PharmaSUG 2026 - Paper 128

Unleashing Open-Source Potential in SAS: The PharmaForest Ecosystem

(proc pharmaforest data=open_source out=… ;)

Sharad Chhetri, Takeda Pharmaceuticals, Cambridge, MA
Ryo Nakaya, Takeda Pharmaceuticals, Osaka, Japan

ABSTRACT

The recent surge in the use of R and open-source software in drug development has sparked a powerful movement focused on shared knowledge, transparency, flexibility, and community-driven innovation. Notably, this shift has produced an unexpected—but positive—side effect: it has inspired many long-time SAS users to adopt the same spirit of openness and community sharing of SAS codes which they have developed.

Increasingly, SAS professionals are recognizing the strength of collective knowledge and the value of contributing their own work as open-source code. Emerging from this momentum is PharmaForest—a unified repository of SAS packages built upon the SAS Packages Framework (SPF) (https://github.com/yabwon/SAS_PACKAGES)[1]. This initiative is dedicated to accelerating open collaboration among SAS users and cultivating a vibrant, sustainable community where ideas, code, and best practices can flourish together—like a forest ecosystem.

This presentation will introduce PharmaForest, explore how it bridges the traditional SAS ecosystem with the open-source movement, and highlight its transformative role in the future of collaborative pharmaceutical programming.

INTRODUCTION

Over the past years, the pharmaceutical industry has witnessed a rapid and widespread adoption of R and open-source tools, fundamentally reshaping how clinical data are analyzed, validated, and shared. In particular, the growth of community-driven initiatives such as the pharmaverse (https://pharmaverse.org/) has demonstrated how open collaboration can dramatically improve productivity, transparency, and methodological consistency across organizations. By providing shared standards, reusable code, and openly discussed best practices, these communities have lowered barriers to innovation which often happens with open and inclusive discussions across organizations and reduced duplication of effort that has traditionally characterized drug development workflows in each company.

Several key drivers underpin this transformation. First, potential efficiency has become a critical priority, as development timelines are tightened caused by lower success rate of drug development and the cost of repeated programming efforts and validation activities grows increasingly unsustainable. Second, collaboration—both within and across companies—has proven essential for tackling complex regulatory and analytical challenges such as e-Data submission in CDISC format and implementing the latest statistical methodologies. Finally, the open-source model has accelerated innovation, enabling rapid iteration, peer review, and continuous improvement that would be difficult to achieve in isolated, proprietary environments.

This cultural shift has not remained closed within the R ecosystem. Its influence is now clearly reaching long-established SAS users, producing two notable responses. One group has begun actively exploring R itself in clinical programming to move from proprietary software SAS, motivated by the benefits of open-tooling and community support. Another group, however, has taken another path: rather than replacing SAS entirely, they are blending the open-source mindset with the SAS environment, embracing

[1] Bartosz Jabłonski, "SAS Packages: The Way to Share (a How To)", SAS Global Forum 2020 Proceedings, 4725-2020

transparency, reuse, and community contribution while aiming to maximize potential of SAS and R in a hybrid way.

PharmaForest was created to turn this growing momentum into a unified and sustainable framework for SAS users. Built on the SAS Packages Framework (SPF), PharmaForest brings open-source ideas—such as shared ownership, modular design, and collaborative development—into the traditional SAS environment. Its goal is to reduce duplicated work, share knowledge more effectively, and build an active community where SAS-based pharmaceutical programming can continue to grow alongside the broader open-source ecosystem.

PHARMAFOREST

PharmaForest is a collaborative open-source initiative focused on expanding and maintaining a repository of SAS packages (https://github.com/PharmaForest) for the pharmaceutical industry. The initiative was launched in 2025 to encourage a more open-source mindset in the SAS user community by sharing as many SAS packages as possible and growing its collaborator base over time. It operates as a working group within PHUSE Data Visualization & Open Source Technology and serves as a community-driven platform for sharing SAS-based tools and best practices. As of January 2026, PharmaForest hosts 44 packages spanning a broad range of pharmaceutical programming use cases and is growing rapidly.

The objectives of the PharmaForest initiative are clearly outlined on its GitHub site. First, it aims to promote the sharing of SAS expertise that has traditionally remained siloed within individual programmers and organizations. Second, it seeks to consolidate this dispersed know-how into a unified knowledge hub—improving productivity, supporting quality through standardization, and strengthening engagement across the SAS pharmaceutical programming community.

These objectives clearly express the priorities of the initiative, placing primary emphasis on open-source sharing. Thus, lowering barriers to participation and broadening the contributor base. Building on this open foundation, the initiative then focuses on improving code quality, advancing standardization, and creating long-term, sustainable value for the community.

Recognizing the SAS Packages Framework (SPF) as the foundation of PharmaForest, this paper briefly touches on SPF, provides an overview of the packages currently available in the PharmaForest repository, and discusses growth to date as well as expectations for future development.

SAS PACKAGES FRAMEWORK (SPF)

The SAS Packages Framework (SPF) is a collection of core SAS macros developed and maintained under the MIT license (https://github.com/yabwon/SAS_PACKAGES). SPF provides a foundation for a new modular component system in SAS, similar in spirit to the package frameworks found in R and Python. SPF enables the creation, distribution, installation, and use of self-contained SAS packages, thereby facilitating modular, reusable, and shareable SAS code in a systematic and standardized way.

Since its initial release in 2019, SPF has matured and expanded to include macros supporting critical aspects of package lifecycle management. The suite currently offers the following:

- Generating package bundles.
- Installing packages to computing environment and loading into a SAS session.
- Validating and organizing collections of packages for reproducible environments.

Core macros of SPF as of January 2026 are listed below.

Name	Description
%bundlePackages()	Bundle a bunch of SAS packages into a single file (a SAS packages bundle), just like a snapshot.
%extendPackagesFileref()	Extends the existing packages fileref by adding additional directories, enabling management of multiple package locations.
%generatePackage()	The core developer macro that generates a standalone SAS package ZIP file based on the prepared folder structure, description file, and package contents.
%helpPackage()	Displays help information about a package and its components (macros, functions, datasets, etc.) in the SAS log.
%installPackage()	Downloads and installs a SAS package ZIP file from a repository (e.g., SASPAC) into the packages directory.
%isPackagesFilerefOK()	Checks whether the packages fileref is properly assigned and configured.
%listPackages()	Lists all available SAS packages found in the directories assigned to the packages fileref.
%loadPackage()	Loads a specified SAS package into the current SAS session. Handles dependencies, lazy data loading, version checking, and selective (cherry-pick) loading.
%loadPackageS()	A wrapper macro that allows multiple packages to be loaded in a single execution (extension of %loadPackage).
%loadPackageAddCnt()	Extracts and loads additional content (such as PDFs or external files) included in a package.
%previewPackage()	Displays the internal SAS code of a package directly in the SAS log without manually unzipping the file. Useful for code inspection.
%relocatePackage()	Moves a package ZIP file to a different packages directory location.
%unbundlePackages()	Separates bundled packages into individual package files.
%sasPackagesFrameworkNotes()	Prints notes and additional information about the SAS Packages Framework.
%splitCodeForPackage()	Assists developers by splitting a single SAS program into multiple files and directories.
%unloadPackage()	Removes a loaded package and its components from the current SAS session.
%verifyPackage()	Verifies the integrity of a package by checking its SHA256 hash to ensure it has not been modified.

The addition of %bundlePackages() and %unbundlePackages() in the 2025-12-31 release (https://github.com/yabwon/SAS_PACKAGES/releases/tag/20251231) expands the framework's utility by enabling users to group multiple packages into a single bundle for distribution and reproducible deployment in team or shared environments.

PHARMAFOREST PACKAGES

As of January 2026, PharmaForest hosts 44 SAS packages (https://github.com/PharmaForest). These packages consist of both original contributions developed and maintained within the PharmaForest initiative and donated packages contributed by individual developers and organizations. Together, they form a shared repository of reusable SAS components for pharmaceutical programming.

The packages are organized into four functional categories: Output/Visualization, Data Utility, Checker, and Other. This categorization is maintained and publicly available on PharmaForest's GitHub repository and the gallery page, allowing users to easily explore available tools by use case (https://pharmaforest.github.io/).

PACKAGE CATEGORIES

- **Output/Visualization** category includes packages designed to support common reporting and visualization needs in clinical development. Examples include *OncoPlotter*, which generates standard oncology-specific visualizations; *rtfcreator*, which supports the creation of tables and listings in RTF format; and *trial_flow*, which produces clinical trial flow diagrams in accordance with ICH E3 guidelines.
- **Data Utility** category contains a wide range of packages aimed at improving efficiency in data handling and transformation. Representative examples include *Sashash*, which simplifies the use of SAS hash objects; *sas_dataset_json*, which converts CDISC datasets into JSON format; and *adamski*, which supports the creation of CDISC ADaM datasets.
- **Checker** category focuses on quality control and validation. Packages in this group include *sas_logchecker* for automated SAS log review, *sas_compare* for dataset comparison, and *valivali* for validating SAS packages and ensuring compliance with defined standards.
- **Other** category includes packages that serve a variety of supporting roles, such as facilitating the development of SAS packages, sharing practical guidance and maxims for SAS programmers, and providing miscellaneous utility functions that do not fall into the other categories.

PharmaForest packages are distributed and consumed using the SAS Packages Framework (SPF). A typical workflow for installing and loading a package named `xxx` is shown below:

```
/* Setup for installing and loading SAS package */

%installPackage(xxx, mirror=pharmaforest)

%loadPackage(xxx)
```

GROWTH AND EXPECTATIONS

Since its initiation in June 2025, PharmaForest has experienced rapid growth in the number of available packages within a relatively short period. This early expansion was driven by a deliberate strategy to increase visibility and awareness of the repository by prioritizing the accumulation of packages. In practice, the availability of more than 40 packages within approximately six months successfully attracted attention from the broader SAS community. As a result, an initiative that initially began as a locally organized activity within a **PHUSE working group in Japan has evolved into a globally oriented effort.

At the same time, this rapid growth highlights the need to consider subsequent development phases sooner rather than later. One of the most important challenges to be addressed is **package validation**. In the R ecosystem, initiatives such as **pharmaverse** provide packages with established validation processes aligned with **CRAN** requirements, enabling their use in regulatory submissions subject to inspection. However, the cultural and historical foundations of package usage in R and SAS differ substantially. As of 2026, more than 20,000 packages are registered on CRAN, and the concept of

reusable packages is widely recognized and embedded in the R community. In contrast, SAS does not yet have a comparably comprehensive or widely adopted package ecosystem.

Although the **SAS Packages Archive (SASPAC)** (https://github.com/SASPAC) provides a central repository, broader adoption of SAS packages—particularly within the pharmaceutical industry—remains an ongoing challenge. In this context, PharmaForest adopts a dual strategy. On one hand, it aims to provide high-quality, potentially regulatory-compliant packages that can be validated and inspected. On the other hand, it intentionally promotes the publication of a wide range of packages to lower barriers to participation and to encourage broader adoption of the SAS package concept itself. This emphasis on expanding engagement distinguishes PharmaForest from pharmaverse in R, where validation and regulatory readiness are primary entry criteria from the outset. Nevertheless, ensuring quality through systematic validation remains a critical objective for future phases.

Another important next step involves **organizational participation**. To date, PharmaForest has primarily attracted interest and contributions from individual developers, while company-level involvement has not yet been formally established. Encouraging organizational participation will require clearly defined scopes of sharing, as well as agreed-upon best practices for contribution, review, and maintenance that align with corporate governance and compliance requirements. It is important to distinguish between proprietary and open-source tools within organizations. A clear strategy is needed to balance both, and PharmaForest serves as a platform for sharing open-source tools across organizations.

In summary, PharmaForest is nearing completion of its first development phase, which focused on rapidly increasing the number of available packages to establish visibility and community awareness. The next phase will center on defining governance structures and implementing collaborative validation processes to support quality and sustainability. Through these stepwise milestones, PharmaForest aims to evolve into a trusted hub that not only provides a broad range of practical SAS tools but also fosters collaboration and contributes to industry-wide efficiency in clinical programming.

DIVERSE PROGRAMMING ENVIRONMENT

Once these initial goals are achieved, the next level of growth will increasingly depend on **organizational decisions regarding computational environments**. Historically, SAS has served as the central programming platform for clinical data processing and reporting over several decades. More recently, R has emerged as a viable alternative, raising the possibility that SAS could be partially or fully replaced by R for clinical programming in some organizations. At the same time, other organizations may choose to continue relying exclusively on SAS, while still others may adopt a **hybrid model** that combines both SAS and R.

In practice, organizations may select from several viable approaches:

- exclusive use of R
- exclusive use of SAS
- combined use of SAS and R
- other

The choice of computational environment is ultimately an organizational decision driven by internal strategy, regulatory considerations, and existing infrastructure. Within this landscape, the role of **PharmaForest** is not to prescribe a single technological path, but rather to **expand the available options**, particularly for organizations that continue to use SAS or adopt hybrid SAS–R models. By introducing open-source principles into the SAS ecosystem, PharmaForest enables organizations to benefit from practices that are already well established in open-source communities.

A key contribution of the open-source mindset is improved efficiency through the reduction of repeated efforts, achieved by sharing techniques, code, and best practices across organizational boundaries. This efficiency gain is especially critical under current industry conditions, where development timelines are tightening and resources are increasingly constrained.

At the same time, Python has become widely adopted for general-purpose programming, particularly in the areas of artificial intelligence and machine learning. As a result, many organizations now operate multi-language environments in which Python is used for AI/ML applications, while SAS and/or R are used for clinical programming. Viewed in this broader context, the coexistence of SAS and R within clinical programming workflows is both realistic and sustainable, provided that clear rationale and complementary roles are defined.

From the perspective of end users, the availability of multiple programming languages represents a significant advantage. Rather than being constrained to a single technology, programmers can select the most appropriate tool for a given task, thereby improving productivity, flexibility, and long-term adaptability. By supporting this plurality of approaches, PharmaForest contributes to a more resilient and efficient clinical programming ecosystem.

CONCLUSION

In conclusion, the rise of open-source collaboration has not only transformed how R users build and share tools but has also begun to reshape the SAS community itself. PharmaForest stands as a symbolic bridge between tradition and innovation – bringing open, community-driven principles into the long-established SAS ecosystem. It is acting as a catalyst for the next generation of pharmaceutical programming where code transparency can assist in quality improvement, reusability and flexibility.

By embracing transparency, reusability, and shared growth, SAS professionals are redefining what collaboration means in regulated environments. The journey has only just begun, but the collective roots of PharmaForest continue to grow—nurturing a future where R, SAS, and open source coexist, complement, and evolve together for the betterment of the entire pharmaceutical analytics ecosystem.

As a SAS user, we need to take a careful look at how we use PharmaForest procedure based on source data of open-source mind for further clarified output.

proc pharmaforest data=open_source out=community asset;

run;

ACKNOWLEDGMENTS

The authors wish to express sincere gratitude to Nori Oharu and Yohei Takanami for their guidance, valuable review and insightful feedback on this paper.

REFERENCES

Bartosz Jabłonski. 2020. ”SAS Packages: The Way to Share (a How To)”, *SAS Global Forum 2020 Proceedings*, 4725-2020

Pharmaverse. “pharmaverse” Accessed February 15, 2026. https://pharmaverse.org

Bartosz Jabłonski. “SASPAC GitHub repository”. Accessed February 15, 2026. https://github.com/SASPAC

Katja Glass. “OPEN SOURCE PORTAL”. Accessed February 15, 2026. www.glacon.eu/portal

Open source guides community. “Open Source Guides” Accessed February 15, 2026. https://opensource.guide/

CONTACT INFORMATION

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

Sharad Chhetri
Takeda Pharmaceuticals
sharad.chhetri@takeda.com

Ryo Nakaya
Takeda Pharmaceuticals
ryou.nakaya@takeda.com