

1 **HOW TO PREPARE A GENERIC SCIENTIFIC MANUSCRIPT**
2 **FOR SUBMISSION USING MARKDOWN AND PANDOC**

3 First A. Author¹, Second B. Author², Third Author³, Last Author^{1,2,3}

- 4 1. University of Pandoc, Randwick, NSW, Australia
5 2. Markdown Institute, Ottawa, ON, Canada
6 3. LaTeX Research Institute, Kingston, On, Canada

7

8

9

10 **Corresponding author**

11 Dr. First A. Author

12 University of Pandoc

13 42 Printers Lane

14 2031, Randwick, NSW, Australia

15 email: fa.author@pandoc.au

16 ph: +61-1-2345-6789

17

18

19 **Keywords:** Pandoc, Markdown, Latex

20 **Running title:** Markdown manuscript

21 **Wordcount:** 4,321

22 **Abstract**

23 **Purpose.** If journal requires abstract sections, these can be included and made bold by
24 including double asterisks `**section_name**` before and after each header. If these are
25 not needed, simply do not include them, simple. **Methods.** Because of our selected font,
26 we can easily include special characters directly into our markdown file and these will
27 be rendered properly. This provides some motivation to find a text editor that has a good
28 *character map* plugin. Personally, I have grown rather fond of the Atom editor, which has
29 a *character-map* plugin that does the trick. by clicking on a hotkey, `Alt-1`, I can type the
30 name of the symbol I am looking and it is inserted into my document. **Results.** So lets
31 say I just pressed `Alt-1` and typed `plusminus`. The item at the top of the list would be \pm , as
32 expected. I hit `Enter` and it is inserted into my document. This makes it easy to say that,
33 on average, my level of frustration drafting a manuscript has decreased by $77\% \pm 4\%$
34 since learning about Markdown and Pandoc. Also, scientific papers always sound fancier
35 when you use Greek letters. So I will search for the letter `beta`, which gives me the special
36 character β . Also, because Pandoc converts Markdown to PDF using LaTeX, it is amazing
37 support for mathematical equations and the like. Most basic tutorials on LaTeX should
38 cover the basics of math-mode (i.e. `$math stuff$`). **Conclusion.** This dummy manuscript
39 and its accompanying files includes everything you need to generate a basic manuscript
40 for submission. The rest of this paper will provide a brief overview of Markdown, Pandoc
41 and the various parts that accompany this manuscript to generate the final result. I
42 chose a simple approach and targeted generic formatting. Much more sophisticated
43 manuscripts can be generated, but it is always a good idea to start small and simple.

44 Introduction

45 A guiding principle of Pandoc and LaTeX is to separate content from style. While
46 this can be achieved in LaTeX, the actual document in which you write can be
47 rather intimidating for the uninitiated. It is filled with commands, for example
48 `\usepackage[margin=3cm]{geometry}`, and typing even a simple document can become clut-
49 tered with function calls to make text bold (`\textbf{my text}`) or italic (`\emph{my text}`),
50 or just trying to write a percent sign (`\%`). While LaTeX is more powerful and is what is
51 used by many publishers to typeset the journals we publish, it can be overwhelming for
52 people who have any used Microsoft Office or Libre Office.

53 Markdown is a simple markup language initially created to write content for the web.
54 Pandoc, on the other hand, is a powerful Swiss Army knife of a tool that converts docu-
55 ments from one format to another. Importantly, Pandoc adds functionality that allows
56 us, researchers and scientists, to generate professional looking manuscripts. And yes,
57 Pandoc handles references and citations in a clean and efficient way; more on this later.
58 This means that we can now focus on the content of our manuscripts when we are writ-
59 ing them. We will let Pandoc do the heavy lifting of creating a professional looking
60 manuscript that we can share with co-authors and submit for review. And since we are
61 using Markdown to write our manuscript, we can use a simple text editor, and the text
62 that we write will be quite straight forward, free of LaTeX-style function calls.

63 While we may use various formatting styles when writing notes, reports, letters, etc.,
64 things are rather straightforward when preparing scientific manuscripts.¹ The main
65 types of inline formatting I use are italics, super- and subscript. I also include tables
66 and figures, and very occasionally use footnotes. For the footnote I just used, I typed
67 `[1]` in the spot where I want the superscript number 1 to appear, and then, somewhere
68 else in the document (I chose immediately after the paragraph as this makes it easy
69 to find if I need to change it) add `[1]`: At least for the type of papers I find myself
70 writing. This text has to be on its own line, with an empty line before and after. To make
71 something italic, simply add an asterisks on either side. For example, typing this, `*this`
72 `will be slanted*` will result it text that appears as *this will be slanted*. Superscript is

¹At least for the type of papers I find myself writing.

73 achieved by surrounding the text with the carat symbol. For example, `sum^adults^` will
74 generate $\text{sum}^{\text{adults}}$. Similarly, the tilda symbol is used for subscript: `sum~children~` will
75 appear as $\text{sum}_{\text{children}}$ in the final document. We will cover tables and figures later in this
76 document. The key thing here is that making these style changes can be done later, *en*
77 *masse*. If you don't remember how to make something italic, bold or superscript, don't
78 worry. Just leave yourself a note [TODO: make this text bold] and keep your writing flow
79 going. Remember, we are trying to separate content from style. Stay in the flow and
80 don't get distracted. I have included a brief review of key inline formatting (Table 1) and
81 special characters (Table 2) at the end of this paper.

82 **Methods**

83 It turns out we need to add and specify a few things to Pandoc in order to obtain a
84 presentable manuscript. As you can see, these details are not included in this file, the
85 `manuscript.md` file. That is because we are trying to separate content from style (and
86 distractions). So this file is where we should write our manuscript. The details that
87 specify things like the font, whether or not we want numbered sections, the addition of
88 line numbers, etc are in small helpers files. Lets go over these now.

89 **Participants**

90 To generate the PDF output, we simply need to run `pandoc -d header.yaml` on the com-
91 mand line. The content of this file could be placed in a `yaml` header at the top of this
92 document, but that would be distracting to use and confusing to our co-authors who
93 might not be familiar with Markdown and Pandoc. So, we extracted the `yaml` header and
94 placed it in its own file: `header.yaml`. Note that `yaml` files need to be formatted based on
95 some simple rules, so if you want to change anything, make sure you adhere to proper
96 formatting style; a quick Google search will locate a quick tutorial if you need one.

97 In `header.yaml`, we find two commands related to referencing. One specifies the Bibtex
98 file that contains our references the other specifies the referencing style we want to use
99 to format our references (note that a line that starts with `#` is a comment and has no
100 effect).

```
# References and Reference Style  
csl: journal-of-neurophysiology.csl  
bibliography: references.bib  
citeproc: true # required in pandoc 2.14
```

101 csl stands for Citation Style Language. As stated on its website:

102 “Welcome to the open source Citation Style Language (CSL) project! Our goal
103 is to facilitate scholarly publishing by automating the formatting of citations
104 and bibliographies. We develop the Citation Style Language and maintain a
105 crowdsourced repository with over 10,000 free CSL citation styles.”

106 We will have more to say about references later in this article.

107 The next part of the header .yaml file specifies the input and output files.

```
# Names of Input and Output Files  
input-file: manuscript.md  
output-file: manuscript.pdf
```

108 These will need to be changed if you decide to rename the base Markdown file, or if you
109 want a specific name for the outputted PDF file.

110 Then our header .yaml file has three difference ‘include’ statements. These allow us to
111 include the content of specified files into the header, or preamble, of the LaTeX file that
112 will generate our article, as well as before and after the actual content of our article,
113 the stuff if manuscript.md is added to the LaTeX file. What these files contain will be
114 described in the following sub-sections.

```
# Addition to LaTeX Template  
include-in-header: header.tex # Change margin and line spacing  
include-before-body: before_body.tex  
include-after-body: after_body.tex
```

115 The next section in the header .yaml file specifies some Pandoc variables that will control
116 various aspects of how are article is generated. In order to properly process special
117 characters typed directly into our text editor, rather than specifying a special LaTeX

118 command, we are using the `xelatex` pdf-engine. To have these special characters appear
119 in our This font was selected as it has good support for special characters.

```
# LaTeX Specifications  
pdf-engine: xelatex  
variables:  
  documentclass: article  
  mainfont: "DejaVu Serif"  
  sansfont: Arial  
  classoption:  
    - 10pt # 11pt, 12pt
```

120 The final part of header `.yaml` contains instructions and a commented out line related to
121 adding numbered sections to our article.

```
# To have numbered sections, uncomment the following line.  
# But we don't want our Title or Abstract to be numbered,  
# so we will add {.unnumbered} next to our Title and Abstract  
# headers, separated by one space.  
# Example: # Abstract {.unnumbered}  
  
# number-sections: true
```

122 **Experimental set-up**

123 The file called header `.tex` contains LaTeX commands that Pandoc will insert in the pream-
124 ble of the LaTeX it generates to make our manuscript.

```
\usepackage[margin=3cm]{geometry}  
\usepackage{lineno}  
\usepackage{setspace}  
\doublespacing  
%\singlespacing  
%\onehalfspacing
```

125 If you want to change the margins of the PDF document that is generated, you can
126 change the value passed to the `geometry` packaged. Next, we instruct LaTeX to use the
127 `lineno` package, which will allow us to add line numbers to our document. Finally, we
128 load the `setspace` package, which gives us access to three functions that we can use to
129 specify the line spacing we want for our manuscript. The default is double line spacing,
130 but by commenting out `\doublespacing` with a `%` and uncommenting one of the other lines
131 we can use single line spacing or one and a half line spacing.

132 **Protocol**

133 The next file we have is `before_body.tex`. The content of this file is inserted after the
134 `\begin{document}` command in the LaTeX file, but before the actual content -the stuff in
135 this file- is added (by the way, I used two dashes in a row `--` to get the two emdashes in
136 this sentence).

```
\thispagestyle{empty}  
\vspace*{2 cm}  
\begin{linenumbers}
```

137 The first line tells LaTeX to not add a page number to the first page, our title page. The
138 second line adds some vertical space, which allows the title of our article to not be at
139 the very top of our title page. The final line tells LaTeX to start numbering the lines in
140 our document.

141 If you did not want numbered lines, you could delete or comment out `\begin{linenumbers}`
142 in this file, as well as `\end{linenumbers}` that we add to the very end of our document via
143 the `after_body.tex` file.

144 **Statistical analysis**

145 The file `after_body.tex` includes a single LaTeX command: `\end{linenumbers}`. As men-
146 tioned above, delete or comment out this command if you do not want line numbers

147 **Results**

148 **Effect of font on blood pressure**

149 Many researchers love to hate referencing in manuscripts. Many people default to pro-
150 prietary software like Endnote or Reference Manager, but many good quality free alter-
151 natives are available such as Mendeley, JabRef, Zotero, etc. Regardless of what refer-
152 ence manager program you choose, the key functionality that we want if for the program
153 to automatically generate and update a .bib file that contains our references.

154 I often manually generate my .bib Bibtex files when I know they will be relatively small.
155 However, I have recently (re)started using Zotero, and I am licking it. I can add refer-
156 ences when I a browsing for articles, using the Zotero plugin for the Firefox web browser.
157 I can also search for references from within the Zotero desktop application. Also, having
158 an online accounts allows me to view and sync my library on any computer. I recommend
159 the Better Bibtex plugin for Zotero, which adds additional functionality.

160 The key things is that you have a Bibtex file that contains your references. Each ref-
161 erence has a key, which we use to refer to the reference. Below is an example of a
162 reference in a Bibtex file:

```
@Article{Diong2015,  
  Author="Diong, J. and Herbert, R. D. ",  
  Title="{[I]s ankle contracture after stroke due to abnormal intermuscular force transmission?}",  
  Journal="J. Appl. Biomech.",  
  Year="2015",  
  Volume="31",  
  Pages="13--18",  
}
```

163 The key of this reference is `Diong2015`. Therefore, I can refer to this paper by preceding
164 it's key with the `@` symbol. To include a reference to the article that will be included
165 in parentheses, I would type `[@Diong2015]`; this would result in the following (Diong and
166 Herbert, 2015). If was to refer to the paper by Diong and Herbert (2015) in a sentence, as
167 I just did, I would omit the square brackets (i.e. `@Diong2015`). You can also include several

168 reference keys in the square brackets. For example, typing [`@Diong2012a; @Diong2012b;`
169 `@Diong2015; @Diong2019]` will generate (Diong et al., 2012a, 2019; Diong et al., 2012b;
170 Diong and Herbert, 2015). Note the semi-colon and space between each reference key.

171 Sometimes we want to add text in the parentheses with our reference. This can be
172 achieved by simply including the desired text before or after the reference. As an
173 example, lets look at the following line of text: `contractures are a major problem [see`
174 `@Diong2015 for a review];` it looks like this when processed by Pandoc: `contractures are`
175 `a major problem (see Diong and Herbert, 2015 for a review).` More complex examples
176 are also possible, such as `[see @Huijing2003, Figure 1; also @Bojsen-Moller2010, Table`
177 `2],` which results in the following (see Huijing and Baan, 2003; also Bojsen-Moller et al.,
178 2010, Table 2). Some of these styles of referencing will be depending on the referencing
179 style you are using.

180 Now, if you skip to the end of this manuscript, you will find a References section where
181 all the references I just cited will be properly formatted according to the style that I
182 specified in the header `.yaml` file.

```
# References and Reference Style  
csl: jneurophysiol.csl  
bibliography: references.bib
```

183 Here we are using a Citation Style Language file that formats our references for sub-
184 mission to the Journal of Neurophysiology. There are literally thousands of such files
185 available for download, or you can generate your own. That means, if your paper gets
186 rejected from one journal and the next journal you want to submit to use a different
187 referencing style, simply find the appropriate `.csl` file, add it to the project folder and
188 change the appropriate line in your header `.yaml` file.

189 **The relationship between hyphens and cholesterol**

190 It is worth mentioning that no reference manager software is perfect. When references
191 are downloaded from the web, the formatting is not always correct, or they may not
192 include the abbreviated version of a journal title. Thus, it is up to the authors to ensure
193 the content of the `.bib` file is correct. For example, the tile of the paper sometimes comes

194 in title case, where each word is capitalised. This needs to be fixed in the Zotero and
195 the `.bib` file updated.

196 **Causal relationship between Markdown use and happiness**

197 In many text editors, you can split your screen and have two documents open. Thus,
198 you could have your paper open on one side and your `.bib` file open on the other. And
199 when you want to enter a reference, you could search the `.bib` file for the reference and
200 identify its key. As you probably have already realised, this is less than ideal.

201 A better way to work is to use the Atom editor and the `autocomplete-bibtex` plugin. By
202 pointing this plugin to the `.bib` file that is generated and regularly updated by Zotero,
203 we can add references without having to move our hands from the keyboard. All we
204 have to do is type `@` and start typing and a list of possible references will appear. We
205 can scroll through the available choices or we can keep on typing to narrow down our
206 search. The, when we have found the reference we were looking for, we simply click on
207 Enter and the reference key is added. We can see an example of this is action in Figure
208 1, where I am adding this reference (Diong and Herbert, 2015).

209 **Discussion**

210 This group of files provides a relatively simple approach to drafting scientific articles. It
211 can be adapted to suite various requirements, yet remains relatively simple and clutter-
212 free. Because of this, it remains a simple article template and may not suite everyone's
213 needs. In preparing this group of files, several other examples were reviewed; some
214 were relatively simple while others were sophisticated and generated publication-quality
215 PDF documents, some based on the official LaTeX packages provided by publishers. The
216 benefit of this series of files is that it can easily be adapted to other types of documents,
217 such as study notes, lab notebooks, blog posts, etc. The building blocks are simple and
218 most key parts are explained in the present article.

219 **Not using maketitle**

220 Many introductory tutorials on using Markdown and Pandoc to generate nice looking
221 PDF add the title, authors, abstract and date in the `yaml` header. This way, these entries
222 are passed to LaTeX and used as part of the `\maketitle` command.

223 I tried several ways to get `maketitle` to generate something that would be acceptable as
224 a manuscript title page. Using an additional LaTeX package I was able to specify more
225 than one affiliation per author. However, allowing room to specify the details of the
226 corresponding author and also various other items such as word count, running title, key
227 words, etc was not straightforward. There were hacks that hijacked the `date` variable,
228 but these were less than optimal. Also, I was aiming for a simple approach that did not
229 require writing a new LaTeX template file with newly defined (or redefined) functions.

230 Therefore, I opted to bypass `maketitle` completely. This allows for great flexibility as to
231 what appears on the title page. The downside is that I had to use several manual line
232 breaks (i.e. `\`) and a call to `\newpage` in this, the main Markdown document.

233 **Tables**

234 Tables can be found at the end of the manuscript, and they are rather easy to prepare,
235 especially if you use Pandoc-style tables (see `manuscript.md` for example of how to prepare
236 tables). To add a table caption, simply allow for one empty line and start the caption
237 with `“Table:”`; this will be recognised by Pandoc as a table caption and will be properly
238 formatted in LaTeX. To ensure LaTeX produces the table in the expected location, use
239 `\clearpage` between each new table you add.

240 **Figures**

241 Figure are also found at the end of the manuscript and they too are easy to add to the
242 manuscript.

```
! [Caption_goes_here] (figure.png) { width=10cm }  
! [Caption_goes_here] (img/figure.png) { height=10cm }
```

243 Add the full caption to your figure in the square brackets, but unlike tables, there is no

244 need to add 'Figure' or something similar at the start. Pandoc, via LaTeX, will automati-
245 cally add 'Figure' and the appropriate figure number on the final PDF.

246 The figure can be located in the same folder as the Markdown file where you are drafting
247 your article, but it can also be located in a dedicated folder, such as `img`. In this case,
248 simply include the folder name before the figure name (the forward slash may need to
249 be backslash for Windows users).

250 Finally, if you did not make your figures to exact size you want them to appear in the
251 figure published manuscript, you really should do this, you can specify the size of the
252 figure by adding an additional entry. Note that the entry goes between curly braces,
253 with no space between the closing parenthesis and the opening curly brace, and with
254 spaces on either side of the size command. Various options are possible, but the most
255 sensible for manuscripts are to specify either the width or height of the figure.

256 **Writing with co-authors**

257 Now comes the \$1M dollar question: how to write a paper using this template with
258 co-authors who only know Microsoft Word? Well, there are a few options.

259 The best way would be to have your co-authors work directly on the Markdown file
260 (`manuscript.md`), where comments could be left in an agreed upon format, for example
261 in typewriter font. In papers than do not contain references to software or code, it is
262 uncommon to use such a font, thus it would be an easy way to leave comments `|MH: I`
263 `was wondering if we could expand on this point a little, maybe providing an example`
264 `of what such a comment would look like|`.

265 Also, rather than emailing the files around, the manuscript could be version controlled
266 using `git` and hosted on GitHub or an internal GitLab server. With several co-authors,
267 the person who is likely to give you the best and detailed feedback should go first, and
268 then other co-authors could add their changes and comments after a few initial rounds
269 between you and the other key authors (often your supervisor). It would also be possible
270 to have each co-authors create a branch in the `git` repository, do their work their, and
271 then make a pull-request. This allows for a nice, civilised conversation between the two
272 parties. A workflow around `git` is definitely not common in many fields, including my

273 own, but it makes total sense to have a lasting history of the manuscript in all its forms,
274 and this without having dozens of files with co-author initials appended at the end or v1,
275 v2, v3final, v3finalfinal appended at the end.

276 An alternative to having your co-authors use git is to email them the Markdown file and
277 have them email back their changes and comments. You can incorporate these changes
278 yourself on a git branch, and then merge them in to the main document. This will likely
279 be needed for co-authors not familiar with git.

280 Another less-than-optimal alternative is to use Pandoc to output a Microsoft Word .docx
281 file. While the formatting is not perfect, it is surprisingly good; definitely enough to have
282 your co-authors revise the manuscript. But what do you do when the document is sent
283 back to you?

284 We can use Pandoc to convert our document back into a Markdown file. If you co-author
285 used tracked-changes, you have to option to accept them all or reject them all. You
286 will likely want to accept them all and add the modified file to a dedicated git branch,
287 where you will be able to merge these changes (accepting and rejecting the changes
288 individually). One thing to not is that any comment you co-authors left for you as Word
289 comments will be lost.

```
% Example where all changes are accepted
```

```
pandoc --from=docx manuscript.docx --track-changes=accept -o revised_manuscript.md
```

```
% Example where all changes are rejected
```

```
pandoc --from=docx manuscript.docx --track-changes=reject -o revised_manuscript.md
```

290 Another option is to accept all changes, including comments, by using `--track-changes=all`.
291 This is likely not you want to do, as it actually embeds detailed notes about each change
292 and comment made in the document. Here is an example:

```
If [I added stuff.]{.insertion author="Martin Heroux" date="2021-07-08T15:22:24Z"}
```

This means that we can now focus on the content of our manuscripts when we are writing them.

```
[A comment.]{.comment-start id="0" author="Martin Heroux" date="2021-07-08T15:25:08Z"}
```

```
[[]]{.comment-end id="0"}]{.insertion author="Martin Heroux" date="2021-07-08T15:25:08Z"}
```

293 **Workflow in Atom**

294 If you are new to all of this and don't have a preferred writing environment, you might
295 be curious about my current tooling. Well, as you can see in Figure 1, I use the Atom
296 editor. This is an open-source editor that to which you can add plugins (from the folks at
297 Atom or the Atom community). You can use Atom to do your actual coding (if you code),
298 but you can also use it to write your papers! As you can see, I have my files and folders
299 visible on the left side, my main `manuscript.md` file open in one window, a rendered PDF
300 in another window, and, along the bottom, a command-line terminal.

301 When I am in the flow of writing, I don't have the PDF visible, as it is distracting. It is too
302 tempting to compile our cool document and see what it looks like. However, when you
303 are in the final phase of preparing your manuscript, or when you are learning Markdown
304 and Pandoc, it can be useful to see what the PDF output looks like.

305 As explained above, I run `pandoc -d header.yaml` on the command line to generate my
306 PDF manuscript. With the PDF viewer plugin that I use (see Table 3) for details, the
307 PDF auto-updates.

308 **Submitting your revised, final manuscript**

309 While you can usually submit a single PDF document to journals when first submitting
310 your paper, they usually want your paper in a different format, including individual, high-
311 quality figures. Oddly enough, publishers almost all use LaTeX to typeset journals, yet
312 many journals do not accept LaTeX files to be submitted, or simply text file or Mark-
313 down files. Rather, they prefer Microsoft Word `.docx` files, a complex file type that mixes
314 formatting and content, where the formatting will be stripped away by the typesetters.

315 So, what to do after you have received reviewer feedback on your first submission and
316 the editor new demands the source files? You are lucky if you are allowed to submit a
317 LaTeX file. Then, you can use Pandoc to output a `.tex` file. The next easiest thing to do is
318 to use Pandoc to convert your Markdown manuscript file into a Word file. The formatting
319 should be good enough, but you might have to tweak your tables a little (see Figure 2

320 for an example). While this seems like an unnecessary and somewhat tedious step, you
321 should only have to do it once per manuscript. And make sure you do it at the very end,
322 when you and your colleagues have make all the required changes.

323 Someday, we may actually be able to provide Markdown or text files as the final product.
324 But we should not let a journal or publisher, and their idiosyncratic choice of preferred fi-
325 nal file, influence how we spend the vast majority of our time working on our manuscript.
326 We will spend countless hours writing and revising our paper; I would much prefer work
327 in a simple, clean format like Markdown for those hours, and then spend a somewhat
328 painful 5-10 min tweaking a Word document if my manuscript has passed a first round
329 of reviews and I have been asked to submit a revised version.

330 **Marked-up version of your manuscript**

331 When submitting a revised version of your manuscript, journals often ask for a marked-
332 up version, to show the various additions and deletions that have been made. How do
333 we do this in the current workflow?

334 There is a wonderful LaTeX package called `latexdiff` that does a great job when you
335 have two versions of the same `.tex` file. A call to this program would look like this:

```
336 latexdiff manuscript.tex revised_manuscript.tex > diff.tex
```

337 Thus, we could have Pandoc output two `.tex` files, one from our original submission and
338 one from our revised version, and get the marked-up version that can be processed using
339 `pdflatex` (or `xelatex`) to get a nice looking PDF. An example of what such a PDF document
340 looks like is shown in Figure 3.

341 Another option that is in theory simpler is to use ~~strikeout text for the things that you~~
342 ~~want to delete~~ and **bold text for any additions**. This is can be achieved using basic Pan-
343 doc Markdown: `~~strikeout text for the things that you want to delete~~` and `**bold`
344 `text for any additions.**` While this approach may seem simpler because it does not
345 require running files through `latex-diff` and recompiling a PDF document, it is actually
346 much more work. Consider a document that has been through multiple rounds of re-
347 visions by you and your co-authors, how will you know what has been added and what

348 has been deleted? The only way for this to work is for you and your co-authors to agree
349 on the convention to manually strikeout text using `~~text~~` and manually bold new next
350 with `**text**`. It could work, but because it is not common practice, it is quite likely that
351 someone will forget to do it. Best to learn to work with `latex-diff` if you ask me.

352 **Conclusion**

353 The approach presented here is somewhat simplistic. However, it provides a nice in-
354 troduction to Pandoc and Markdown for academic writing that does not use much or
355 any mathematical notation or formulas, or embedded code. A slightly different flavour
356 of Markdown, called R-Markdown is popular amongst users of the R programming lan-
357 guage. When combined with `knitr`, it can be used to write papers in various journal-
358 specific styles. It can even work with Python code. However, my approach is generally
359 to start simple and use more sophisticated tools when I need them. The current approach
360 will do me just fine for the majority of my papers.

³⁶¹ **Acknowledgements**

³⁶² I would like to thank the participants of the study, and to Dr. Honeycut for his wisdom
³⁶³ and guidance.

³⁶⁴ **Funding**

³⁶⁵ The present work was supported by the two gold coins I found the other day when I was
³⁶⁶ cleaning between the sofa cushions.

³⁶⁷ **Author Contributions**

³⁶⁸ Authors agree to attest that they have contributed sufficiently to be listed as an author.

Table 1: Fundamental inline formatting in Pandoc-flavoured Markdown

Inline formatting	Pandoc Markdown command	Result
Bold	<code>**bold**</code>	bold
Italic	<code>*italic*</code>	<i>italic</i>
Superscript	<code>x^2^</code>	x^2
Subscript	<code>x~2~</code>	x_2
Emdash	<code>--</code>	-

Table 2: Example of special characters that can be typed directly into a text editor and are properly rendered by Pandoc, without using the LaTeX math-mode or other packages.

Character name	output
Degree symbol	°
Greek letters	β α
plus-minus	±
Sumamtion	Σ
Trademark	™
Copyright	©
Money	¢ £ €

Table 3: Atom plugins that make writing (Markdown) documents a joy.

Atom plugins	Functionality
<code>autocomplete-bibtex</code>	Amazing plugin that simplifies entering reference key
<code>character-table</code>	For insert special characters
<code>language-markdown</code>	To allow language highlighting for Markdown
<code>pdf-view</code>	Provides in-editor PDF reviewer
<code>platformio-ide-terminal</code>	An embedded terminal window to run your Pandoc commands
<code>tool-bar-markdown-writer</code>	Helpful when first learning Markdown and its commands
<code>markdown-preview</code>	Provides real-time HTML rendered view of Markdown document
<code>wordcount</code>	Wordcount of current document
<code>markdown-fold</code>	Can fold sections, and thus focus on parts of document

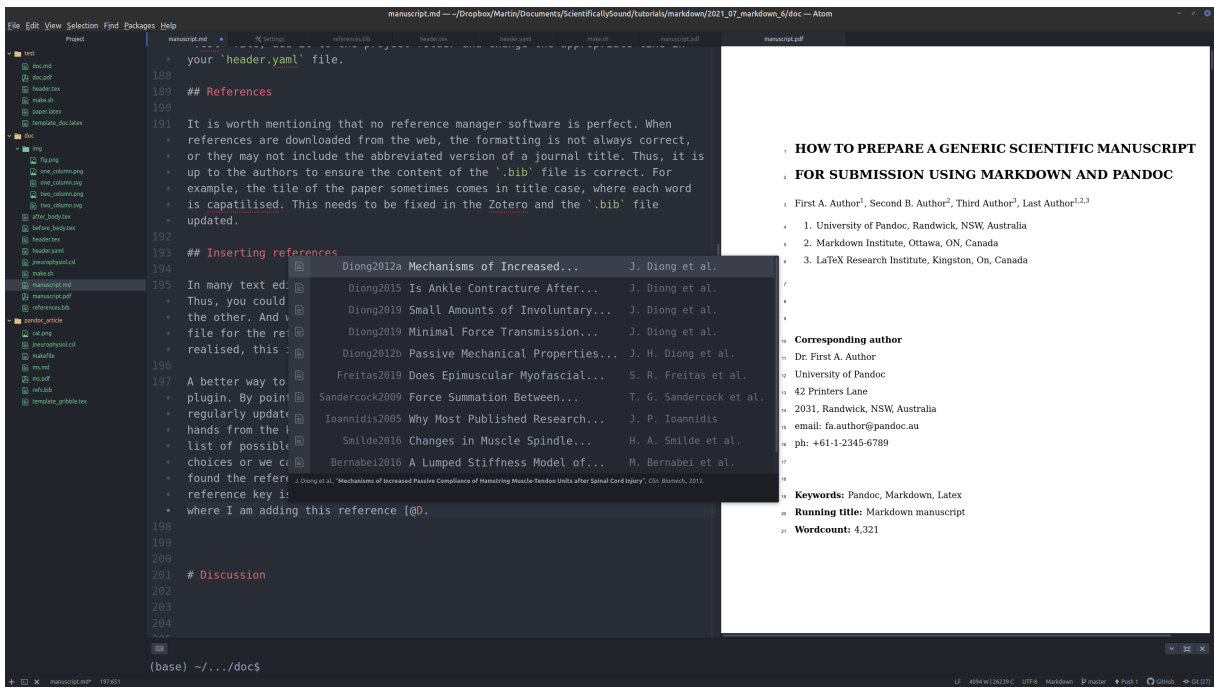


Figure 1: Autocomplete Bibtext in action. After typing the @ symbol and typing the first letter of the reference we wanted to enter, a list of possible references popped up. We can continue typing to narrow down the choice of available references, or we can use the up and down key to scroll through the references. Note that at the bottom of the pop-up screen there is a formatted version of the currently selected reference.

HOW TO PREPARE A GENERIC SCIENTIFIC MANUSCRIPT FOR SUBMISSION USING MARKDOWN AND PANDOC

First A. Author¹, Second B. Author², Third Author³, Last Author^{1,2,3}

1. University of Pandoc, Randwick, NSW, Australia
2. Markdown Institute, Ottawa, ON, Canada
3. LaTeX Research Institute, Kingston, On, Canada

Corresponding author

Dr. First A. Author
University of Pandoc
42 Printers Lane
2031, Randwick, NSW, Australia
email: fa.author@pandoc.au
ph: +61-1-2345-6789

Keywords: Pandoc, Markdown, Latex

Running title: Markdown manuscript

Wordcount: 4,321

Abstract

Purpose. If journal requires abstract sections, these can be included and made bold by including double asterisks `**<section_name>**` before and after each header. If these are not needed, simply do not include them, simple. **Methods.** Because of our selected font, we can easily include special characters directly into our markdown file and these will be rendered properly. This provides some motivation to find a text editor that has a good *character map* plugin. Personally, I have grown rather fond of the Atom editor, which has a `character-map` plugin that does the trick. by clicking on a hotkey, `Alt-l`, I can type the name of the symbol I am looking and it is inserted into my document. **Results.** So lets say I just pressed `Alt-l` and typed `plu` \pm `minus`. The item at the top of the list would be \pm , as expected. I hit `Enter` and it is inserted into my document. This makes it easy to say that, on average, my level of frustration drafting a manuscript has decreased by $77\% \pm 4\%$ since learning about Markdown and Pandoc. Also, scientific papers always sound fancier when you use Greek letters. So I will search for the letter beta, which gives me the special character β . Also, because Pandoc converts Markdown to PDF using LaTeX, it is amazing support for mathematical equations and the like. Most basic tutorials on LaTeX should cover the basics of math-mode (i.e. `$stuff$`). **Conclusion.** This dummy

Figure 2: Word document. What the first page of the current document looks like when it is converted to .docx using Pandoc. Much of the formatting is correct, including the references list.

Draft-Revision Title

Pratik Patel and Another Author

February 9, 2013

This is an example of a ~~draft-revision~~ article. These are some types of changes to ~~expect~~be expected. Here is how it deals with equations:

$$y = \int (x^2 + 32) dx \quad (1)$$

When you do not include your collaborator's name in the document, they might get upset with you. But inclusion of their name in the final version will settle all scores.

Figure 3: latex-diff. An example of generating a PDF document of the differences between two documents.

References

- 369 **References**
- 370 **Bojsen-Moller J, Schwartz S, Kalliokoski KK, Finni T, Magnusson SP.** Intermus-
371 cular force transmission between human plantarflexor muscles in vivo. *J. Appl. Physiol.*
372 109: 1608-1618, 2010.
- 373 **Diong J, Gandevia SC, Nguyen D, Foo Y, Kastre C, Andersson K, Butler JE, Héroux**
374 **ME.** Small amounts of involuntary muscle activity reduce passive joint range of motion.
375 *J Appl Physiol* 127: 229-234, 2019.
- 376 **Diong J, Herbert RD.** Is ankle contracture after stroke due to abnormal intermuscular
377 force transmission? *J. Appl. Biomech.* 31: 13-18, 2015.
- 378 **Diong J, Herbert RD, Kwah LK, Clarke JL, Harvey LA.** Mechanisms of increased
379 passive compliance of hamstring muscle-tendon units after spinal cord injury. *Clin.*
380 *Biomech.* 27: 893-898, 2012a.
- 381 **Diong JH, Herbert RD, Harvey LA, Kwah LK, Clarke JL, Hoang PD, Martin JH,**
382 **Clarke EC, Bilston LE, Gandevia SC.** Passive mechanical properties of the gastrocne-
383 mius after spinal cord injury. *Muscle Nerve* 46: 237-245, 2012b.
- 384 **Huijing PA, Baan GC.** Myofascial force transmission: muscle relative position and
385 length determine agonist and synergist muscle force. *J. Appl. Physiol.* 94: 1092-1107,
386 2003.