


DATA + CODE + SOFTWARE = PDF:

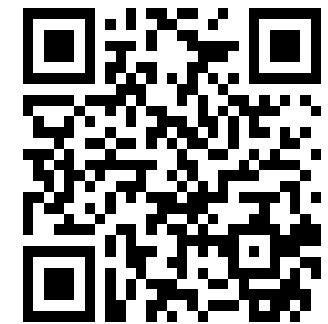
HOW TO WRITE A REPRODUCIBLE RESEARCH ARTICLE

Adina Wagner

 @AdinaKrik

Psychoinformatics lab,
Institute of Neuroscience and Medicine, Brain & Behavior (INM-7)
Research Center Jülich
ReproNim/INCF fellow

Slides: DOI 10.5281/zenodo.5508797 (Scan the QR code)
Sources: github.com/adswa/talk-reproducible-research-articles

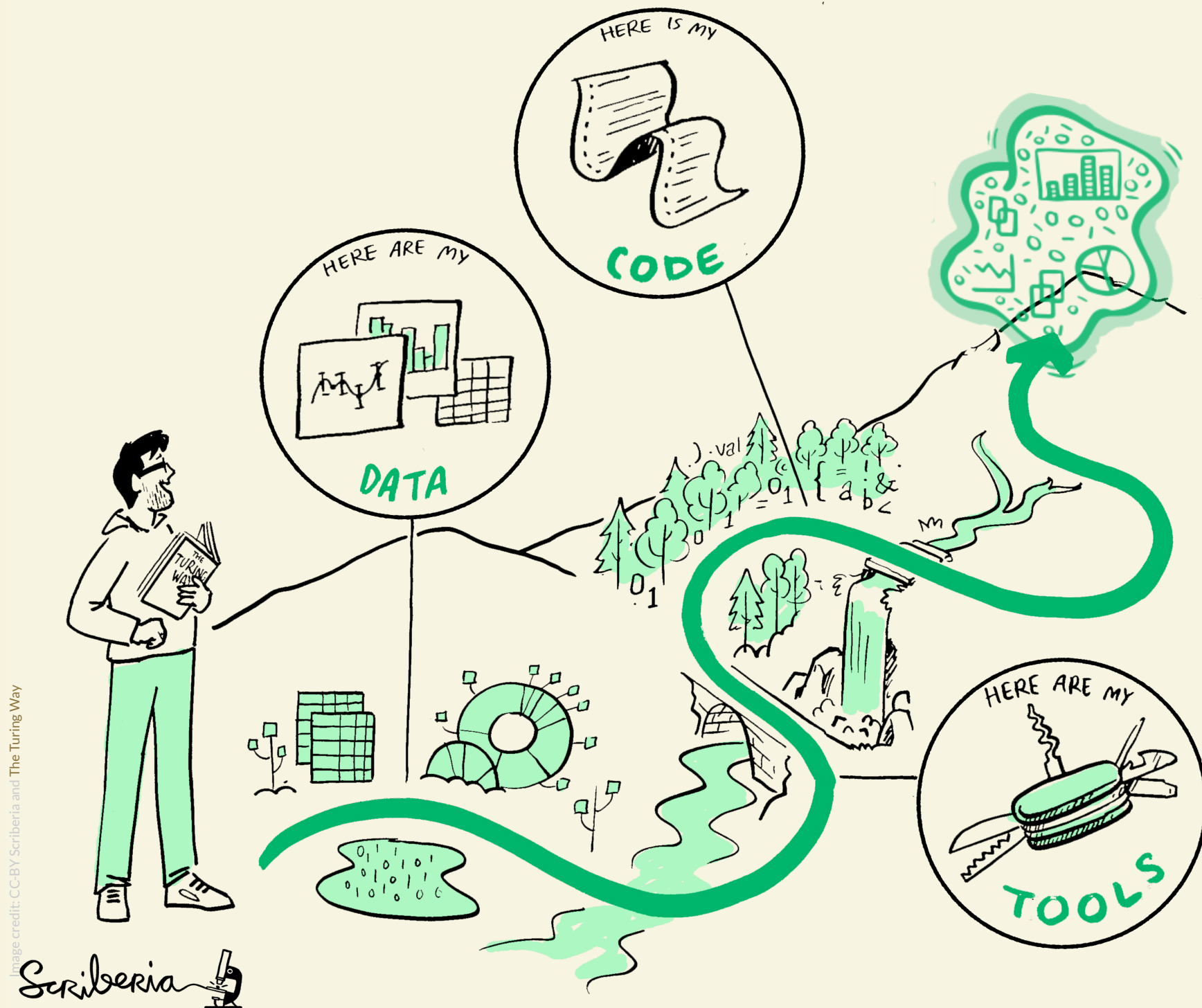


“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

Jon Claerbout (paraphrased)

SCIENCE IS MORE THAN A PDF

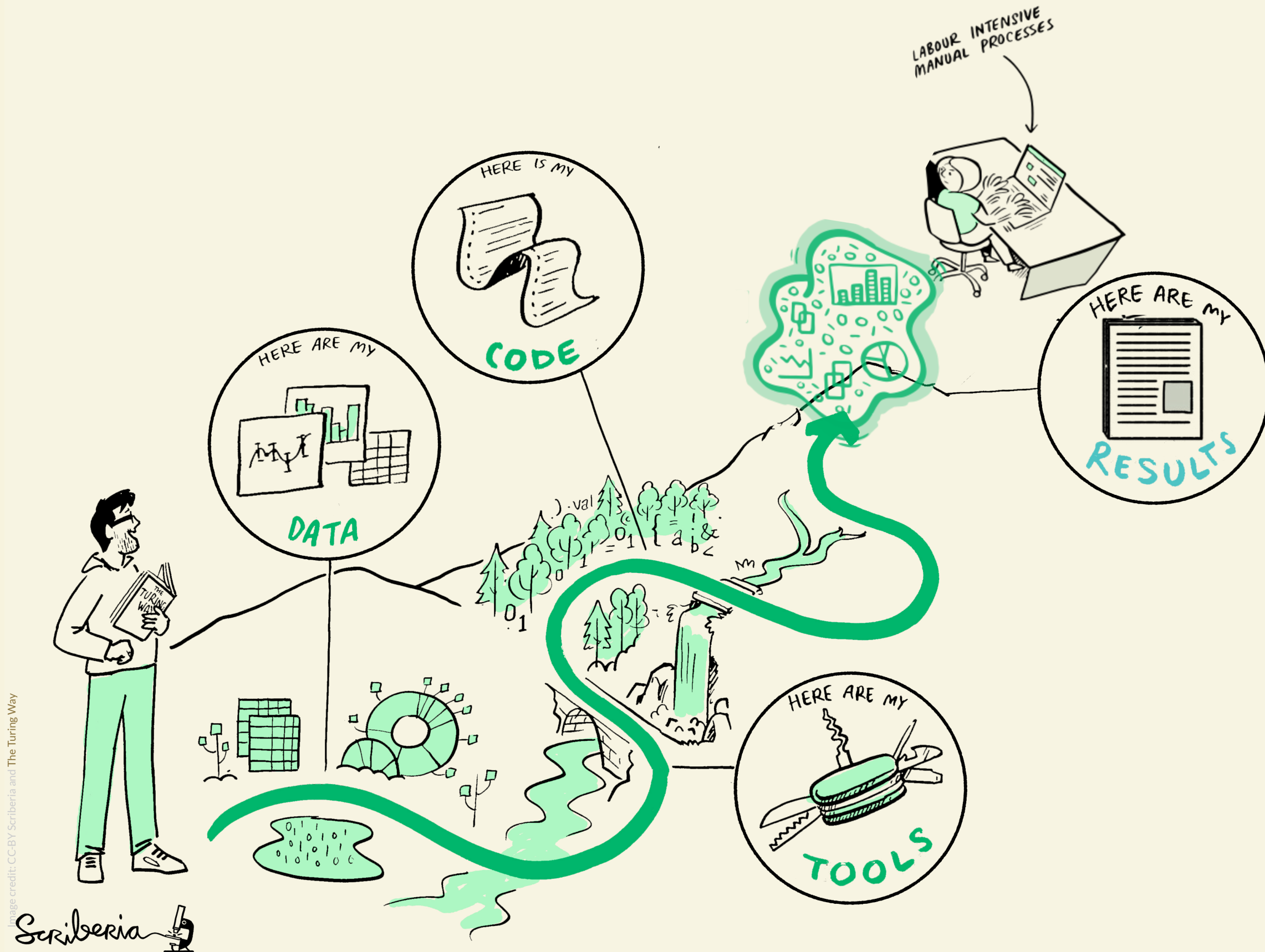
2.2



SCIENCE IS MORE THAN A PDF



SCIENCE IS MORE THAN A PDF



RESEARCH INGREDIENTS



But what about the recipe?



OPEN SHARING MAY NOT BE ENOUGH

Hardwicke et al., 2018:

- Open data of 38% of N=174 studies were not "in principle reusable"
- 24 out of 35 studies with reusable data: irreproducible main results without assistance of the original authors,
- 13 out of 24: not exactly reproducible even with assistance

Obels et al., 2020:

- N=62 Registered Reports
- 36 articles shared both data and code to reproduce results
- Out of those, main findings of 15 articles could not be reproduced

Data Management

- multiple versions of code/data
- suboptimal data curation
- unclear execution order/entry points
- non-portable code

Technical problems

- wrong software environment
- proprietary software
- link rot/accessibility issues

Human errors

- reporting (copy-paste) errors
- ambiguous analysis specification

BE KIND TO YOUR FUTURE SELF, WORK REPRODUCIBLY



Karen Cranston

@kcranstn



@mtholder motivating git: You mostly collaborate with yourself, and me-from-two-months-ago never responds to email. @swcarpentry

[Tweet übersetzen](#)

4:23 nachm. · 23. Aug. 2013 · TweetDeck

A REPRODUCIBLE PAPER IN THE WILD

The screenshot shows the GitHub interface for the repository 'psychoinformatics-de/paper-remodnav'. The repository is public and has 11 stars, 2 forks, and 2 tags. The main content area displays a list of files and folders with their commit history. The files include 'code', 'data', 'img', 'remodnav @ d289118', '.gitignore', '.gitmodules', 'COPYING', 'Makefile', 'README.md', 'main.tex', 'references.bib', 'results_def.tex', 'spbasic.bst', 'svglov3.dio', and 'svjour3.cls'. The commit history for each file is shown, including the commit message and the time since the last commit. The right sidebar contains sections for 'About', 'Releases', 'Packages', and 'Contributors'. The 'About' section mentions the code, data, and manuscript for a paper, with a DOI link. The 'Releases' section shows 2 tags. The 'Packages' section indicates no packages are published. The 'Contributors' section lists three contributors: adswa, mih, and AsimHDar.

Search or jump to... Pull requests Issues Marketplace Explore

psychoinformatics-de / paper-remodnav Public

Unwatch 11 Star 2 Fork 2

<> Code Issues Pull requests Actions Projects Wiki Security Insights

master 6 branches 2 tags

Go to file Add file Code

adswa Merge pull request #19 from psychoinformatics-de/reproducibility-impr... 5 hours ago 451 commits

File	Commit Message	Time
code	Generate figures with deterministic metadata for comp. reproducibility	yesterday
data	Point to latest label dataset	2 years ago
img	Include original SVGs into the repo to allow immediate manuscript ren	8 hours ago
remodnav @ d289118	Update remodnav with latest test dataset	2 years ago
.gitignore	Prevent permanent rebuilds of the figures	2 years ago
.gitmodules	[DATALAD] modified subdataset properties	5 months ago
COPYING	Declare CC-BY license	2 years ago
Makefile	Specify Python package versions in the state of final submission June	8 hours ago
README.md	DOC: improve readme, differentiate between recompile and recompute	8 hours ago
main.tex	Minor edits as suggested by reviewer 2	17 months ago
references.bib	velocity versus dispersion-based: cite a few algorithms and an evalua...	2 years ago
results_def.tex	Put generated results back into Git	2 years ago
spbasic.bst	Basic switch to new layout	8 years ago
svglov3.dio	Basic switch to new layout	8 years ago
svjour3.cls	Basic switch to new layout	8 years ago

0:00 / 2:13

About

Code, data and manuscript for <https://doi.org/10.1101/619254>

Readme

CC-BY-4.0 License

Releases

2 tags

Create a new release

Packages

No packages published

Publish your first package

Contributors 3

adswa Adina Wagner

mih Michael Hanke

AsimHDar Asim H Dar

You can find this recording on YouTube: <https://www.youtube.com/watch?v=nhLqmF58SLQ>

BUILDING BLOCKS OF THE PUBLICATION

Disclaimer: This is not the only way to write a reproducible paper!

- Original reproducible paper: <https://github.com/psychoinformatics-de/paper-remodnav>
- Resulting publication: <https://doi.org/10.3758/s13428-020-01428-x>
- Step-by-Step instructions: handbook.datalad.org/usecases/reproducible-paper.html
- Tutorial: github.com/datalad-handbook/repro-paper-sketch

Specific workflow and software choices:

- Version control and data transport: Git & DataLad
- Workflow management: Make
- Software management: Virtual environments
- Dynamic document generation: LaTeX

WHY: VERSION CONTROL AND DATA TRANSPORT?

The building blocks of a scientific result are rarely static

Analysis code evolves

(Fix bugs, add functions, refactor, ...)

WHY: VERSION CONTROL AND DATA TRANSPORT?

The building blocks of a scientific result are rarely static

Data changes

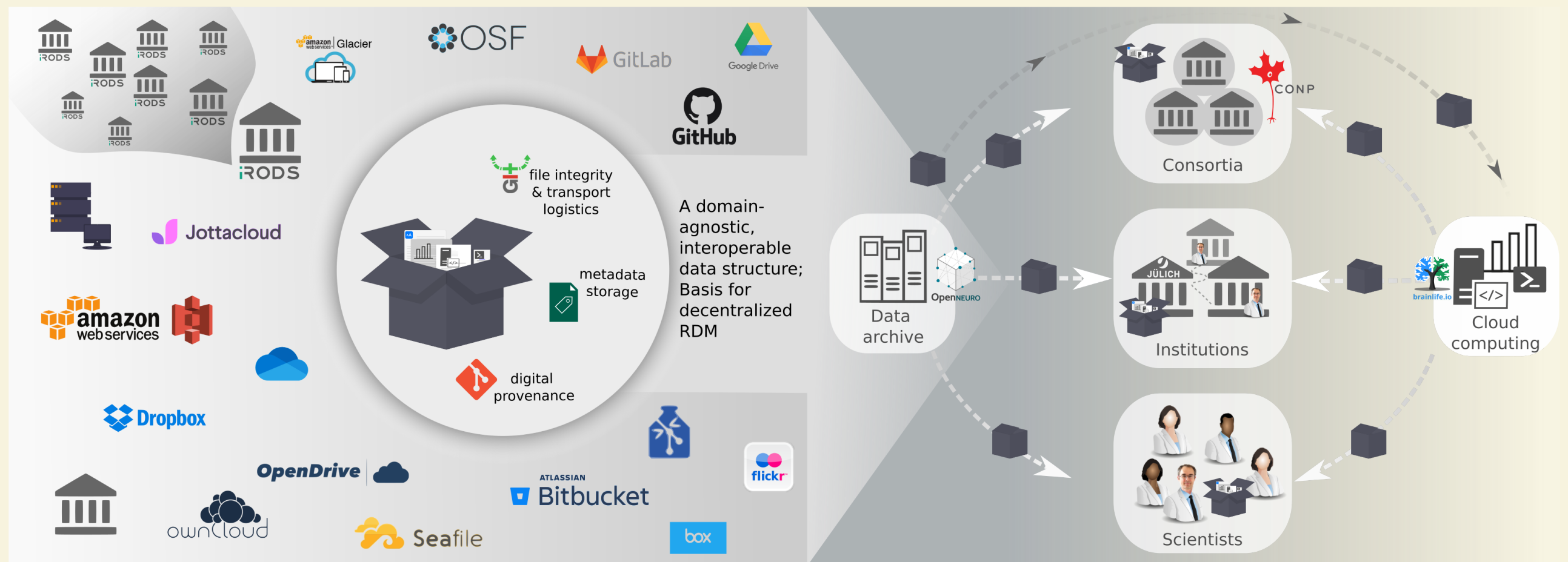
(errors are fixed, data is extended,
naming standards change, an analysis
requires only a subset of your data...)

HOW IT WORKS: VERSION CONTROL AND DATA TRANSPORT

- DataLad (Halchenko et al): Open source, domain-agnostic data management & data publication tool, build on top of Git and Git-annex (Joey Hess)
- **Main features used in this publication:**
 - version-controlling and linking arbitrarily large digital objects (data, software, code, results),
 - transport mechanisms for sharing, obtaining & collaborating on data (analysis) using existing infrastructure without a central service

HOW IT WORKS: VERSION CONTROL AND DATA TRANSPORT

- DataLad ([Halchenko et al](#)): Open source, domain-agnostic data management & data publication tool, build on top of [Git](#) and [Git-annex](#) (Joey Hess)
- Main features used in this publication:
 - version-controlling and linking arbitrarily large digital objects (data, software, code, results),
 - transport mechanisms for sharing, obtaining & collaborating on data (analysis) using existing infrastructure without a central service



HOW IT WORKS: VERSION CONTROL AND DATA TRANSPORT

- DataLad ([Halchenko et al](#)): Open source, domain-agnostic data management & data publication tool, build on top of [Git](#) and [Git-annex](#) (Joey Hess)
- **Main features used in this publication:**
 - version-controlling and linking arbitrarily large digital objects (data, software, code, results),
 - transport mechanisms for sharing, obtaining & collaborating on data (analysis) using existing infrastructure without a central service

```
1 (remodnav) adina@muninn in ~/repos/papers/paper-remodnav on git:master
2 # How large is the repository after cloning?
3 $ du -sh      # show disk usage in human readable sizes
4 16M
```

Let's take a look into how the script retrieves data:

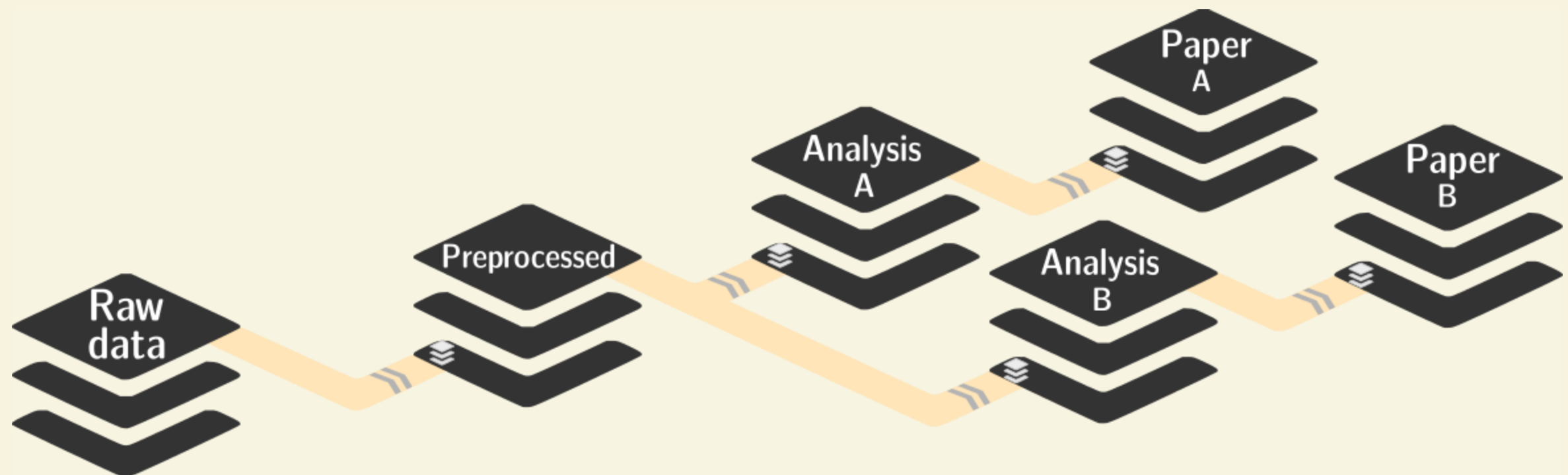
```
1 vim code/mk_figuresnstats.py
2
3 from datalad.api import get as datalad_get
4 [...]
5 infiles = [op.join('data', 'raw_eyegaze', 'sub-32', 'beh', 'sub-32_beh_movie_1m_3_recording-eyegaze_ph
6 for f in infiles:
7     datalad get(f)
8     data = np.recfromcsv(f)
```

Import datalad API

Get individual files

HOW IT WORKS: VERSION CONTROL AND DATA TRANSPORT

- DataLad ([Halchenko et al](#)): Open source, domain-agnostic data management & data publication tool, build on top of [Git](#) and [Git-annex](#) (Joey Hess)
- **Main features used in this publication:**
 - version-controlling and linking arbitrarily large digital objects (data, software, code, results),
 - transport mechanisms for sharing, obtaining & collaborating on data (analysis) using existing infrastructure without a central service



HOW IT WORKS: VERSION CONTROL AND DATA TRANSPORT

- DataLad ([Halchenko et al](#)): Open source, domain-agnostic data management & data publication tool, build on top of [Git](#) and [Git-annex](#) (Joey Hess)
- **Main features used in this publication:**
 - version-controlling and linking arbitrarily large digital objects (data, software, code, results),
 - transport mechanisms for sharing, obtaining & collaborating on data (analysis) using existing infrastructure without a central service

```

1 $ (remodnav) adina@muninn in ~/repos/papers/paper-remodnav on git:master
2 > datalad subdatasets --recursive
3 subdataset(ok): data/raw_eyegaze (dataset)
4 subdataset(ok): data/raw_eyegaze/src/lab-eyetracking (dataset)
5 subdataset(ok): data/studyforrest-data-eyemovementlabels (dataset)
6 subdataset(ok): data/studyforrest-data-eyemovementlabels/code/remodnav (dataset)
7 subdataset(ok): data/studyforrest-data-eyemovementlabels/inputs/raw_eyegaze (dataset)
8 subdataset(ok): remodnav (dataset)
9 subdataset(ok): remodnav/remodnav/tests/data/anderson_etal (dataset)
10 subdataset(ok): remodnav/remodnav/tests/data/studyforrest (dataset)
11
12 subdataset (ok: 8)

```

Raw data

Software package

HOW IT WORKS: VERSION CONTROL AND DATA TRANSPORT

2020-06-05 10:58	+0200	Adina Wagner	M	[master] {upstream/master} {upstream/HEAD} Merge pull request #12 from psychoinformatics-d
2020-06-05 08:24	+0200	Adina Wagner	o	[finalround] {upstream/finalround} add results from computing with mean instead of median
2020-06-05 09:09	+0200	Michael Hanke	o	Change wording, clarify comment
2020-06-05 07:26	+0200	Michael Hanke	M	Merge remote-tracking branch 'gh-mine/finalround'
2020-05-28 16:39	+0200	Asim H Dar	o	Added datalad.get() so S2SRMS() pulls data and can run standalone
2020-05-18 08:25	+0200	Adina Wagner	o	{gh-asim/finalround} S2SRMS: example implementation of the S2SRMS method suggested by R2
2020-05-01 17:38	+0200	Adina Wagner	o	Minor edits as suggested by reviewer 2
2020-05-29 09:04	+0200	Adina Wagner	M	Merge pull request #13 from psychoinformatics-de/adswa-patch-1
2020-05-24 09:15	+0200	Adina Wagner	o	{upstream/adswa-patch-1} Fix installation instructions
2020-05-24 09:53	+0200	Adina Wagner	M	Merge pull request #14 from psychoinformatics-de/bf-data
2020-05-24 09:33	+0200	Adina Wagner	o	[bf-data] One-time datalad import
2020-05-24 09:32	+0200	Adina Wagner	o	install and get relevant subdataset data
2020-03-18 10:19	+0100	Michael Hanke	M	Merge pull request #8 from psychoinformatics-de/adswa-patch-1
2019-12-19 10:22	+0100	Adina Wagner	o	{gh-asim/adswa-patch-1} add sklearn to requirements
2020-03-18 10:13	+0100	Michael Hanke	o	Tune new figure caption
2020-03-18 10:03	+0100	Michael Hanke	M	Merge pull request #11 from ElectronicTeaCup/revision_2
2020-03-18 09:59	+0100	Adina Wagner	M	[revision_2] {gh-asim/revision_2} Merge branch 'revision_2' of github.com:ElectronicTe
2020-03-18 09:58	+0100	Michael Hanke	o	Last detections
2020-03-18 09:59	+0100	Adina Wagner	o	name parameter in caption

Date

Author

Change summary (commit message)

WHY: WORKFLOW MANAGEMENT?

"Shit, which parametrization of which script produced these outputs from which subset of what data in which order?"



HOW IT WORKS: WORKFLOW MANAGEMENT

Makefiles!

```

1 # use `chronic` to make output look neater, if available
2 CHRONIC=$(shell which chronic || echo '')
3 PYTHON=python
4
5 all: main.pdf
6
7 # important to process stats and figures first, such that
8 # up-to-date versions are compiled into the manuscript
9 main.pdf: main.tex results_def.tex references.bib
10     @echo "# Render figures"
11     $(MAKE) -C img
12     @echo "# Render manuscript"
13     @$ (CHRONIC) latexmk -pdf -g $<
14
15 # the stats-script outputs all scores and figures
16 results_def.tex: code/mk_figuresnstats.py
17     @test -z "$$VIRTUAL_ENV" && \
18         echo "ERROR: must be executed in a virtual env (set VIRTUAL_ENV to fake one)" && \
19         exit 1 || true
20     @echo "# Ensure REMODNAV installation"
21     @python -m pip install pandas==1.0.5 seaborn==0.10.1 scikit-learn==0.23.0 datalad
22     @datalad get -n remodnav
23     @$ (CHRONIC) pip install -e remodnav
24     @rm -f $@
25     @REMODNAV_RESULTS=$@ $(PYTHON) code/mk_figuresnstats.py -s -f -r -m
26
27 clean:
28     rm -f main.bbl main.aux main.blg main.log main.out main.pdf main.tdo \
29         main.fls main.fdb_latexmk texput.log \
30         results_def.tex
31     $(MAKE) -C img clean
32
33 virtualenv:
34
35 .PHONY: clean
36

```

The final result: A manuscript PDF

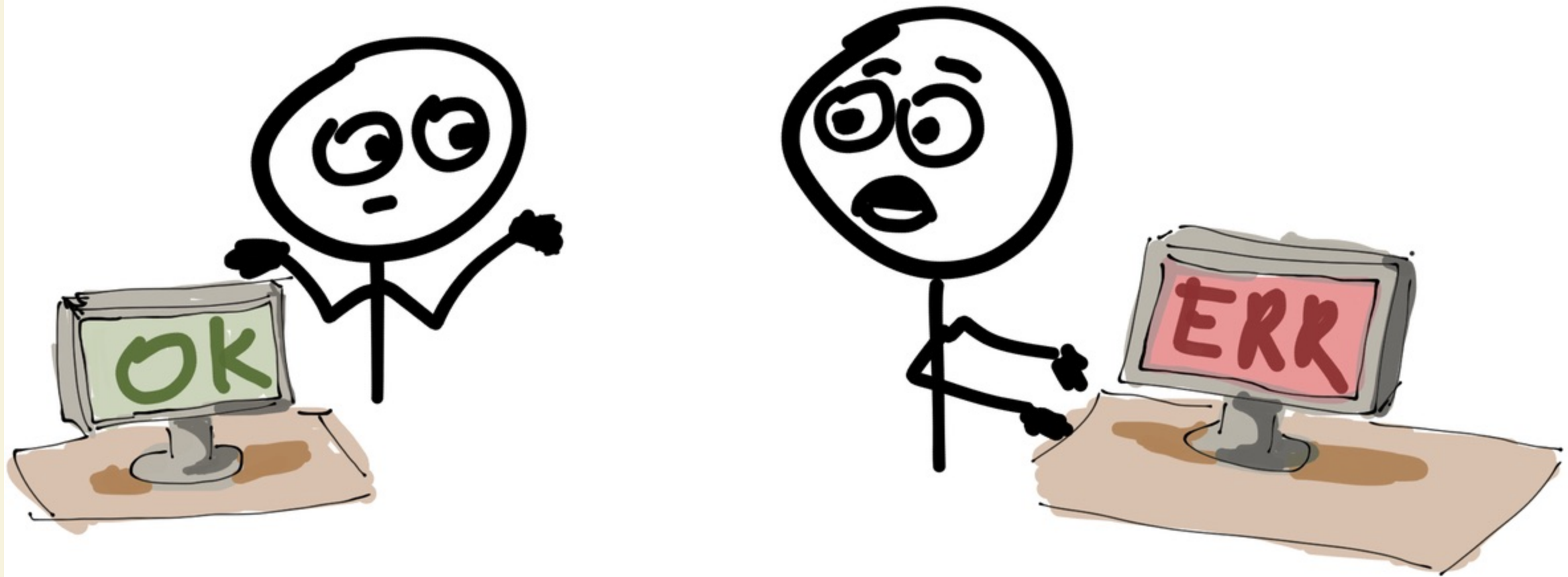
Its ingredients: Manuscript sources and results. Only render the manuscript when those exist.

How to generate results and figures

Clean-up routine

WHY: SOFTWARE MANAGEMENT?

"Works on my machine"



WHY: SOFTWARE MANAGEMENT?

"This used to work on my machine..."




```
1 # use `chronic` to make output look neater, if available
2 CHRONIC=$(shell which chronic || echo '')
3 PYTHON=python
4
5 all: main.pdf
6
7 # important to process stats and figures first, such that
8 # up-to-date versions are compiled into the manuscript
9 main.pdf: main.tex results_def.tex references.bib
10     @echo "# Render figures"
11     $(MAKE) -C img
12     @echo "# Render manuscript"
13     @$ (CHRONIC) latexmk -pdf -g $<
14
15 # the stats-script outputs all scores and figures
16 results_def.tex: code/mk_figuresnstats.py
17     @Set up software environment prior to computing
18     @exit 1 || true
19     @echo "# Ensure REMODNAV installation"
20     @python -m pip install pandas==1.0.5 seaborn==0.10.1 scikit-learn==0.23.0 datalad
21     @datalad get -n remodnav
22     @$ (CHRONIC) pip install -e remodnav
23     @rm -f $@
24     @REMODNAV_RESULTS=$@ $(PYTHON) code/mk_figuresnstats.py -s -f -r -m
25
26
27 clean:
28     rm -f main.bbl main.aux main.blg main.log main.out main.pdf main.tdo \
29         main.fls main.fdb_latexmk texput.log \
30         results_def.tex
31     $(MAKE) -C img clean
32
33 virtualenv:
34
35 .PHONY: clean
36
```

WHY: DYNAMIC DOCUMENT GENERATION?



HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

Let's take a look into how results and figures are saved by the script:

```
1 def mk_confusion_figures(fig, stat):
2     """
3     small helper function to save all confusion matrices
4     """
5     max_mclf = 0
6     for pair in itertools.combinations(['MN', 'RA', 'AL'], 2):
7         plt.figure(
8             # fake size to get the font size down in relation
9             figsize=(14, 3),
10            dpi=120,
11            frameon=False)
12        cur_max_mclf = confusion(pair[0],
13                                pair[1],
14                                fig,
15                                stat)
16        plt.savefig(
17            op.join('img', 'confusion_{}_{}.svg'.format(*pair)),
18            transparent=True,
19            bbox_inches="tight",
20            metadata={'Date': None})
21        plt.close()
22        if cur_max_mclf > max_mclf:
23            max_mclf = cur_max_mclf
24    if stat:
25        rsout('\newcommand{\maxmclf}{%s}'
26             % ('%.1f' % max_mclf))
```

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

Let's take a look into how results and figures are saved by the script:

```
1 def mk_confusion_figures(fig, stat):
2     """
3     small helper function to save all confusion matrices
4     """
5     max_mclf = 0
6     for pair in itertools.combinations(['MN', 'RA', 'AL'], 2):
7         plt.figure(
8             # fake size to get the font size down in relation
9             figsize=(14, 3),
10            dpi=120,
11            frameon=False)
12        cur_max_mclf = confusion(pair[0],
13                                pair[1],
14                                fig,
15                                stat)
16        plt.savefig(
17            # box_inches="tight",
18            metadata={'Date': None})
19        plt.close()
20        if cur_max_mclf > max_mclf:
21            max_mclf = cur_max_mclf
22    if stat:
23        rsout('\newcommand{\maxmclf}{%s}'
24              % ('%.1f' % max_mclf))
```

Figures are saved into a relative project path

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

The LaTeX sources can directly embed this figure:

```

1 \begin{figure*}
2   \includegraphics[trim=0 0 0 0,clip,width=1\textwidth]{img/confusion_MN_RA.pdf} \\
3   \includegraphics[trim=0 0 0 6.6mm,clip,width=1\textwidth]{img/confusion_MN_AL.pdf} \\
4   \includegraphics[trim=0 0 0 6.6mm,clip,width=1\textwidth]{img/confusion_RA_AL.pdf}
5   \caption{Confusion patterns for pairwise eye movement classification
6     comparison of both human raters \citep[MN and RA: 1][Andersson2017] and the
7     \remodnav\ algorithm}
8   images (left column,
9     column). All matrices present gaze sample based Jaccard indices \citep[JI;
10    ][][jaccard1901etude]. Consequently, the diagonals depict the fraction of
11    time points labeled congruently by both raters in relation to the number of
12    timepoints assigned to a particular event category by any rater.}
13   % Give a unique label
14   \label{fig:conf}
15 \end{figure*}
16

```

Figures are embedded into the manuscript sources

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

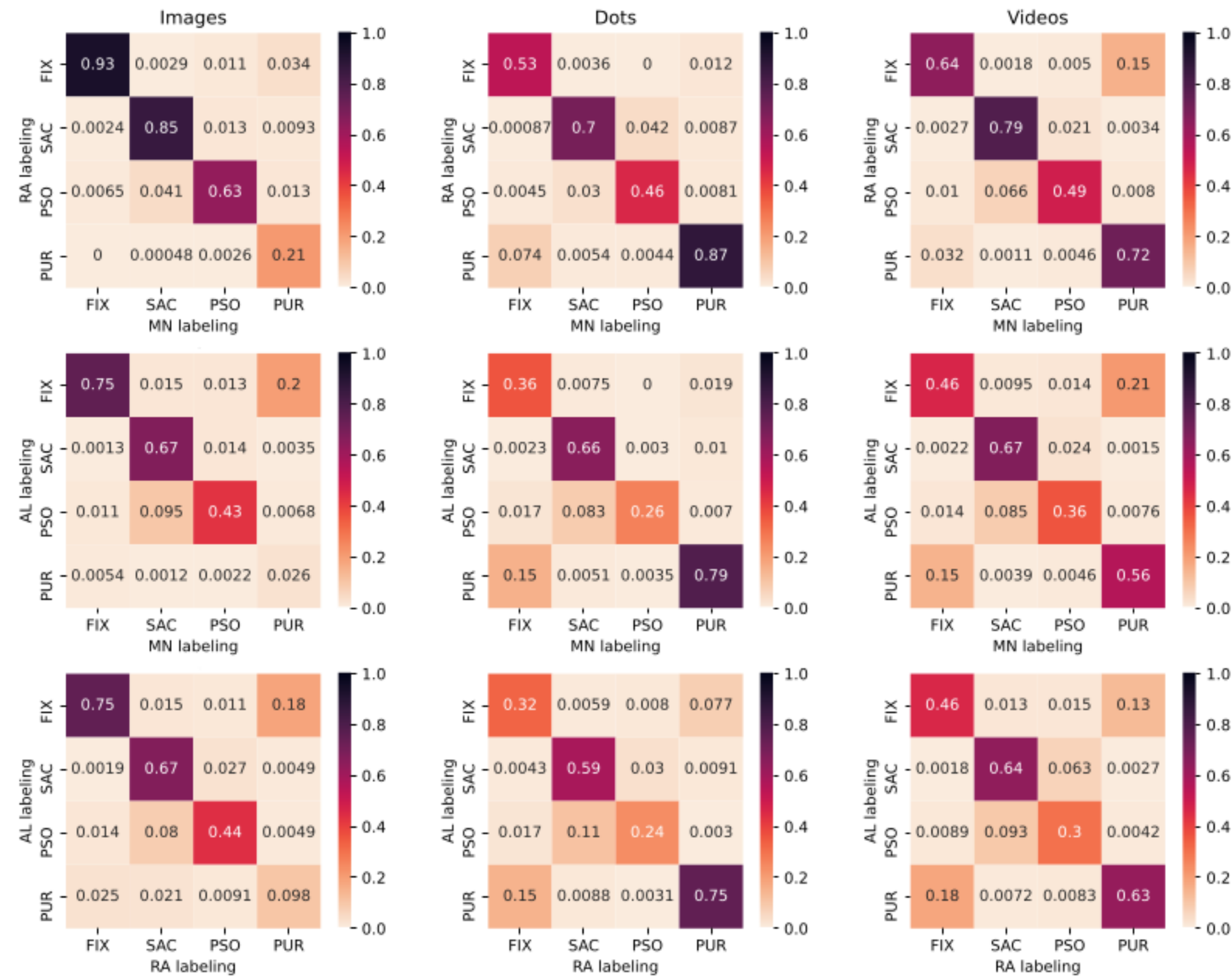


Fig. 2 Confusion patterns for pairwise eye movement classification comparison of both human raters (MN and RA; Andersson et al., 2017) and the REMoDNaV algorithm (AL) for gaze recordings from stimulation with static images (left column), moving dots (middle column), and video clips (right column). All matrices present gaze sample based Jaccard indices (JI; Jaccard, 1901). Consequently, the diagonals depict the fraction of time points labeled congruently by both raters in relation to the number of timepoints assigned to a particular event category by any rater.

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

What about tables?

```

1 def mk_confusion_figures(fig, stat):
2     """
3     small helper function to save all confusion matrices
4     """
5     max_mclf = 0
6     for pair in itertools.combinations(['MN', 'RA', 'AL'], 2):
7         plt.figure(
8             # fake size to get the font size down in relation
9             figsize=(14, 3),
10            dpi=120,
11            frameon=False)
12        cur_max_mclf = confusion(pair[0],
13                                pair[1],
14                                fig,
15                                stat)
16        plt.savefig(
17            op.join('img', 'confusion_{}_{}.svg'.format(*pair)),
18            transparent=True,
19            bbox_inches="tight",
20            metadata={'Date': None})
21        plt.close()
22        if cur_max_mclf > max_mclf:
23            max_mclf = cur_max_mclf
24    if stat:
25        rsout('\newcommand{\maxmclf}{%s}'
26            % ('%.1f' % max_mclf))

```

We print them under a unique label as a LaTeX `\newcommand` variable
 For example, `\newcommand{\maxmclf}{10.0}`

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

Output is automatically written to `results_def.tex`...

```
1 $ cat results_def.tex
2 \newcommand{\imgMNRAMCLF}{6.1}
3 \newcommand{\imgMNRAMclfwOP}{3.0}
4 \newcommand{\imgMNRAFXref}{70}
5 \newcommand{\imgMNRASACref}{9}
6 \newcommand{\imgMNRAPSOref}{21}
7 \newcommand{\imgMNRASPr
8 \newcommand{\imgMNRAFX
9 \newcommand{\imgMNRASACcod}{15}
10 \newcommand{\imgMNRAPSOcod}{20}
11 \newcommand{\imgMNRASPcod}{53}
12 [...]
13
```

The value 6.1 will be in a LaTeX variable `\imgMNRAMCLF`

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

...and can be used in the LaTeX sources:

```

1 \begin{document}
2 \input{results_def.tex}
3 \onecolumn
4 \title{REMoDNaV: Robust Eye-Movement Classification for Dynamic Stimulation} &\ (remodnav)
5
6 [...]
7
8 \begin{table}[tbp]
9   \caption{Cohen's Kappa reliability between human coders (MN, RA), and \remodnav\ (AL)
10   with each of the human coders.}
11   \label{tab:kappa}
12   \begin{tabular*}{0.5\textwidth}{c @{\extracolsep{\fill}}llll}
13     \textbf{Fixations} & & & & \
14     \hline\noalign{\smallskip}
15     Comparison & Images & Dots & Videos & \
16     \noalign{\smallskip}\hline\noalign{\smallskip}
17     MN versus RA & & \kappaRAMNimgFix & \kappaRAMNdotsFix & \kappaRAMNvideoFix \
18     AL versus RA & & \kappaALRAimgFix & \kappaALRADotsFix & \kappaALRAvideoFix \
19     AL versus MN & & \kappaALMNimgFix & \kappaALMNdotsFix & \kappaALMNvideoFix \
20     \noalign{\smallskip}\hline
21   \end{tabular*}
22 \end{table}

```

Use them to embed the results in a table

HOW IT WORKS: DYNAMIC DOCUMENT GENERATION

Table 2 Proportion of samples in each stimulus category classified in disagreement between human coders (MN, RA) and the REMoDNaV algorithm (AL). The MC (misclassification) column lists proportions considering all four event categories (fixation, saccade, PSO, pursuit), while the w/oP (without pursuit) column excludes pursuit events for a direct comparison with Andersson et al. (2017, Tables 8-10). The remaining columns show the percentage of labels assigned to incongruent time points by each rater (deviation of their sum from 100% is due to rounding).

Images							
Comp	MC	w/oP	Coder	Fix	Sac	PSO	SP
MN-RA	6.1	3.0	MN	70	9	21	0
—	—	—	RA	13	15	20	53
MN-AL	23.1	6.5	MN	86	2	11	2
—	—	—	AL	5	13	6	75
RA-AL	22.8	6.4	RA	77	3	11	9
—	—	—	AL	13	13	6	68
Dots							
Comp	MC	w/oP	Coder	Fix	Sac	PSO	SP
MN-RA	10.7	4.2	MN	11	10	9	71
—	—	—	RA	64	7	6	23
MN-AL	18.6	8.2	MN	9	5	8	78
—	—	—	AL	77	6	2	15
RA-AL	22.8	10.8	RA	28	4	6	61
—	—	—	AL	59	7	2	31
Videos							
Comp	MC	w/oP	Coder	Fix	Sac	PSO	SP
MN-RA	18.5	4.0	MN	75	3	8	15
—	—	—	RA	16	4	3	77
MN-AL	31.5	7.9	MN	57	1	6	36
—	—	—	AL	36	5	3	55
RA-AL	28.5	9.1	RA	38	3	5	55
—	—	—	AL	53	6	5	35

ADD A FEW BASIC RDM STRATEGIES

- Create an intuitive structure, ideally comply to standards

```
|— code/           # scripts
|— data/          # datasets
|— img/           # computed figures
|— main.tex       # manuscript sources
|— Makefile       # documentation for machines
|— README.md      # documentation for humans
|— references.bib
|— remodnav/      # software
|— results_def.tex # numerical results
```

- Ensure completeness and portability:

```
# ANTIPATTERN
# read important configuration
path = '/nfs/e2/somewhere/outside/theproject/veryimportant.info'
```

```
# include important configuration, read from a relative path
path = 'code/veryimportant.info'
```

WHY WOULD YOU WANT TO WORK THIS WAY?

Data changes (for real)

(errors are fixed, data is extended,
naming standards change, ...)

Example:

I found an error in one of the public datasets -
just before submission

```
2019-03-08 12:38 +0100 Richard Andersson M [master] {origin/master} {origin/HEAD} Merge pull request #3 from AdinaWagner/datafix
2019-03-08 11:35 +0100 Adina Wagner | o ENH/FIX: relabel erroneous saccades to fixations, closes #2.
2018-12-05 15:27 +0100 Richard Andersson o Uploaded a folder consting only of the data used in the original article
2017-08-22 19:40 +0200 Richard Andersson o Code added
2016-12-14 10:35 +0100 richardandersson o Added currently shared data and stimuli.
2016-12-14 10:29 +0100 richardandersson o Deleted -- to be reuploaded
2016-12-14 10:08 +0100 Richard Andersson o Add files via upload
2016-12-14 10:07 +0100 Richard Andersson o Add files via upload
2016-12-14 10:04 +0100 Richard Andersson I Initial commit
```

Fixed data

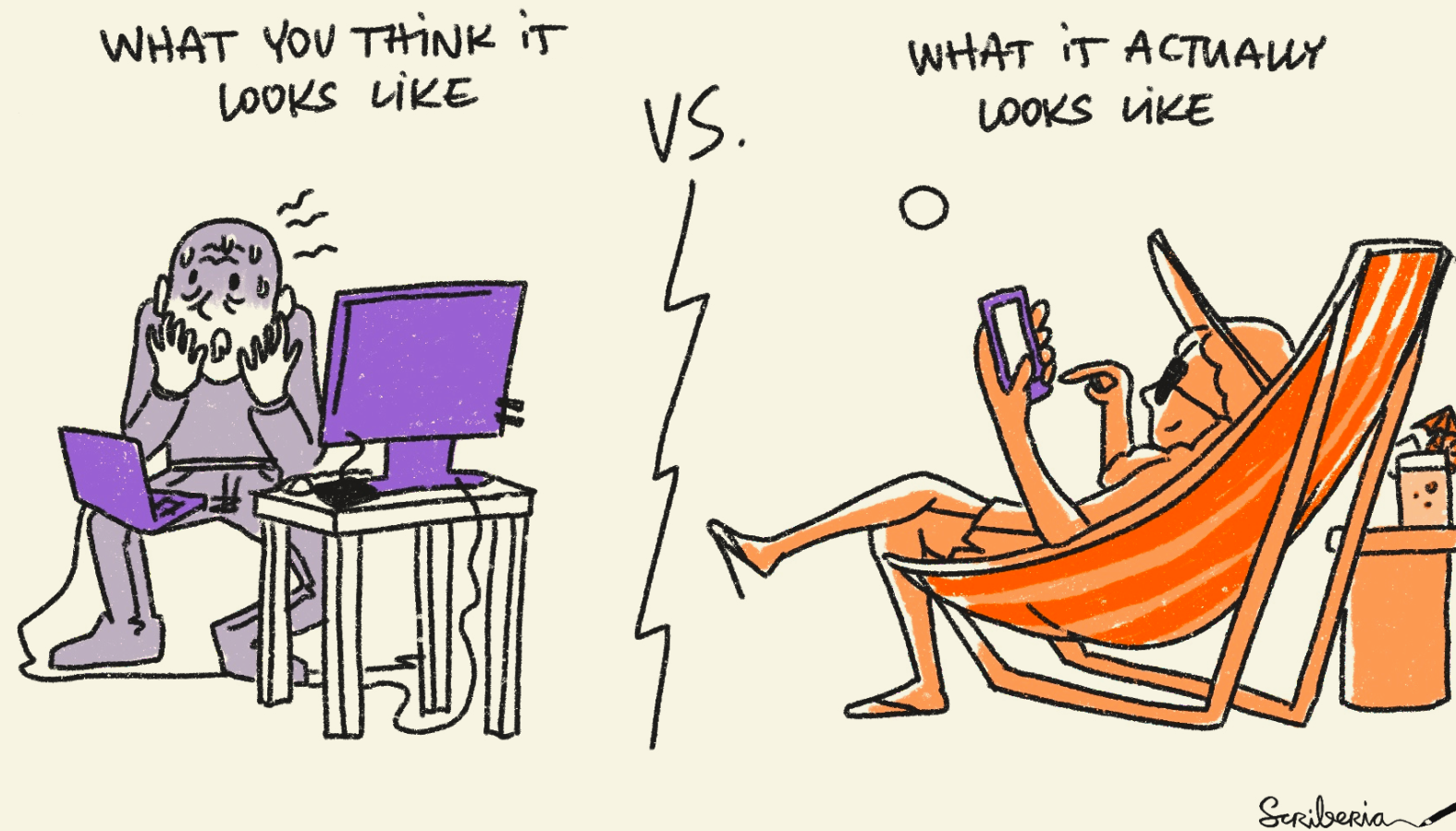
Original data

```
commit d28911838f91f33af0db57332f55ccc442d24f66
Refs: [master], 1.0-2-gd289118
Author: Michael Hanke <michael.hanke@gmail.com>
AuthorDate: Thu Apr 25 17:54:06 2019 +0200
Commit: Michael Hanke <michael.hanke@gmail.com>
CommitDate: Thu Apr 25 17:54:06 2019 +0200

    Update Anderson et al test dataset with Zembls labeling fix
---
remodnav/tests/data/anderson_etal | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

diff --git a/remodnav/tests/data/anderson_etal b/remodnav/tests/data/anderson_etal
index c6d0253..0e6f827 160000
--- a/remodnav/tests/data/anderson_etal
+++ b/remodnav/tests/data/anderson_etal
@@ -1,1 @@
-Subproject commit c6d02539712d12d7bea96912521a43cb84e7a7b8
+Subproject commit 0e6f82708e10b48039763aa1078696e802260674
```

WHY WOULD YOU WANT TO WORK THIS WAY?



- Save time
- Have a framework for collaboration
- Gain confidence in the validity of your results
- Make a more convincing case with your research
- Make your research more accessible
- Increase your work's reusability

"OH GOD, BUT I DON'T KNOW ANY OF THE TOOLS..."

Use other tools! (RMarkdown/Bookdown, Snakemake, Bash scripts for orchestration, Guix, Software containers, ...). What matters is the result, not how you get there

There are even complete frameworks, such as WORCS ([Van Lissa et al., 2020](#)) or [Peikert & Brandmaier, 2021](#)

There are tutorials and documentation everywhere, e.g., for [our paper](#), using [Make](#), in [R](#), etc.

ACKNOWLEDGEMENTS

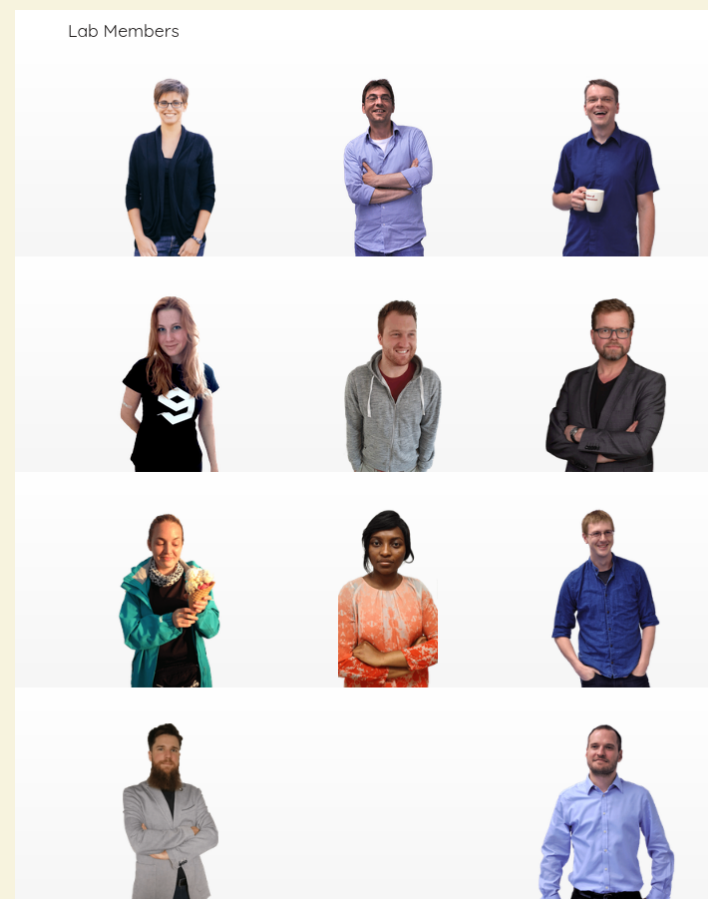
Mentors + Co-authors

- Michael Hanke & Asim H. Dar
- Psychoinformatics Lab

Illustrations and Tutorials

- The Turing Way + Scriberia

Creators and Maintainers of our ecosystem of open source tools



THANKS FOR YOUR ATTENTION



Slides at DOI [10.5281/zenodo.5508797](https://doi.org/10.5281/zenodo.5508797)

Women neuroscientists are [underrepresented in neuroscience](#). You can use the [Repository for Women in Neuroscience](#) to find and recommend neuroscientists for conferences, symposia or collaborations, and help making neuroscience more open & divers.