Six Misunderstandings about Scholarly Markdown

Martin Fenner, Gobbledygook

March 3, 2014

In this post I want to talk about some of the misunderstandings I frequently encounter when discussing markdown as a format for authoring scholarly documents.

Scholars will always use Microsoft Word

Microsoft Word is of course what almost all authors use in the life sciences and many other disciplines. One big reason for this is the file formats accepted my manuscript submission systems. By limiting the options to Microsoft Word (and maybe LaTeX), you make it impossible for authors to use other tools, even if they wanted to. Publishers should accept manuscripts in any reasonable file format, as I have argued before.

My markup language is better than markdown

There are of course numerous alternatives to markdown, including Textile, AsciiDoc, MediaWiki Markup and reStructuredText. There will always be features that are better implemented in one of these languages, but I don't think there is room for more than one major initiative for a scholarly markup language. And markdown has the right mix of features and broad support by tools and the community.

Related to this there is the argument against markdown that the format is a mess and that there are too many versions (or flavors) of it. While that is certainly a big problem with markdown, I would argue that with Pandoc we have a nice standard and reference implementation for Scholarly Markdown. Pandoc is constantly evolving, and the addition of support for arbitrary YAML metadata was the biggest new feature in 2013 for me.

Scholarly Markdown is too complex and we might as well use LaTeX

"LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation" (The LaTeX Team, 2014). Although LaTeX has solved many of the problems Scholarly Markdown tries to tackle a long time ago, it is still something else. LaTeX at its core is a typesetting system, which is not something Scholarly Markdown cares about for two reasons: a) the focus is on authoring documents, which are then submitted to other systems at publishers and elsewhere that are specialized in producing the final document, and b) the focus is on HTML and the web as this is where we want most of the interactions with scholarly documents to take place. This means that

- Markdown is a great input format to convert into other formats, including XML (see for example my pandoc-jats).
- LaTeX will always be the best choice for some content, e.g. documents rich in mathematical formulas
- If the ultimate goal was to produce high-quality PDF documents, Scholarly Markdown would be a bad choice. It is the right format for HTML and the related ePub.

We have to be very careful that we keep the right balance of simplicity and features in Scholarly Markdown. This means that sometimes we should just include the LaTeX code, e.g. for math.

Scientists need a WYSIWYG Editor, and then the file format doesn't matter

WYSIWYG - What You See Is What You Get - is a user interface metaphor that is both a blessing and a curse. We desperately need better writing tools, and this of course also means user interfaces that help with that task. But the focus on creating a new authoring environment that focusses too much on WYSIWYG creates several problems:

- WYSIWYG is not always a good metaphor for scholarly documents. Typographic features such as fonts, line spacing, etc. are not something that belong into an authoring environment - this is done during the publishing step, as is the formatting of references according to a specific citation style.
- WYSIWYG is for human interactions, but content in scholarly documents is increasingly created by computers. Two good examples are statistics and figures created in R/knitr or iPython Notebook. Scholarly Markdown works perfectly with these workflows.
- WYSIWYG authoring environments run the high risk of vendor lock-in. This is understandable if you run a startup and want to promote your tool, but is not in the best interest of the scholarly community.

Version control via git is central to Scholarly Markdown, and this can also be challenging for a WYSIWYG environment. But there are many good examples of how to make this work.

Scientists should submit their manuscripts in JATS XML, the standard format for scholarly documents

At the end of the day most scholarly publications in the life sciences are converted into JATS XML. Unfortunately central aspects of the format (e.g. the required document structure or required attributes) are difficult to enforce in an authoring environment. Even if you build a tool that can nicely handle this, I'm not so sure we want to burden an author with this, especially since the manuscript will usually undergo a lot of changes before it is accepted and then published.

The future is HTML

Although the future for consuming scholarly documents is clearly HTML (and ePub), and there are great HTML editors, I'm not so sure that HTML will become the default for authoring environments. This is the reason why markdown and related markup languages were invented, and even with modern WYSIWYG editors working directly with HTML is not always the best choice. HTML has two problems: a) it is not as human-readable as markdown and therefore requires an additional layer for authoring, and b) it is not as structured as XML, which makes it difficult to create some of the rigid document structure required for scholarly documents. O'Reilly is trying to get more structure into HTML for print and digital books with HTMLBook, but with too much structure you might run into similar problems for authoring as discussed above for JATS XML. And of course you can include HTML in markdown documents.

References

The LaTeX Team. (2014). LaTeX - a document preparation system. http://www.latex-project.org/. Retrieved from http://www.latex-project.org/