Paper OS12

From Frozen to Fluid: How the {slushy} R Package Meets the Unique Needs of Study Teams

Becca Krouse, GSK

ABSTRACT

In the pharma industry, stable and secure R environments are crucial for our work. The ideal solution has often been the use of locked down, "frozen" environments, which are carefully curated and tested by tech teams. However, the fast-paced nature of R development means these environments can become outdated quickly, preventing teams from utilizing the latest advances. Enter {slushy}: a team-friendly R package powered by {renv} and Posit Package Manager. {slushy} was designed to quickly mimic a future controlled environment, with the easy ability to time travel between CRAN snapshots. However {slushy} offers more than just a glimpse into the future. It caters to the unique needs of studies by empowering teams to define and assess the quality of their own custom R environments. Attendees will discover how {slushy} exceeded our expectations by proving its value in a variety of scenarios at GSK, in turn bolstering our R adoption efforts.

INTRODUCTION

In the pharmaceutical industry, where traceability is of utmost importance, maintaining stable and secure R programming environments is a critical component of success. Traditionally, organizations have relied on central "frozen" environments for production programming. These environments are meticulously curated and tested by central governance and tech teams, ensuring broad applicability and reliability. However, these carefully maintained environments quickly become outdated due to the rapid pace of R development, hindering the ability of teams to harness the latest tools and innovations. This paper will describe how {slushy} enables teams to anticipate and prepare for future frozen environments. It will also explore the potential of {slushy} in empowering teams that desire greater control over their environments.

{SLUSHY} AS A BRIDGE TO THE FUTURE

To support teams eager to leverage the latest R functionalities, we created {slushy}, an R package that allows teams to quickly mimic a controlled environment. Specifically, {slushy} supports the following:

- Alignment of all team members to the same set of "approved" packages
- Consistent use of the same package versions by all team members
- Ensuring compatibility among packages by fixing their versions to a specific date in time ("CRAN snapshot")
- Monitoring the use of non-approved packages

The {slushy} package was initially designed to support teams between releases of frozen environments, allowing users to effectively program towards the future release. By leveraging {renv} as its engine, {slushy} facilitates reproducibility and consistency among team members. Additionally, integration with Posit Package Manager's CRAN snapshots ensures package stability and the ease of transitioning to newer tools over time. {slushy} provides additional support to keep all team members aligned with an agreed-upon set of packages.

In practice, a team could request a new package for the next frozen environment and start using it immediately. In the short term, {slushy} keeps the team aligned with the future frozen environment, ensuring a smooth transition when the new environment is released. This flexibility has been crucial in meeting the diverse needs of our studies and allowing tech and study teams to work in parallel towards a common future. We discussed benefits and the adoption of {slushy} in a recent Posit webinar: GSK's R Journey: From Pilot Projects to Enterprise Adoption.

More information about the {slushy} package can be found on its website and in a recorded presentation from the 2023 Posit Conference: {slushy}: A Bridge to the Future.

PACKAGE REQUEST PILE-UP

For organizations with centrally deployed frozen environments, central governance teams may need to prioritize incoming package requests. This prioritization may be based on the broad applicability of a package or anticipated risk level. As R gains traction, the volume of package requests is likely to increase. Some teams might require packages for unique needs that are not widely applicable and, therefore, could be deprioritized for future frozen environments. This prioritization can conflict with the study team's requirements, especially when deadlines are approaching. In these cases, {slushy} remains a valuable tool for providing stability and reliability within an open R environment, although there is less certainty about whether a future frozen environment will meet their needs and deadlines.

COULD THERE BE ANOTHER WAY?

While organizations address the scaling of frozen environments, the external community is establishing best practices in establishing trust and traceability in R. The R Validation Hub has published numerous resources, including a white paper on approaches to package assessment and system qualification. Further, the R Submissions Working Group seeks to determine the acceptable requirements for R-based submissions, with {renv} as a core component for R package management. The lock file produced by {renv} allows reviewers to easily reproduce the R environment used for code development.

These community-driven efforts are critical not only for addressing current challenges but also for illuminating a path forward. This collective knowledge demystifies the intricacies of R-based submissions, providing a clearer picture of compliance and quality standards. With increased comfort and clarity around requirements, there is an opportunity to rethink what is possible when it comes to R environments. Imagine a scenario where the trust of a centralized environment is achieved in a more custom, individualized setting. In such a situation, teams could define their own environments to suit their study needs, making independent risk assessments tailored to their specific package and function usage. As a result, any bottlenecks from a centralized assessment process could be eliminated, allowing teams to use R with less friction.

{SLUSHY} AS A PATHWAY FOR THE PRESENT

Traditionally, {slushy} served as a temporary working environment, and the official final runs of programs occurred in the central frozen environment. In this hypothetical scenario, {slushy} supports the entire project life cycle, including final stages.

BEGIN BY ALIGNING WITH THE CENTRAL TEAM

Instead of taking on the full risk of the full {slushy} environment, teams adopting this model would ideally build on the central team's efforts, including vetted R packages. Suppose the central team installed a curated set of 20 R packages to the latest frozen environment. In line with industry thinking (i.e., the R Validation Hub white paper), these packages have been assessed and deemed trustworthy by the central team. Thus, the {study} team initiates their {slushy} environment using these 20 R packages and there is no need for reassessment of risk.

AUGMENT WHERE NEEDED

{slushy} permits flexibility in incorporating additional packages while retaining traceability and reliability. Additional packages require a risk assessment and supporting documentation, tailored to the team's intended use.

COMPLETE PROGRAMMING TASKS AS USUAL

After the {slushy} environment is initialized and modified according to the study's needs, it serves as the R environment throughout study. All team members access the same packages/versions, receiving notifications if they fall out of sync. With all packages installed from a single CRAN snapshot, teams can be assured that interdependent packages function as expected.

ARTIFACTS FOR TRACEABILITY

{slushy} generates artifacts for traceability, including a lock file containing metadata about packages/versions, which is the same as that produced by {renv} in the R Submissions Working Group pilots. Another important artifact is {slushy}'s "log": a report comparing the team's local environment with the specs of the centrally defined environment.
While a centralized model is highly important and applicable to most R use cases, {slushy} environments offer a promising alternative for teams seeking to use R on their own terms.

FURTHER CONSIDERATIONS

In this hypothetical scenario, the project or organizational context can dictate varying levels of risk tolerance, which impacts package selection and additional testing/documentation required to ensure the {slushy} environment's trustworthiness. By aligning closely with the work of the central governance team, teams using {slushy} can avoid unnecessary rework on package assessments.
Teams may also need to prove the packages have been installed correctly and perform as expected. While success is partially assessed implicitly through study code, some teams may opt for their own system qualification, possibly applying the central team's testing procedures on the custom environment for reassurance. In situations necessitating full qualifications, establishing best practices that reduce team burdens requires further consideration.

CONCLUSION

The open-source ecosystem is constantly evolving, enhancing our understanding of quality and risk management. Tools like {slushy} can complement central environments, effectively meeting the diverse and specific needs of individual teams.

REFERENCES

{slushy}: https://gsk-biostatistics.github.io/slushy/
Talk at Posit::conf 2023 – {slushy}: A Bridge to the Future

Posit Webinar – GSK's R Journey: From Pilot Projects to Enterprise Adoption

DISCLAIMER

The content of this paper is my personal opinion and not that of GSK.

CONTACT INFORMATION

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

Author Name: Becca Krouse
Company: GSK
Email: becca.z.krouse@gsk.com

Brand and product names are trademarks of their respective companies.