

reportifyr: R-based Open-Source Tool for Automated Reporting

Jacob Dumbleton¹, Matthew D. Smith¹, Agnes Kim¹, Fernando Carreno², Lia Liefwaard³, Tarjinder Sahota³, Maxwell Chirehwa³, Kashyap Patel⁴, Devin Pastoor¹, Theodoros Papathanasiou⁵
¹A2-Ai LLC, South State Street, Suite 301, Ann Arbor, MI, USA 48104; ²Clinical Pharmacology Modelling & Simulation, GSK, Upper Providence, PA, USA; ³Clinical Pharmacology Modelling & Simulation, GSK, Stevenage, UK; ⁴Clinical Pharmacology Modelling & Simulation & Ethnopharmacology, GSK, Australia; ⁵Clinical Pharmacology Modelling & Simulation, GSK, Baar, Switzerland

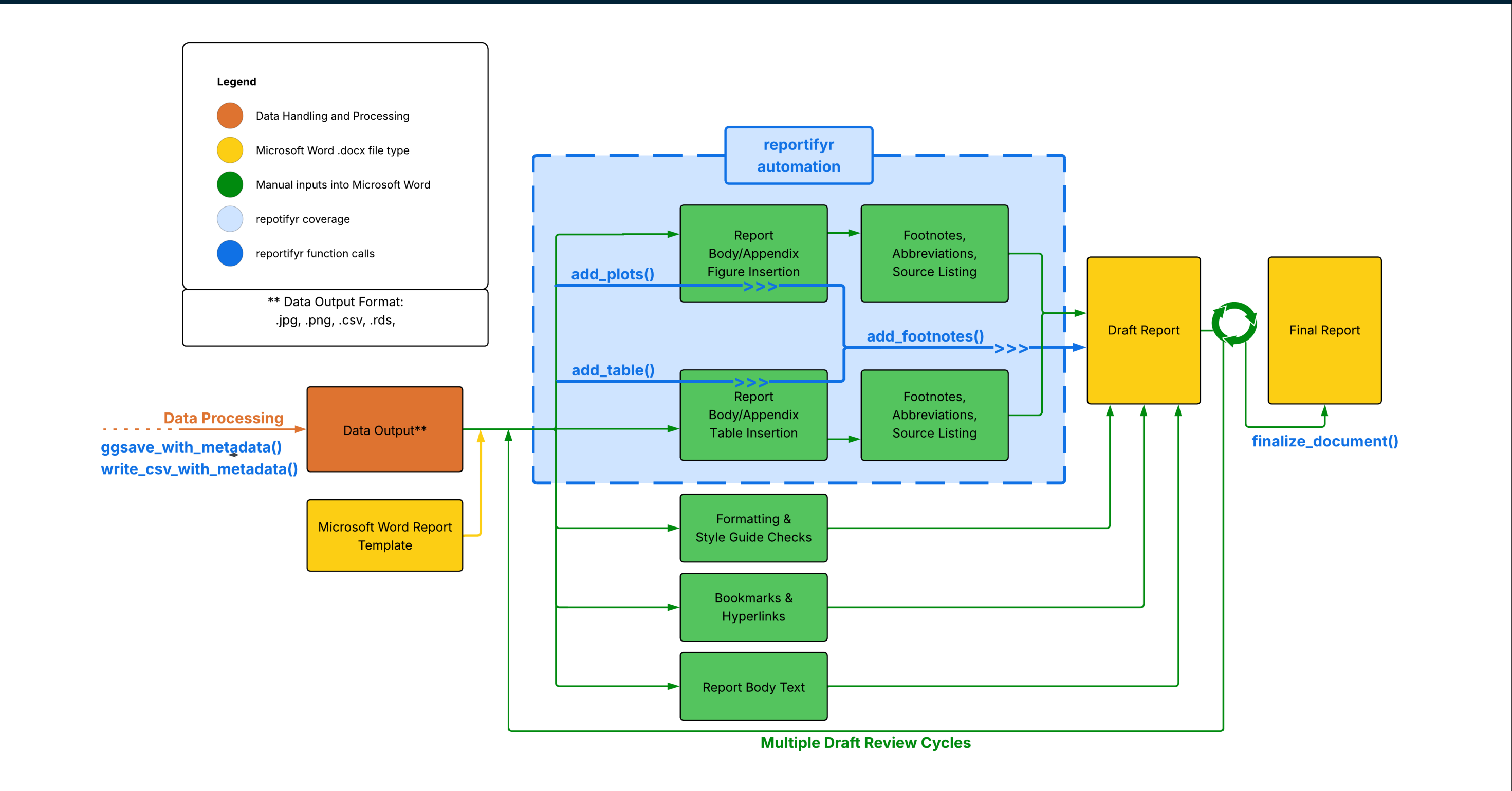
Background

- Generating structured, reproducible, and high-quality reports is a critical yet time-consuming task in data analysis and regulatory submissions.
- While R Markdown[1] and LaTeX-based[2] solutions offer automation for reporting in Microsoft Word documents, they rely on their own markup standard for input files, limiting the seamless iterative review cycles required in industry.

Objective

- reportifyr is an open-source R package created to streamline review and edit cycles in Microsoft Word documents by supporting iterative updates of figure and tables (artifacts) in accordance with pre-existing, complex company style guides.

Figure 1. Reporting Workflow



Note: Figure 1 depicts a standard reporting workflow post-data analysis/processing. Areas of coverage by reportifyr are depicted in blue demonstrating how iterative review cycles may be expedited with simple reportifyr function calls.

Setup

- The computational environment used consisted of an RStudio Pro Server (version 2024.09.0 - “Cranberry Hibiscus”), R (version 4.4.1), Python (version 3.12.4) and reportifyr (version 0.2.6) running under Ubuntu (version 22.04.5 LTS) on an AWS cluster.
- reportifyr automates the setup of the Python environment via a shell script that installs uv, which in turn manages Python and the required dependencies, including python-docx (v1.1.2) and PyYAML (v6.0.2).

Analysis – Artifact Capture

- Plots and tables are generated in R and are saved to the local file system using reportifyr's wrapper functions.

Save Plots	Save Tables
<pre>figures_path <- here("OUTPUTS", "figures") plot_file_name <- "theoph-pk-plot.png" ggsave_with_metadata(filename = file.path(figures_path, plot_file_name), plot = p, meta_type = meta_type\$`conc-time-trajectories`, meta_equations = NULL, meta_notes = NULL, meta_abbrevs = c("Cmax", "Tmax"), height = 4, width = 6)</pre>	<pre>tables_path <- here("OUTPUTS", "tables") table_file_name <- "theoph-pk-parameters.csv" write_csv_with_metadata(object = pk_params, file = file.path(tables_path, table_file_name), meta_type = meta_type\$`parameter-summary`, meta_equations = NULL, meta_notes = "AUC was calculated using linear trapezoidal integration for ascending and non- loggable segments and log-linear trapezoidal integration for descending segments.", meta_abbrevs = c(meta_abbrevs\$AUC, "Cmax", "Tmax"), table1_format = FALSE, row.names = FALSE)</pre>

Figure 2. Example of Report Template Setup

Figure 1: Observed Theophylline Concentration-time Profiles	Table 1: Pharmacokinetic Parameters of Theophylline
{rpfy}: theoph-pk-plot.png	{rpfy}:theoph-pk-parameters.csv

Note: Figure 2 illustrates the expected format of a magic string, within the Microsoft Word document, for both a figure and table.

Report Drafting – Report Generation

- Microsoft Word report templates, based on internal standards, were used as the foundation for this workflow. These templates can be customized by users to fit their specific formatting needs and should contain reportifyr's magic strings (Figure 2).
- Call reportifyr's wrapper function to insert figures, tables, and then footnotes into template document.

Finalize Document – Final Revision

- Once review and edit cycles are complete, the document can be finalized using reportifyr's final wrapper function.
- The wrapper function removes all reportifyr magic strings, effectively severing the link between the document and reportifyr.

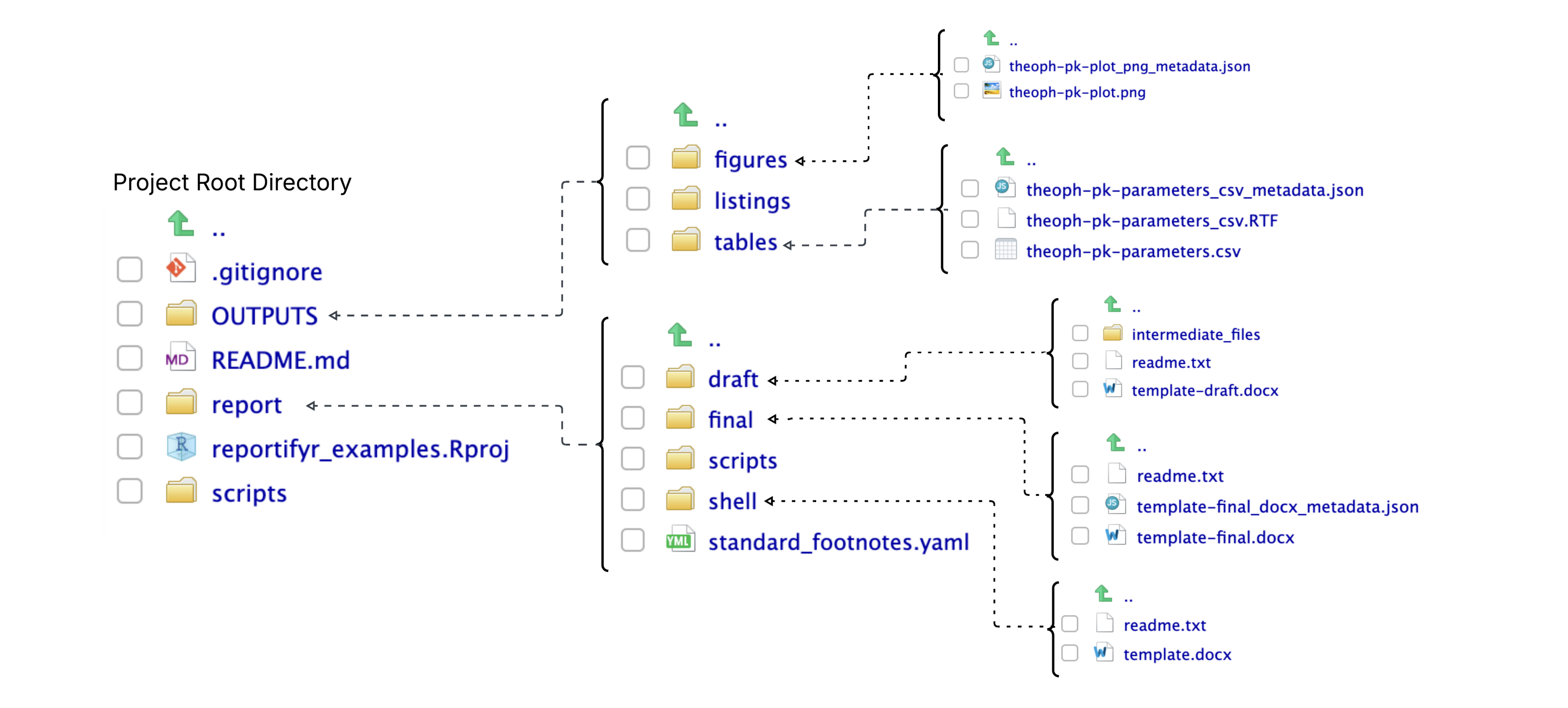
Report Drafting

```
build_report(
  docx_in = here("report", "shell",
    "template.docx"),
  docx_out = here("report", "draft", "template-
    draft.docx"),
  figures_path = here("OUTPUTS", "figures"),
  tables_path = here("OUTPUTS", "tables"),
  standard_footnotes_yaml = here("report",
    "standard_footnotes.yaml"),
  add_footnotes = TRUE,
  include_object_path = FALSE,
  footnotes_fail_on_missing_metadata = TRUE
)
```

Finalize Document

```
finalize_document(
  docx_in = here::here("report", "draft",
    "template-draft.docx"),
  docx_out = here::here("report", "final",
    "template-final.docx")
)
```

Figure 3. Folder Structure



Note: Figure 3 demonstrates the directory hierarchy established by reportifyr. Artifacts are saved to their relevant OUTPUTS subdirectory, while document versions are saved accordingly to drafts, final, or shell subdirectories.

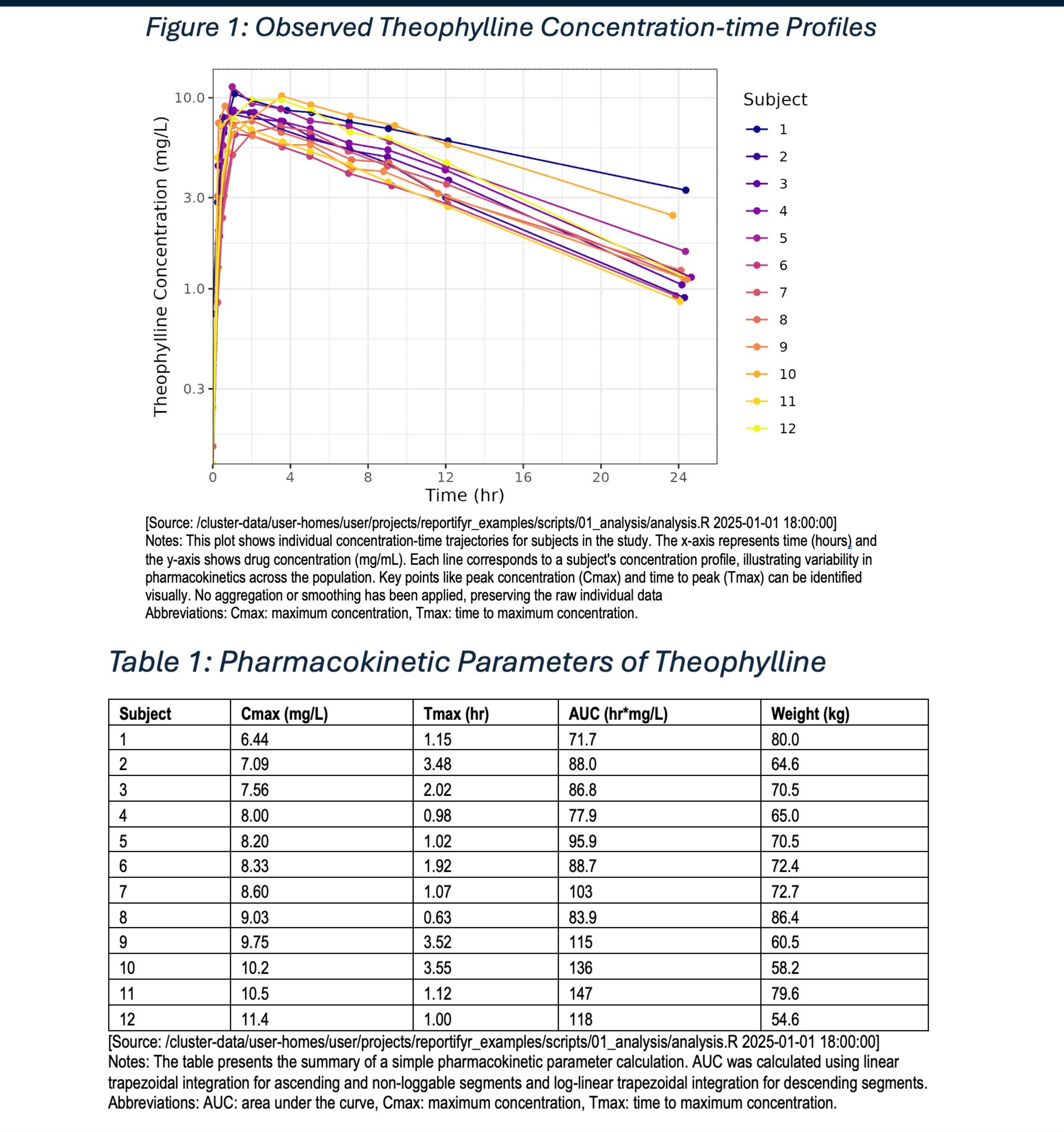
Figure 4. Figure Metadata

```
...
"source_meta": {
  "creation_author": "user <user@a2-ai.com>",
  "latest_author": "user <user@a2-ai.com>",
  "path": "/cluster-data/user-homes/user/projects/reportifyr_examples/scripts/01_analysis/analysis.R",
  "creation_time": "2025-01-01 18:00:00",
  "latest_time": "2025-01-01 18:30:00"
},
"object_meta": {
  "author": "user <user@a2-ai.com>",
  "path": "/cluster-data/user-homes/user/projects/reportifyr_examples/OUTPUTS/figures/theoph-pk-plot.png",
  "creation_time": "2025-01-01 18:30:00",
  "file_type": "png",
  "meta_type": "conc-time-trajectories",
  "hash": "66e89e9b383493e53fbf51417720dde079b10b66a05fd0eb86c8ac50dbc04f0",
  "table1": false,
  "footnotes": {
    "equations": [],
    "notes": [],
    "abbreviations": [
      "Cmax",
      "Tmax"
    ]
  }
}
...

```

Note: Figure 4 represents a direct chunk of the metadata .json file captured from “Analysis – Save Plots”.
... denotes additional lines of metadata omitted for visual representation.

Figure 5. Finalized Report



Note: Figure 5 depicts the finalized document following iterative review cycles. All artifacts and their corresponding footnotes have been inserted, while reportifyr magic strings have been removed. Note how the metadata from Figure 4 corresponds to the footnotes for “Observed Theophylline Concentration-time Profiles”.

Conclusions & Outlook

- reportifyr supports structured reports and automatic artifact replacements under a versatile framework, eliminating the need to convert document content into custom markup, while also enabling early drafting in organizational Microsoft Word templates to accelerate review cycles and streamline finalization after analysis.
- As part of its ongoing development, reportifyr is being integrated with Quarto[3], a next-generation document publishing system for initial document creation.
- Integration with Quarto introduces a hybrid workflow in which users generate an initial Microsoft Word draft via Quarto, while reportifyr facilitates structured updates throughout the review and revision process.
- By automating content updates within pre-existing Microsoft Word documents, this approach ensures that reports remain dynamic, editable, and reproducible while maintaining full compatibility with regulatory and collaborative workflows.



Disclosure: reportifyr was created, and is currently maintained by A2-Ai LLC, sponsored by GSK plc, and open-sourced for free public access and use. FC, LL, TS, MC, KP and TP are employees of, and shareholders of GSK.
References: [1]: <https://rmarkdown.rstudio.com/> [2]: Rasmussen CH, Smith MK, Ito K, Sundararajan V, Magnusson MO, Niclas Jonsson E, Fostvedt L, Burger P, McFadyen L, Tensfeldt TG, Nicholas T. PharmTeX: a LaTeX-Based Open-Source Platform for Automated Reporting Workflow. AAPS J. 2018 Mar 16;20(3):52. doi: 10.1208/s12248-018-0202-0. PMID: 29549459. [3]: Allaire, J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2024). Quarto (Version 1.6) [Computer software]. <https://doi.org/10.5281/zenodo.5960048>

