Paper OS10

{r}among: Towards a common, open-source solution for programming AMNOG-style EU-HTA analyses

Matthew Phelps, Novo Nordisk, Copenhagen, Denmark
Christian Haargaard Olsen, Novo Nordisk, Copenhagen, Denmark

ABSTRACT

The upcoming EU HTA requirements of the EU Joint Clinical Assessment (JCA) represents both a challenge and an opportunity. It's a challenge because draft guidelines indicate significant overlap with the analysis requirements for Germany reimbursement (AMNOG), well known for extensive workloads under challenging timelines. What infrastructure exists to handle the AMNOG analyses? Is it EU HTA proof? At Novo Nordisk, we have open-sourced an ecosystem of R packages to substantially reduce the programming time required for AMNOG style submissions. We aim to engage the wider community in further development, so they work across the industry. Therin lies the opportunity – to move towards a common way of programming AMNOG style submissions, so each company doesn't have to re-invent the wheel. While this presentation will explore the technical aspects, it will also address the "softer" aspects of developing open-source solutions, including how to build the engagement and community needed to ensure success.

R Package URL: https://hta-pharma.github.io/ramnog/

INTRODUCTION

In the pharmaceutical industry, the distinction between health technology assessment (HTA) and regulatory submissions is pivotal, yet often misunderstood. Regulatory submissions primarily address the question: "Can we use the treatment?" This involves a rigorous and predictable process in which the benefits are weighed against the risks, and the quality of the product is thoroughly scrutinized. Questions and trial designs are typically agreed upon with regulators at an early stage, ensuring a streamlined and relatively straightforward path to approval.

On the other hand, HTA submissions tackle a different question: "Should we use the treatment?" This encompasses evaluations of value, comparative effectiveness, and cost-effectiveness, often in a dynamic and evolving landscape. Unlike regulatory submissions, HTA assessments must adapt to new products entering the market, evolving treatment strategies, and a multitude of other variables, making the process inherently messy and dynamic. Decisions in HTA cannot be postponed; they must be made using the best available evidence, which is not always derived from randomized controlled trials (RCTs).

One prominent example of an HTA approach is that of the National Institute for Health and Care Excellence (NICE) in the UK, which often considers published results and employs indirect treatment comparison methods. In contrast, Germany's HTA process, governed by the Arzneimittelmarkt-Neuordnungsgesetz (AMNOG), requires a more localized and detailed assessment. The German benefit assessment focuses on the relevance of the drug to the German population, emphasizing specific endpoints and analyses that often deviate from the clinical study report (CSR). This necessitates substantial additional work, including re-analysis to ensure alignment with the characteristics and treatment patterns of German patients.

With the impending implementation of the EU Joint Clinical Assessment (JCA), a pan-European approach to HTA will come into effect, influenced heavily by the AMNOG framework. This new system, set to roll out in phases from 2025 for oncology drugs, 2028 for orphan drugs, and by 2030 for other treatments, aims to harmonize HTA processes across EU member states.

The motivation behind our initiative arises from the anticipated increase in workloads for statistical programmers due to the upcoming EU HTA requirements. The traditional methods of handling these analyses are unlikely to withstand the additional pressures. To address this, we at Novo Nordisk have developed and open-sourced an ecosystem of R packages designed to substantially reduce the programming time required for AMNOG-style submissions. Our goal is to engage the wider community in further development, fostering a common approach to programming these submissions across the industry. This initiative represents a significant opportunity to streamline processes, enhance efficiency, and avoid the redundancy of each company reinventing the wheel.

While this paper delves into the technical aspects of our open-source solution, it also highlights the "softer" aspects essential for the success of such initiatives, including community engagement and collaborative development. By doing so, we aim to build a robust, community-driven framework that not only meets the immediate needs posed by the EU HTA but also paves the way for future advancements in the field.

THE TECHNICAL PROBLEM TO SOLVE

AMNOG submissions are notoriously complex and voluminous, often entailing the generation of thousands of tables, figures, and listings (TFLs). To illustrate the scale, one recent AMNOG submission equated to approximately 3,300 clinical study report (CSR) summaries. This immense volume of required outputs presents a significant logistical challenge, as summarized by one of our HTA statisticians: "To produce all the outputs required for AMNOG is a logistical nightmare."

The core technical problem we are aiming to solve is the overwhelming burden placed on statisticians and programmers who must navigate this extensive output generation. The traditional approach not only demands substantial time and resources but also diverts focus from statistical and scientific considerations to more mundane bookkeeping and computational tasks.

OBJECTIVES OF RAMNOG

Our primary objective was to eliminate the "logistical nightmare" associated with AMNOG submissions and reduced the number of FTE (full-time employee) hours required per submission. We aimed to develop a framework that allows statistical programmers to concentrate on the essential statistical and scientific aspects of their work, rather than being bogged down by logistical and computational issues. As a by-product of improved efficiency and reduced manual input, we saw a reduction in the potential for errors.

RAMNOG FRAMEWORK OVERVIEW

In brief, ramnog is a collection of R packages designed to streamline the setup of pipelines for AMNOG-style HTA analyses.[1] The primary goal of ramnog is to enable statisticians and programmers to configure an AMNOG-type analysis with minimal familiarity with R.

For each study, the programmer will need to create, adjust, or verify the following four types of code:

1. Definition of Each Endpoint (or Group of Endpoints): This involves specifying the outcomes that need to be analyzed.
2. ADaM Functions: These functions modify existing ADaM datasets or create new ones if they do not exist for the required outputs. For example, creating new age groupings in the ADSL dataset.
3. Statistical Functions: A specification of the statistical methods used to summarize or analyze the data. The {chefStats} package provides a library of these functions.
4. (*optional)* Criteria for Endpoint Inclusion (if needed): A set of rules determining when an endpoint should be included in the results. These criteria are stored in the companion package {chefCriteria}.

The ramnog ecosystem of R packages is summarized as follows:

- **chef**: The core package that aids in creating endpoint specifications, building the analysis pipeline, and coordinating the execution of the analyses.
- **chefStats**: A collection of statistical methods used to make summarizations or inferences for each endpoint or outcome.
- **chefCriteria**: A collection of criteria used in AMNOG dossiers to determine when specific analyses or statistics should be calculated.
- **ramnog**: Contains all the CI/CD infrastructure and most of the help documentation for the ecosystem.

The framework has several key features that abstract away the logistical and computational complexities while maintaining extensibility:

- **Compact yet Flexible Endpoint Specification**: Allows users to define endpoints in a streamlined manner.
- **Analysis pipelining and Caching of Results**: Enables users to focus on one set of endpoints at a time without re-running the entire submission, while ensuring results are always up to date.
- **Plug-and-Play Library of Pre-Validated Statistical Methods**: Simplifies the use of statistical methods.

- **Custom User-Supplied Statistical Methods**: Provides the flexibility to include custom methods as needed.

DETAILS OF SELECTED RAMNOG FEATURES

SPECIFYING ENDPOINTS

The endpoint specifications are a central input to the {chef} package, containing the instructions to produce the statistics for endpoints. These instructions include:

- Data to Use for the Endpoint: Specifies which dataset will be used.
- Statistics to Include: Details the types of statistical outputs required.
- Calculation Methods: Describes how the statistics are calculated.
- Population Grouping: Defines how the population should be grouped.
- Inclusion Criteria: Criteria that must be met to include the endpoint in the final dossier.

Many components in the endpoint specifications reference custom functions outside {chef}, providing high flexibility in defining the endpoints. Both the endpoint specifications and associated custom functions must be supplied by the user as inputs to {chef} and are not part of the package itself.

Consider a typical safety section of an AMNOG submission, focusing on adverse event reporting. They might request measures such as:

- Number of subjects per arm
- Number of events per arm
- Number of subjects with events per arm
- Risk Ratio (RR) and Odds Ratio (OR) between arms
- Statistical significance between arms

This would result in a table with a total of 9 endpoints (endpoints in this context refer to cells in an output table). To specify this in the framework, you would use the following structure: {figure}

```
ep_spec <- chef::mk_endpoint_str(
  data_prepare = mk_adae,
  treatment_var = "TRT01A",
  treatment_refval = "High Dose",
  pop_var = "SAFFL",
  pop_value = "Y",
  stat_by_strata_by_trt = list(
    chefStats::n_subj,
    chefStats::n_subj_event,
    cheStats::p_subj_event
  ),
  stat_by_strata_across_trt = list(
    chefStats::RR,
    chefStats::OR,
    chefStats::p_val
  )
 );
```

A more typical request would also include stratification by some variables (e.g. age group, sex, race, etc.) and by the cross of severity level and System Organ Class (SOC). This quickly multiplies the number of endpoints to >4,000. However, to expand the specification to contain this complexity, only a few additional lines (highlighted in yellow) are needed:

```
ep_spec <- chef::mk_endpoint_str(
  data_prepare = mk_adae,
  treatment_var = "TRT01A",
  treatment_refval = "High Dose",
  pop_var = "SAFFL",
  pop_value = "Y",
```

```
  stat_by_strata_by_trt = list(
    chefStats::n_subj,
    chefStats::n_subj_event,
    cheStats::p_subj_event
  ),
  stat_by_strata_across_trt = list(
    chefStats::RR,
    chefStats::OR,
    chefStats::p_val
  ),
  stratify_by = list(c("AGEGR", "SEX", "RACE")),
  group_by = list(list(ASEV = c(), AESOC = c())),

);
```

ANALYSIS PIPELINING

The endpoint specification demonstrated above shows how to quickly specify a group of related endpoints. However, AMNOG submissions consist of many of these groups of endpoints. Some endpoints can be time-consuming to calculate, and running all the endpoints can take many hours for some submissions.

To allow the statistician to focus only on a subset of analyses, the framework allows each endpoint group to be worked on independently of each other. Once an endpoint group is run, the results are cached, and this group is not re-run again unless one of its dependencies is outdated. The user can go back and modify an already-run endpoint group, and only the modified endpoints will be re-run.

The core functionality of this pipelining is provided by the {targets} package by Will Landau at Eli Lilly[2].

EXTENSIBILITY

To maintain the extensibility of the ecosystem and to easily allow it to work for many different users, we offloaded into separate packages the aspects of the codebase most likely to vary between use cases. The core functionality handling the endpoint specifications and the analysis pipelining is housed in the {chef} package, while the library of statistical methods and the analysis filters are maintained in the {chefStats} and {chefCriteria} packages, respectively.

Keeping the statistical methods and analysis filters in separate packages allows users to more easily add new methods. This is because they only need to worry about how {chefStats} or {chefCriteria} functions interface with {chef}. They do not need to worry about touching the codebase of chef, which is more complex. These packages offer helper functions to guide the user when writing new functions to help ensure compatibility with the ramnog way of writing statistical or analysis filter functions.

For more ad hoc extensibility, users can directly supply their own custom functions to any argument in the endpoint specification that accepts functions as input. This allows users to quickly add functions that might only be relevant to their project and are not good candidates to be added to the standard libraries in {chefStats} or {chefCriteria}.

Additionally, there is an opportunity to add functionality both to the input and the output of the current ecosystem. We envision potential input modules helping design the analysis specifications, while also providing the specifications in machine-readable format. This would reduce the coding required even further. Output modules could take the results datasets and format them to meet the requirements of regulatory authorities, or even into interactive formats, such as web apps.

This focus on extensibility aligns well with our goal of promoting community driven development as it makes it easier for other developers to start contributing. But open-sourcing involves much more than extensible-oriented code design.

OPEN SOURCE

When working with R open source quickly seems like the default mode of operation for bundled functionality. Many others have made contributions that makes R the great tool it is, and it seems only fair that any new value one can add would be fed back to the pool for the greater good. While this view spoke to our own values, open source also quickly proved to be viable from a business perspective.

Chef was created out of a necessity of being able to do among analysis more efficient. Looking at package like *admiral* we could see the huge benefits that collaboration can bring in tool development: more people to scope and design the architecture well, more people to help in the implementation and more users to ensure thorough testing and bug fixing. This in turn can help drive more users and people with good ideas to the project, potentially starting a positive feedback loop of interest and improvements.

When conducting statistical analyses, trust is paramount for the usability of the results. Open source provides a good foundation for building trustworthy tools. At the core open source ensures transparency as all code is available for inspection, but the tools available on typical open-source projects (version control, automated testing, issue tracking) also promotes transparency about decision making and implementation as well as robustness of the tool.

In addition, participation in open-source development can also serve as motivation for the individual. Collaboration gives a sense of community, discussing with others provide great learning opportunities, while public listed contributions can serve as a proof of possessed skills. Finally, developing tools available for all drug developers also allows the individual to have a positive impact on the life of patients across all diseases.

Summarizing, we group the advantages of open source into the following categories.

1. **Collaboration**: Open-source frameworks encourage collaboration across the industry. By inviting contributions from a diverse group of stakeholders, we can leverage a wide range of expertise and perspectives to continuously improve the framework.
2. **Quality**: The open-source model naturally fosters higher quality through peer review and community scrutiny. Bugs and issues can be identified and resolved more quickly, ensuring a more reliable and robust solution.
3. **Trust:** Transparency is a cornerstone of open-source projects. Users can inspect the code, understand its functionality, and trust that it operates as intended. This transparency builds confidence among users and stakeholders.
4. **Sharing**: Open source promotes the sharing of knowledge and best practices. As more organizations adopt and contribute to the framework, the collective knowledge base grows, benefiting the entire community.

The open source journey of {r}amnog

While we are dedicated to the open source vision of our tool, it was not born as a collaborative effort – and has not yet become one. And while we always had open source requirements in the back of our heads while we developed it, we focused primarily on building a tool that could help us with the task at hand. Along the way we developed helper functions for easy setup, made unit tests and build vignettes for easy access. First for our internal colleagues, but also with external collaborators in mind later. We ended with an MVP that was reliant quite a bit of our internal computational setup, but which nevertheless proved the value of the idea.

After the initial MVP and the first test run, we identified several opportunities for improvements that lead to a rewrite of the internal structure. At the same time, we focused more on the requirements for making the tool available outside of Novo Nordisk. This meant that dependencies on our internal methods for accessing data had to be removed, many function names had to be changed, etc. We also greatly improved the documentation, as well as automating testing of the package through CI/CD pipelines on Github.

License

While there exist standard and well-established open-source licenses, such as APACHE, MIT and GPL, it was difficult to get good input from our internal organisation on choice of license. And it took quite a while before we got a green light to make our tool public. We chose the permissive MIT license, because we heard from other pharma colleagues that their company policies are to only use tools with permissive licenses.

Hosting

In addition to consider how choice of license might limit how others would use our tool, we also considered factors that might keep others from collaboration. This led us to a decision about hosting the code under the *HTA-R* workstream in *openstatsware*. Choosing infrastructure on a neutral ground means giving up some of the control of the package, but it also means that it is freer. It is worth noting that while an open-source license does give everyone the right to freely use and distribute the code – it does not guarantee the access to it. Should our company in the future decide no longer to sponsor open-source projects and close our organisations Github page, {r}among will still be available in the same location for anyone using it. With all the infrastructure that goes with it. We encourage others to choose similar neutral grounds for hosting and development, to ensure that we can count on continuity in the availability of our tools.

A different strategy for collaborations

Looking back at our journey, there is a few things we would different if we had to do it all over again. We did have some concern about when the tool and code would be mature enough that we could share it with others. Looking back, we should have started developing directly in the public. It would have made it easier from a practical point of view – not having to move our code and infrastructure from one platform to another. But, more important, it would have made it possible to invite collaborators much earlier in the process. Overall, we would have put more emphasis on building collaborations early on and making the tool development a joint venture.

Looking forward towards the increased HTA requirements in the JCA there is a need for tools to support these new processes. As these are new and similar requirements for all pharmaceutical companies, there is a large potential for doing strategic collaborations to build solutions that will work for everyone. This is exactly what is called for in the new HTA Working group in the R-consortium and it will be interesting to see which initiatives are born from it.

CONCLUSION

The upcoming EU Joint Clinical Assessment (JCA) presents both significant challenges and remarkable opportunities for the pharmaceutical industry. We have proactively addressed these challenges by developing and open-sourcing an ecosystem of R packages designed to streamline the process of AMNOG-style submissions. This initiative not only aims to reduce the programming time and logistical burden associated with these submissions but also seeks to foster a collaborative community across the industry focused on solving this challenge. By promoting a common framework and encouraging contributions from a diverse group of stakeholders, we can enhance the efficiency, quality, and reliability of HTA analyses. This effort has the potential to set a new standard in the industry, ensuring that we are better prepared to meet the evolving requirements of the EU HTA, however, we first need to secure more engagement from the community.

REFERENCES

1. Matthew David Phelps, Nicolai Skov Johnsen, Henrik Sparre Spiegelhauer, Christian Haargaard Olsen (2024). ramnog: Facilitating AMNOG style analysis pipelines. R package version 0.1.2, https://github.com/hta-pharma/ramnog, https://app.codecov.io/gh/hta-pharma/ramnog, https://hta-pharma.github.io/ramnog/.
2. Landau, W. M., (2021). The targets R package: a dynamic Make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. Journal of Open Source Software, 6(57), 2959, https://doi.org/10.21105/joss.02959

CONTACT INFORMATION

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

Author Name: Matthew Phelps
Company: Novo Nordisk
Address: Vandtårnsvej 114 Søborg, Denmark
Work Phone: +45 4444 8888
Email: mewp@novonordisk.com
Website: https://hta-pharma.github.io/ramnog/index.html

Author Name: Christian Haargaard Olsen
Company: Novo Nordisk
Address: Vandtårnsvej 114 Søborg, Denmark
Work Phone: +45 4444 8888
Email: cino@novonordisk.com
Website: https://hta-pharma.github.io/ramnog/index.html