Paper RE02

Leveraging Julia for Advanced Data Visualization in Pharmaceutical Research

Ranjeet Kumar, Bristol Myers Squibb, Hyderabad, India
Deepak Vaishnav, Bristol Myers Squibb, Hyderabad, India

Abstract

The pharmaceutical industry increasingly relies on data-driven insights derived from complex clinical and operational datasets. As study sizes, data dimensions, and analytical expectations continue to grow, traditional visualization approaches often struggle to deliver performance, interactivity, and scalability simultaneously. This paper explores the Julia programming language as a modern, high-performance platform for advanced data visualization in pharmaceutical research. Julia's ability to combine computational efficiency with expressive visualization frameworks positions it as a strong complementary technology to existing ecosystems such as SAS®, R®, and Python®. The paper discusses practical visualization use cases, performance considerations, regulatory boundaries, and future potential, with a focus on safe and phased adoption within regulated environments.

Introduction to Julia Programming Language

Julia® is an open-source, high-performance programming language designed specifically for numerical and scientific computing. It was created to address the so-called 'two-language problem', where analysts prototype in high-level languages and later rewrite performance-critical code in lower-level languages. Julia offers a single-language workflow that combines expressive syntax with near C-level execution speed, making it particularly suitable for data-intensive analytics and visualization workloads.

Data Visualization in Pharmaceutical Research

Data visualization is vital throughout the pharmaceutical research lifecycle, aiding in study planning, execution, safety monitoring, and decision support. With complex datasets involving demographics, adverse events, laboratory results, and operational metrics collected longitudinally, effective visualization allows stakeholders to quickly interpret trends, identify anomalies, and communicate findings across teams. Modern analytics demand interactive, scalable tools that go beyond static plots and summary tables, supporting rapid iteration while maintaining regulatory discipline. Visualization techniques uncover patterns not apparent in tabular outputs, enhance transparency and traceability for data review and audit readiness, and accelerate insight generation—enabling faster, more informed decisions during clinical development.

Current Visualization Landscape

The current pharmaceutical analytics landscape is dominated by SAS, R, and Python. SAS remains widely used for clinical reporting and submission deliverables due to its validated environment and long-standing regulatory acceptance. R and Python are widely adopted for exploratory analytics and visualization, supported by rich open-source ecosystems. Despite their strengths, performance and scalability challenges often emerge when handling large datasets or demanding interactive visualizations, particularly when frequent data refreshes are required.

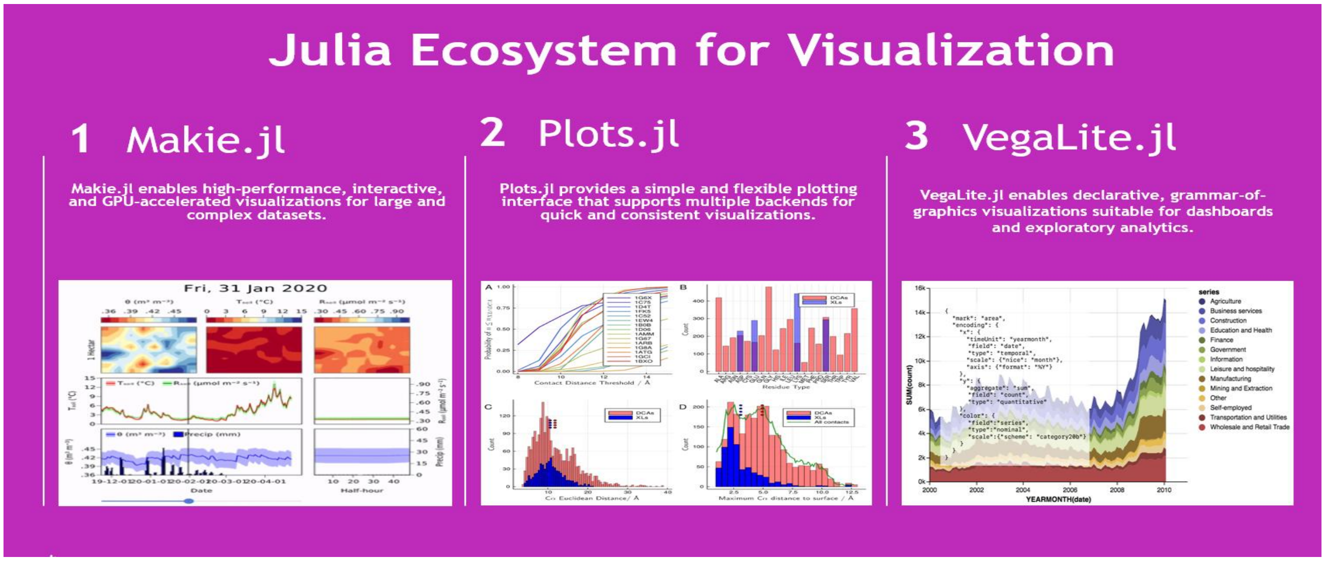
Why Julia?

Julia's primary strength lies in its native performance, achieved without sacrificing developer productivity. Unlike many high-level languages, Julia allows users to write concise, readable codes that execute efficiently at scale. In addition, Julia supports parallelism, efficient memory management, and seamless integration with modern visualization libraries. These characteristics make Julia an attractive option for scenarios where both computational efficiency and interactive visualization are required.

Julia Ecosystem for Visualization

- ➢ Julia offers a growing and mature visualization ecosystem tailored for different analytical needs. Makie.jl provides high-performance, interactive, and GPU-accelerated visualizations suitable for large and complex datasets.

- ➢ Plots.jl offers a flexible and user-friendly plotting interface that supports multiple backends, enabling quick generation of standard analytical graphics. VegaLite.jl supports declarative, grammar-of-graphics visualizations that are particularly well suited for dashboards and exploratory analytics.



Julia Ecosystem for Visualization

1 Makie.jl

Makie.jl enables high-performance, interactive, and GPU-accelerated visualizations for large and complex datasets.

2 Plots.jl

Plots.jl provides a simple and flexible plotting interface that supports multiple backends for quick and consistent visualizations.

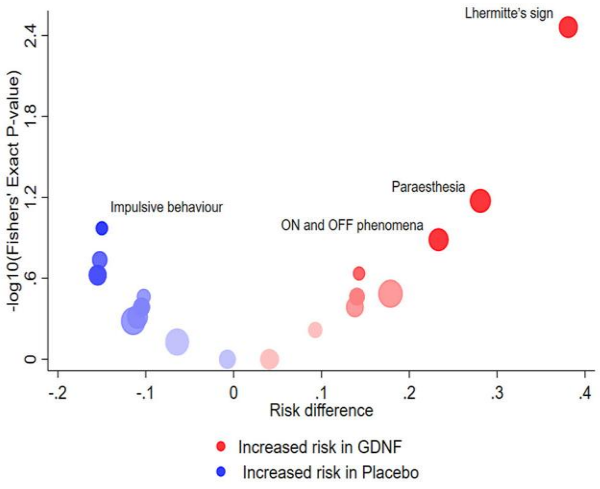
3 VegaLite.jl

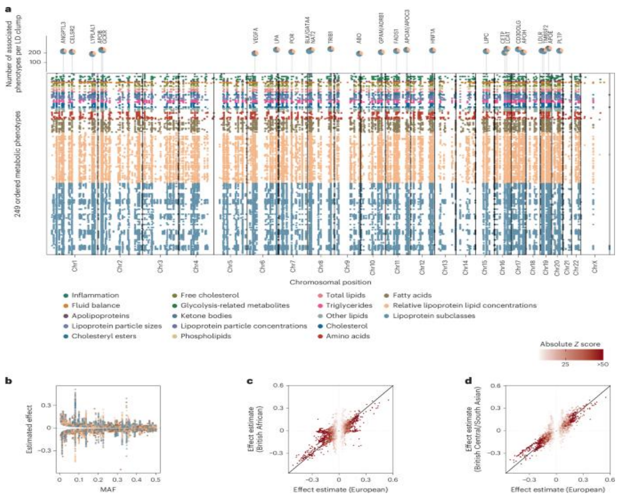
VegaLite.jl enables declarative, grammar-of-graphics visualizations suitable for dashboards and exploratory analytics.

Source: MakieOrg/Makie.jl

Clinical Trial Data Visualization

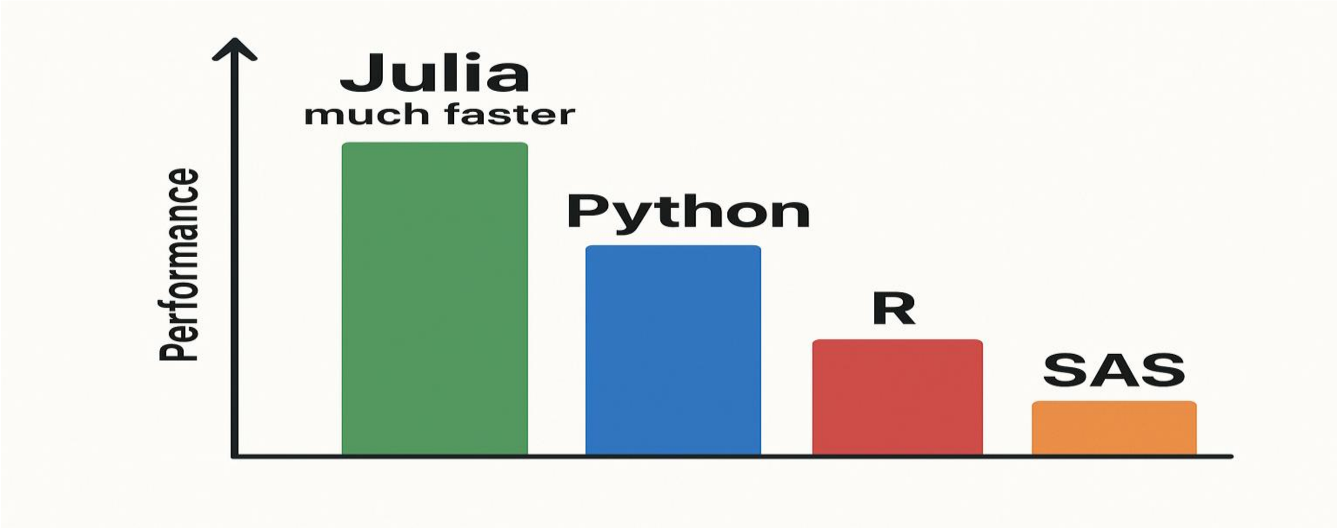
- ➢ In clinical research, visualization is essential for monitoring trial progress and ensuring patient safety. Julia can support advanced visualization of patient enrollment trends, enabling real-time assessment of recruitment performance across sites and regions.
- ➢ Similarly, adverse event distributions and laboratory value trends can be explored interactively, allowing analysts to identify safety signals, outliers, and clinically relevant shifts earlier in the trial lifecycle.





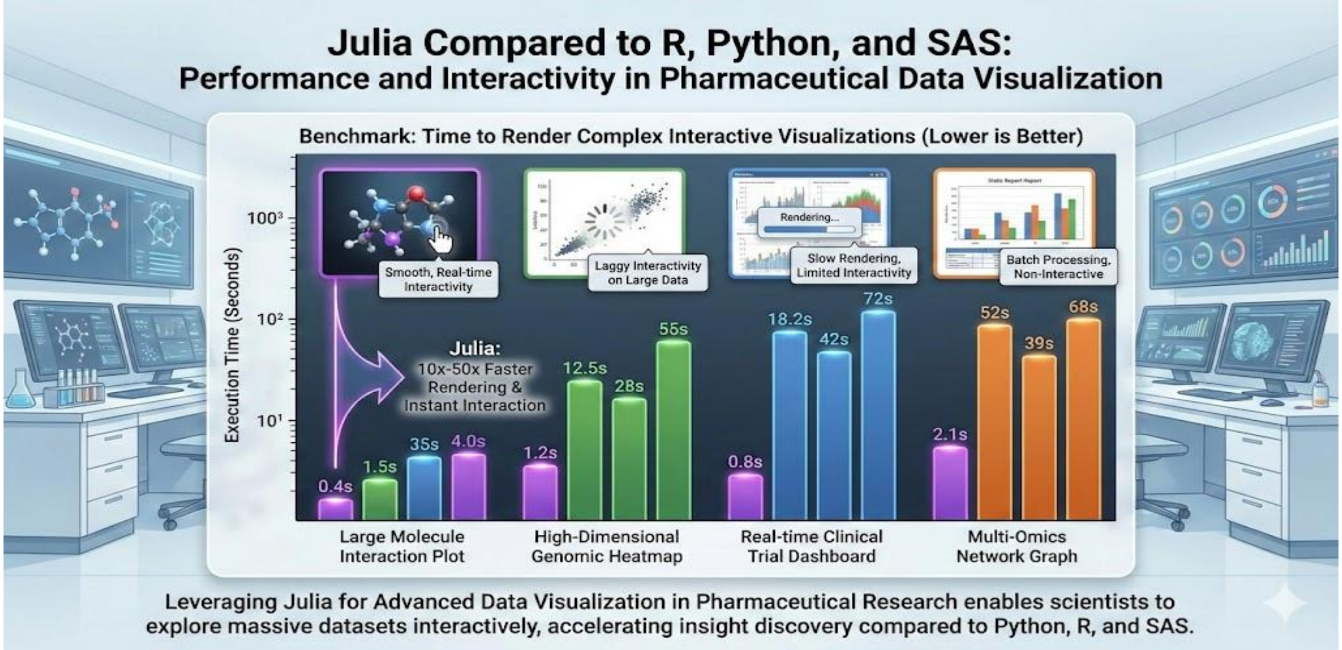
Performance Comparison

- Performance is a critical consideration when working with large clinical datasets. Julia's execution model allows it to scale efficiently as data size and analytical complexity increase.
- Compared to traditional scripting approaches, Julia reduces the need for external performance optimizations, enabling smoother visualization workflows even when datasets are large and frequently updated.



Julia Compared to R and Python: Performance and Interactivity

- R and Python rely heavily on optimized libraries and external backends to achieve acceptable performance for large-scale analytics. While this approach is effective, it can introduce complexity when building interactive visualization workflows.
- Julia's design allows performance and interactivity to coexist within the same environment, enabling responsive data exploration without extensive tuning or language switching.



Pseudo Code Example: Load, Transform, Visualize

- A typical Julia-based visualization workflow begins with loading clinical or operational data from structured sources. The data is then transformed, summarized, and prepared for analysis using efficient data manipulation techniques.
- Finally, interactive visualizations are generated using Julia's visualization libraries, allowing analysts to explore trends and patterns dynamically. This unified workflow simplifies development and supports rapid iteration.

Regulatory Considerations

Within regulated pharmaceutical environments, it is essential to clearly distinguish between exploratory analytics and validated submission workflows. Julia is best positioned for exploratory visualization, innovation initiatives, and analytical research. Appropriate governance, documentation, version control, and validation strategies must be applied to ensure compliance with GxP expectations while enabling innovation.

Where Julia Fits Today

Today, Julia fits well within innovation labs, proof-of-concept initiatives, and advanced exploration analytics teams. It can be used alongside existing tools to address performance-intensive visualization challenges without disrupting established regulatory workflows.

Future Potential

As the Julia ecosystem continues to mature, its role within pharmaceutical analytics is expected to expand. Hybrid workflows that integrate Julia with R and Python are likely to become more common, enabling organizations to leverage the strengths of each technology. In the longer term, Julia has the potential to support high-performance analytics platforms and advanced visualization solutions as data complexity continues to grow.

Key Takeaways

Julia represents a serious and emerging option for advanced data visualization in pharmaceutical research. Its combination of performance, interactivity, and expressive visualization frameworks address many challenges associated with large and complex datasets. A phased and governed adoption approach is recommended to ensure regulatory compliance while unlocking innovation.

Conclusion

Advanced data visualization is becoming increasingly critical in pharmaceutical research as data volume, velocity, and complexity continue to increase. Julia offers a compelling complementary solution that addresses performance and interactivity limitations observed in traditional visualization approaches. By adopting Julia in a controlled and exploratory manner, pharmaceutical organizations can enhance their analytical capabilities while maintaining regulatory rigor and future readiness.

Contact Information

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

Author: Ranjeet Kumar
Company: Bristol Myers Squibb
Address: International Tech Park (CapitaLand), Inorbit Mall Road,
Madhapur, Hyderabad-500081, Telangana.
Email: ranjeet.kumar3@bms.com
Website: www.bms.com

Co-Author: Deepak Vaishnav
Company: Bristol Myers Squibb
Address: International Tech Park (CapitaLand), Inorbit Mall Road,
Madhapur, Hyderabad-500081, Telangana.
Email: deepak.vaishnav@bms.com
Website: www.bms.com