on refining this functionality, ensuring a seamless experience for users with similar need.

Another significant collaboration has been with **Sanofi**, who has developed on **`teal.builder`** - a package that streamlines the process of constructing {teal} applications. This great effort significantly lowered the learning curve, enabling app developers to create tailored applications more efficiently without starting from scratch, ultimately streamlining the development workflow.

Additionally, **Boehringer Ingelheim (BI)** has developed **`teal.modules.bsafe`**, a R Shiny app designed for statistical analysis of adverse event summary data with consultation from the {teal} core development team. B-SAFE

is built with the {teal} framework, and it enhances descriptive safety analyses by incorporating historical data across one or more treatment arms, improving precision in safety assessments for ongoing trials.

Beyond these specific collaborations, other pharmaceutical companies have also actively engaged in {teal}'s adoption and development, reinforcing the value of cross-industry collaboration. The continued success of {teal} is driven by an active and growing community across the pharmaceutical industry. To foster collaboration, we have established multiple channels where industry professionals can share insights and experience, contribute, and shape the future of {teal} together. One major initiative is the PHUSE Working Group on {teal} Enhancements, which aims to drive cross-industry adoption by identifying common needs and developing new features collaboratively. Additionally, {teal} will be a major topic in the PHUSE Computational Science Symposium (CSS) - DVOST stream in 2025, where we hope to collectively share our experiences and insights. Beyond PHUSE, the Teal Community Meeting serves as a regular forum where contributors and users can share updates, provide feedback, and discuss upcoming developments. The #teal Slack Channel under pharmaverse workspace further supports real-time communication, allowing developers and users to collaborate, troubleshoot, and exchange ideas efficiently. These engagement channels play a vital role in ensuring that {teal} remains a truly collaborative and community-driven framework, reflecting the needs of the broader industry while fostering transparency, innovation, and shared learning.

CONCLUSION

The introduction of new features such as data transformators, output decorators, using teal as a module, etc., has significantly enhanced usability and scalability, as well as adaptability across different workflows. The evolution of {teal} from an internal tool to an open-source framework has unlocked new opportunities for collaboration and innovation, where collective expertise can lead to continuous improvements. By fostering partnerships with other companies across the industry, {teal} has further expanded into a flexible and scalable solution for interactive clinical data analysis. We welcome industry partners and contributors to join us and help expand {teal}'s capabilities and collectively move towards more efficient and insightful clinical data analysis.

ACKNOWLEDGMENTS

We would like to thank all past and present developers that contributed to any of the `teal` framework R packages. We would like to thank all collaborators and community members across the industry.

RECOMMENDED READING

- PHUSE US Connect 2024 Presentation: DV04 - Revolutionize Clinical Trial Data Exploration with teal, presentation material and paper
- `teal` video: A Complete Guide to Getting Started with teal
- `teal` web manual: https://insightsengineering.github.io/teal
- `teal` Github repository: https://github.com/insightsengineering/teal
- `teal.gallery` demo apps: https://insightsengineering.github.io/teal.gallery/
- NEST R product family website: https://insightsengineering.github.io/nest/

CONTACT INFORMATION

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

Author Name: Nina Qi
Company: Roche/Genentech
Email: qi.ting@gene.com

Co-Author Name: Dony Unardi
Company: Roche/Genentech
Email: unardi.dony@gene.com