# A template for writing scientific papers

**Andrew G. York**[1*]

*[1]Calico Life Sciences LLC, South San Francisco, CA 94080, USA*

*[*]Permanent email: andrewg.york+template@gmail.com*

**A template for formatting scientific papers using HTML, CSS, and Javascript.**

doi: 10.5281/zenodo.231328

Note that this is a limited PDF version; animated and interactive figures are disabled. For the full version of this article, please visit http://andrewgyork.github.io/publication_template

## Introduction

Version control tools like Git, Github, and Github Pages are great for developing, sharing, and documenting code. Why not for science, too? Consider the following workflow:

- **Discover or invent something.** This is supposed to be the hard part. For me, this invariably involves writing wild and wooly exploratory Python code, which goes in a private Github repository, if I'm behaving myself.
- **Collect results into intelligible figures.** I usually do this via Python code, which also belongs in a private Github repository.

- **Write explanatory text.** I've always found this step is a mess. No set of tools I've tried (e.g. Word, LaTeX, Google Docs) fit my workflow (e.g. collaboration, version control, formatting). An HTML/CSS template with math typesetting, code syntax highlighting, and automatic reference listing (via Javascript) fits me perfectly - and allows collaborative writing via a private Github repo!
- **Circulate preprints.** Existing options like email, ArXiv, or BioArXiv are great, but it's 2016 - why are we using PDFs? The web makes sharing data, code, animations, and interactive figures straightforward. If the code, data, and HTML are already on Github, then sharing a preprint is as simple as activing Github Pages.
- **Submit for peer review.** Historically, a huge mess. Much has been written about the unfortunate state of academic publishing. Why not cut out the middleman and solicit review directly from qualified colleagues? (A much longer discussion, and beyond the scope of this document - but I want to try it.)
- **Archive for posterity.** I want my work to exist after I die, in an open, available and citable form. Github, Github Pages, and Zenodo address these problems cleanly (although there's room for improvement).

I've been careful about dependencies; Git and Github are helpful, but not required. You can download a `.zip` of this repository, edit `index.html` with a text editor, and view the results with a web browser. I avoid Javascript which loads via the web, so you can work offline. My figure generation code depends on the Scipy stack, but this is easy to swap out; all you have to do is generate static images, put them in the proper folder, and link to them from `index.html`. You can generate your static images using any tool you choose. If you happen to use code to generate your figures, though, great! You can put this code in the `figure_generation` directory, enabling version control and simple reproduction of your results by others.

Writing directly in HTML/CSS/Javascript isn't for everyone. It's fiddly, has a learning curve, and requires obsessive attention to detail. In my experience, writing a scientific publication already fits this description, so it's a small price to pay, and one I'm already paying.

## Math typesetting

I use MathJax for typesetting. I want to be able to work offline, but a local MathJax copy is huge, so I followed the recipe described by Tibor Simon to make a minimal version. The recipe's pretty cool; you delete most of MathJax, try to load it, watch your browser console, and copy files back one at a time until your browser stops complaining. I only support LaTex input; if you want more, you can follow Tibor's recipe and replace the contents of `/javascript/Minimal-MathJax/`.

Examples of paragraph equations:

$$\prod_{\substack{1\leq i\leq n\\1\leq j\leq m}} M_{i,j}$$

$$L' = L\sqrt{1-\frac{v^2}{c^2}}$$

$$\begin{aligned} B' &= -\nabla \times E,\\ E' &= \nabla \times B - 4\pi j, \end{aligned}$$

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2+\frac{1}{a_3+a_4}}}$$

Example of inline math: $\frac{a^3}{b}$

## Code syntax highlighting

Prism.js makes HTML syntax highlighting easy. I only included Python highlighting; if you want more, download a freshly customized version from prismjs.com and replace `/javascript/python-highlighting/prims.js`.

Example python syntax highlighting:

```
import antigravity
antigravity.fly()
print("Whee!")
```

## Automatic reference list

An example of how to include a citation: [Eswaramoorthy2014]. If you inspect the HTML, you'll see that inserting the citation is super clumsy, but at least you don't have to keep track of uniqueness, ordering, etc.

## Static and interactive figures

Version control isn't a great way to organize images, but you can get away with it if there aren't too many, and they don't change too often. Figure 1 uses an image stored in the local repository.

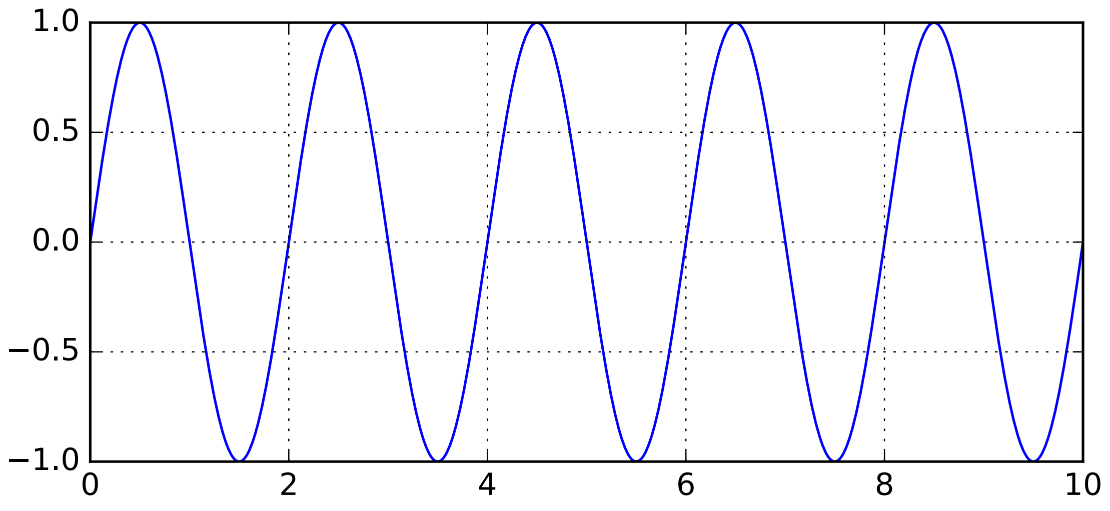
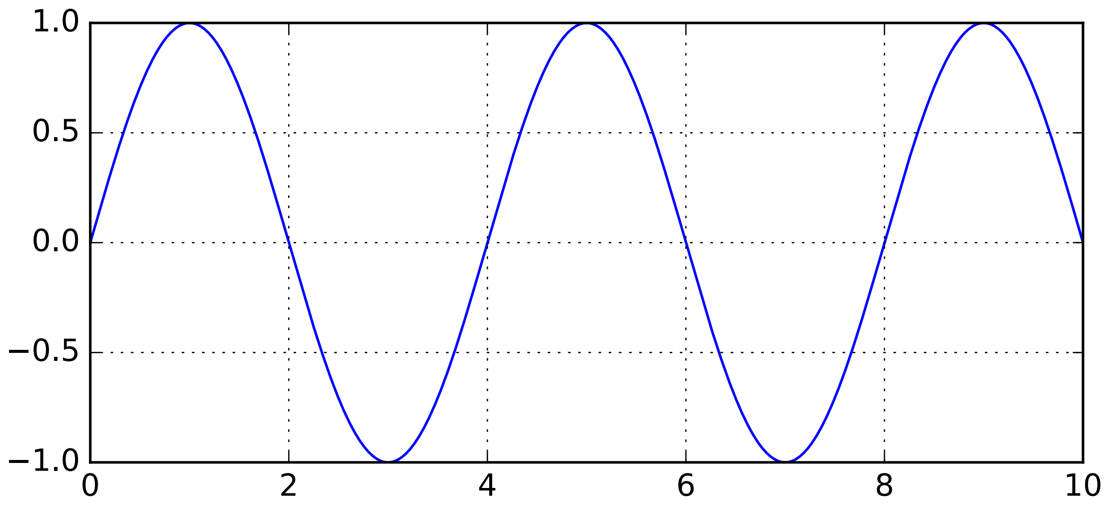


**Figure 1: A static figure using a local image.** This image was generated by `/figure_generation/figure_1.py`.

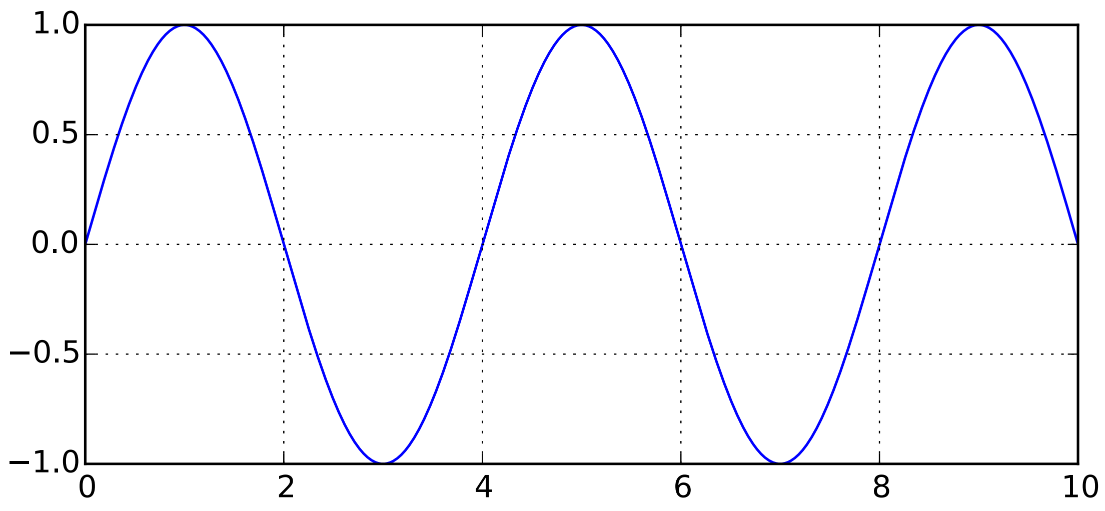
Figure 2 is interactive; the images are pre-computed (by Python code, in this case) and stored in the local repository. Simple javascript changes the figure's `img.src` when the figure's `select` changes. The images are small and don't change much, so it's not horrible to store them in this repository. A similar figure with substantially more images would stretch the limits of this approach.



| | | |
|---|---|---|
| Period: | 4 ▼ | (change this to adjust the period of the sinusoidal plot) |

**Figure 2: An interactive figure using local images.** This figure's images were pre-generated by `/figure_generation/figure_2.py`, and switched dynamically by `/javascript/update_figures.js`.

Interactive figures can potentially contain huge numbers of images, which are clumsy to store in a version-controlled repository. My current preferred solution is to store such images in an auxiliary location. During the writing process, we generate these images automatically using the figure generation code and store them locally. Once the paper is ready to publish, we host a copy of these images in a second auxiliary repository. The figure update Javascript finds the appropriate image source for us.



Period: 4 ▾ (change this to adjust the period of the sinusoidal plot)

**Figure 3: An interactive figure using images that aren't stored in this repository.** This figure's images were pre-generated by `/figure_generation/figure_3.py`, and switched dynamically by `/javascript/update_figures.js`.

## References

1. [Eswaramoorthy2014] Asymmetric Division and Differential Gene Expression during a Bacterial Developmental Program Requires DivIVA; Prahathees Eswaramoorthy, Peter W. Winter, Peter Wawrzusin, Andrew G. York, Hari Shroff, Kumaran S. Ramamurthi; PLoS Genet 10 (8), e1004526 http://dx.doi.org/10.1371/journal.pgen.1004526