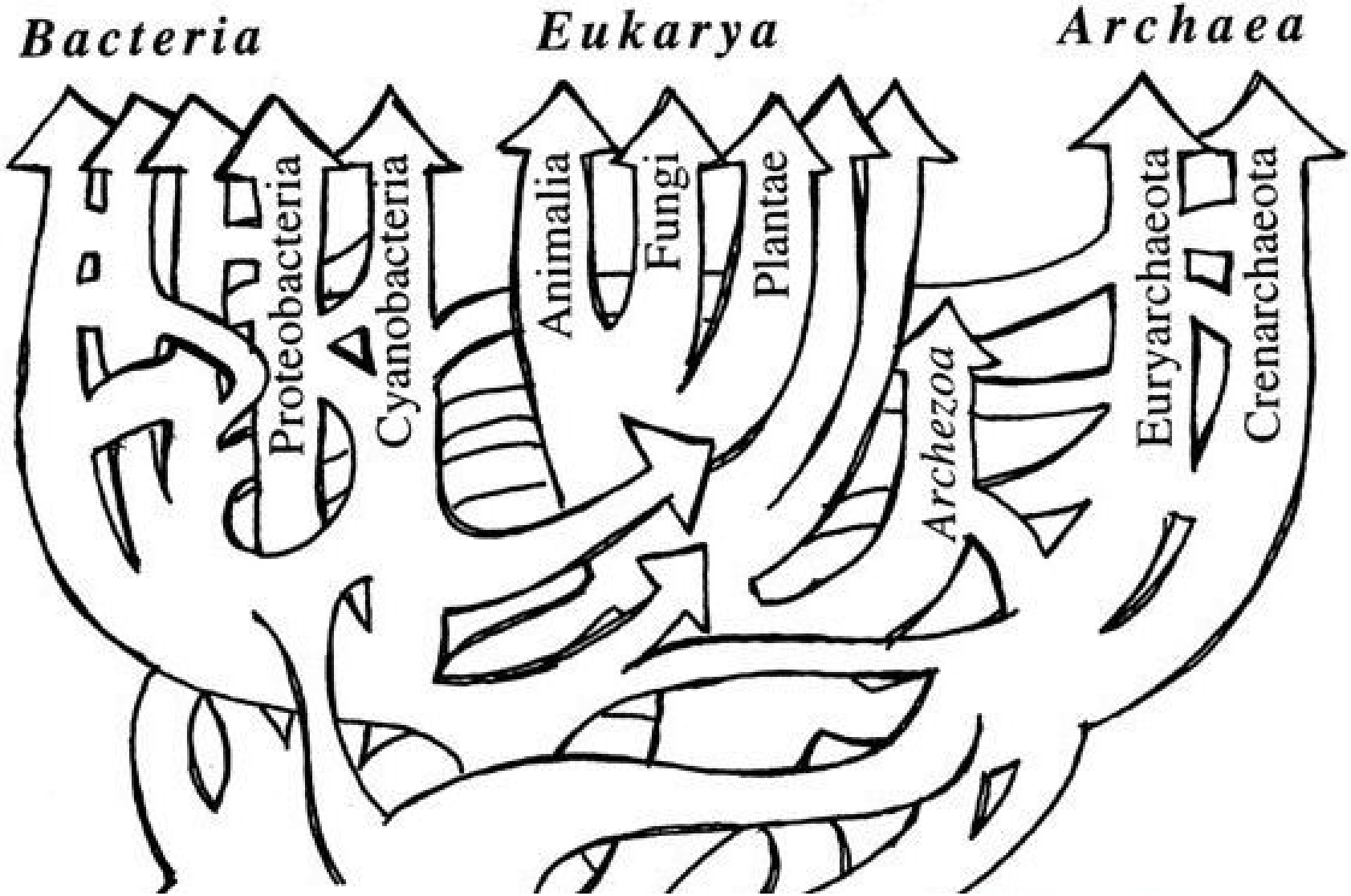
Digital Lab Journalling with Git

Katrin Leinweber

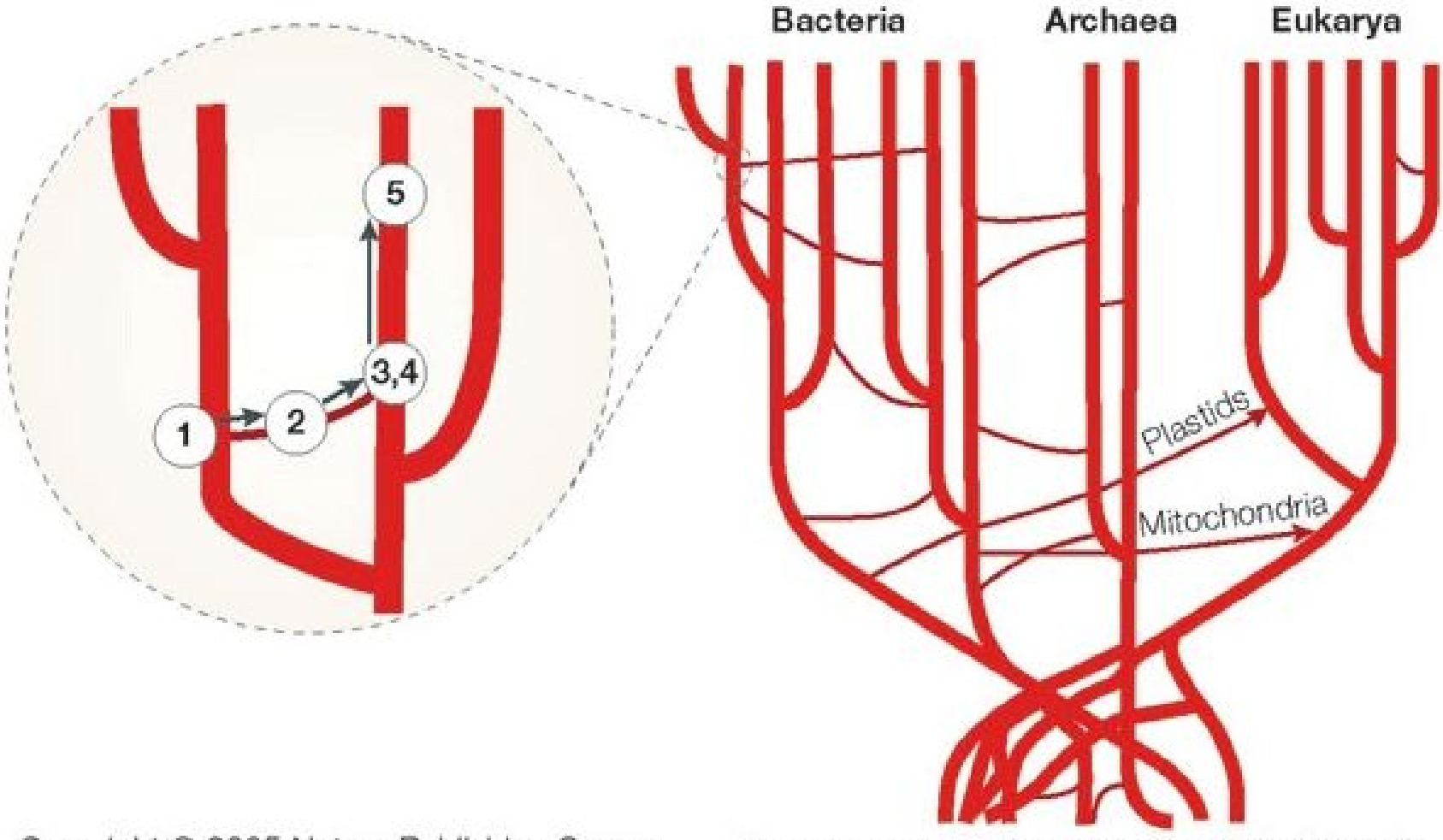
2015-Apr-13

Slides of this presentation with recorded audio here

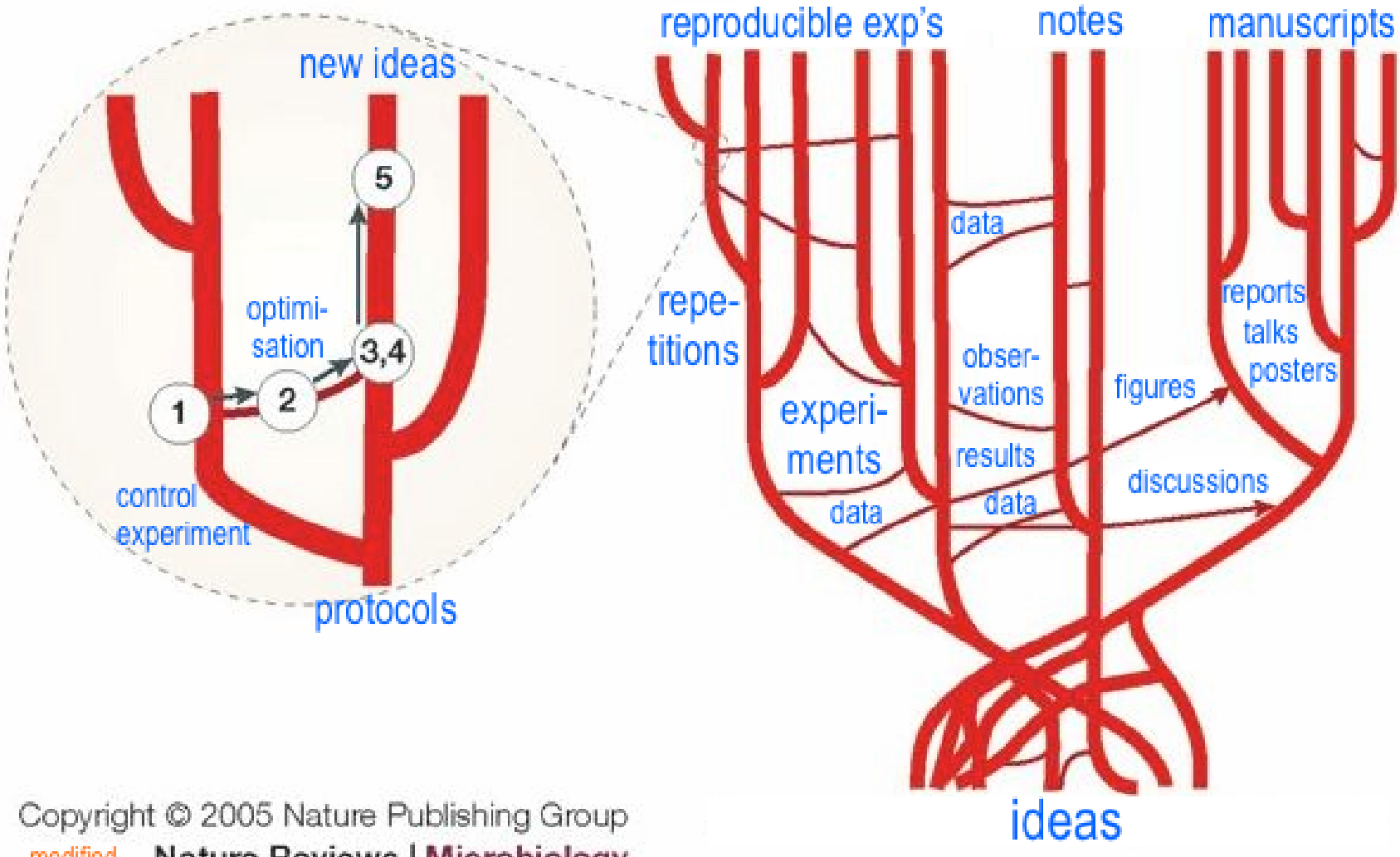
Tree of Life


Bacteria
Eukarya
Archaea
Proteobacteria
Cyanobacteria
Animalia
Fungi
Plantae
Archezoa
Euryarchaeota
Crenarchaeota

Tree of Life



Tree of Projects



Git tracks digital evolution of files

- highlights exactly what changed, line by line
- makes you comment file changes

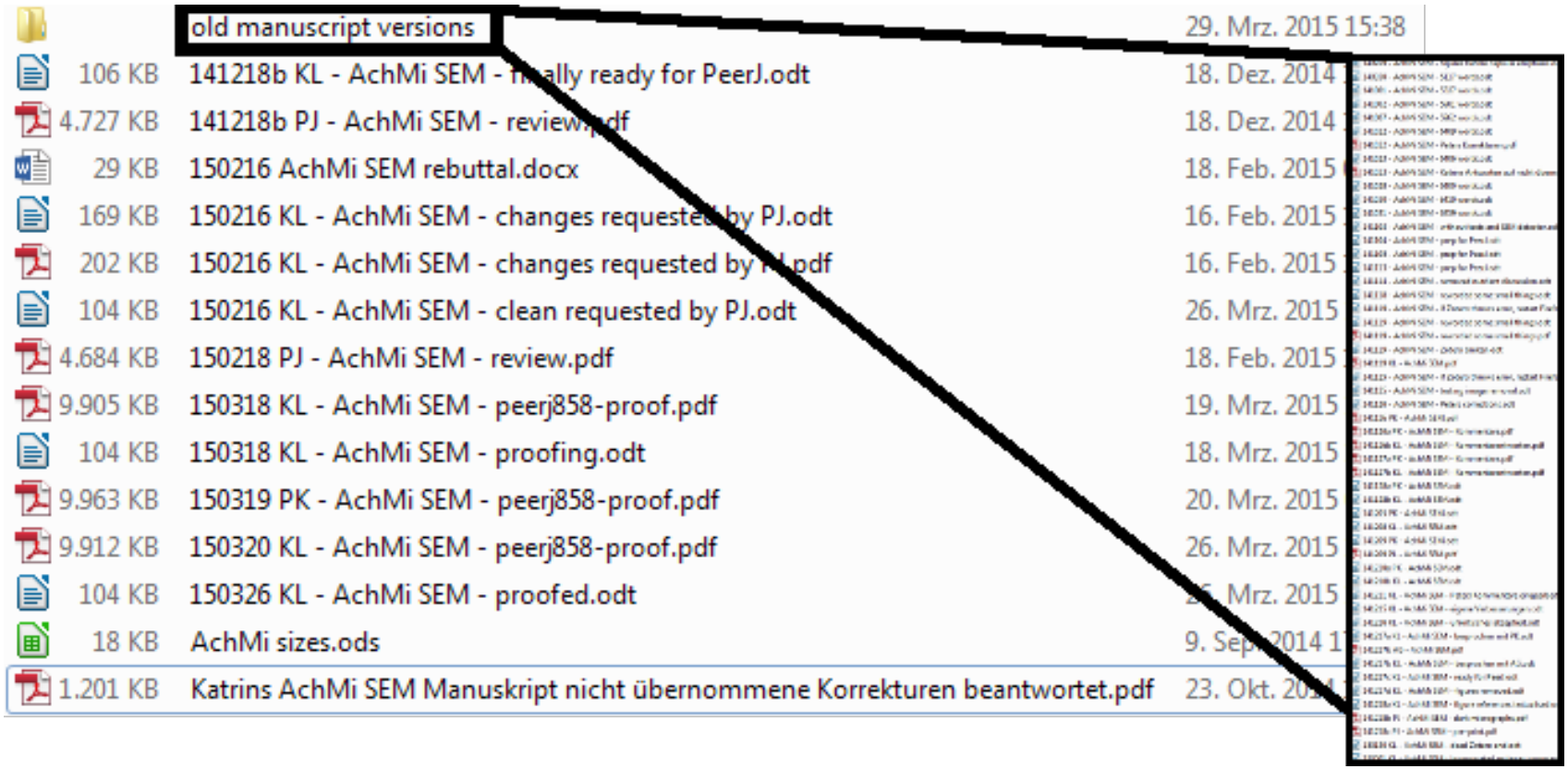
proof-read

 Katrin Leinweber 158fc2f

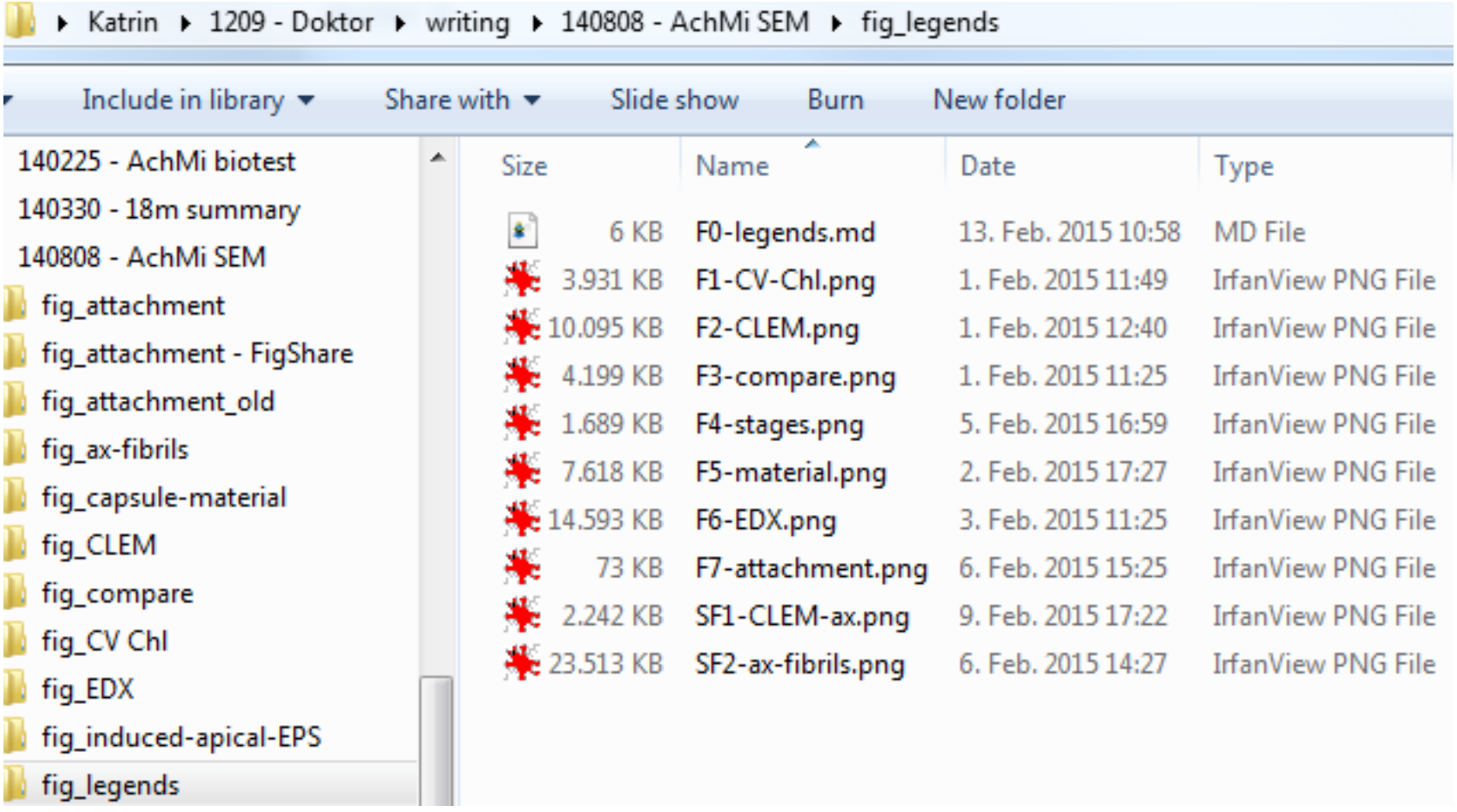
```
12  12
13      - Identification of *A. minutissimum* capsules (asterisks)
          by subsequent observation of cell clusters by both
          bright-field and scanning electron microscopy of xenic
          biofilm (scale bars: 5 µm).
    13  + Identification of *A. minutissimum* capsules (asterisks)
          by successive observation of cell clusters by first
          bright-field and then scanning electron microscopy of
          xenic biofilm (scale bars: 5 µm).
14  14
```

Git is your file watchdog.

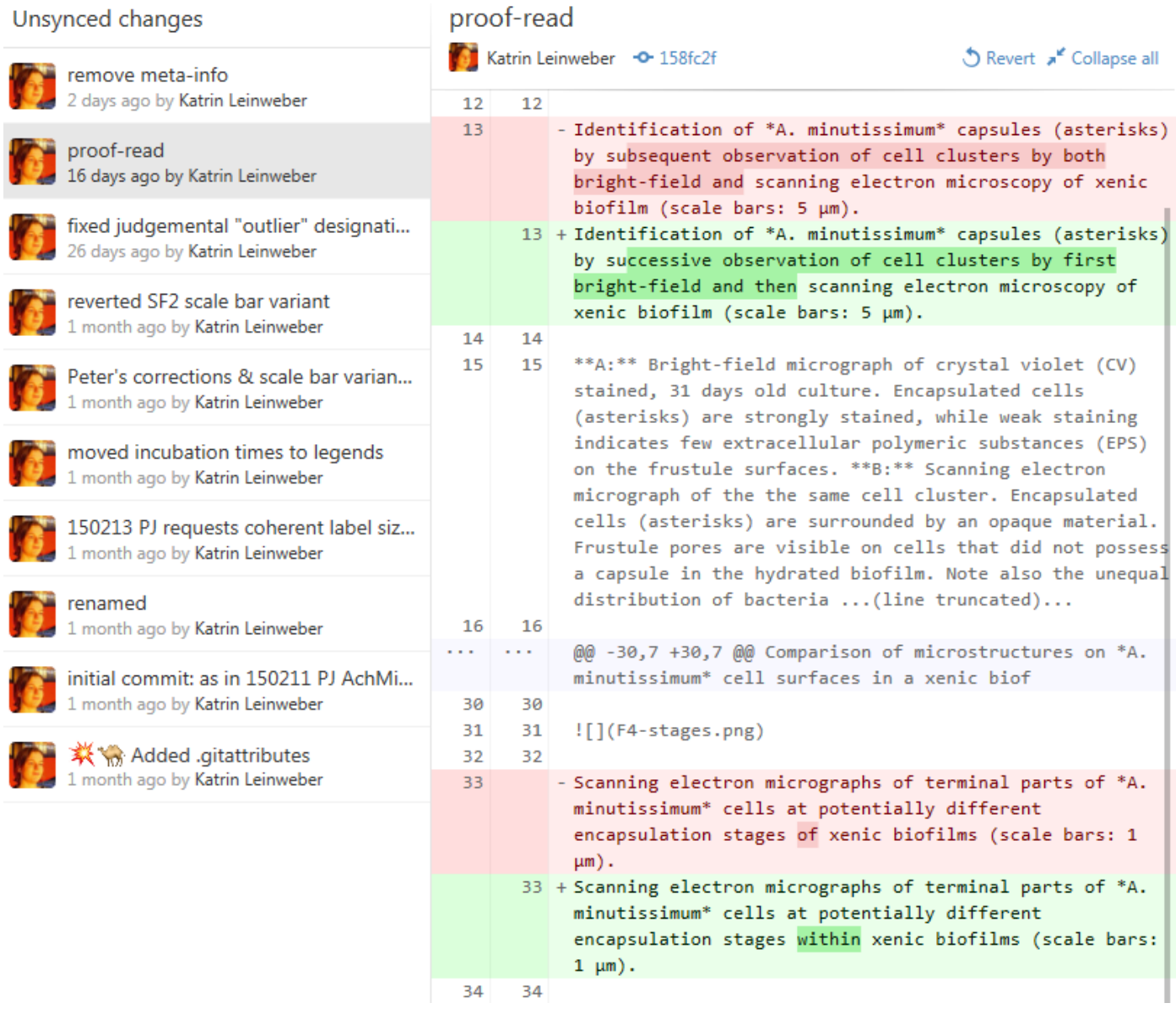
Good against: too many files



Good for: tidy project folders, but. . .



Good for: . . . version history still accessible



Version control with Git

Technical

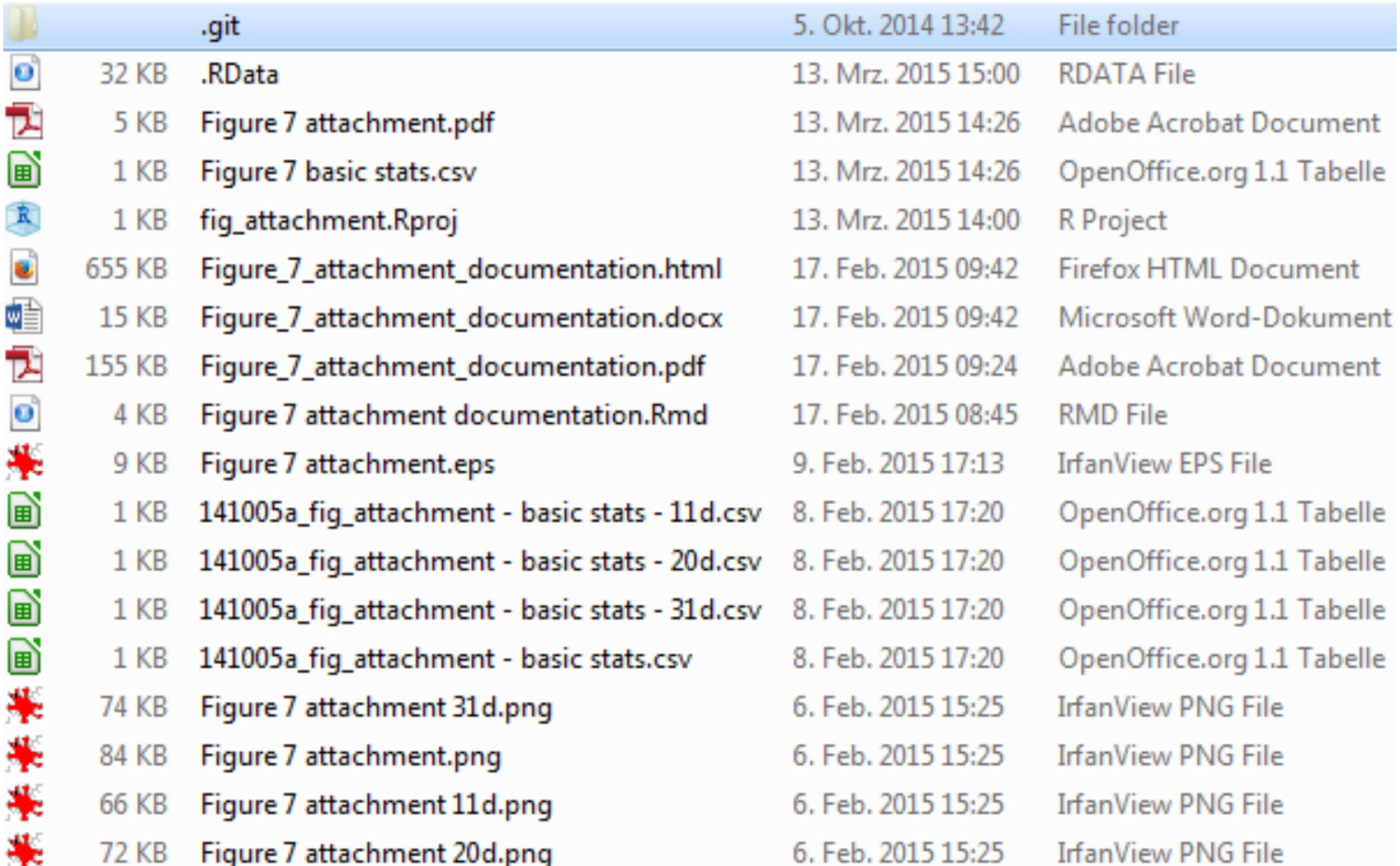
- works best for line- or paragraph-based files
- built for source code, but also useful for texts and simple images
- less useful for complex or large file types
- not a substitute for full backups

Cultural

- helps build self-explanatory & verifiable project documentation (for your future self, colleagues, students, reviewers. . .)
- enables less painful collaboration on same set of files
- conflicts only if same lines in same file get changed
- for text files: better than syncing tools

Basic Git vocabulary: repository/repo

- project folder watched by Git
- is still a normal folder: add, edit & delete files normally
- database of file versions in hidden subfolder



	.git	5. Okt. 2014 13:42	File folder
32 KB	.RData	13. Mrz. 2015 15:00	RDATA File
5 KB	Figure 7 attachment.pdf	13. Mrz. 2015 14:26	Adobe Acrobat Document
1 KB	Figure 7 basic stats.csv	13. Mrz. 2015 14:26	OpenOffice.org 1.1 Tabelle
1 KB	fig_attachment.Rproj	13. Mrz. 2015 14:00	R Project
655 KB	Figure_7_attachment_documentation.html	17. Feb. 2015 09:42	Firefox HTML Document
15 KB	Figure_7_attachment_documentation.docx	17. Feb. 2015 09:42	Microsoft Word-Dokument
155 KB	Figure_7_attachment_documentation.pdf	17. Feb. 2015 09:24	Adobe Acrobat Document
4 KB	Figure 7 attachment documentation.Rmd	17. Feb. 2015 08:45	RMD File
9 KB	Figure 7 attachment.eps	9. Feb. 2015 17:13	IrfanView EPS File
1 KB	141005a_fig_attachment - basic stats - 11d.csv	8. Feb. 2015 17:20	OpenOffice.org 1.1 Tabelle
1 KB	141005a_fig_attachment - basic stats - 20d.csv	8. Feb. 2015 17:20	OpenOffice.org 1.1 Tabelle
1 KB	141005a_fig_attachment - basic stats - 31d.csv	8. Feb. 2015 17:20	OpenOffice.org 1.1 Tabelle
1 KB	141005a_fig_attachment - basic stats.csv	8. Feb. 2015 17:20	OpenOffice.org 1.1 Tabelle
74 KB	Figure 7 attachment 31d.png	6. Feb. 2015 15:25	IrfanView PNG File
84 KB	Figure 7 attachment.png	6. Feb. 2015 15:25	IrfanView PNG File
66 KB	Figure 7 attachment 11d.png	6. Feb. 2015 15:25	IrfanView PNG File
72 KB	Figure 7 attachment 20d.png	6. Feb. 2015 15:25	IrfanView PNG File

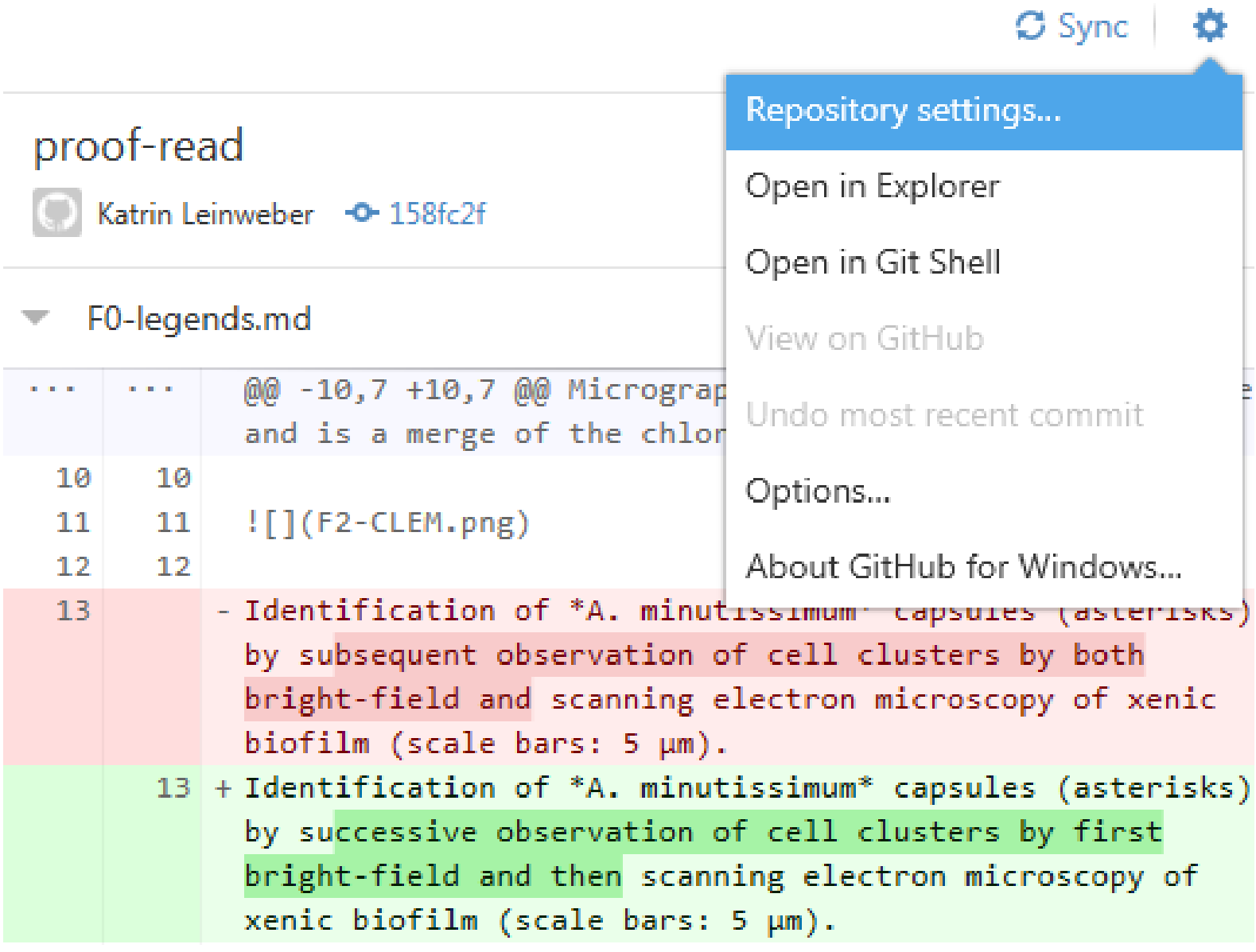
Basic Git vocabulary: committing

- saving a logical set of file changes in Git, together with meaningful comment
- changes can be within single file, or across different ones
- **commit history:** self-explanatory & verifiable documentation

Basic Git vocabulary: .gitignore

- file that contains a repo's rules for ignoring files
- No preview of highlighted changes? Large file? Auto-generated? Rather ignore those!

Basic Git vocabulary: .gitignore in GitHub for Windows



Vocabulary summary

repository/repo

- project folder watched by Git, plus hidden database of file versions

committing

- composing logical set of file changes and meaningful commit message

.gitignore

- a repo's ignore rules for non-essential files & file types

How to start using Git?

1. download client from windows.github.com (any other from git-scm.com/downloads/guis) & install
2. start with small sub-project (protocol optimisation, diagram in R, report, etc.)
3. **besides working on the files themselves:** commit logically connected changes often
4. **recommended:** start writing texts in Markdown[1] format
5. **optional (for collaboration etc.):** create account with Git hosting service like GitLab or GitHub

[1] Possible in Word with Writage add-in

Thanks to & Further Reading

Writing with Git

- "Markdown Basics" and "Mastering Markdown" by GitHub
- "Scientific Markdown" by Jens Erat
- "Paper Now" by PeerJ

Git in Science

- "Git/GitHub: a Primer for Researchers" by Carly Strasser
- "We Need a Github of Science" by Marcio von Muhlen
- "Git for Scientists" by Molly Gibson

Git concepts & tech

- "Learn Version Control with Git" by Fournova
- "10 Years of Git: An Interview with Git Creator Linus Torvalds" by Jennifer Cloer

Thanks for your attention! Questions?