Template to prepare preprints and manuscripts using markdown and github actions

Timothée Poisot [1,2,‡], Peregrin Took [3,4], Merriadoc Brandybuck [4,5,‡]

[1] Université de Montréal; [2] Québec Centre for Biodiversity Sciences; [3] Inn of the Prancing Pony; [4] Fellowship of the Ring; [5] Green Dragon Inn
[‡] Equal contributions

Correspondance to:
Timothée Poisot — timothee.poisot@umontreal.ca

Purpose: This template provides a series of scripts to render a markdown document into an interactive website and a series of PDFs.
Internals: GitHub actions and a series of python scritpts. The markdown is handled with `pandoc`.
Motivation: It makes collaborating on text with GitHub easier, and means that we never need to think about the output.

Keywords:
pandoc
pandoc-crossref
github actions

This template templates turns README files hosted in a GitHub repo into a formatted manuscript. In practice, this involves converting the `README.md` file into various LaTeX files, an interactive website, and an OpenDocument text file (for use in Word, or to generate track changes).

The workflow is *entirely* GitHub-based, so the manuscript file is contained entirely in the README. There are a few differences with "normal" markdown (or with GFM, the most commonly used variant). First, because manuscripts require extensive metadata, the metadata are stored in a JSON file, as opposed to a YAML front-matter. Second, we use the `pandoc-crossref` filter to generate references to figures and tables, which are not rendered when looking at the README on Github. Everything else is standard markdown.

There are *two* important folders: `figures` and `appendix`. Both of them will remain in the repository even if they are empty (they come with a `.gitkeep` file). The `figures` folder is where figures are stored, and the `appendix` folder is where you can put markdown files for the appendices. Note that the appendices will come with their *own* bibliography (which draws from the global `references.bib` file at the root of the project).

1 The typesetting

The *actual* typesetting is handled by another repository, which is located at PoisotLab/manuscript-typesetter. If you don't feel like running foreign code on your repo, you are absolutely correct. This is why the GitHub action file associated to this template will specify which release of the typesetter will be downloaded, so that you can inspect what the typesetting steps are doing. The typesetting versions follow semantic versioning, so as long the *major* (as in `major.minor.patch`) version remains the same, you will not need to change your `README.md` or `metadata.json`.

In short: they convert the bibliography into a CSL JSON, reformat the metadata so they are usable with pandoc templates, downloads a whole bunch of binaries to do the typesetting (as well as the TeX Gyre fonts we use for text, and the `JuliaMono` font we use for code), and then return everything as a compressed file, which is then deployed using GitHub pages.

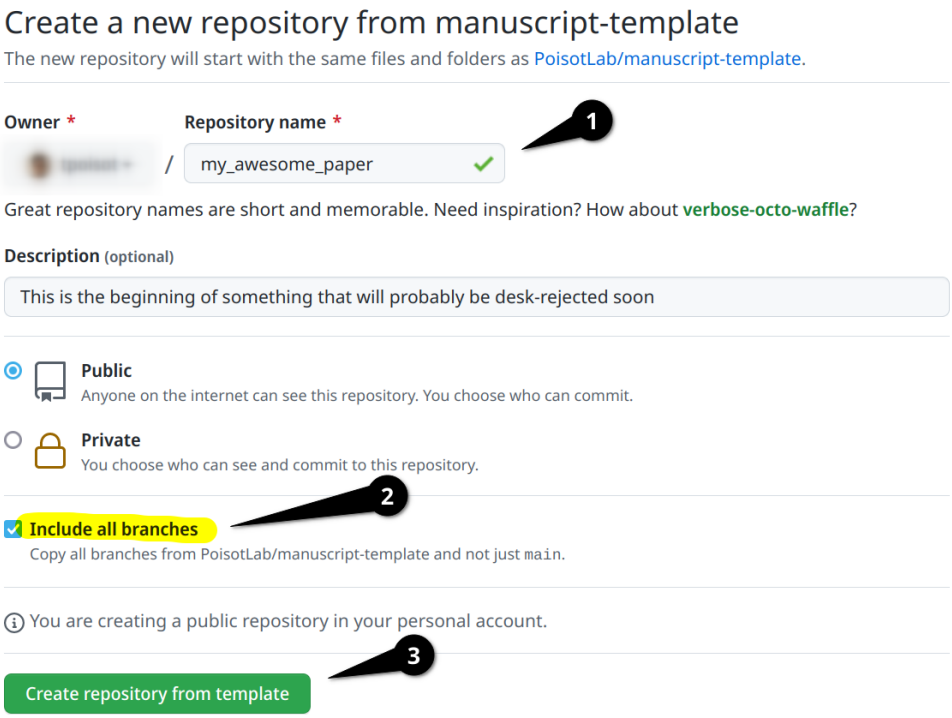

Figure 1 The important steps in using this template are (1) to pick the new name for your repository, (2) to make sure "Include all branches" is checked, and (3) to finish the process by clicking on "Create repository from template".

All documents will be deployed to `gh-pages` *only* on push events from the `main` branch. All of the artifacts will be built when doing pull requests, so you can check that merging a branch is *not* going to cause the compilation of the documents to fail; indeed, you can download the artifacts produced during the run, to check the PDF and html files. The website is only updated from the `main` branch. When doing a pull request, you will be notified of the word count of the manuscript (approximate, and excluding references).

If you want to use the typesetting system locally, *you can*. We do not recommend it, but you absolutely can. All you need to do is make sure it is downloaded in the `.typesetter` folder in your repository (which is going to be ignored by `git` by default), and then install `Julia` (we know…), and then run the top-level script. Again, we definitely advise against running things locally.

2 Deploying the template

The process of deploying this template has been *greatly* streamlined from previous versions. Click on the "Use this template" button at the top of the Github repo, making sure to check the option to import all branches (this will import `gh-pages` and allow deploys to start immediately). An example is given in fig. 1.

When this is done, you can open your own repository, edit `README.md` with your own text, commit, and push. This will trigger the website build, which will become available at `http://you.github.io/repo-name/`. You can then edit the `references.bib` file, change the `logo.png` file, and update the `metadata.json` file. Every time you push a new commit, everything will be updated.

3 The metadata file

The `metadata.json` file provided with this template is reasonably complete, but the full format is explained in the Supp. Mat. on the metadata specification. From the website, Supp. Mat. can be accessed at the very bottom of the page. # References management

The references are managed by `pandoc`. References *must* be stored in a `references.bib` file, and that it would make sense to order it alphabetically by key, but this makes no difference.

We use Zotero for references management, and for the lab's manuscripts, we work from folders in a shared library (with a folder for every manuscript).

We recommend the Better BibTeX plugin for citation key generations, and auto-export of the shared library to the `references.bib` file. We use a citation key format meant to convey information on the author (first author full name), date (complet year), and title (first non-stop-word word of the title). It must be set in the Better BibTeX preferences as (you might need to remove the line changes):

```
auth.fold.fold + year + title.fold.nopunctordash.skipwords.lower.select(1,1).capitalize()
```

It is a good idea to configure Better BibTeX to auto-export on change, and to remove a lot of fields that are not strictly speaking required for references. The list of fields we usually ignore is:

```
abstract,copyright,annotation,file,pmid,month,shorttitle,keywords
```

The citations are done using the normal markdown syntax, where `@Elton1927AniEco` produces Elton (1927), and `[@Camerano1880EquViv]` produces (Camerano 1880).

4 Figures, Tables, and other things

Note that you can wrap the text of legends for both figures and tables. This avoids the issue of having very long lines.

4.1. Mathematics

The following equation

$$J'(p) = \frac{1}{\log(S)} \times \left(-\sum p \times \log(p)\right) \tag{1}$$

is produced using

```
$$J'(p) = \frac{1}{\text{log}(S)}\times ... $$ {#eq:eq1}
```

and can be referenced using `@eq:eq1`, which will result in eq. 1. Note that because we use `pandoc-crossref`, the label "eq." will be generated automatically.

4.2. Tables

Table legends go on the line after the table itself. To generate a reference to the table, use `{#tbl:id}` – then, in the text, you can use `{@tbl:id}` to refer to the table. For example, the table below is tbl. 1. You can remove the *table* in front by using `!@tbl:id`, or force it to be capitalized with `\*tbl:id`.

Table 1 This is a table, and its identifier is `id` – we can refer to it using `{@tbl:id}`. Note that even if the table legend is written below the table itself, it will appear on top in the PDF document.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

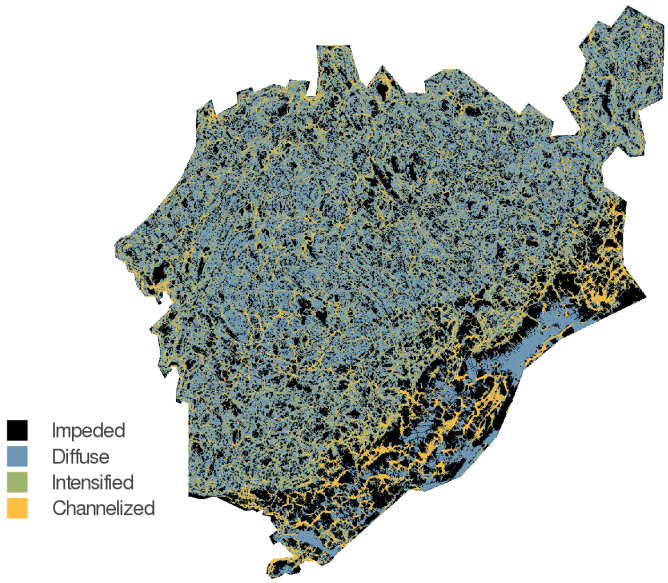

Figure 2 This is the legend of the figure, which will be shown in the margin in preprint mode, and underneath the figure in draft mode. The legend can contain references, etc. It is advised to use a resolution of at least 600dpi for the figures.

5

Figures

Figures can have a legend – all figures *must* be in the `figures/` folder of the project, as it is also used for the website. We recommend to use good resolution images, rather than PDFs, or at least to have multiple versions available.

```
![This is the legend of the figure...](figures/figure.png){#fig:figure}
```

We can now use `@fig:figure` to refer to fig. 2.

6

Unicode support

The PDFs are rendered using the excellent `tectonic` engine, and use `fontspec` for better looking fonts. The text is set in TeX Gyre Termes, the titles are set in TeX Gyre Heros, and the code is set in JuliaMono.

The text can use unicode: for example, this α is written as is in the main text. The code can, similarly, use unicode. For example, we can use the one-liner neural network from the beautiful algorithms repository:

```
neural_network(x, V, w, φ, g) = w ⋅ map(vⱼ → g(vⱼ ⋅ φ(x)), V)
```

Every character that `JuliaMono` supports (which is *a lot* can be used in code).

References

Camerano, L. (1880). Dell'equilibrio dei viventi merce la reciproca distruzione. *Atti Della R. Accad. Delle Sci. Torino*, 15, 393–414.

Elton, C.S. (1927). *Animal ecology*. University of Chicago Press.