Paper OS18

Check Yourself: Edit Checks and Queries Made Easy with Shiny

Maya Gans, Atorus Research
Kirsty Lauderdale, Atorus Research

Abstract:

Effective data management is critical in clinical trials to ensure clean, accurate, and compliant datasets. Shiny allows us to build applications tailored for clinical data management and enables CDMs to move beyond static listings and spreadsheets. By integrating edit check logic, discrepancies and flags can be raised in real time with minimal manual effort. Automated queries generated based on predefined rules/dynamic thresholds supporting consistent and efficient data cleaning. Customizable filters, drill-down views, and responsive visualizations make it easier for CDMs/DMs to explore key issues across subject-level and study-wide datasets. To ensure performance and usability, strategies such as modular design, reactive programming, and server-side processing are employed to ensure smooth performance with large clinical datasets. Features such as audit logging and built-in metrics enhance traceability and oversight. See Shiny go beyond visualizations—serving as a practical, efficient platform for improving the speed, accuracy, and transparency of clinical data review.

Introduction

In clinical trials, the accuracy and completeness of data underpin every stage of statistical analysis and regulatory submission. Data Management (DM) teams invest substantial effort to detect discrepancies, raise and resolve queries, and ensure compliance with global standards such as CDISC and 21 CFR Part 11. Traditionally, these activities rely on static listings, spreadsheets, or SAS-based edit check programs, approaches that, while reliable, can be slow to iterate and difficult to scale.

Shiny, an R-based framework for building interactive web applications, offers an opportunity to transform this paradigm. By combining dynamic visuals, color-coded cues, and interactive dashboards, a Shiny application enables teams to quickly spot anomalies across large datasets. Users can immediately drill down into individual records, raise queries, and resolve issues all within the same interface, eliminating the need for cumbersome multi-step processes. This approach not only reduces review time, providing clear return on investment (ROI), but also fosters a more automated, traceable, and collaborative workflow for ensuring data quality.

This paper presents the design and implementation of a Shiny-based application for edit check automation and query management. It explores architectural decisions, user experience considerations, and performance strategies that allow the tool to function efficiently across large clinical datasets.

Why Shiny?

The volume, diversity, and velocity of data in modern clinical trials have increased dramatically, with sources now including EDC, laboratory systems, ePRO, wearable devices, and external data feeds. With these growing complexities, data management must ensure that quality controls keep pace, maintaining accuracy, while increasing efficiencies and focusing on adaptability.

Regulatory expectations emphasize traceability and reproducibility, requiring that data review processes are documented, consistent, and auditable. However, manual reviews and Excel-based tracking often lead to inconsistent queries, redundant effort, and limited oversight. Generic tools can overwhelm teams with buried features, creating “dashboard fatigue” and limiting adoption.

R and Shiny provide a modern, agile approach to data review. Built on open-source principles, R offers powerful data handling, statistical modeling, and error-checking capabilities. Shiny translates these into sleek, interactive web applications. Together, they allow the creation of bespoke solutions tailored to a Data Management team’s exact workflow, enabling not just a customized interface, but also customized error detection, validation rules, and modeling systems. Users can interactively explore data, raise queries, and address discrepancies in real time. Looking ahead,

such frameworks could even integrate AI to anticipate anomalies and suggest corrective actions, further enhancing efficiency and data quality.

Building the Shiny Solution

Through close engagement with our internal Data Management team to gain a deep understanding of their experiences monitoring and cleaning clinical data, we identified opportunities to optimize workflows and minimise frustrations. This insight informed the development of a modular Proof of Concept (PoC) Shiny application designed to support end-to-end data cleaning. The solution utilises raw data but can also integrate seamlessly with CDISC-formatted datasets, including SDTM and ADaM, and leverages configurable rule definitions to automate edit checks and query generation, enhancing both speed and accuracy.

At a strategic level, the application is built on four core architectural layers, each addressing a critical component of the workflow. These layers were developed concurrently by the team to ensure cohesive functionality, robust performance, and alignment with user needs and regulatory standards:

- **Data Access and Ingestion**: Securely loads clinical datasets and metadata into memory or a database.
- **Visualization Layer**: Provides interactive dashboards, summaries, and filters for data review.
- **Rule Engine**: Executes logical checks against the data to identify discrepancies.
- **Query Module**: Generates and manages queries based on rule results.

Interoperability with other systems, such as EDC platforms, metadata repositories, and issue trackers, is achieved through REST APIs or standardized exports, enabling integration into existing workflows.

Data Access and Ingestion

Data access and ingestion form the foundation of any Shiny application designed for clinical trial analytics. These processes govern how raw and derived data, from SDTM, ADaM, or other operational sources, are securely retrieved, standardized, and made available for real-time exploration and analysis.

Properly defining data access ensures that team members interact only with data appropriate to their role, maintaining strict compliance with regulatory, privacy, and governance requirements. Role-based access controls safeguard sensitive patient and study information while preserving the integrity of ongoing analyses.

Effective data ingestion, in turn, enables seamless integration and transformation of raw clinical data into analysis-ready structures without manual intervention. When implemented correctly, this framework ensures that users can confidently visualize, analyze, and interpret up-to-date information within Shiny, supporting faster decision-making, enhanced data integrity, and full traceability from ingestion to insight.

With the above in mind, we established a robust backend architecture capable of handling raw and query-level data through API connections to the study database. This provided a dynamic data layer that kept the application synchronized with live data updates and ensured authorised users were always reviewing the latest information.

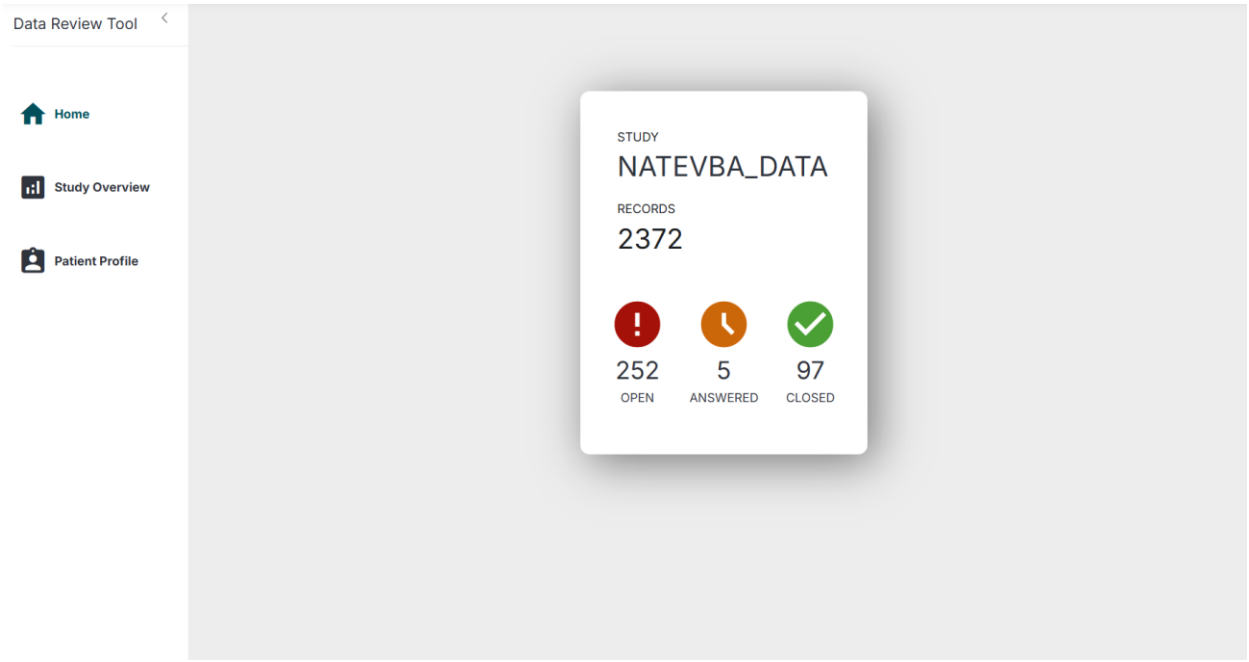

Fig.1: Landing page of studies available to current user, summarizing the number of open, answered, and closed queries

Visualization Layer

The development of the application began through an iterative, user-centered design approach focused on aligning technology with real-world clinical data management workflows. Each week, our team engaged directly with Data Management end users who were accustomed to working across multiple disconnected systems, each with unique quirks, interfaces and navigation challenges. Through observation and discussion, we mapped their existing review workflows, tracking how many steps, clicks, and screens were required to complete common tasks such as identifying discrepancies, raising queries, and cross-checking listings. These sessions revealed significant inefficiencies, as well as key opportunities to streamline and centralize the entire data cleaning process within a single, unified interface.

Using this feedback, we moved into structured wireframing and prototyping sessions, which allowed users to visualize and interact with early design concepts before development started. Wireframing proved invaluable in aligning expectations, validating user flows, and refining features based on practical feedback. By incorporating user review at each stage, we ensured the final application was not only functional, but also intuitive and efficient, reducing cognitive load, focusing on minimizing clicks and consolidating views, and improving data review accuracy. This collaborative, feedback-driven process helped keep the users oriented as they navigated between study, subject- and record-level perspectives.

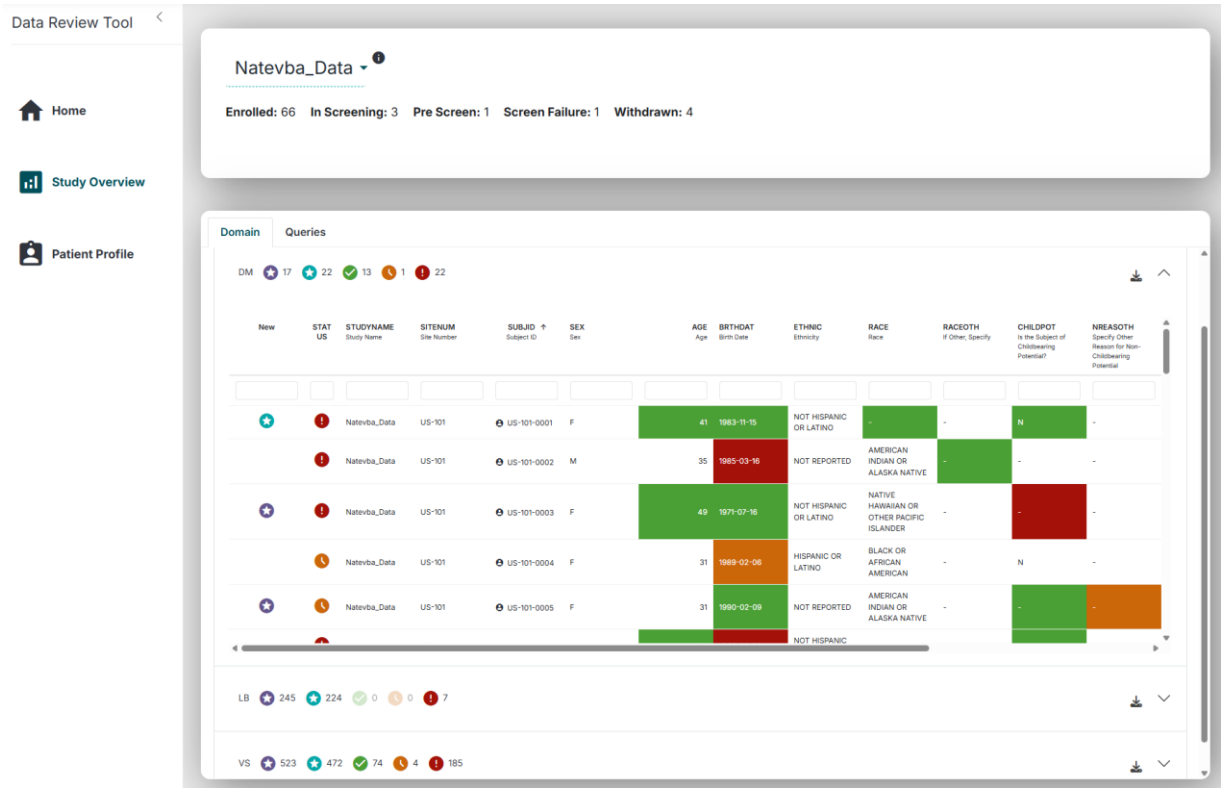

Fig.2: Study Overview page: data is visualized in the form of listings and cells with queries that are closed, open, or answered and coloured using the traffic light coloring.

Rule Engine

At the core of the application lies a configurable rule engine that drives the edit check logic and query management process. The rule engine was designed to be flexible and support both static and dynamic logic checks, enabling data managers to identify issues ranging from simple missing values to complex cross-domain inconsistencies. Rules are authored in a flexible, human-readable configuration format such as YAML or JSON, or directly in R for more advanced logic, allowing for transparency, auditability, and reuse across studies.

Each rule definition captures essential metadata, including the datasets and variables involved, the condition that triggers a discrepancy, its severity level, and suggested query text for investigator follow-up. Once deployed, the Shiny application executes these rules reactively, automatically flagging discrepancies in real time as data or parameters are updated. When a rule is triggered, the system dynamically generates a corresponding query populated with study metadata and standardized templating, then tracks its lifecycle through resolution.

By centralizing edit check logic within a unified, configurable framework, the system reduces manual review effort, enforces consistency, and accelerates data cleaning. Cross-domain checks are supported through relational joins, and dependencies are intelligently managed to maintain logical execution order. The result is a transparent, auditable, and efficient data management process that aligns with clinical data standards while empowering teams to focus on proactive data review rather than reactive cleanup.

Query Module

The final phase of development focused on integrating automated query generation and tracking across all data views, both tabular and graphical, giving the data management team a cohesive experience tailored towards their own data review preferences. This integration fundamentally reshaped how queries are created, reviewed, and resolved, ensuring every interaction with trial data is fully traceable and compliant with regulatory standards. Real-time visualizations and trend summaries enabled teams to monitor data quality, query volumes, and resolution progress proactively, rather than reactively.

By connecting querying activities directly to the data they reference, the system established clear relationships between discrepancies, actions, and outcomes, providing complete transparency throughout the data cleaning lifecycle. Each query, from detection to closure, is logged within a robust audit trail, supporting accountability and reproducibility at every stage. The result is a measurable improvement in efficiency and data quality: a streamlined, collaborative environment where data cleaning becomes faster, more consistent, and intrinsically aligned with the principles of traceability and regulatory readiness.

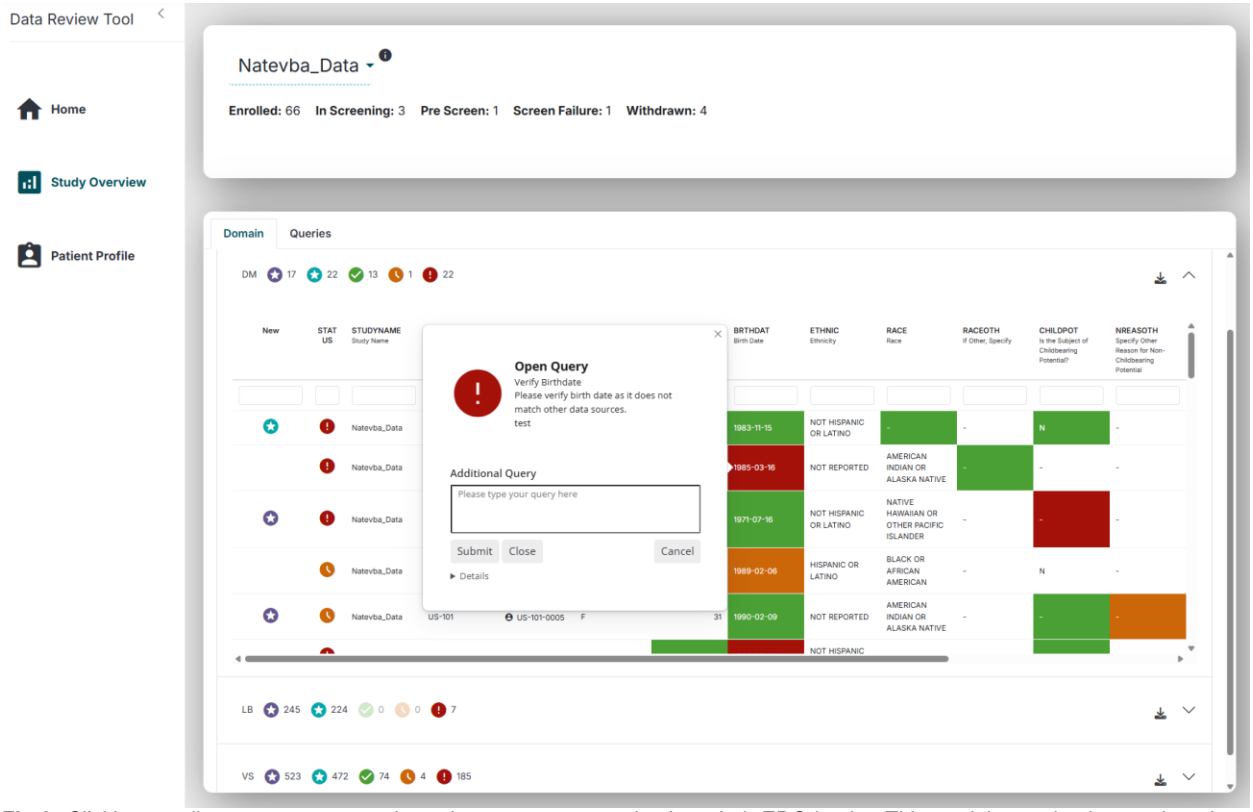

Fig.3: Clicking a cell opens a popover where the user can query back to their EDC in situ. This modal can also be used to close queries. Access controls ensure all actions are logged automatically, capturing who performed them and the associated observation metadata for full traceability.

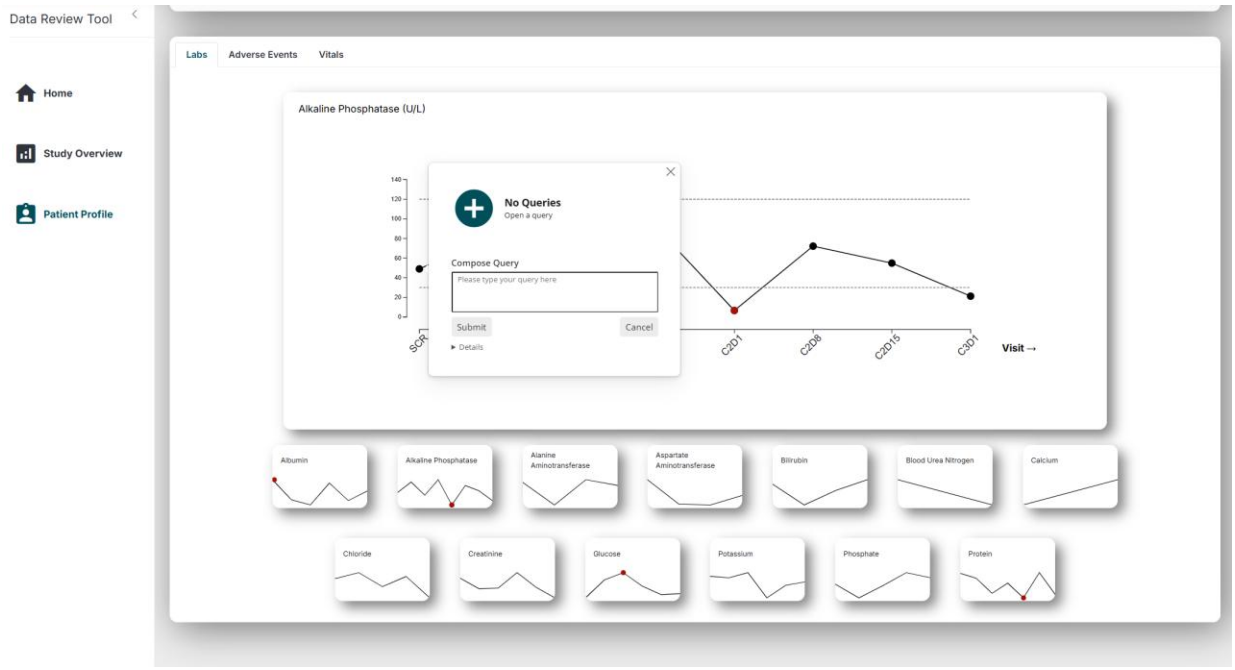

Fig.4: The graphical view offers the same querying capabilities as the listing view, with the added advantage of visually identifying outliers, anomalies and performing rapid spot checks directly within line charts.

Application Features and User Experience

From the user's perspective, the application transforms the data review experience into something intuitive, efficient, and transparent. Customizable filters allow our clinical Data Managers to quickly narrow focus by domain, subject, visit, or rule category, while seamless drill-down navigation makes it easy to move from study-level summaries to individual subject records in just a few clicks.

Dynamic visualizations highlight patterns and outliers instantly, helping users identify and act on issues faster than ever before. Routine reviews that once required navigating multiple systems can now be completed in a fraction of the time within a single, cohesive interface. Built-in audit trails automatically capture every action, ensuring traceability without adding administrative burden.

The result is a streamlined, user-friendly application that simplifies data cleaning, accelerates decision-making, and gives teams confidence that every review is complete, compliant, and efficient.

Technical Implementation

From a technical standpoint, the application leverages a custom API to retrieve source data and monitor changes in real time, providing a live feed of new and updated records. Initially, much of the data processing was programmed independently, but we later had the opportunity to test APIs with Veeva and ultimately decided to work with CDB to support automated query setup. This dynamic data layer supports a modular, hierarchical application architecture, enabling users to navigate from high-level study summaries down to individual patient views.

Tables and visualizations are backed by a custom server-side architecture, allowing interactive elements (such as clicking on a table cell or plot point) to trigger a live query back to the data source, ensuring the most current information is always displayed.

The application is designed following best practices for scalable and maintainable Shiny applications. Key implementation strategies include:

- **Reactive Programming:** All outputs are efficiently updated when data or inputs change, minimizing redundant computations and keeping the interface responsive.

- **Server-Side Processing:** Heavy data manipulation and aggregation are performed outside of the Shiny application, allowing the app itself to remain lightweight and highly performant, even when working with large clinical datasets.

- **Caching and Lazy Loading:** Shiny's native caching, combined with modular lazy loading strategies, ensures rapid access to frequently used datasets and reduces unnecessary API calls.

- **Interactive Tables and Visualizations:** Reactable tables and custom D3 graphics in the form of HTMLWidgets allow users to drill down from high-level study summaries to individual patient records. Clicking on a table cell or a D3 plot element triggers a live query back to the source data or the EDC, enabling seamless issue creation and resolution directly from the interface.

- **Access Control and Security:** Integration with Connect provides role-based access controls, user authentication, and secure API connections, ensuring compliance with regulatory requirements.

- **CI/CD and Reproducibility:** Continuous integration pipelines automate testing and deployment, while Docker-based containerization ensures consistent, reproducible environments. Git version control supports auditability and collaborative development.

- **Extensibility for Future Enhancements:** The modular architecture supports integration with AI-based anomaly detection, automated query prioritization, and additional custom modules without extensive rework of the core framework.

This combination of tools and design principles enables a responsive, scalable, and user-friendly application tailored specifically for clinical data management workflows, balancing performance, compliance, and flexibility.

Challenges, Value Add and Future Thoughts

Shiny's flexibility and accessibility have made it a valuable addition to the clinical data management toolkit. While originally developed for statistical visualization, its reactivity and open-source nature make it equally effective for operational data review.

Key lessons learned included the importance of defining rule ownership early, modularizing rule logic, and validating the performance with production-scale datasets.

Deploying Shiny in a regulated context requires a careful balance between agility and compliance. Organizations must establish governance for version control, validation, and user access while maintaining the agility that makes Shiny so powerful.

Future enhancements could extend beyond rule-based checks toward machine learning–assisted data review, identifying anomalies or patterns not covered by predefined rules. Integration with metadata-driven standards (e.g., define.xml, ODM, or MDR APIs) could further automate rule configuration.

Additionally, scaling to multi-user environments through ShinyProxy or Posit Connect would allow collaborative, portfolio-level monitoring across studies. Enhancements to visualization and real-time metrics could turn the application into a comprehensive data cleaning dashboard.

Conclusion

By combining automation, interactivity, and auditability, Shiny redefines how clinical data management teams approach data cleaning and quality review. The presented application demonstrates that edit checks and query workflows can be streamlined while maintaining compliance and oversight.

Shiny's versatility allows it to go beyond visualizations, serving as a practical, efficient, and transparent platform for managing the integrity of clinical trial data. By enabling faster identification of anomalies, reducing redundant manual reviews, and minimizing the need for external tools or repeated data extractions, the app delivers measurable time and cost savings.

Additional efficiencies include automating repetitive queries, consolidating dashboards into a single interface to prevent “dashboard fatigue,” and providing a scalable framework that can accommodate future enhancements such as AI-driven anomaly detection.

Together, these improvements increase team productivity, accelerate study timelines, and reduce operational costs, providing a strong return on investment for the clinical trial program.

Acknowledgements

We would like to express our gratitude to the Atorus data management team, for their time and patience, and our Atorus colleagues who helped with the development of the PoC, it is our privilege to work alongside such a great team.

Contact Information

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

Maya Gans
Atorus Research (https://www.atorusresearch.com/)
Maya.Gans@Atorusresearch.com

Kirsty Lauderdale
Atorus Research (https://www.atorusresearch.com/)
Kirsty.Lauderdale@Atorusresearch.com