Making research reproducible using literate programming

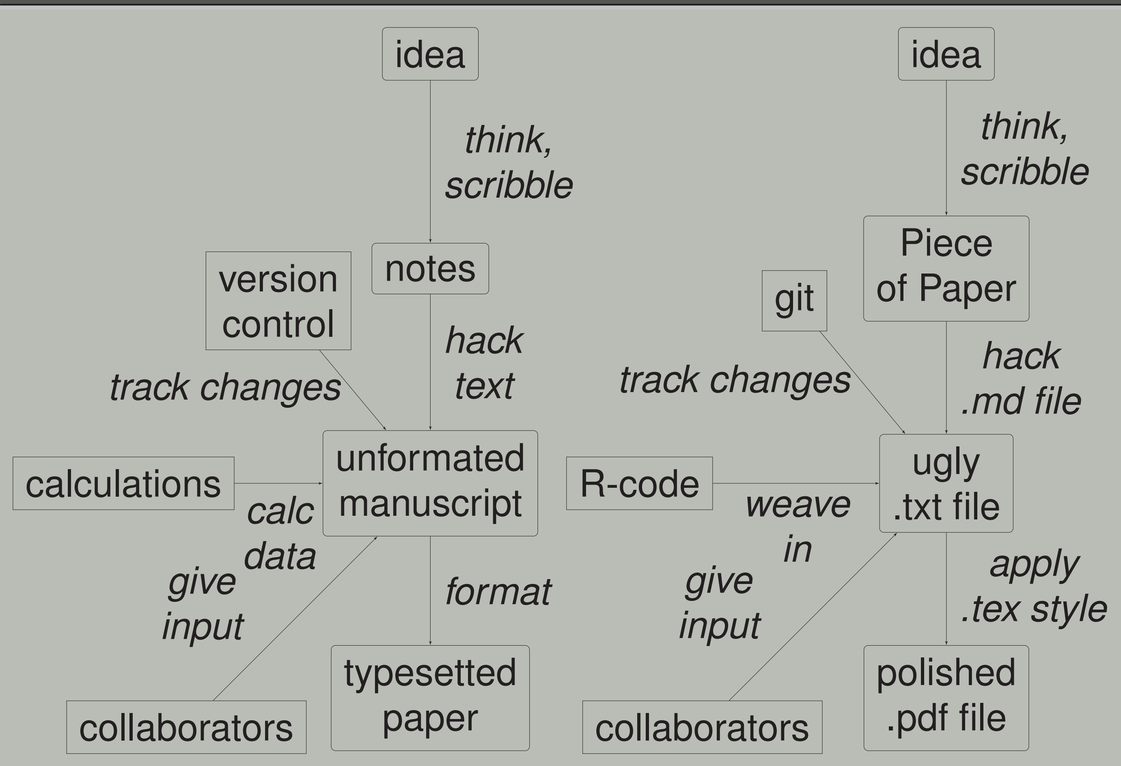
Sebastian Sauer, Sandra Sülzenbrück, Yvonne Ferreira, and Christoph Kurz

FOM University of Applied Sciences, Helmholtz Zentrum München

Reproducibility: What and why?

There is an increasing concern about the reliability of research results [3]. One reason is that it has been found that many published results cannot be replicated [2]. In parts, this can be due to the fact that it is often hardly possible for independent researchers to confirm (ie., reproduce) the results of a research paper. Thus, making research more reproducible seems a pressing need [3]. Note that reproducibility means transparency, and lack of transparency is a serious threat to science. Here, we present a "recipe" for making (your) research papers more reproducible using literate programming. Literate programming refers to weaving programming code (ie., statistical calculations) with the paper text, ie., the normal text of a paper.

Workflow for reproducible paper writing



The *left* diagram shows the workflow *in theory*. The *right* diagram exemplifies useful tools for each step.

Criteria to help with writing scientific papers reproducibly

- **Simple:** The tool should be easy to learn and use (flat learning curve). Life is short.
- **Aesthetic:** The tool should set the text in an eye pleasing way, eg., no holes in block aligned text.
- **Plain:** The tool should work with plain text files, so that they can be readable in future. In addition, text file are more compatible with other tools, such as version control.
- **Versionized:** The tool should support versionizing (tracking changes), including changes from collaborators.
- **Citable:** The tool should be able to manage (scholarly) citations.
- **Flexible:** The tool should allow for flexible layout, ie., many possibilities for rendering the final, formatted output. This is often in outright contradiction to simplicity.

Which tools make the researcher happy?

	Word	Latex	Markdown	WebApps
simple	●●●	●	●●	●●
beautiful	●	●●●	●●●	●●
plain	●	●●●	●●●	●
versionized	●	●●●	●●●	●●●
citable	●●●	●●●	●●●	●●●
flexible	●●●	●●	●	●●
Σ	13	17	18	16

This table provides (subjective) evaluation of what a researcher needs for writing her paper in a reproducible and efficient way. Note: *Word* refers not only to MS Word, but to similar WYSIWYG text processors as well. *WebApps* refer to scholarly writing tools such as *Authorea*. *Markdown* refers to the *Pandoc* dialect (and extensions) of Markdown.

Stating the problem

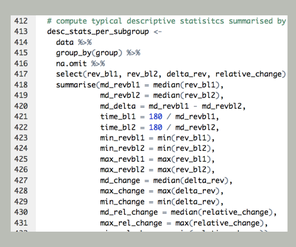


Figure: ∼ 1000 lines of code in your paper

Figure: ∼ 20 numbers as results of computations somewhere in your paper

Assume there is not a direct link of your (statistical) computations to the respective number in the text. It will be hard to understand for someone unfamiliar with your work which calculations have yielded some figure in your paper. So confirming that the calculations are valid will be hard.

Software tools for reproducible writing

Logo	Name	Description
	R	Environment for statistical calculations and programming [4]
	R-Studio	IDE for R including features for version control (eg git), and literal programming (Knitr, Pandoc, Markdown)
	Knitr	R-package to "knit" (or "weave") R-code into plain text as a way of functional programming [5]
	Git	version control tool, handles multiples collaborators, tracks changes in text files
PANDOC	**Pandoc**	"swiss army knife" for converting markup text, eg. Markdown → PDF
	Markdown	Similar aims as Latex, but you learn it in 5 minutes.

Main commands to turn R+Markup-Text into polished PDF-paper

- Weave code in text using **knitr**, giving a R-Markdown (.rmd) (or R-Latex, .rnw) file: "Mean reaction time was `` `r round(mean(rt), 2)` `` ms". → "Mean reaction time was 433.30 ms".
- Knit R-Text-Mixup to pure text plus markup using R-package **knitr:**
  `knitr(source.rmd)`
- Convert Markdown to Latex-PDF with **Pandoc:**
  `pandoc source.md --output paper.pdf ...`

Conclusion

- Note that complex cognitive steps such as creative thinking, drafting a paper outline, writing an argument, debugging code etc. are best dealt with sequentially, as depicted in the workflow. Our brain is not so fit for multitasking [1].
- A great combination of tools for efficient and reproducible writing papers is: R+RStudio+knitr+markdown+pandoc+git.
- Writing in `Markdown` is nice, because little markup clutters the view. Pandoc then compiles to latex, applying some Tex-template of your flavor.
- However, `Pandoc` is not (yet) flexible enough to adjust every layout detail one may think of.
- Slides using `Markdown` or Latex/Beamer are no pleasure to the eay.
- Looking forward to the time when it is possible to write in R+Markdown and compile to Tex easily with all kinds of style-templates!

References

[1] Wesley C Clapp, Michael T Rubens, Jasdeep Sabharwal, and Adam Gazzaley. Deficit in switching between functional brain networks underlies the impact of multitasking on working memory in older adults. *Proceedings of the National Academy of Sciences*, 108(17):7212–7217, apr 2011.

[2] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251), aug 2015.

[3] Roger Peng. The reproducibility crisis in science: A statistical counterattack. *Significance*, 12(3):30–32, jun 2015.

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[5] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2016.