PharmaSUG China 2025 - Paper AA-168

AutoList: AI-Enhanced End-to-End Automation for Clinical Trial Listings Generation

Jiacheng Ni, Jiangsu Hengrui Pharmaceuticals Co., Ltd.

ABSTRACT

The automation of Tables, Figures, and Listings (TFL) remains a focal point in clinical programming. Among TFL components, listings hold relatively lower importance compared to tables and figures, featuring simpler structures. However, current macro-assisted listing generation processes still demands significant programmer effort in variable handling, formatting adjustments, and structural refinements, making listings a prime target for end-to-end automation.

This paper introduces AutoList, an innovative end-to-end automation tool designed to transform TFL shells directly into production-ready listings with minimal manual intervention. Leveraging a hybrid framework, AutoList integrates rule-based programming with cutting-edge large language models (LLMs) to address traditionally intractable challenges, such as Variable Identification & Derivation, Dynamic Formatting Optimization. In a pilot implementation, AutoList significantly reduced manual intervention. The tool's modular architecture introduced in this paper also allow further extension to Tables and Figures, establishing a scalable paradigm for AI-augmented automation in statistical programming.

INTRODUCTION

With increasing industry pressure to reduce costs and enhance efficiency, the automation of TFLs (Tables, Figures, and Listings) – the core deliverables of clinical programming – has become critically important. Achieving true end-to-end automation, capable of transforming TFL shells directly into production-ready outputs, represents the ultimate objective. In this paper, listings have been selected as the initial proof of concept for this end-to-end approach based on two considerations. First, the volume of listings required across clinical trial studies, often exceeding 40 in Phase II/III studies, presents a substantial burden despite their relatively lower analytical criticality compared to tables and figures. Second, listings are structurally more tractable. While tables and figures serve complex analytical needs demanding intricate statistical logic, listings primarily fulfil data presentation requirements and generally exhibit more consistent structural elements.

PREREQUISITES: FOUNDATIONAL STANDARDIZATION

The generation of any TFL necessitates a foundational three-step process. First, TFL specifications must be extracted from the TFL mock shell, capturing essential elements such as the TFL serial number, title, footnotes, and structural definitions (column headers, groupings). Second, an analysis dataset tailored to the TFL requirements must be constructed using the relevant source ADaM datasets. Finally, dedicated SAS macros utilize the specifications and the prepared dataset to generate both a final TFL dataset and a draft TFL output. This draft then undergoes formatting adjustments, including pagination, alignment, and presentation refinement, to achieve the final version.

As underscored by this process, realizing true end-to-end TFL automation fundamentally depends on the department-level implementation of highly standardized practices across these three critical elements.

- Shell Standardization: All TFLs in the mock shell are categorized by therapeutic area (e.g., oncology/non-oncology) and analysis type (e.g. efficacy/safety), assigned a Unique Identifier (UID), and structured to contain consistent metadata (including title, column headers, footnotes).
- ADaM Standardization: Key ADaM variables have standardized naming conventions and derivation logic.
- Macro Standardization: Template macros and batch execution tools enable programmatic TFL generation.

AUTOLIST WORKFLOW

Having established this essential standardization, the path towards automation requires systematically addressing each of these three key steps through dedicated automation solutions. AutoList is designed precisely for this purpose, executing a well-defined three-stage workflow.

STEP 1: SPECIFICATIONS EXTRACTION FROM MOCK SHELLS

Leveraging standardized structures of the study-level TFL shell, AutoList extracts critical listing specifications – including serial number, title, footnotes, column headers, and UID – with deterministic extraction algorithms. The standardization eliminates the need for manual retrieval, ensuring consistency and efficiency from the outset.

STEP 2: TFL DATASET GENERATION FROM ADAM SOURCES

This step, transforming source ADaM datasets into analysis-ready TFL datasets, presents the most significant automation complexity. Key challenges include the context-dependent logic required for tasks such as selecting the appropriate source ADaM datasets, defining population inclusions, and mapping variables, which demands semantic interpretation beyond simple rules. Furthermore, listings often require dynamic variable derivations not present in the source data (e.g., concatenated dates related variables like ASTDT and ASTDY to present “date/study day” column in listing).

Centralized JSON Metadata Repository

To address these challenges, we build a Centralized JSON Metadata Repository. Within this structured database, each standardized listing shell corresponds to a unique JSON object identified by its UID. The structure of an example object in this database is shown as below:

```
{
    "UID": "l_onc_mh_16_2_4_3",
    "Listing_No": "16.2.4.3",
    "title": {
          "CN": "受试者既往病史列表",
          "EN": "Medical History (ITT)"
      },
    "treatmentArea": "Oncology",
    "section": "16.2.4",
    "txtname": "l-mh",
    "sourceda": "ADMH",
    "columns":[
          {
                "variable": "SUBJID",
                "origin": "Predecessor",
                "derivation": "",
                "label_cn": ["受试者号", "受试者编号"],
                "label_en": ["Subject ID"]
          },
          {...
          },
          {
                "variable": "AENDT_DY",
                "origin": "Derived",
                "derivation": "if ^missing(AENDT) and ^missing(AENDY) then
AENDT_DY=put(AENDT, YYMMDD10.)||'/'||strip(put(AENDY, best.)); else if
missing(AENDT) and ^missing(AENDY) then AENDT_DY=put(AENDT, YYMMDD10.);
else if missing(AENDT) and missing(AENDY) and ^missing(MHENDTC) then
AENDT_DY=MHENDTC;",
                "label_cn": ["结束日期/研究天"],
```

```
                "label_en": ["End Date/Study Day"]
            }
        ]
    },
```

Listings in the shell are linked to specific metadata objects within the JSON Repository using the UID and column headers. This metadata identifies the source ADaM datasets, the program filenames, and variable-level details such as variable names, origins, and derivation codes where applicable. This metadata repository incorporates several key innovations:

- Fuzzy Matching: Accommodates variations in column headers names encountered across different studies.
- Predefined Derivations: Embeds standardized derivation logic for commonly required listing variables directly within the repository schema.
- LLM-Assisted Fallback Mechanisms: Columns that cannot be mapped deterministically through the repository trigger AI-assisted code suggestions generated by large language models (requiring programmer validation due to non-deterministic outputs).

Excel-based Listing specification

Once uploading the study-level TFL shell to the Autolist, it will generate a Excel-based specification file serving as the bridge between metadata repository and execution programs. This file comprises two primary sections. The very first Table of Contents (TOC) Sheet (Figure 1) indexes all listings, providing UIDs, titles, analysis populations, footnotes, and dataset sources. The following Listing-Specific Sheets (Figure 2) contain detailed variable-level controls, including column headsers, underlying variable names, origin (Predecessor/Derived), derivation code, and display settings such as sorting criteria, page break triggers, and column width allocation (typically equal width by default). Production programmers select the specific listings for generation in the TOC sheet and, if necessary, make adjustments to subset filtering conditions or derivations and display settings in the listing-specific sheets. AutoList then parse the edited specification file, generating the SAS program files in batches. These programs are then executed via batch utilities to produce draft listing outputs.

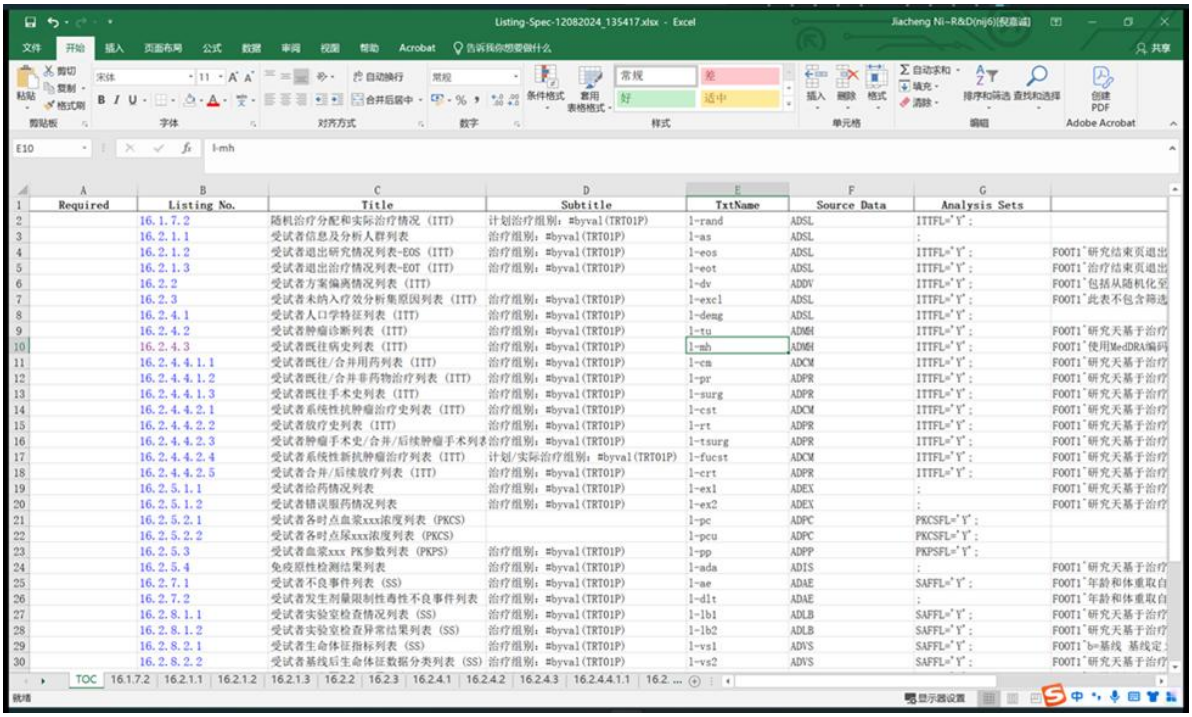

Figure 1. TOC Sheet of the Listing Specification File

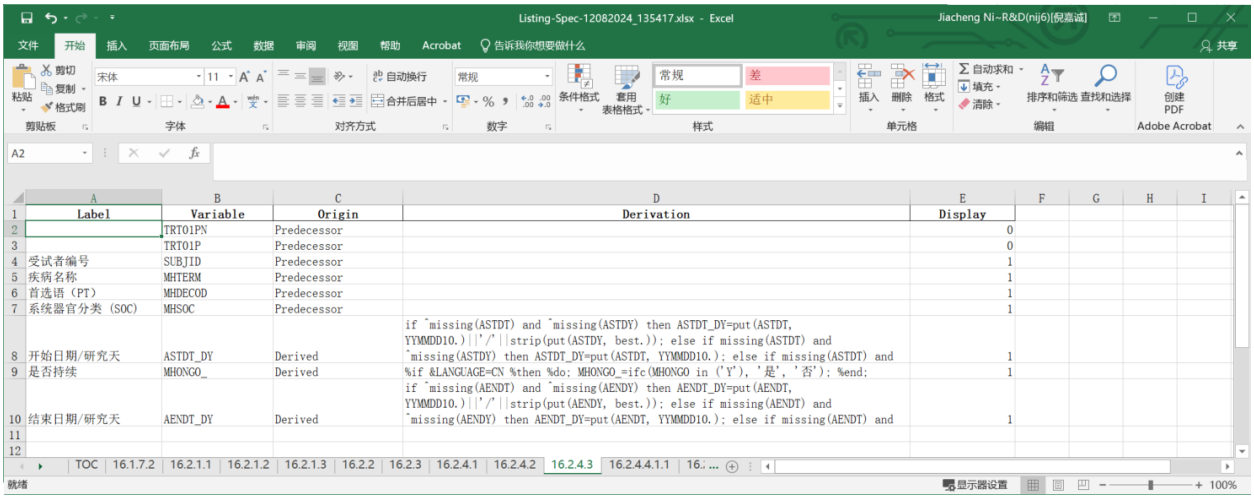

Figure 2. Listing-Specific Sheet of the Listing Specification File

STEP 3: AI-ENHANCED LAYOUT OPTIMIZATION

While the draft listings ensure content accuracy, the optimization of layout for readability– encompassing column wrapping, vertical alignment, spacing, and pagination – traditionally demands considerable manual effort. Programmatic solutions face inherent difficulties with the dynamic text rendering inherent in RTF outputs and context-dependent aesthetic considerations (e.g., justifying some columns while allowing others variable width). For this kind of problem without a fixed standard solution, AutoList proposes to integrate LLM-based formatting advisors. The AutoList utilizes a local-deployed DeepSeek R1 model to analyze the generated draft TFL dataset and the intended display requirements (from the specification file) to provide intelligent suggestions for alignment rules, spacing adjustments, and pagination control. An example of the AI model suggestion is shown below (Output 1). This AI-assisted approach aims to further reduce the formatting burden on programmers.

```
在 A4 横向纸张中，可用宽度计算如下：
- **A4 横向纸张宽度**：29.7 厘米
- **左右页边距**：2.54 厘米 × 2 = 5.08 厘米
- **可用宽度**：29.7 厘米 - 5.08 厘米 = **24.62 厘米**
字体为宋体，字号为 5 号（约 10.5 磅），根据内容长度和字体大小，以下是每一列的推荐宽度（以厘米为单位）：
---
### 列宽推荐
| 列名                    | 推荐宽度| 说明                          |
|-------------------------|---------|-------------------------------|
| 受试者号                | 2.2     | 内容较短（如“CN001001”）      |
| 系统器官分类(SOC)       | 4.5     | 内容较长（如“胃肠系统疾病”）  |
| 首选术语(PT)            | 3.0     | 内容中等（如“恶心*”）         |
| 不良事件名称            | 3.0     | 内容中等（如“恶心”）          |
| 药物名称和剂量#         | 5.0     | 内容较长（如“ABC-0001 片 30mg”）|
| 开始日期/研究天         | 3.0     | 内容较短（如“2024-06-07/84”） |
| 结束日期/研究天         | 3.0     | 内容较短（如“2024-09-20/189”）|
---
### 总宽度计算
2.2 + 4.5 + 3.0 + 3.0 + 5.0 + 3.0 + 3.0 = **23.7 厘米**
```

Output 1. Example Outputs of Local LLM model

In summary, the workflow of AutoList is illustrated in the figure below.

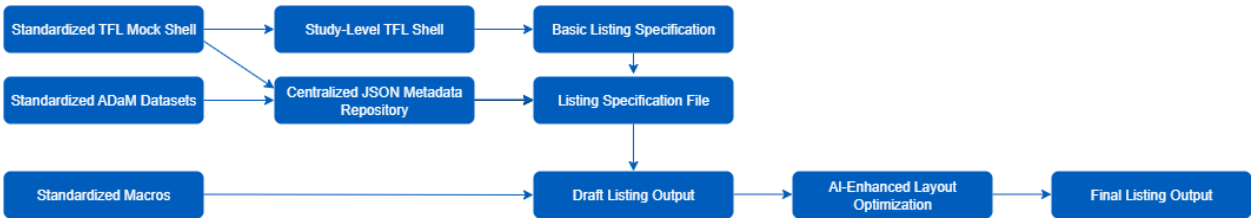

Figure 3. Workflow of Autolist

DISCUSSION

AutoList successfully demonstrates the technical feasibility and operational potential of end-to-end listings automation. However, sustained success hinges on effectively managing ongoing maintenance requirements and inherent limitations associated with the approach. Maintaining the integrity of the Centralized JSON Metadata Repository demands rigorous synchronization with evolving TFL shell standards and derivation logic. While LLM-assisted components significantly mitigate gaps in deriving non-standard variables and optimizing layout, their probabilistic nature inherently carries the risk of AI-generated errors ("hallucinations"). These errors may manifest as invalid SAS syntax in suggested derivation code or contextually suboptimal formatting rules. Consequently, careful programmer validation remains an indispensable element of the workflow.

A key strategic advantage of AutoList lies in its extensible, modular architecture. The core principles and infrastructure established for listings can be generalized to automate tables—components presenting greater analytical complexity but sharing common foundational elements like mock shells and ADaM datasets. Specifically, future iterations can leverage the same standardized shell metadata capturing critical analytical requirements unique to tables.

The development and implementation of AutoList underscore several critical principles for automation in clinical programming. First, robust, department-wide standardization of ADaM datasets and TFL shells is a non-negotiable prerequisite; automation effectively codifies established standards. Second, pragmatic AI integration is paramount. Large language models serve as powerful augmentative tools, enhancing efficiency and tackling ambiguity, but cannot replace domain expertise and programmer judgment. Automation solutions must therefore be designed to keep the programmer firmly in control, prioritizing tangible value creation over indiscriminate application of the technology ("AI for AI's sake").

CONCLUSION

AutoList establishes the feasibility of end-to-end listing automation through hybrid rule-based and AI-driven design. By significantly reducing manual effort, it demonstrates the potential for scalable AI adoption in clinical reporting. Its modular framework paves the way for fully automated TFL generation upon foundational standardization.

CONTACT INFORMATION

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

Jiacheng Ni
Jiangsu Hengrui Pharmaceuticals Co., Ltd.
18962166286
jiacheng.ni.jn6@hengrui.com