Formatting Open Science: agile creation of multiple document types by writing academic manuscripts in pandoc markdown

Albert Krewinkel[1] **and Robert Winkler**[2,⋆]

Affiliations: ¹ TBD (Pandoc Development Team), ² CINVESTAV Unidad Irapuato, Department of Biochemistry and Biotechnology, Laboratory of Biochemical and Instrumental Analysis, Km. 9.6 Libramiento Norte Carr. Irapuato-León, 36821 Irapuato, Gto. Mexico

Correspondence: Prof. Dr. Robert Winkler, `robert.winkler@cinvestav.mx`

Keywords: open science, document formats, markdown, latex, publishing, typesetting

ABSTRACT

The timely publication of scientific results is essential for dynamic advances in science. The ubiquitous availability of computers which are connected to a global network made the rapid and low-cost distribution of information through electronic channels possible. New concepts, such as Open Access publishing and preprint servers are currently changing the traditional print media business towards a community-driven peer production. However, the cost of scientific literature generation, which is either charged to readers, authors or sponsors, is still high. The main active participants in the authoring and evaluation of scientific manuscripts are volunteers, and the cost for online publishing infrastructure is close to negligible. A major time and cost factor though is the formatting of manuscripts in the production stage. In this article we demonstrate the feasibility to write scientific manuscripts in plain markdown (MD) text files, which can be easily converted into common publication formats, such as PDF, HTML or EPUB, using pandoc. The simple syntax of markdown assures the long-term readability of raw files and the development of software and workflows. We show the implementation of typical elements of scientific manuscripts – formulas, tables, code blocks and citations – and present tools for editing, collaborative writing and version control. We give an example on how to prepare a manuscript with distinct output formats, a DOCX file for submission to a journal and a LATEX/PDF version for deposition as a PeerJ preprint. Reducing the work spent on manuscript formatting translates directly to time and cost savings for writers, publishers, readers and sponsors. Therefore, the adoption of the MD format contributes to the agile production of open science literature.

INTRODUCTION

Agile development of science depends on the continuous exchange of information between the researchers (Woelfle, Olliaro & Todd, 2011). In the past, physical copies of scientific works had to be produced and distributed. Therefore, publishers needed to invest considerable economical resources for typesetting and printing. Since the journals were mainly financed by their subscribers, their editors not only had to decide on the scientific quality of a submitted manuscript, but also on the potential interest for their readers. The availability of globally connected computers enabled the rapid exchange of information at low cost. Yochai Benkler (2006) predicts important changes in the information production economy, which are based on three observations:

1. A nonmarket motivation in areas such as education, arts, science, politics and theology.
2. The actual rise of nonmarket production, made possible through networked individuals and coordinate effects.
3. The emergence of large-scale peer production, e.g. of software and encyclopaedias.

Immaterial goods such as knowledge and culture are not lost, when consumed or shared – they are 'non-rival' –, and they enable a networked information economy, which is not commercially driven (Benkler, 2006).

Preprints and e-prints

In some areas of science already existed a preprint culture, i.e. a paper-based exchange system of research ideas and results, when Paul Ginsparg in 1991 initiated a server for the distribution of electronic preprints – 'e-prints' – about high-energy particle theory at the Los Alamos National Laboratory (LANL), USA (Ginsparg, 1994). Later, the LANL server moved with Ginsparg to Cornell University, USA, and was renamed to arXiv (Butler, 2001). Currently, arXiv (`https://arxiv.org/`) publishes e-prints related to physics, mathematics, computer science, quantitative biology quantitative finance and statistics. Just a few years after the start of the first preprint servers, their important contribution to scientific communication was evident (Ginsparg, 1994; Youngen, 1998; Brown, 2001). In 2014, arXiv reached the impressive number of 1 million e-prints (Van Noorden, 2014). In more conservative areas, such as chemistry and biology, accepting the publishing prior peer-review took more time (Brown, 2003). A preprint server for life sciences (`http://biorxiv.org/`) was launched by the Cold Spring Habor Laboratory, USA, in 2013 (Callaway, 2013). *PeerJ preprints* (`https://peerj.com/preprints/`), started in the same year, accepts manuscripts from biological sciences, medical sciences, health sciences and computer sciences. The terms 'preprints' and 'e-prints' are used synonymously, since the physical distribution of preprints has become obsolete. A major drawback of preprint publishing are the sometimes restrictive policies of scientific publishers. The SHERPA/RoMEO project informs about copyright policies and self-archiving options of individual publishers (`http://www.sherpa.ac.uk/romeo/`).

Open Access

The term *'Open Access'* was introduced 2002 by the Budapest Open Access Initiative and was defined as:

"Barrier-free access to online works and other resources. OA literature is digital, online, free of charge (gratis OA), and free of needless copyright and licensing restrictions (libre OA)." (Suber, 2012)

Frustrated by the difficulty to access even digitalized scientific literature, three scientists founded the *Public Library of Science (PLoS)*. In 2003, *PLoS Biology* was published as the first fully Open Access (OA) journal for biology (Brown, Eisen & Varmus, 2003; Eisen, 2003). Thanks to the great success of OA publishing, many conventional print publishers now offer a so-called 'Open Access option', i.e. to make accepted articles free to read for an additional payment. The copyright in this hybrid models might remain with the publisher, whilst fully OA usually provide a liberal license, such as the Creative Commons Attribution 4.0 International (CC BY 4.0, `https://creativecommons.org/licenses/by/4.0/`). OA literature is only one component of a more general *open* philosophy, which also includes the access to scholarships, software, and data (Willinsky, 2005). Interestingly, there are several different 'schools' of thinking on how to understand and define *Open Science*, as well the position that any science is open by definition, because of its objective to make generated knowledge public (Fecher & Friesike, 2014).

Cost of journal article production

In a recent study, the article processing charges (APCs) for research intensive universities in the USA and Canada were estimated to be about 1,800 USD for fully OA journals and 3,000 USD for hybrid OA journals (Solomon & Björk, 2016). PeerJ (`https://peerj.com/`), an OA journal for biological and computer sciences launched 2013, drastically reduced the publishing cost and offers its members a life-time publishing plan for a small registration fee (Van Noorden, 2012); alternatively the authors can choose to pay an APC of 1,095 USD, which may be cheaper, if multiple co-authors participate. Examples such as the *Journal of Statistical Software* (*JSS*, `https://www.jstatsoft.org/`) and *eLife* (`https://elifesciences.org/`) demonstrate the possibility of completely community-supported OA publications. **Fig. 1** compares the APCs of different OA publishing business models. *JSS* and *eLife* are peer-reviewed and indexed by Thomson Reuters. Both journals are located in the Q1 quality quartile in all their registered subject categories of the Scimago Journal & Country Rank (`http://www.scimagojr.com/`), demonstrating that high-quality publications can be produced without charging the scientific authors or readers.

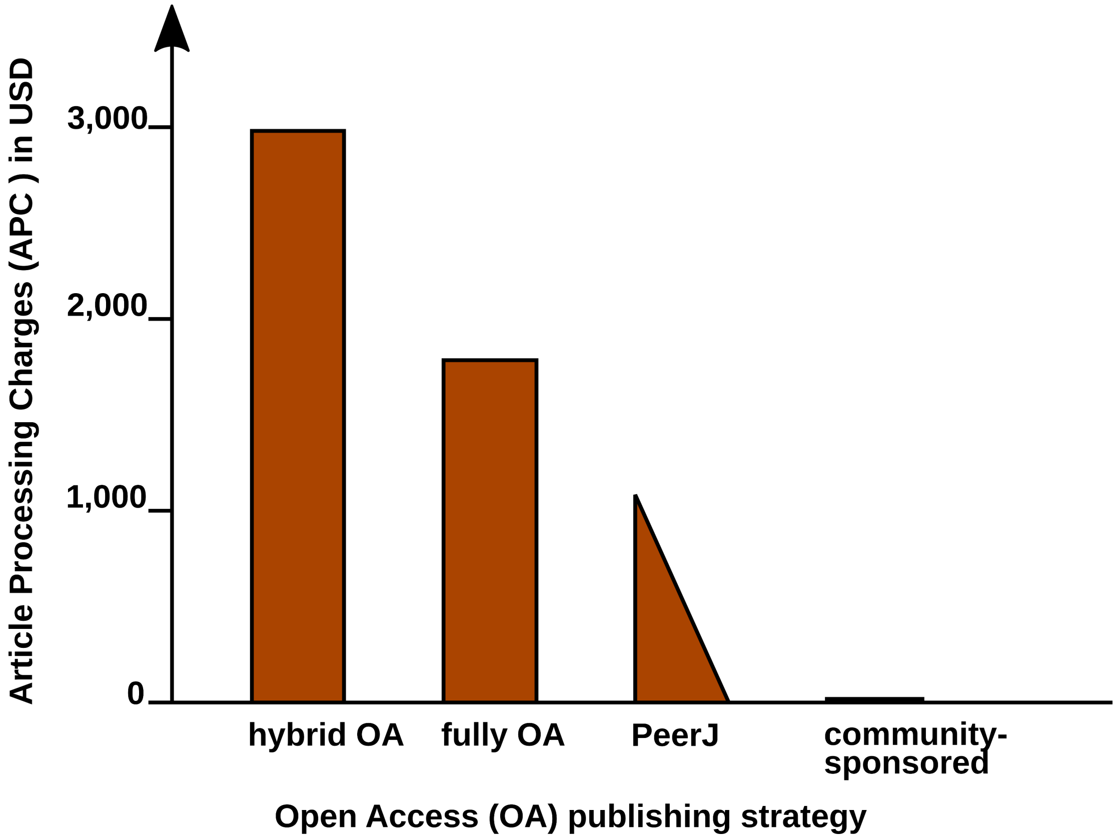

Figure 1. Article Processing Charge (APCs) that authors have to pay for with different Open Access (OA) publishing models. Data from (Solomon & Björk, 2016) and journal webpages.

In 2009, a study was carried concerning the *"Economic Implications of Alternative Scholarly Publishing Models"*, which demonstrates an overall societal benefit by using OA publishing model (Houghton et al., 2009). In the same report, the real publication costs are evaluated. The relative costs of an article for the publisher are represented in **Fig. 2**.

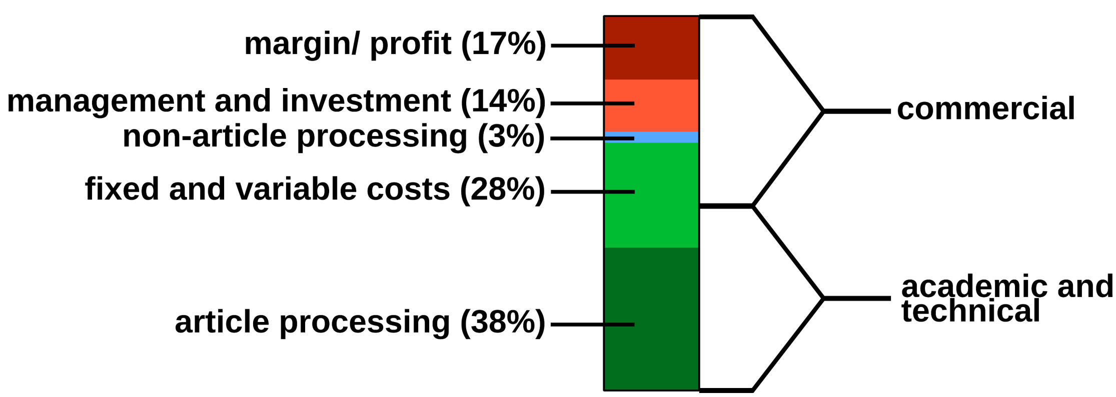

Figure 2. Estimated publishing cost for a 'hybrid' journal (conventional with Open Access option). Data from (Houghton et al., 2009).

Conventional publishers justify their high subscription or APC prices with the added value, e.g. journalism (stated in the graphics as 'non-article processing'). But also stakeholder profits, which could be as high as 50%, must be considered, and are withdraw from the science budget (Van Noorden, 2013). Generally, the production costs of an article could be roughly divided into commercial and academic/ technical costs (**Fig. 2**). For nonmarket production, the commercial costs such as margins/ profits, management etc. can be drastically reduced. Hardware and services for hosting an editorial system, such as Open Journal Systems of the Public Knowledge Project (`https://pkp.sfu.ca/ojs/`) can be provided by public institutions. Employed scholars can perform editor and reviewer activities without additional cost for the journals. Nevertheless, 'article processing', which includes the manuscript handling during peer review and production represents the most expensive part. Therefore, we investigated a strategy for the efficient formatting of scientific manuscripts.

Current standard publishing formats

Generally speaking, a scientific manuscript is composed from contents and formatting. Whilst the content, i.e. text, figures, tables, citations etc., may remain the same between different publishing forms and journal styles, the formatting can be very different. Most publishers require the formatting of submitted manuscripts in a certain format. Ignoring this **Guide for Authors**, e.g. by submitting a manuscript with a different reference style, gives a negative impression with a journal's editorial staff. Too carelessly prepared manuscripts can even provoke a straight 'desk-reject' (Volmer & Stokes, 2016). Currently DOC(X), LATEX and/ or PDF file formats are the most frequently used for journal submission platforms. But even if the content of a submitted manuscript might be accepted during the peer review 'as is' (very rare), the format still needs to be adjusted to the particular publication style in the production stage. For the electronic distribution of scientific works, which is gaining more and more importance, additional formats (EPUB, (X)HTML) need to be generated. **Tab. 1** lists the file formats which are currently most relevant for scientific publishing.

Table 1. Current standard formats for scientific publishing.

Type	Description	Use	Syntax	Reference
DOCX	Office Open XML	WYSIWYG editing	XML, ZIP	(Ngo, 2006)
ODT	OpenDocument	WYSIWYG editing	XML, ZIP	(Brauer et al., 2005)
PDF	portable document	print replacement	PDF	(International Organization for Standardization, 2013)
EPUB	electronic publishing	ebooks	HTML5, ZIP	(Eikebrokk, Dahl & Kessel, 2014)
LATEX	typesetting system	high-quality print	TEX	(Lamport, 1994)
HTML	hypertext markup	websites	(X)HTML	(Raggett et al., 1999; Hickson et al., 2014)
MD	Markdown	lightweight markup	plain text MD	(Ovadia, 2014; Leonard, 2016)

Although be content elements of the documents such as title, author, abstract, text, figures, tables, etc. remain the same, the syntax of the file formats is rather different. **Tab. 2** demonstrates some simple examples of differeces in different markup languages.

Table 2. Examples for formatting elements and their implementations in different markup languages types.

Element	Markdown	LATEX	HTML
structure			
section	`# Intro`	`\section{Intro}`	`<h1><Intro></h1>`
subsection	`## History`	`\subsection {History}`	`<h2><History></h2>`
text style			
bold	`**text**`	`\textbf{text}`	`<b>text</b>**`

Element	Markdown	LATEX	HTML
italics	`*text*`	`\textit{text}`	`<i>text</i>`
links			
http link	`<https://arxiv.org/>`	`\usepackage{url}` `\url{https://arxiv.org/}`	`<a href="https://arxiv.org/"></a>`

Documents with the commonly used Office Open XML (DOCX Microsoft Word files) and OpenDocument (ODT LibreOffice) file formats can be opened in a standard text editor after unzipping. However, content and formatting information is distributed into various folders and files. Practically speaking, those file formats require the use of special word processing software. From a writer's perspective, the use of *What You See Is What You Get (WYSIWYG)* programs such as Microsoft Word, WPS Office or LibreOffice might be convinient, because the formatting of the document is directly visible. But the complicated syntax specifications often result in problems when using different versions, in collaborative writing, and simple conversions between file formats can be difficult or impossible. In worst case, 'old' files cannot be opened any more. In some parts of the scientific community therefore LATEX, a typesetting program in plain text format, is very popular. With LATEX, documents with highest typographic quality can be produced. However, the source files are cluttered with LATEX commands and the source text can be complicated to read. Compilation errors in LATEX are sometimes difficult to find. Therefore, LATEX is not very user friendly, especially for casual writers or beginners. In academic publishing, additionally the creation of different output formats from the same source text is desirable:

- For the publishing of a book, with a print version in PDF and an electronic version in EPUB.
- For distributing of a seminar script, with an online version in HTML and a print version in PDF.
- For submitting a journal manuscript for peer-review in DOCX, as well as a pre-print version with another journal style in PDF.

Some of the task can be performed e.g. with LATEX, but an integrated solution remains a challenge. Several programs for the conversion between documents formats exist, such as the e-book library program calibre `https://code.google.com/archive/p/faenza-icon-theme/`. But the results of such conversions are often not satisfactory and require substancial manual corrections. Therefore, we were looking for a solution, which enables the creation of scientific manuscripts in a simple format, and the subsequent generation of multiple output formats. The need for hybrid publishing has been recognized outside of science (Kielhorn, 2011; DPT Collective, 2015), but the requirements specific to scientific publishing have not been addressed so far. Therefore, we investigated the possibility to generate multiple publication formats from a simple manuscript source file.

CONCEPTS OF MARKDOWN AND PANDOC

Markdown was originally developed by John Gruber in collaboration with Aaron Swartz, with the goal to simplify the writing of HTML documents `http://daringfireball.net/projects/markdown/`. Instead of coding a file in HTML syntax, the content of a document is written in plain text and annotated with simple tags which define the formatting. Subsequently, this markdown (MD) file are parsed to generate the final HTML document. With this concept, the source file remains easily readable and the author can focus on the contents rather than formatting. Despite its original focus on the web, the MD format has been proven to be well suited for academic writing (Ovadia, 2014). In particular, pandoc MD (`http://pandoc.org/`) adds several extensions which facilitate the authoring of academic documents and their conversion into multiple output formats. **Tab. 2** demonstrates the simplicity of MD compared to other markup languages. **Fig. 3** illustrates the generation of various formatted documents from a manuscript in pandoc MD. Some relevant functions for scientific texts are explained below in more detail.

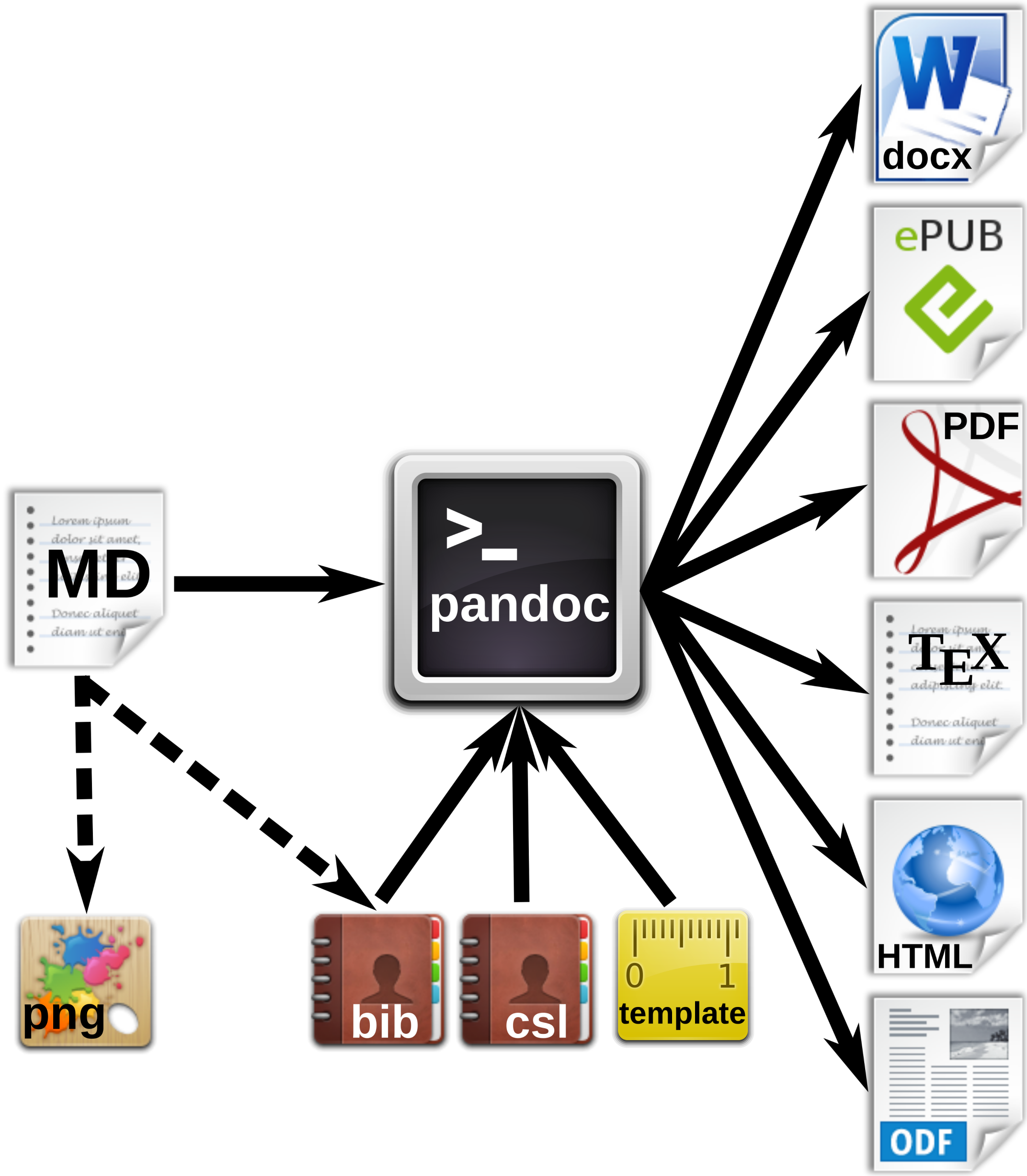

Figure 3. Workfow for the generation of multiple document formats with pandoc.

MARKDOWN EDITORS AND ONLINE EDITING

For the end user, the convenience of work with text, either writing alone or with several co-authors is important. Therefore, in this section we present software and strategies for different scenarios. **Fig. 4** summarized various options for local or networked editing of MD files.

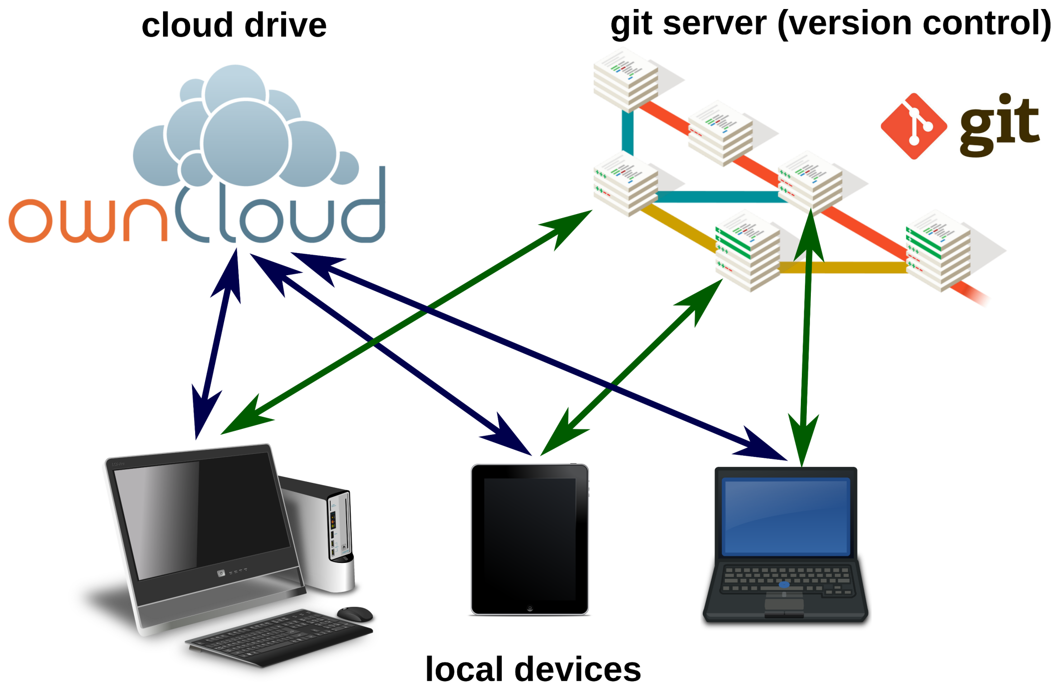

Figure 4. Markdown files can be edited on local devices or on cloud drives. A local or remote git repository enables advanced advanced version control.

Markdown editors

Because of the simple MD syntax, basically any text editor is suitable for editing markdown files. The formatting tags are written in plain text and easy to remember. Therefore, the author is not distracted by looking around for layout options with the mouse. For several popular text editors, such as vim (`http://www.vim.org/`), GNU Emacs (`https://www.gnu.org/software/emacs/`), atom (`https://atom.io/`) or geany (`http://www.geany.org/`), plugins provide additional functionality for markdown editing, e.g. syntax highlighting, command helpers, live preview or structure browsing. Also various dedicated mardown editors have been published. Many of those are cross-platform compatible, such as Abricotine (`http://abricotine.brrd.fr/`), Ghostview (`https://github.com/wereturtle/ghostwriter`) and CuteMarkEd (`https://cloose.github.io/CuteMarkEd/`). The lightweight format is also ideal for writing on mobile devices. Numerous applications are available on the App stors for Android and iOS systems. The programs Swype and Dragon (`http://www.nuance.com/`) facilitate the input of text on such devices by guessing words from gestures and speach recognition (dictation). **Fig. 5.** shows the editing of a text with the markdown editor CuteMarkEd.

Figure 5. Editing window, HTML preview and table of contents using the CuteMarkEd editor.

Online editing and collaborative writing

Storing manuscripts on network drives (*The Cloud*) has become popular because of several reasons: Protection against data loss, synchronization of documents between several devices and collaborative editing options. Markdown files on a Google Drive (`https://drive.google.com`) for instance can be edited online with StackEdit (`https://stackedit.io`). m can be used for editing markdown files . **Fig. 6** demonstrates the online editing of a markdown file on an OwnCloud (`https://owncloud.com/`) installation, using a plugin.

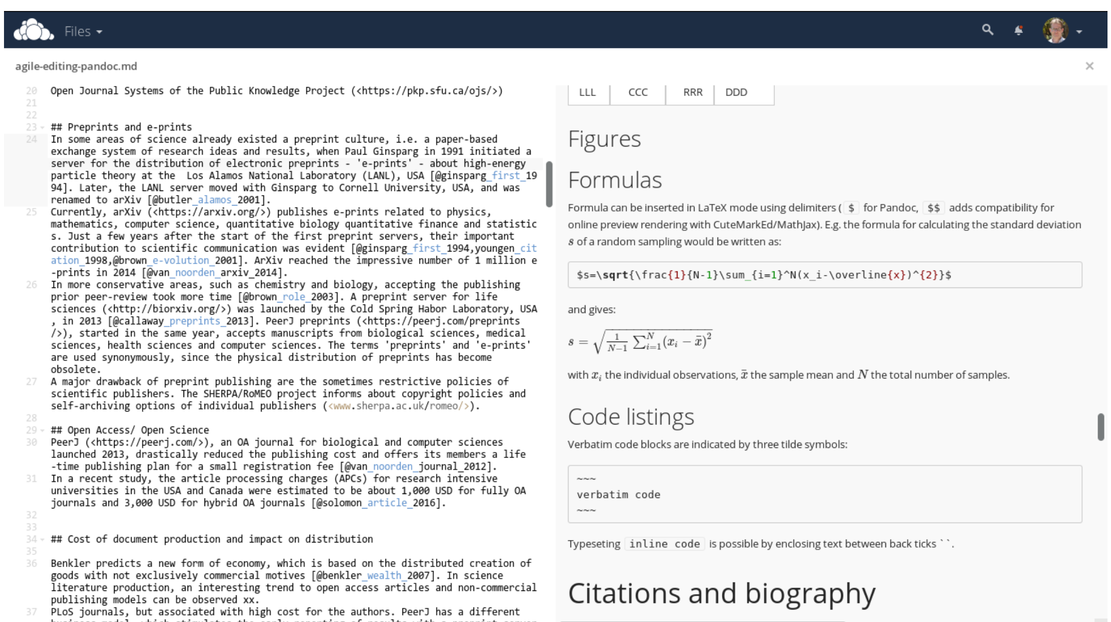

Figure 6. Direct online editing of this manuscript with live preview using the ownCloud Markdown Editor plugin by Robin Appelman.

Even formalas are rendered correctly in the HTML live preview window of the OwnCloud markdown plugin (**Fig. 6**).

Document versioning and change control

Programmers, especially when working in distributed teams, rely on version control systems to manage changes of code. Currently, Git (`https://git-scm.com/`), which is also used e.g. for the development of the Linux kernel, is one of the most employed software solutions for versioning. Git allows the parallel work of collaborators and has an efficient merging and conflict resolution system. A Git respository may be used from a single local author to keep track of changes, or by a team with a remote repository, e.g. on github (`https://github.com/`) or bitbucket (`https://bitbucket.org/`).

Figure 7. Version control and collaborative editing using a git repository on bitbucket.

For the writing of the present article, the co-authors (Germany and Mexico) used a remote Git repository on bitbucket. The plain text syntax of markdown facilitates the visualization of differences of document versions, as shown in **Fig. 7**.

PANDOC MARKDOWN FOR SCIENTIFIC TEXTS

Following, the potential of typesetting scientific manuscripts with pandoc is demonstrated with examples for typical document elements, such as tables, figures, formulas, code listings and references. A brief introduction is given by (Dominici, 2014). The complete Pandoc User's Manual is available at `http://pandoc.org/MANUAL.html`.

Tables

There are several options to write tables in markdown. The most flexible alternative - which was also used for this article - are pipe tables. The contents of different cells are separated by pipe symbols (`|`):

```
Left | Center | Right | Default
:-----|:-------:|------:|---------
 LLL  | CCC     | RRR   | DDD
```

gives

Left	Center	Right	Default
LLL	CCC	RRR	DDD

The headings and the alignment of the cells is given in the first two lines. The cell width is variable. The pandoc parameter `--columns=NUM` can be used to define the length of lines in characters. If contents do not fit, they will be wrapped.

Figures

Figures are inserted as follows:

```
![alt text](image location/ name)
```

e.g.

```
![Publishing costs](fig-hybrid-publishing-costs.png)
```

The `alt text` is used e.g. in HTML output. Additional parameters such as image width are possible.

Symbols

Scientific texts often require special characters, e.g. Greek letters, mathematical and physical symbols etc.

The UTF-8 standard, developed and maintained by *Unicode Consortium*, enables the use of characters across languages and computer platforms. The encoding is defined as RFC document 3629 of the Network Working group (Yergeau, 2003) and as ISO standard ISO/IEC 10646:2014 (International Organization for Standardization, 2014). Specifications of Unicode and code charts are provided on the Unicode homopage (`http://www.unicode.org/`).

In pandoc mardown documents, Unicode characters such as °, α , ä , Å can be inserted directly and passed to the different output documents. For the correct processing of UTF-8 encoding in LATEX, the use of the `--latex-engine=xelatex` option is necessary, further the use of an appropiate font. The Times-like XITS font (`https://github.com/khaledhosny/xits-math`) for high quality typesetting of scientific texts can be set in the LATEX template:

```
\usepackage{unicode-math}
\setmainfont
[    Extension = .otf,
   UprightFont = *-regular,
      BoldFont = *-bold,
    ItalicFont = *-italic,
BoldItalicFont = *-bolditalic,
]{xits}
\setmathfont
[    Extension = .otf,
      BoldFont = *bold,
]{xits-math}
```

To facilitate the input of specific characters, so-called mnemonics can be enabled in some editors (e.g. in atom by the `character-table` package). For example, the 2-character Mnemonics ‘:u’ gives ‘ü’ (diaeresis), or ’D*’ the greek Δ. The possible character mnemonics and character sets are listed in RFC 1345 (Simonsen, 1992).

Formulas

Formula are written in LATEX mode using the delimiters `$`. E.g. the formula for calculating the standard deviation s of a random sampling would be written as:

```
$s=\sqrt{\frac{1}{N-1}\sum_{i=1}^N(x_i-\overline{x})^{2}}$
```

and gives:

$s=\sqrt{\frac{1}{N-1}\sum_{i=1}^N(x_i-\overline{x})^{2}}$

with x_i the individual observations, $\overline{x}$ the sample mean and N the total number of samples.

Pandoc parses formulas into internal structures and allows conversion into formats other than LATEX. This allows for format-specific formula representation and enables computational analysis of the formulas (Corbí & Burgos, 2015).

Code listings

Verbatim code blocks are indicated by three tilde symbols:

```
~~~
verbatim code
~~~
```

Typeseting `inline code` is possible by enclosing text between back ticks.

```
`inline code`
```

Other document elements

Those examples are only a short demonstration of the capacities of pandoc concerning scientific documents. For more detailed information, we refer to the official manual (http://pandoc.org/MANUAL.html).

CITATIONS AND BIOGRAPHY

The efficient organization and typesetting of citations and bibliographies is crucial for academic writing. Pandoc supports various strategies for managing references. For processing the citations and the creation of the bibliography, the command line parameter `--filter pandoc-citeproc` is used, with variables for the reference database and the bibliography style. The bibliography will be located automatically at the header `# References` or `# Bibliography`.

Reference databases

Pandoc is able to process all mainstream literature database formats, such as RIS, BIB, etc. However, for maintaining compatibility with LATEX/ BIBTEX, the use of BIB databases is recommended. The used database either can be defined in the YAML metablock of the MD file (see below) or it can be passed as parameter when calling pandoc.

Inserting citations

For inserting a reference, the database key is given within square brackets, and indicated by an '@'. It is also possible to add information, such as page:

```
[@suber_open_2012; @benkler_wealth_2006, 57 ff.]
```

gives (Benkler, 2006, p. 57 ff.; Suber, 2012).

Styles

The Citation Style Language (CSL) http://citationstyles.org/ is used for the citations and bibliographies. This file format is supported e.g. by the reference management programs Mendeley https://www.mendeley.com/, Papers http://papersapp.com/ and Zotero https://www.zotero.org/. CSL styles for particular journals can be found from the Zotero style repository https://www.zotero.org/styles. The bibliography style, which pandoc should use for the target document can be chosen or in the YAML block of the markdown document or can be passed as an command line option. The later is more recommendable, because distinct bibliography style may be used for different documents.

Creation of LATEX `natbib` citations

For citations in scientific manuscripts written in LATEX, the natbib package is widely used. To create a LATEX output file with natbib citations, pandoc simply has to be run with the `--natbib` option, but without the `--filter pandoc-citeproc` parameter.

Database of cited references

To share the bibliography for a certain manuscript with co-authors or the publisher's production team, it is often desirable to generate a subset of a larger database, which only contains cited references. If LATEX output was generated with the `--natbib`, the compilation of the file with LATEX gives an AUX file (in the example named `md-article.aux`), which subsequently can be extracted using BibTool `https://github.com/ge-ne/bibtool`:

```
~~~
bibtool -x md-article.aux -o bibshort.bib
~~~
```

In this example, the article database will be called `bibshort.bib`.

For the direct creation of an article specific BIB database without using LATEX, we wrote a simple Perl script `mdbibexport` (`https://github.com/robert-winkler/mdbibexport`).

META INFORMATION OF THE DOCUMENT

Document information such as title, authors, abstract etc. can be defined in a metadata block written in YAML syntax. YAML ("YAML Ain't Markup Language", `http://yaml.org/`) is a data serialization standard with simple, human readable format. Variables defined in the YAML section are processed by pandoc and integrated into the generated documents. The YAML metadata block is recognized by three hyphens (`---`) at the beginning, and three hyphens or dots (`...`) at the end, e.g.:

```
---
title: Formatting Open Science
author: 'Albert Krewinkel$^1$ and Robert Winkler$^{2,\star}$'
bibliography: agile-markdown.bib
---
```

Using the LATEX syntax for superscripts (`$^{2,*}$`) enables the correct processing for different output formats.

EXAMPLE: MANUSCRIPT WITH OUTPUT OF DOCX/ ODT FORMAT AND LATEX/ PDF FOR SUBMISSION TO DIFFERENT JOURNALS.

At this moment, DOCX the most common format for manuscript submission. Some publishers also ask for LATEX or accept ODT. In this example, we want to create a manuscript for a *PLoS* journal, in DOCX and ODT for WYSIWYG word processors. Further, a version in LATEX/ PDF should be produced for PeerJ submission and archiving at the PeerJ preprint server.

Development of DOCX template

A first DOCX document with bibliography in *PLoS* format is created with pandoc DOCX output:

```
pandoc -S -s --csl=plos.csl --filter pandoc-citeproc
-o pandoc-manuscript.docx agile-editing-pandoc.md
```

The document settings and styles of the resulting file `pandoc-manuscript.docx` can be modified, and following it can be used as document template (`--reference-docx=pandoc-manuscript.docx`).

```
pandoc -S -s --reference-docx=pandoc-manuscript.docx
--csl=apa.csl --filter pandoc-citeproc -o outfile.docx agile-editing-pandoc.md
```

It is also possible to directly re-use a previous output file as template (i.e. template and output file have the same file name):

```
pandoc -S -s --columns=10 --reference-docx= pandoc-manuscript.docx --csl=apa.csl --filter par
```

In this way, the template can be incrementally adjusted to the desired document formatting. The final document may be employed later as pandoc template for other manuscripts with the same specifications. In this case, running pandoc the first time with the template, the contents of the new manuscript would be filled into the provided DOCX template. A page with DOCX manuscript formatting of this article is shown in **Fig. 8**.

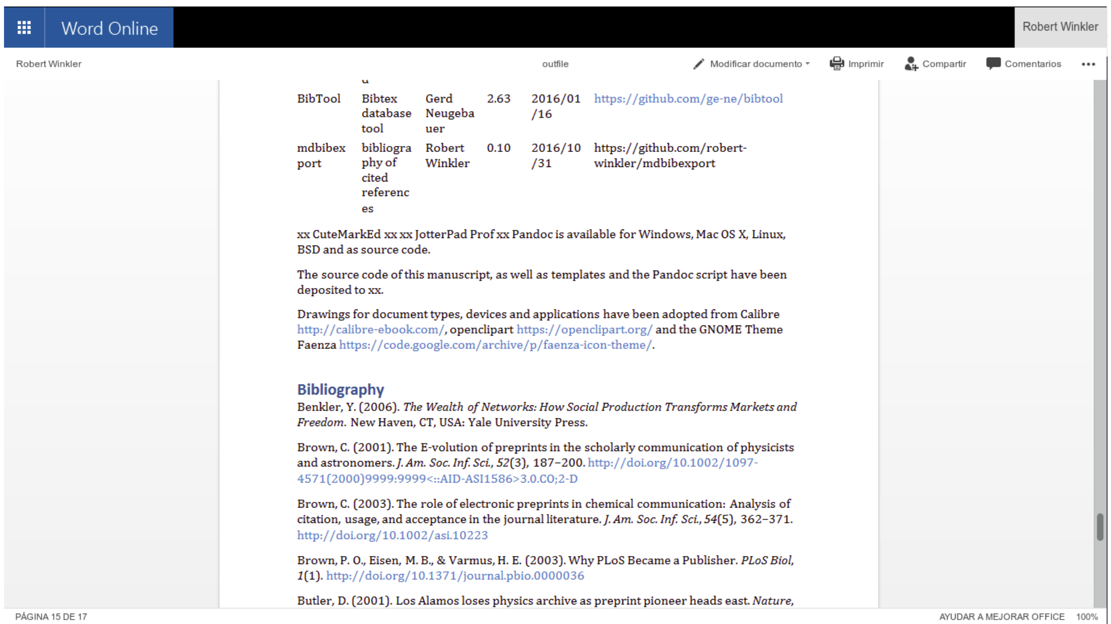

Figure 8. Editing a pandoc generated DOCX in Office 365.

The same proceedure can be applied for an ODT formatted document.

Development of a TEX/PDF template

```
pandoc -D latex > template-peerj.latex
```

AUTOMATING DOCUMENT PRODUCTION

The commands necessary to produce the document in a specific formats or styles can be defined in a simple `Makefile`. An example `Makefile` is included in the source code of this preprint/ . The desired output file format can be chosen when calling `make`. E.g. `make outfile.pdf` produces this preprint in PDF format.Calling `make` without any option creates all listed document types.

Cross-platform compatibility

The `make` process was tested on Windows 10 and Linux 64 bit. All documents – DOCX, ODT, LATEX, PDF, EPUB and HTML – were generated successfully, which demonstrates the cross-platform compatibility of the workflow.

CONCLUSIONS

Authoring scientific manuscripts in markdown (MD) format is straight-forward, and manual formatting is reduced to a minimum. The simple syntax of MD facilitates the document editing and collaborative writing. The rapid conversion of MD to multiple formats such as DOCX, LATEX, PDF, EPUB and HTML can be done easily using pandoc, and templates enable the automated generation of documents according to specific journal styles. Altogether, the MD format supports the agile writing and fast production of scientific literature. The associated time and cost reduction especially favours community-driven publication strategies.

ACKNOWLEDGMENTS

We cordially thank Dr. Gerd Neugebauer for his help in creating a subset of a bibtex data base using BibTool and Dr. Ricardo A. Chávez Montes for comments on the manuscript. The work was funded by the Consejo Nacional de Ciencia y Tecnología (CONACyT) Mexico, with the grant FRONTERAS 2015-2/814 and by institutional funding of the Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV).

SOFTWARE AND CODE AVAILABILITY

The relevant software for creating this manuscript used is cited according to (Smith, Katz & Niemeyer, 2016) and listed in **Tab. 3**. Since unique identifiers are missing for most software projects, we only refer to the project homepages or software repositories:

Table 3. Relevant software used for this article.

Software	Use	Authors	Version	Release	Homepage/ repository
pandoc	universal markup converter	John MacFarlane	1.16.0.2	16/01/13	`http://www.pandoc.org`
pandoc-citeproc	library for CSL citations with pandoc	John MacFarlane, Andrea Rossato	0.9.1	16/03/19	`https://github.com/jgm/pandoc-citeproc`
ownCloud	personal cloud software	ownCloud GmbH, Community	9.1.1	16/09/20	`https://owncloud.org/`
Markdown Editor	plugin for ownCloud	Robin Appelman	0.1	16/03/08	`https://github.com/icewind1991/files_markdown`
BibTool	Bibtex database tool	Gerd Neugebauer	2.63	16/01/16	`https://github.com/ge-ne/bibtool`

The source code of this manuscript, as well as templates and the pandoc Makefile have been deposited to https://github.com/robert-winkler/scientific-articles-markdown/.

Drawings for document types, devices and applications have been adopted from Calibre `http://calibre-ebook.com/`, openclipart `https://openclipart.org/` and the GNOME Theme Faenza `https://code.google.com/archive/p/faenza-icon-theme/`.

BIBLIOGRAPHY

Benkler Y. 2006. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven, CT, USA: Yale University Press.

Brauer M., Durusau P., Edwards G., Faure D., Magliery T., Vogelheim D. 2005. *Open Document Format for Office Applications (OpenDocument) v1.0*. OASIS.

Brown C. 2001. The E-Volution of Preprints in the Scholarly Communication of Physicists and Astronomers. *J. Am. Soc. Inf. Sci.* 52:187–200. DOI: 10.1002/1097-4571(2000)9999:9999<::AID-ASI1586>3.0.CO;2-D.

Brown C. 2003. The Role of Electronic Preprints in Chemical Communication: Analysis of Citation, Usage, and Acceptance in the Journal Literature. *J. Am. Soc. Inf. Sci.* 54:362–371. DOI: 10.1002/asi.10223.

Brown PO., Eisen MB., Varmus HE. 2003. Why PLoS Became a Publisher. *PLoS Biol* 1. DOI: 10.1371/journal.pbio.0000036.

Butler D. 2001. Los Alamos Loses Physics Archive as Preprint Pioneer Heads East. *Nature* 412:3–4. DOI: 10.1038/35083708.

Callaway E. 2013. Preprints Come to Life. *Nature News* 503:180. DOI: 10.1038/503180a.

Corbí A., Burgos D. 2015. Semi-Automated Correction Tools for Mathematics-Based Exercises in MOOC Environments. *International Journal of Interactive Multimedia and Artificial Intelligence* 3:89–95. DOI: 10.9781/ijimai.2015.3312.

Dominici M. 2014. An overview of Pandoc. *TUGboat* 35:44–50.

DPT Collective. 2015. From Print to Ebooks: A Hybrid Publishing Toolkit for the Arts. In: Monk J, Rasch M, Cramer F, Wu A eds. Institute of Network Cultures,

Eikebrokk T., Dahl TA., Kessel S. 2014. EPUB as Publication Format in Open Access Journals: Tools and Workflow. *Code4Lib*.

Eisen M. 2003. Publish and be praised. *The Guardian*.

Fecher B., Friesike S. 2014. Open Science: One Term, Five Schools of Thought. In: Bartling S, Friesike S eds. *Opening Science*. Springer International Publishing, 17–47.

Ginsparg P. 1994. First Steps Towards Electronic Research Communication. *Computers in Physics* 8:390–396. DOI: 10.1063/1.4823313.

Hickson I., Berjon R., Faulkner S., Leithead T., Navara ED., O'Connor E., Pfeiffer S., Faulkner S., Navara ED., Leithead T., Berjon R., Hickson I., Pfeiffer S., O'Connor T. 2014. *HTML5*. W3C.

Houghton J., Rasmussen B., Sheehan P., Oppenheim C., Morris A., Creaser C., Greenwood H., Summers M., Gourlay A. 2009. Economic implications of alternative scholarly publishing models: Exploring the costs and benefits.

International Organization for Standardization. 2013. ISO 32000-1:2008 - Document management – Portable document format – Part 1: PDF 1.7. *ISO*.

International Organization for Standardization. 2014. ISO/IEC 10646:2014 - Information technology – Universal Coded Character Set (UCS). *ISO*.

Kielhorn A. 2011. Multi-target publishing-Generating ePub, PDF, and more, from Markdown using pandoc. *TUGboat-TeX Users Group* 32:272.

Lamport L. 1994. *LaTeX: A Document Preparation System*. Reading, Mass: Addison-Wesley Professional.

Leonard S. 2016. *Guidance on Markdown: Design Philosophies, Stability Strategies, and Select Regis-*

trations. RFC Editor; Internet Request for Comments.

Ngo T. 2006. *OFFICE OPEN XML OVERVIEW ECMA TC45*. Ecma International.

Ovadia S. 2014. Markdown for Librarians and Academics. *Behavioral & Social Sciences Librarian* 33:120–124. DOI: 10.1080/01639269.2014.904696.

Raggett D., Hors AL., Jacobs I., Le Hors A., Raggett D., Jacobs I. 1999. *HTML 4.01 Specification*. W3C.

Simonsen K. 1992. *Character Mnemonics & Character Sets*. Rationel Almen Planlaegning; Internet Request for Comments.

Smith AM., Katz DS., Niemeyer KE. 2016. Software Citation Principles. *PeerJ Computer Science* 2:e86. DOI: 10.7717/peerj-cs.86.

Solomon D., Björk B-C. 2016. Article Processing Charges for Open Access Publicationthe Situation for Research Intensive Universities in the USA and Canada. *PeerJ* 4:e2264. DOI: 10.7717/peerj.2264.

Suber P. 2012. *Open Access*. Cambridge, Mass: The MIT Press.

Van Noorden R. 2012. Journal Offers Flat Fee for "all You Can Publish". *Nature News* 486:166. DOI: 10.1038/486166a.

Van Noorden R. 2013. Open Access: The True Cost of Science Publishing. *Nature* 495:426–429. DOI: 10.1038/495426a.

Van Noorden R. 2014. The arXiv Preprint Server Hits 1 Million Articles. *Nature News*. DOI: 10.1038/nature.2014.16643.

Volmer DA., Stokes CS. 2016. How to Prepare a Manuscript Fit-for-Purpose for Submission and Avoid Getting a "desk-Reject". *Rapid Commun. Mass Spectrom.*:n/a–n/a. DOI: 10.1002/rcm.7746.

Willinsky J. 2005. The Unacknowledged Convergence of Open Source, Open Access, and Open Science. *First Monday* 10. DOI: 10.5210/fm.v10i8.1265.

Woelfle M., Olliaro P., Todd MH. 2011. Open Science Is a Research Accelerator. *Nat Chem* 3:745–748. DOI: 10.1038/nchem.1149.

Yergeau F. 2003. *UTF-8, a transformation format of ISO 10646*. Alis Technologies.

Youngen GK. 1998. Citation Patterns to Traditional and Electronic Preprints in the Published Literature. *Coll. res. libr.* 59:448–456. DOI: 10.5860/crl.59.5.448.