Paper DV03

Your Outputs Deserve a Second Set of Eyes — Meet ELARA

Mane Sargsyan, STATECS LLC, Yerevan, Armenia
Arman Melkonyan, STATECS LLC, Yerevan, Armenia

ABSTRACT

Validating clinical outputs can be slow, repetitive, and vulnerable to human oversight — especially when reviewing layout details like titles, footnotes, page numbers, and axis labels. ELARA (Expert Layout Analysis and Reporting Assistant) is an R-based solution that uses image recognition to automate this visual checking process. It compares two versions of a report, scanning page by page to detect mismatches in structure, layout, and text placement. Instead of manually reviewing outputs line by line or figure by figure, users receive a clear visual summary of what changed — saving time, reducing errors, and increasing confidence in report delivery. ELARA integrates easily into existing workflows and allows reviewers to focus on content, not formatting. Whether validating development vs. QC outputs or checking final deliverables before submission, ELARA simplifies the process and ensures that no page, no label, and no detail is overlooked.

INTRODUCTION

In clinical programming, validation is not only about checking numbers or datasets. Every report must also look as expected with correct titles, footnotes, and layout. Almost every day, programmers face the same repetitive task: checking multiple outputs one by one, sometimes more than twenty reports per day. This takes a lot of time and makes it easy to miss small layout issues, especially when reports have many pages.

Most existing tools focus on comparing datasets or text content but do not address the **layout validation** part of the process. Small differences in alignment, page structure, or figure placement are often left to manual review. As a result, this step becomes one of the most time-consuming and inconsistent parts of validation.

To address this challenge, a tool was developed to automate these checks and highlight visual differences automatically. This concept led to the creation of **ELARA (Expert Layout Analysis and Reporting Assistant)** an R-based solution that compares reports both visually and textually using image recognition and OCR (Optical Character Recognition). This paper presents the motivation, development, and performance of ELARA and demonstrates how it enables programmers to review outputs more efficiently, with fewer manual steps.

WORKFLOW OVERVIEW

ELARA was built to support different types of validation needs that appear in daily programming work. Depending on the purpose, it can be used in two main modes which are Output-to-Output Comparison and Shell-to-Output Comparison. Both follow the same general flow: converting the reports into a consistent format, extracting text regions, identifying layout differences, and generating clear visual and tabular summaries. (Figure 1)

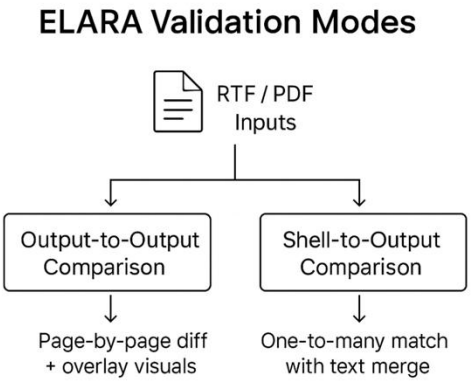

Figure 1 Workflow of ELARA

1. OUTPUT-TO-OUTPUT COMPARISON

This mode targets two fully generated reports. ELARA converts each page to a high-resolution image, extracts text via OCR (Optical Character Recognition), and segments the page into three regions title, main body, and footnote. The title and footnote are compared line-by-line, and results are written to a structured CSV. In parallel, ELARA produces visual diagnostics for the full page so reviewers can immediately see where changes occurred.
Two complementary visuals are generated per page.
The **overlay view** places the two versions on top of each other and highlights all pixel-level differences in red. This is a fast way to scan for any unexpected movement in labels, margins, or pagination.

The **three-panel** view shows the reference page, the difference layer, and the new page side by side; it is easier for reviewers to understand which version introduced a change and whether the change is confined to the intended area or also affects plots or footnotes. Together with the CSV of text mismatches these visuals provide both a quick visual sweep and a traceable record suitable for QC files.

In practice, Output-to-Output is effective when a delivery includes minor layout or text updates while the underlying numbers remain unchanged. For example, if a study title was revised between runs, the overlay will show concentrated red only in the title band, while the three-panel view confirms that figures and footnotes remain stable. When differences extend into the main body, the red overlay and per-page match flag immediately signal pages that require closer review.

Inputs and outputs.

ELARA accepts RTF or PDF as input. Inputs are automatically normalized to images to ensure consistent comparison. Outputs include:
(i) a text-mismatch CSV listing page, region (Title/Footnote), line pairs, and a Y/N mismatch flag;
(ii) per-page visuals (overlay and/or three-panel) that can be embedded in QC documentation;
(iii) an optional summary table indicating whether any visible differences were found and linking to the corresponding images.
ELARA's Output-to-Output comparison runs from a single function call and is designed to be accessible even for reviewers without R programming experience. The user provides a small set of paths that define what to compare and where to store results:

- **mode / view** — defines comparison type; here "OTO" (Output-to-Output) and "threepanel" to generate reference–diff–new composite images.
- **ref_path** – path to the reference (baseline) report, usually the *QC* or *previously validated* version.
- **new_path** – path to the new report being compared, such as the *Production* or *updated* version.
- **ref_png_dir** – directory where ELARA stores per-page PNG images converted from the reference report.
- **new_png_dir** – directory where per-page PNGs from the new report are saved.
- **out_text_diff_csv** – output CSV file summarizing detected text differences in the *title* and *footnote* regions.
- **diff_dir** – folder where visual difference images (overlay or three-panel) are saved.
- **dpi** – image resolution in *dots per inch*; higher values increase visual accuracy but may slow processing slightly.
- **prefix** – prefix applied when naming individual PNG pages (e.g., *"page_001.png"*).
- **overwrite_pngs** – logical flag; when set to TRUE, existing PNGs are overwritten during re-runs.
- **lang** – OCR language setting (typically "eng"); supports multilingual reports if needed.
- **gap_factor** – vertical spacing tolerance used when grouping nearby text lines into logical blocks (e.g., multi-line titles or multi-line footnotes).
- **title_max_y_frac** – defines the **upper boundary** of the page (as a fraction of total height) that will be treated as the *title region*.
- f**oot_min_y_frac** – defines the **starting boundary** of the *footnote region* (as a fraction of total height), extending **from this point down to the end of the page**.

```
elara_validation(
  mode = "OTO", view = "threepanel",
  ref_path   = "Figure1.rtf",
  new_path   = "Figure2.rtf",
  ref_png_dir = "png/ref",
  new_png_dir = "png/new",
  out_text_csv = "results2/elara_text_report.csv",
  diff_dir     = "results2/diff_images",
  dpi = 300, prefix = "page",
  overwrite_pngs = TRUE, lang = "eng",
  gap_factor = 2.0,
  title_max_y_frac = 0.058,
  foot_min_y_frac  = 0.90
)
```

Figure 2 Example of the ELARA call

Once these are defined, the process is fully automated. ELARA converts the reports into consistent image formats, detects title and footnote regions dynamically using text geometry, compares line-by-line content, and generates both visual diagnostics and structured CSVs.

Optional parameters allow control over the page-region thresholds (for unusually tall headers or deep footers), the image DPI (Dots Per Inch) (balancing quality and speed), and the difference sensitivity (how strongly minor visual shifts are highlighted). Default values are tuned to standard clinical-report layouts and typically require no modification.
Internally, ELARA integrates several R packages: **magick** for image rendering and alignment, tesseract for **OCR** (Optical Character Recognition) and **dplyr**, **readr**, **stringr**, and **purr** for data processing and output organization.
Once executed, the comparison produces three key deliverables

1. a text-mismatch CSV for titles and footnotes
2. a visual overlay highlighting all pixel-level differences,
3. a three-panel view for interpretability—making the process transparent, reproducible, and suitable for regulatory QC documentation.

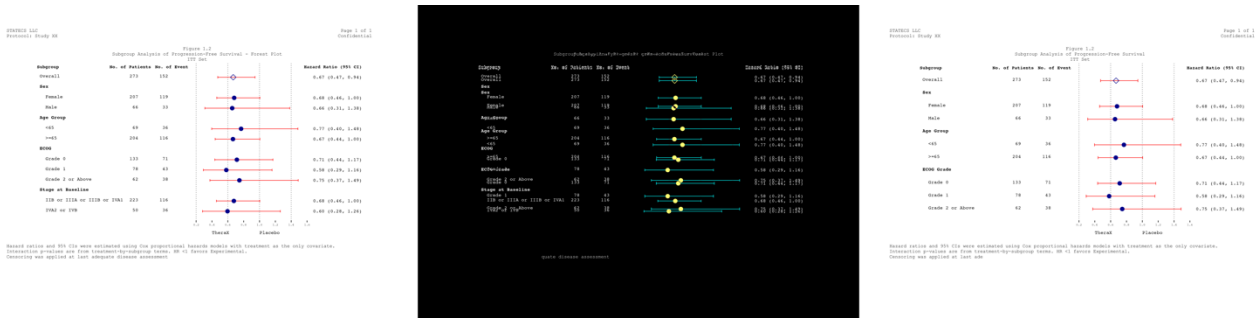

Figure 3 Three-Panel view

2. SHELL-TO-OUTPUT COMPARISON

In many cases, the report layout is defined in a mock shell that serves as a visual specification. However, shells and outputs do not always align page by page, a title might appear on one page, while its footnote continues another. To address this, ELARA compares one or several mock shell pages against all pages of the generated output.
In this mode:

- The mock shell is first cleaned to remove hyperlinks, tracked changes, and other formatting marks.
- Selected pages are cropped to focus only on the relevant regions such as titles and footnotes.
- The extracted text from these pages is combined and then compared with the output report to ensure both structure and content remain consistent.

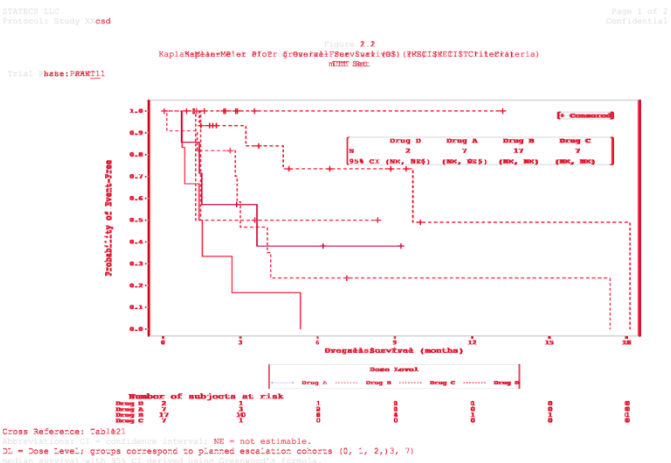

Figure 4 The overlay view

This approach allows ELARA to detect layout differences even when information is spread across pages, making the review process more accurate and less dependent on manual alignment.

The second mode of ELARA focuses on comparing a **mock shell** with a **generated output**. In many projects, mock shells act as the visual blueprint for how a final report should look defining titles, footnotes, and page structure long before any data are added. The goal here is not to compare pixel differences, but to make sure the final report truly follows that original design.

HOW IT WORKS

ELARA first cleans the mock shell file to remove formatting noise such as comments, hyperlinks, and tracked changes. Both the mock shell and output are then converted into a consistent image format and processed using OCR(Optical Character Recognition). From there, ELARA automatically detects the title and footnote regions of each page and compares the extracted text between the two.

Since mock shells and outputs often don't align perfectly page by page, ELARA performs a **one-to-many comparison** scanning each mock shell section across all output pages until it finds the best text match. This makes it possible to confirm that every key phrase or layout element from the mock shell appears in the final report, even if it shifted to another page.

WHAT REVIEWERS RECEIVE

Instead of visual overlays, Shell-to-Output produces a clean, structured **CSV report**. Each row shows a mock shell line, its corresponding match in the output, and a "Y" or "N" flag indicating whether the text is identical. The CSV also notes which page each match was found on, so reviewers can quickly locate it in the output file.

This makes the results easy to review for example, to confirm that titles, subtitles, and disclaimers were copied correctly from the approved mock shell. Even subtle differences such as spacing or punctuation are clearly highlighted, helping ensure visual and textual consistency.

WHEN TO USE IT

This mode is especially useful early in the validation process, when the layout and formatting are being finalized but the underlying data may still change. Reviewers can quickly check whether all structural components match the design specification, long before numerical results are ready for verification. It's also a great fit for confirming sponsor-specific formatting requirements such as capitalization rules, protocol references, or standardized footnotes.

HOW IT'S RUN
Shell-to-Output comparison is executed using the same elara_validation() framework, with a few additional inputs specific to shell files.
The user provides:

- **Mock shell file** (shell_doc_path) — the layout specification in RTF or PDF format.
- **Output report** (figure_doc_path) — the generated figure or report to be checked.
- **Shell and output PNG directories** (shell_png_dir, fig_png_dir) — where per-page images are created for consistent text extraction.
- **Shell page selection** (shell_pages) — allows comparing one or multiple shell pages to all output pages (useful when titles and footnotes span different pages).
- **Cropping and cleaning options** (crop_shell, clean_shell_first) — remove margins, hyperlinks, and tracked changes before analysis.
- **Band definitions** (title_*_frac, foot_*_frac) — specify the top and bottom portions of each page to be treated as title and footnote zones.
- **Output CSV path** (out_text_csv) — stores the merged line-by-line comparison between shell and output text.

Unlike Output-to-Output comparison, this mode **does not produce red overlay images**, as the shell serves only as a visual template. The result is a structured CSV that lists any mismatched titles or footnotes across pages.

Optional parameters can be used to fine-tune OCR(Optical Character Recognition). language, the detection sensitivity for title and footnote regions, or page height thresholds. The process is fully automated after that one function call runs the entire comparison, extracts the text, matches it, and writes the results to CSV.

1. A CSV table showing each mock shell line and its match in the output.
2. "Y"/"N" mismatch flags for every text line. **Y** → mismatch detected in text, formatting, or position (e.g., spacing or capitalization). **N** → exact match between shell and output.
3. Optional logs listing which pages each match was found on.

Together, these results let reviewers confirm that the report layout matches the approved mock shell design — clearly, reproducibly, and without needing to look at a single pixel.

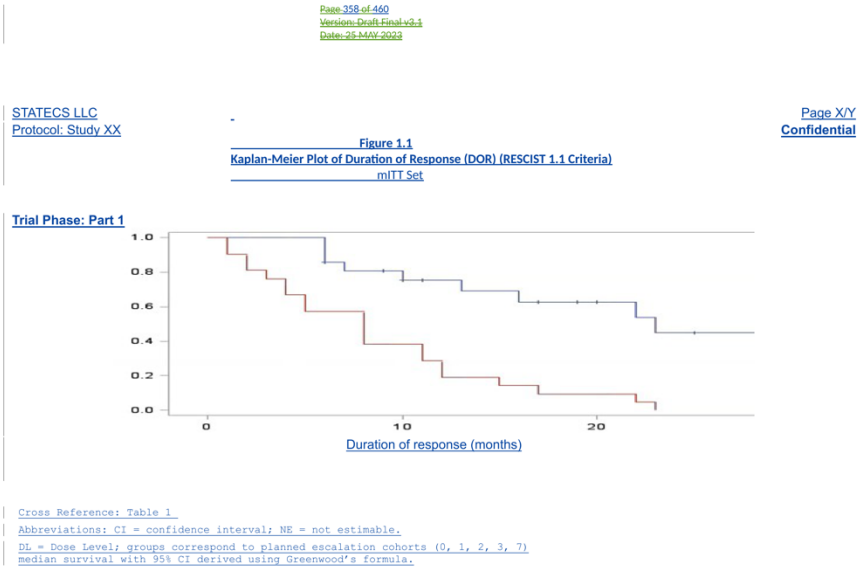

Figure 5

To illustrate how ELARA performs Shell-to-Output validation, Figure 5 shows a mock shell (top) alongside the generated output pages in Figure 6

The mock shell defines the expected layout : including title, trial phase label, and footnote structure while the final outputs are generated programmatically from analysis datasets.
Instead of using pixel overlays, ELARA extracts and aligns text from each section using OCR(Optical Character Recognition), comparing every mock shell line to its match in the output.

This process ensures that key elements such as **titles, protocol references, and footnotes** appear exactly as defined in the approved design, even when the number of pages or layout slightly differs.
All detected matches and mismatches are summarized in a structured CSV, allowing reviewers to confirm that the report adheres to the intended layout and wording before formal QC begins.

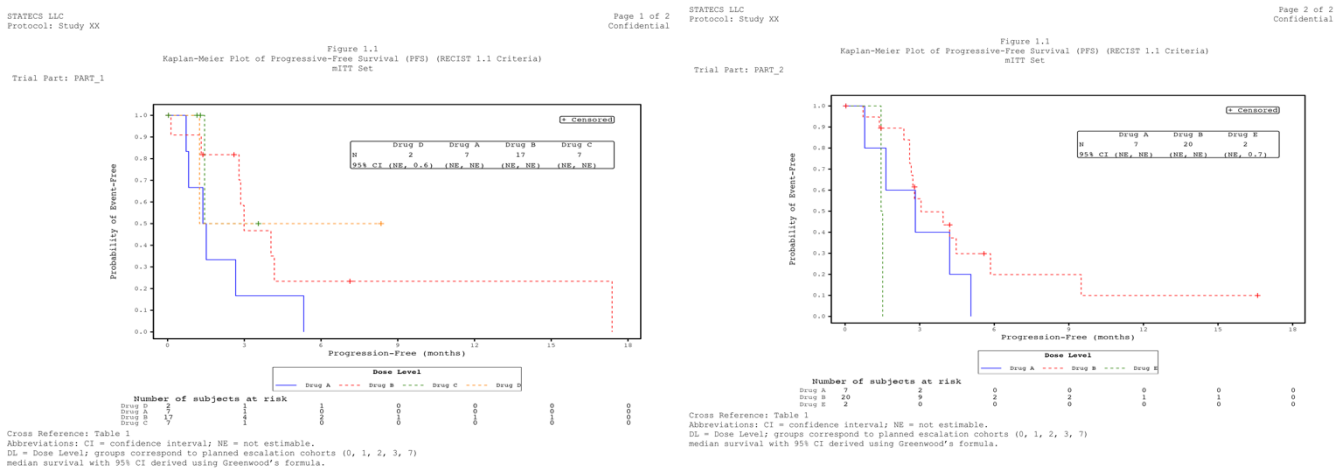

Figure 6

The results are summarized in a structured **CSV report**, which lists each text segment from the mock shell and its matched counterpart in the output.
Each record includes the section type (Title or Footnote), the mock shell text, the matched output line, the page number where the match was found, and a simple “Y/N” flag indicating whether they match exactly.
This tabular output serves as a transparent audit trail — reviewers can quickly confirm that all expected content appears in the final report, or focus immediately on lines marked as mismatched.
Together, the CSV and the corresponding visuals create a comprehensive, reproducible record of layout adherence.

FigurePage	Section	Line	Text_shell	Text_figure	Mismatch(Y/N)
1	TITLE_LINES	1	STATECS LLC . Page X/Y	STATECS LLC Page 1 of 2	Y
1	TITLE_LINES	2	Protocol: Study XX Confidential	Protocol: Study XX Confidential	N
1	TITLE_LINES	3	Figure 1.1	Figure 1.1	N
1	TITLE_LINES	4	Kaplan-Meier Plot of Duration of Response (DOR) (RESCIST 1.1 Criteria)	Kaplan-Meier Plot of Progressive-Free Survival (PFS) (RECIST 1.1 Criteria)	Y
1	TITLE_LINES	5	mITT Set	mITT Set	N
1	TITLE_LINES	6	Trial Phase: Part 1	Trial Part: PART 1	Y
1	FOOTNOTE	1	Cross Reference: Table 1	Cross Reference: Table 1	N
1	FOOTNOTE	2	Abbreviations: CI = confidence interval; NE = not estimable.	Abbreviations: CI = confidence interval; NE = not estimable.	N
1	FOOTNOTE	3	DL = Dose Level; groups correspond to planned escalation cohorts (0, 1, 2, 3, 7)	DL = Dose Level; groups correspond to planned escalation cohorts (0, 1, 2, 3, 7)	N
1	FOOTNOTE	4	median survival with 95% CI derived using Greenwood's formula.	median survival with 95% CI derived using Greenwood's formula.	N
2	TITLE_LINES	1	STATECS LLC . Page X/Y	STATECS LLC Page 2 of 2	Y
2	TITLE_LINES	2	Protocol: Study XX Confidential	Protocol: Study XX Confidential	N
2	TITLE_LINES	3	Figure 1.1	Figure 1.1	N
2	TITLE_LINES	4	Kaplan-Meier Plot of Duration of Response (DOR) (RESCIST 1.1 Criteria)	Kaplan-Meier Plot of Progressive-Free Survival (PFS) (RECIST 1.1 Criteria)	Y
2	TITLE_LINES	5	mITT Set	mITT Set	N
2	TITLE_LINES	6	Trial Phase: Part 1	Trial Part: PART 2	Y
2	FOOTNOTE	1	Cross Reference: Table 1	Cross Reference: Table 1	Y
2	FOOTNOTE	2	Abbreviations: CI = confidence interval; NE = not estimable.	Abbreviations: CI = confidence interval; NE = not estimable.	Y
2	FOOTNOTE	3	DL = Dose Level; groups correspond to planned escalation cohorts (0, 1, 2, 3, 7)	DL = Dose Level; groups correspond to planned escalation cohorts (0, 1, 2, 3, 7)	Y
2	FOOTNOTE	4	median survival with 95% CI derived using Greenwood's formula.	median survival with 95% CI derived using Greenwood's formula.	Y

Figure 7 Result of the validation

CONCLUSION

ELARA was designed to make validation faster, more consistent, and accessible to everyone involved in clinical reporting. What was once a slow, manual process checking every title, footnote, and layout by hand , can now be done in minutes, with clear visual and tabular summaries.

By supporting both **Output-to-Output** and **Shell-to-Output** comparisons, ELARA adapts to real-world needs: confirming consistency between report versions or verifying that a final output matches its mock shell. Its automated visuals, OCR-based text checks, and structured CSV summaries give reviewers an instant, transparent view of any differences.

One of ELARA's key advantages is simplicity it can be used not only by programmers but also by reviewers with no R background. With just one function call and a few input paths, the entire process runs automatically, producing results that are easy to interpret and share.

In practice, ELARA bridges the gap between technical validation and human review. It helps teams save time, reduce oversight, and maintain confidence that every report down to the smallest title or footnote meets its intended design. Looking ahead, its framework can easily expand to more complex figures and reports, continuing to make visual validation faster, smarter, and more inclusive.

REFERENCES

1. Wickham, H., François, R., Henry, L., & Müller, K. (2023). *dplyr: A Grammar of Data Manipulation*. R package version 1.1.4.
2. Ooms, J. (2024). *magick: Advanced Graphics and Image-Processing in R*. R package version 2.8.2.
3. Tesseract OCR. (2023). *Optical Character Recognition Engine*. https://github.com/tesseract-ocr/tesseract

CONTACT INFORMATION

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

Author Name: Mane Sargsyan
Company: STATECS LLC
Email: mane.sargsyan@statecs.com
Website: www.statecs.com