

# 1     SciTeX Writer: A Container-Based Framework for 2     Reproducible Scientific Manuscript Preparation

3             First Author<sup>a</sup>, Second Author<sup>b</sup>, Corresponding Author<sup>a,\*</sup>

4                     <sup>a</sup>*First Institution, Department, City, Country*

5                     <sup>b</sup>*Second Institution, Department, City, Country*

---

## 6     **Abstract**

7  
8     Scientific manuscript preparation requires careful management of doc-  
9     ument structure, version control, and reproducible compilation across di-  
10    verse computing environments. We present SciTeX Writer, a comprehensive  
11    LaTeX-based framework designed to streamline the academic writing work-  
12    flow while maintaining consistency and reproducibility. The system employs  
13    container-based compilation to ensure identical output regardless of the host  
14    environment, eliminating the common "it works on my machine" problem.  
15    Through a modular architecture that separates content from formatting, Sci-  
16    TeX Writer enables researchers to focus on scientific writing while the system  
17    handles document structure, figure format conversion, and version tracking.  
18    The framework supports parallel development of main manuscripts, supple-  
19    mentary materials, and revision documents, all sharing common metadata  
20    from a single source of truth. Automatic handling of diverse image formats  
21    and systematic organization of tables and figures reduces technical overhead.  
22    This self-documenting template demonstrates its own capabilities, providing  
23    researchers with a production-ready system for manuscript preparation that  
24    scales from initial draft to final submission.

25    *Keywords:* keyword one, keyword two, keyword three, keyword four,  
26    keyword five

---

27     ~ 2 figures, 0 tables, 157 words for abstract, and 2072 words for main

---

\*Corresponding author. Email: corresponding.author@institution.edu

28 text

## 29 1. Introduction

30 The preparation of scientific manuscripts involves numerous technical  
31 challenges that extend beyond the intellectual task of communicating re-  
32 search findings. Researchers must navigate complex typesetting systems,  
33 manage multiple document versions, coordinate figures and tables across for-  
34 mats, and ensure reproducible compilation environments. These technical  
35 burdens can distract from the primary goal of clear scientific communica-  
36 tion and often lead to inconsistencies, formatting errors, and wasted time  
37 troubleshooting environment-specific compilation issues.

38 Traditional approaches to manuscript preparation typically rely on local  
39 LaTeX installations, where the specific versions of packages and compilation  
40 tools can vary significantly across different machines and over time. This  
41 variability creates reproducibility challenges, particularly in collaborative en-  
42 vironments where multiple authors work on different systems. Furthermore,  
43 the proliferation of image formats and the need to convert between them  
44 for different submission requirements adds another layer of complexity. Re-  
45 searchers often resort to ad-hoc scripts or manual processes to handle these  
46 conversions, leading to potential errors and inconsistent results.

47 Existing solutions have addressed some aspects of this problem. Over-  
48 leaf and similar cloud-based platforms provide consistent compilation envi-  
49 ronments but require continuous internet connectivity and may not suit all  
50 research workflows. Version control systems like Git effectively track changes  
51 but require researchers to understand both LaTeX and version control simul-  
52 taneously. Template repositories exist for various journals, but they typically  
53 focus on formatting requirements rather than workflow automation and often  
54 duplicate common elements across documents.

55 The fundamental challenge lies in balancing flexibility with consistency.  
56 Researchers need systems that accommodate diverse content types, multi-  
57 ple output documents, and varying journal requirements while maintaining a

single source of truth for shared elements like author lists and bibliographies. The system must be sufficiently automated to reduce technical overhead yet transparent enough that researchers retain full control over their content. Additionally, the solution must work reliably across different computing environments without imposing steep learning curves or workflow disruptions.

SciTeX Writer addresses these challenges through a container-based, modular architecture that separates content management from document compilation. The framework organizes manuscripts into distinct directories for main text, supplementary materials, and revision responses, while maintaining shared metadata in a common location. By leveraging containerization technology, the system guarantees identical compilation results regardless of the host operating system or local software versions. Automatic format conversion for figures and tables eliminates manual preprocessing steps, and built-in version tracking with difference generation facilitates collaborative writing and revision processes. This manuscript serves as a self-documenting example, demonstrating the system’s capabilities through its own structure and compilation.

## 2. Methods

The SciTeX Writer framework implements a modular architecture designed around three core principles: reproducible compilation, content-structure separation, and automated asset management. The system organizes documents into three primary directories, each serving distinct purposes in the manuscript lifecycle while sharing common resources to maintain consistency.

### *2.1. Repository Structure and Organization*

The framework employs a hierarchical directory structure where the `00_shared/` directory serves as the single source of truth for metadata including title, author information, keywords, and bibliographic references. This centralized approach eliminates duplication and ensures consistency across all output documents. The `01_manuscript/` directory contains the main manuscript

87 with subdirectories for content sections, figures, and tables. Similarly, `02_supplementary/`  
88 follows an identical structure for supplementary materials, while `03_revision/`  
89 organizes revision letters by reviewer. Each content section exists as an inde-  
90 pendent LaTeX file, facilitating modular development and enabling multiple  
91 authors to work on different sections simultaneously without merge conflicts.

## 92 *2.2. Container-Based Compilation System*

93 To ensure reproducible builds across diverse computing environments, the  
94 framework leverages both Docker and Singularity containerization technolo-  
95 gies. The compilation environment encapsulates specific versions of TeX  
96 Live and all required packages, eliminating dependency on the host sys-  
97 tem’s LaTeX installation. Users invoke compilation through a simple Make-  
98 file interface that abstracts the container complexity. The command `make`  
99 `manuscript` compiles the main document, while `make all` processes all three  
100 document types in parallel. This containerized approach guarantees that  
101 the same source files produce identical PDFs regardless of the underlying  
102 operating system, making the system equally functional on Linux, macOS,  
103 Windows, and high-performance computing clusters.

## 104 *2.3. Automated Asset Processing*

105 The system implements automatic format conversion for both figures  
106 and tables through preprocessing scripts that execute during compilation.  
107 For figures, the framework accepts common image formats including PNG,  
108 JPEG, SVG, and PDF, automatically converting them to formats optimized  
109 for LaTeX inclusion. Each figure resides in its own subdirectory within  
110 `01_manuscript/contents/figures/caption_and_media/`, with the caption  
111 defined in a corresponding `.tex` file. During compilation, a preprocessing  
112 script scans these directories, generates figure inclusion code, and compiles  
113 all figures into `FINAL.tex` for inclusion in the main document. Tables fol-  
114 low an analogous structure, allowing authors to define complex table layouts  
115 separately from their incorporation into the document flow.

## 116 2.4. Version Control and Difference Tracking

117 The framework integrates with Git to provide systematic version track-  
118 ing and automatic generation of difference documents. When authors cre-  
119 ate a new version through `make archive`, the system archives the current  
120 manuscript with a timestamp and version number. Subsequently, invoking  
121 `make diff` generates a PDF highlighting changes between versions using  
122 the `latexdiff` utility. This functionality proves particularly valuable during  
123 revision processes, where journals often require marked-up versions show-  
124 ing modifications. The revision directory structure accommodates multiple  
125 rounds of review, with separate subdirectories for editor and reviewer re-  
126 sponses, each containing both the original comments and author responses  
127 in a structured format that ensures complete documentation of the revision  
128 process.

## 129 3. Results

130 The SciTeX Writer framework successfully demonstrates comprehensive  
131 manuscript preparation capabilities through its modular design and auto-  
132 mated workflows. This section presents the key features and functionalities  
133 that the system provides to researchers.

### 134 3.1. Cross-Platform Reproducibility

135 The containerized compilation system achieves complete reproducibility  
136 across different operating systems and computing environments. Testing  
137 across Linux distributions, macOS, and Windows Subsystem for Linux con-  
138 firmed that identical source files produce byte-for-byte identical PDF outputs  
139 when compiled using the same container image. This reproducibility extends  
140 to high-performance computing environments where Singularity containers  
141 enable compilation on systems without Docker support. The elimination of  
142 environment-dependent compilation issues represents a significant improve-  
143 ment over traditional local LaTeX installations, where package version mis-  
144 matches frequently cause inconsistent outputs or compilation failures.

### 145 3.2. Automated Figure and Table Management

146 The automatic asset processing system effectively handles diverse input  
147 formats and streamlines figure incorporation. Figure 1 demonstrates the  
148 framework’s capability to include images with properly formatted captions,  
149 while Figure 2 shows how multiple figures can be managed systematically.  
150 The preprocessing pipeline converts source images to optimal formats, main-  
151 taining quality while ensuring compatibility with LaTeX compilation require-  
152 ments. For tables, the system provides structured organization as shown in  
153 Table ??, where complex tabular data can be defined independently and au-  
154 tomatically integrated into the document flow. This separation of content  
155 from presentation enables authors to focus on data rather than formatting  
156 syntax.

### 157 3.3. Modular Content Organization

158 The framework’s modular structure facilitates collaborative writing by  
159 isolating different manuscript components into separate files. Each section,  
160 from the introduction through the discussion, exists as an independent La-  
161 TeX file that can be edited without affecting other sections. This organiza-  
162 tion minimizes merge conflicts in version control systems and allows multiple  
163 authors to work simultaneously on different parts of the manuscript. The  
164 shared metadata system ensures that changes to author lists, affiliations, or  
165 keywords propagate automatically across the main manuscript, supplemen-  
166 tary materials, and revision documents without requiring manual updates in  
167 multiple locations.

### 168 3.4. Version Tracking and Difference Generation

169 The integrated version control system maintains a complete history of  
170 manuscript evolution through the archive mechanism. Each archived version  
171 receives a timestamp and sequential version number, creating a clear audit  
172 trail of document development. The automatic difference generation pro-  
173 duces professionally formatted PDFs highlighting textual changes between

174 versions, using color coding to indicate additions and deletions. This func-  
175 tionality proves particularly valuable during peer review, where revision let-  
176 ters must clearly document modifications made in response to reviewer com-  
177 ments. The system handles this process automatically, requiring only simple  
178 Makefile commands rather than manual execution of `latexdiff` with complex  
179 parameters.

## 180 4. Discussion

181 The SciTeX Writer framework addresses fundamental challenges in scien-  
182 tific manuscript preparation by combining containerized compilation, modu-  
183 lar organization, and automated asset management into a cohesive workflow.  
184 The system demonstrates that technical infrastructure for manuscript writing  
185 can be both powerful and accessible, reducing friction in the research com-  
186 munication process while maintaining the flexibility and control that LaTeX  
187 provides.

### 188 4.1. *Advantages of the Containerized Approach*

189 The container-based compilation system represents a significant depart-  
190 ure from traditional LaTeX workflows and offers substantial practical ben-  
191 efits. By encapsulating the entire compilation environment, the framework  
192 eliminates the common scenario where manuscripts compile successfully on  
193 one author’s machine but fail on collaborators’ systems due to package ver-  
194 sion differences. This reproducibility becomes increasingly important as re-  
195 search teams become more distributed and as long-term document mainte-  
196 nance requires compilation environments to remain stable over years. The  
197 approach also reduces the barrier to entry for researchers new to LaTeX,  
198 as they need not navigate the complexities of installing and configuring a  
199 local TeX distribution. The dual support for Docker and Singularity en-  
200 sures compatibility across institutional computing environments, from per-  
201 sonal workstations to high-performance computing clusters where Docker  
202 may be unavailable for security reasons.

## 203 4.2. *Implications for Collaborative Writing*

204 The modular architecture facilitates collaborative workflows in ways that  
205 traditional monolithic LaTeX documents cannot. By separating content into  
206 individual files for each section and maintaining shared metadata in a central  
207 location, the system minimizes merge conflicts that plague collaborative  
208 document editing. Multiple authors can simultaneously work on different  
209 sections, commit their changes independently, and merge updates without  
210 the conflicts that arise when editing a single large file. The automatic propagation  
211 of metadata changes across multiple output documents ensures consistency  
212 without requiring authors to remember to update information in multiple  
213 locations. This design aligns well with modern software development  
214 practices adapted for scientific writing, where version control and modular  
215 design have become essential for managing complexity.

## 216 4.3. *Comparison with Existing Solutions*

217 Compared to cloud-based platforms like Overleaf, SciTeX Writer offers  
218 greater control over the compilation environment and eliminates dependency  
219 on internet connectivity, which can be crucial for researchers working in  
220 bandwidth-limited environments or on sensitive projects requiring air-gapped  
221 systems. Unlike simple template repositories, the framework provides active  
222 workflow automation through Makefiles and preprocessing scripts rather  
223 than merely offering formatting guidelines. The system complements rather  
224 than replaces Git-based workflows, adding a layer of manuscript-specific tooling  
225 while maintaining compatibility with standard version control practices.  
226 Where other solutions address individual aspects of the manuscript preparation  
227 challenge, SciTeX Writer integrates multiple components into a unified  
228 system.

## 229 4.4. *Limitations and Considerations*

230 The framework requires users to have basic familiarity with command-line  
231 interfaces and Makefiles, which may present a learning curve for researchers  
232 accustomed to graphical editing environments. While the system



233 automates many aspects of document preparation, it remains a LaTeX-based  
234 solution and therefore inherits both the power and complexity of the under-  
235 lying typesetting system. The containerization approach requires Docker or  
236 Singularity installation, adding a dependency that, while increasingly com-  
237 mon in research computing environments, may not be universally available.  
238 The framework is optimized for scientific articles following conventional IM-  
239 RAD structure and may require adaptation for other document types such  
240 as books or technical reports. Future development could address these lim-  
241 itations through optional graphical interfaces, expanded documentation for  
242 LaTeX newcomers, and templates adapted for diverse document formats.

#### 243 *4.5. Future Directions and Extensibility*

244 The modular design of SciTeX Writer enables natural extension points  
245 for additional functionality. Integration with continuous integration systems  
246 could enable automatic compilation and validation of manuscripts upon each  
247 commit, catching formatting errors early in the writing process. Support  
248 for additional output formats beyond PDF, such as HTML for web-based  
249 preprint servers, could be achieved through integration with tools like pan-  
250 doc. The preprocessing scripts could be extended to handle additional asset  
251 types or to perform automated quality checks on figures and tables. The  
252 system could also incorporate automated journal formatting through inte-  
253 gration with journal-specific style files, reducing the effort required to adapt  
254 manuscripts for different submission targets. As the research community  
255 continues to develop tools for reproducible research, SciTeX Writer provides  
256 a foundation that can incorporate emerging best practices while maintaining  
257 backward compatibility with existing manuscripts.

#### 258 *4.6. Conclusions*

259 SciTeX Writer demonstrates that scientific manuscript preparation can be  
260 systematized without sacrificing flexibility or imposing rigid constraints on  
261 content. By addressing reproducibility, modularity, and automation through

262 a unified framework, the system reduces technical overhead and allows re-  
263 searchers to focus on the intellectual work of communicating their findings.  
264 The self-documenting nature of this template provides both an example of  
265 the system’s capabilities and a starting point for new manuscripts. As re-  
266 search communication continues to evolve, frameworks like SciTeX Writer  
267 that prioritize reproducibility and collaborative workflows will become in-  
268 creasingly valuable for maintaining the quality and accessibility of scientific  
269 literature.

## 270 **Data Availability Statement**

271 The NeuroVista dataset used in this study is publicly available through  
272 the International Epilepsy Electrophysiology Portal (IEEG.org) at <https://www.ieeg.org>. Access requires registration and approval for research pur-  
273 poses.

275 The processed PAC databases and analysis code are available at <https://github.com/ywatanabe1989/neurovista>. GPU-accelerated PAC calcu-  
276 lation code is available as a standalone Python package ‘gpac’ at <https://github.com/ywatanabe1989/gPAC>. The SciTeX Python utilities used for re-  
277 producible computing is available at <https://github.com/ywatanabe1989/SciTeX>.

281 For questions regarding data access or analysis procedures, please contact  
282 the corresponding author.

## 283 **Ethics Declarations**

284 All study participants provided their written informed consent ...

## 285 **Author Contributions**

286 Y.W., T.Y., and D.G. conceptualized the study ...

287 **Acknowledgments**

288 This research was funded by funding bodies here

289 **Declaration of Interests**

290 The authors declare that they have no competing interests.

291 **Declaration of Generative AI in Scientific Writing**

292 The authors employed large language models such as Claude (Anthropic  
293 Inc.) for code development and complementing manuscript’s English lan-  
294 guage quality. After incorporating suggested improvements, the authors  
295 meticulously revised the content. Ultimate responsibility for the final content  
296 of this publication rests entirely with the authors.

297 **Tables**

298 **Tables**

**Table 1 – Table 0: Placeholder**

To add tables to your manuscript:

1. Place CSV files in `caption_and_media/` with format `XX_description.csv`
2. Create matching caption files `XX_description.tex`
3. Reference in text using `Table~\ref{tab:XX_description}`

Example: `01_seizure_count.csv` with `01_seizure_count.tex`

---

Step	Instructions
1. Add CSV	Place file like <code>01_data.csv</code> in <code>caption_and_media/</code>
2. Add Caption	Create <code>01_data.tex</code> with table caption
3. Compile	Run <code>./compile -m</code> to process tables
4. Reference	Use <code>\ref{tab:01_data}</code> in manuscript

---

## FIGURE PLACEHOLDER

Figure ID: 01\_example\_figure

Place your image file at:

[01\\_manuscript/contents/figures/caption\\_and\\_media/01\\_example\\_figure.\[png|tif|jpg|svg|mmd|pptx\]](#)

Supported formats: PNG, TIF/TIFF, JPG/JPEG, SVG, MMD, PPTX

Then run: `./compile_manuscript`

Reference in LaTeX as:

`Figure~\ref{fig:1_example_figure}`

This placeholder will be automatically replaced  
when you add the actual image file.

**Figure 1** – Example figure caption. This is a template showing how to include figures in your manuscript. Replace this text with a descriptive caption that explains what the figure shows. Include panel labels (A, B, C) if using multi-panel figures. Explain abbreviations and symbols used in the figure. Provide sufficient detail that readers can understand the figure without referring to the main text.

## FIGURE PLACEHOLDER

Figure ID: 02\_another\_example

Place your image file at:

[01\\_manuscript/contents/figures/caption\\_and\\_media/02\\_another\\_example.\[png|tif|jpg|svg|mmd|p](#)

Supported formats: PNG, TIF/TIFF, JPG/JPEG, SVG, MMD, PPTX

Then run: `./compile_manuscript`

Reference in LaTeX as:

`Figure~\ref{fig:2_another_example}`

This placeholder will be automatically replaced  
when you add the actual image file.

**Figure 2** – Another example figure. Use this template to add additional figures to your manuscript. Each figure should be placed in a separate .tex file in this directory. The compilation system will automatically process and include these figures in your manuscript.