Slides: https://doi.org/10.6084/m9.figshare.10299212.v1

Evaluating the Stability of Neuroimaging Pipelines

Gregory Kiar, with work contributed by

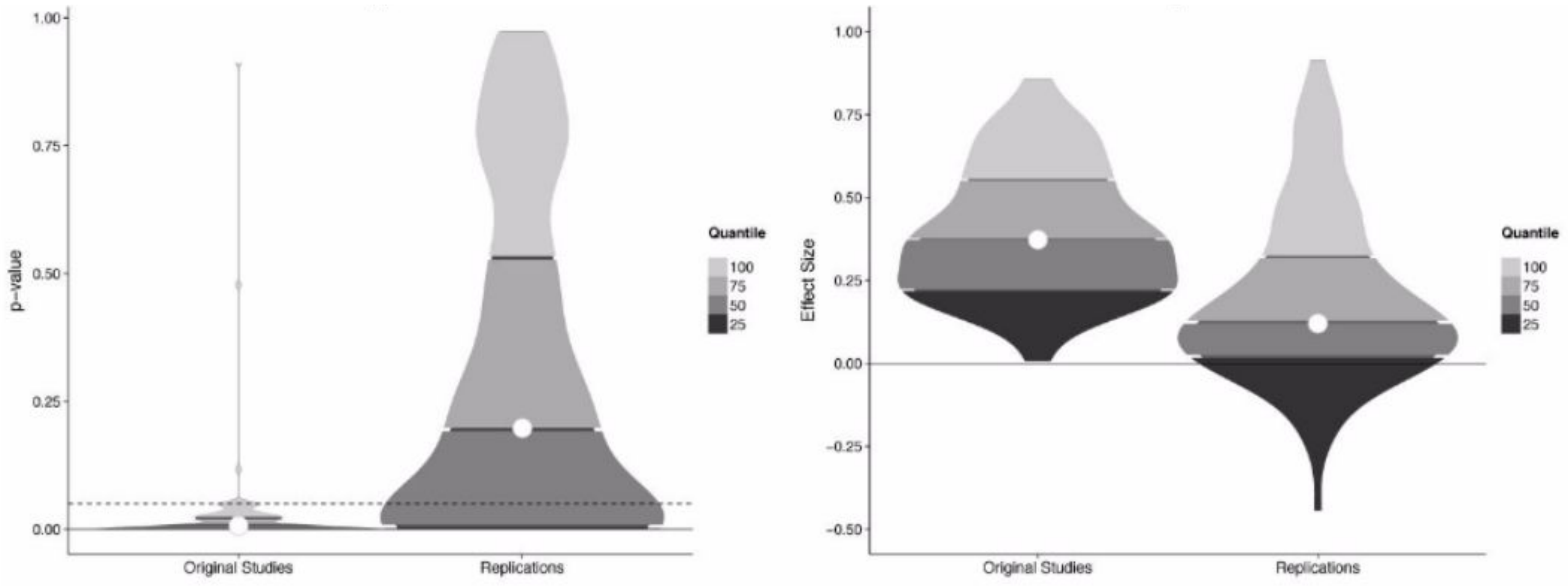
Pablo de Oliveira Castro, Ali Salari, Pierre Rioux, Eric Petit,
Shawn T. Brown, Alan C. Evans, Tristan Glatard

McGill neuro Université Concordia University Exascale computing research

Overview

- Topics in reproducibility
- Operationalizing Stability
- Evaluation of a Neuroimaging Pipeline

Reproducibility is Measurable



(Open Science Collaboration, 2015)

Replication can be difficult...

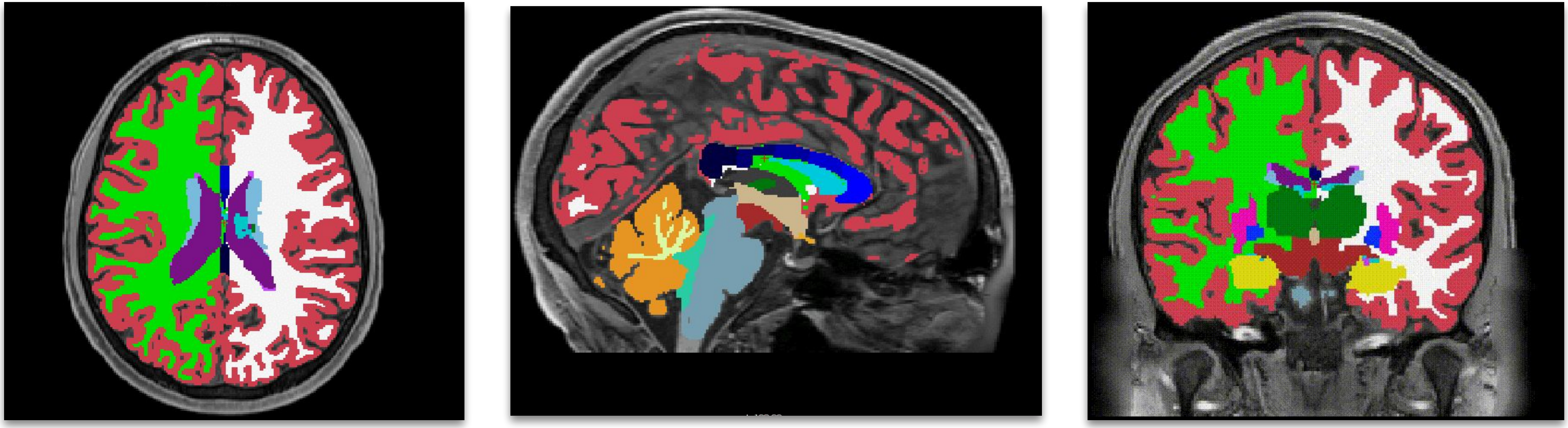
But this talk is NOT about replicability

Let's say we go so far as to get that right, then...

Stability is playing spot the difference

Difference Spotted!   OS

CentOS 6 vs CentOS 7

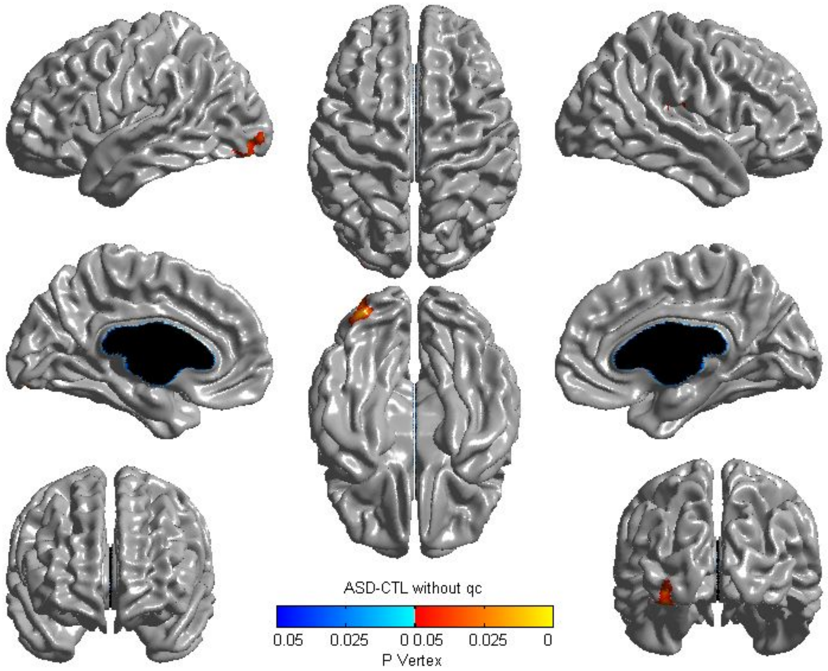


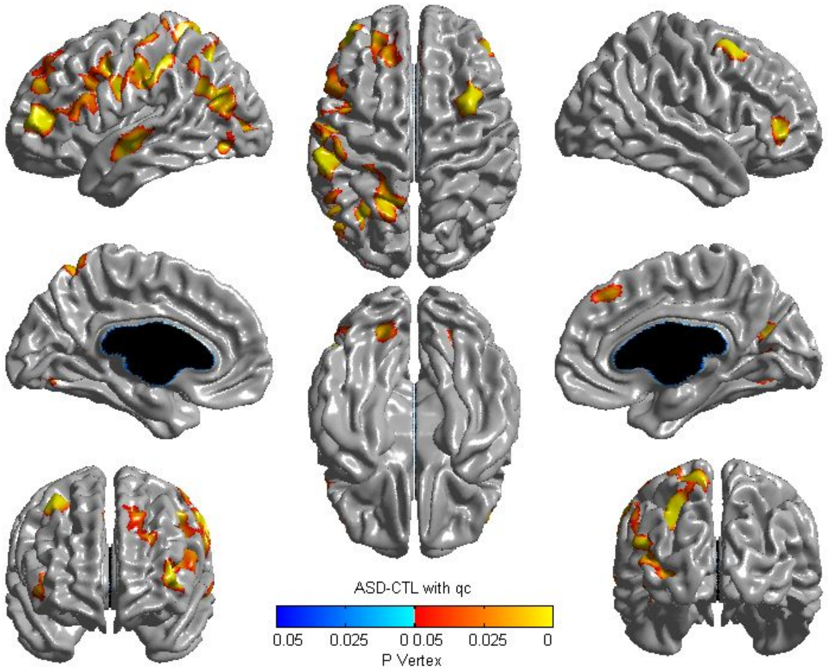
(Scaria, 2017)

(Scaria et al., 2017)

Difference Spotted! Data Quality

Full Dataset (N ~ 1100) vs QC'd Dataset (N ~ 400)



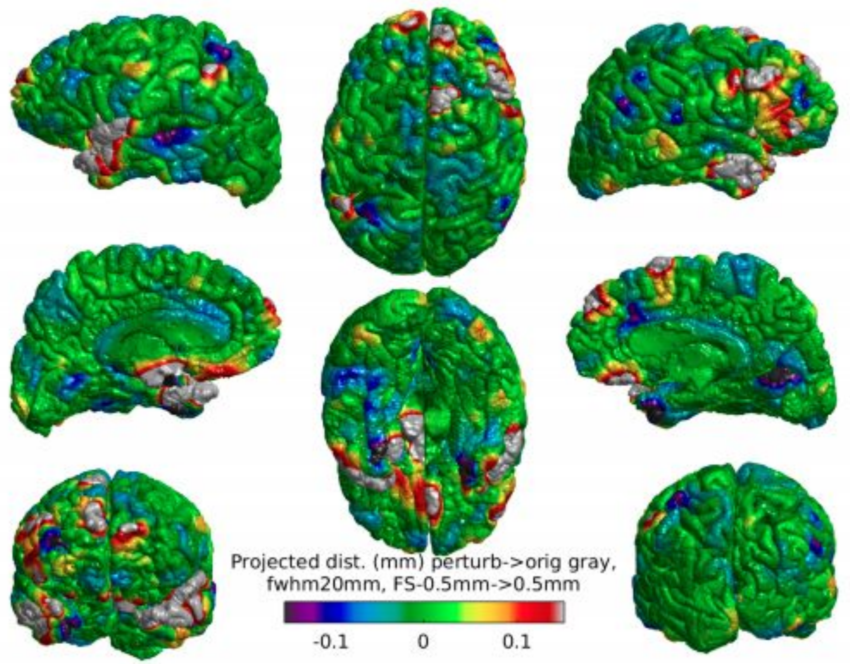


(Scaria, 2017)

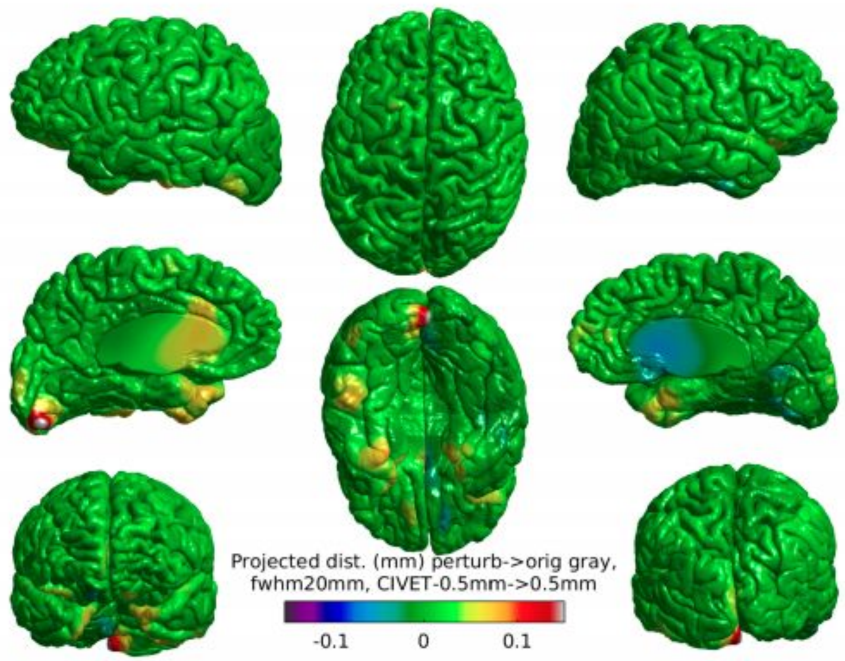
(Khundrakpam et al., 2017)

Difference Spotted! Instability

1-voxel noise injections at 1% intensity



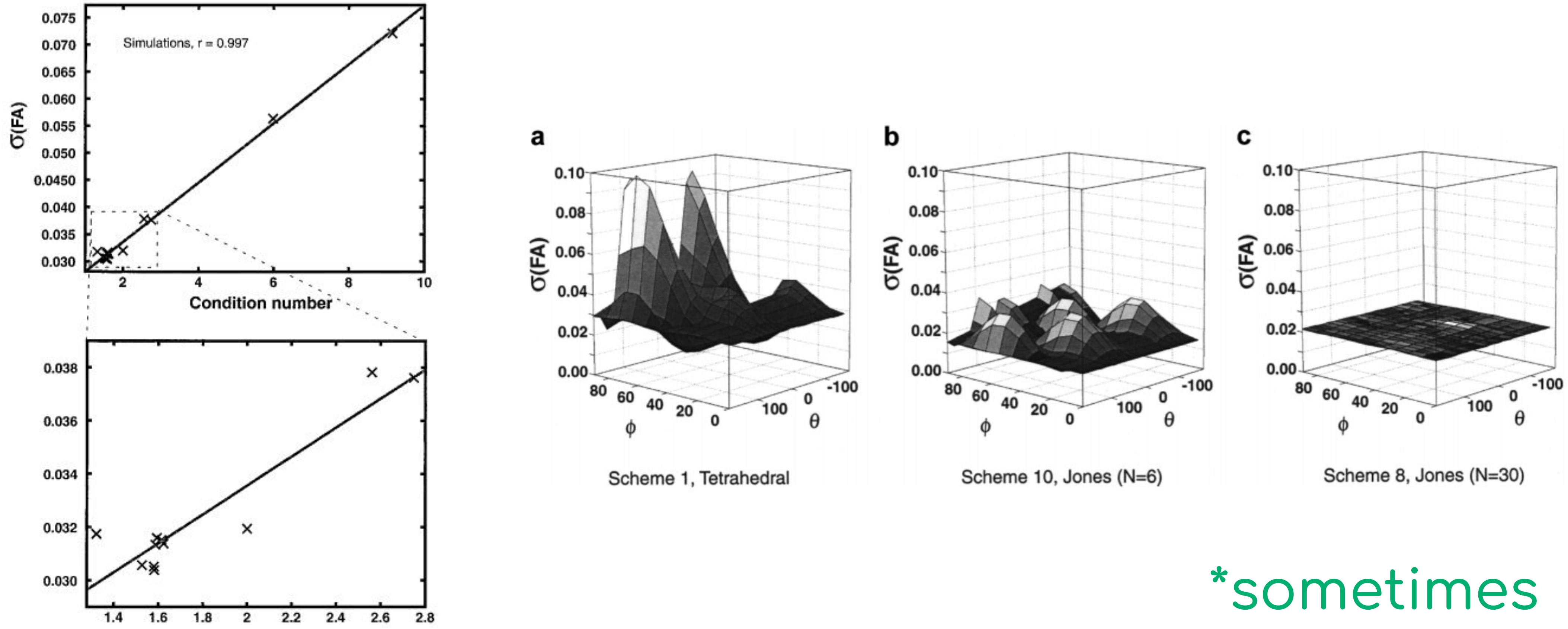
Freesurfer



CIVET

(Lewis et al., 2017)

Instabilities can* be Anticipated



*sometimes

(Skare et al., 2000)

What about the even simpler errors we see *in practice*?

Such as addition in Python…

In [1]:
```
def count_to_one(N):
    step = 1.0 / N
    return step, step*N, sum([step for _ in range(int(N))])
```

- The function above should return the number 1: we're adding N steps of size 1/N together

In [2]:
```
N = 5
print(count_to_one(N))
```

```
(0.2, 1.0, 1.0)
```

- great, it works, now let's try a slightly bigger number

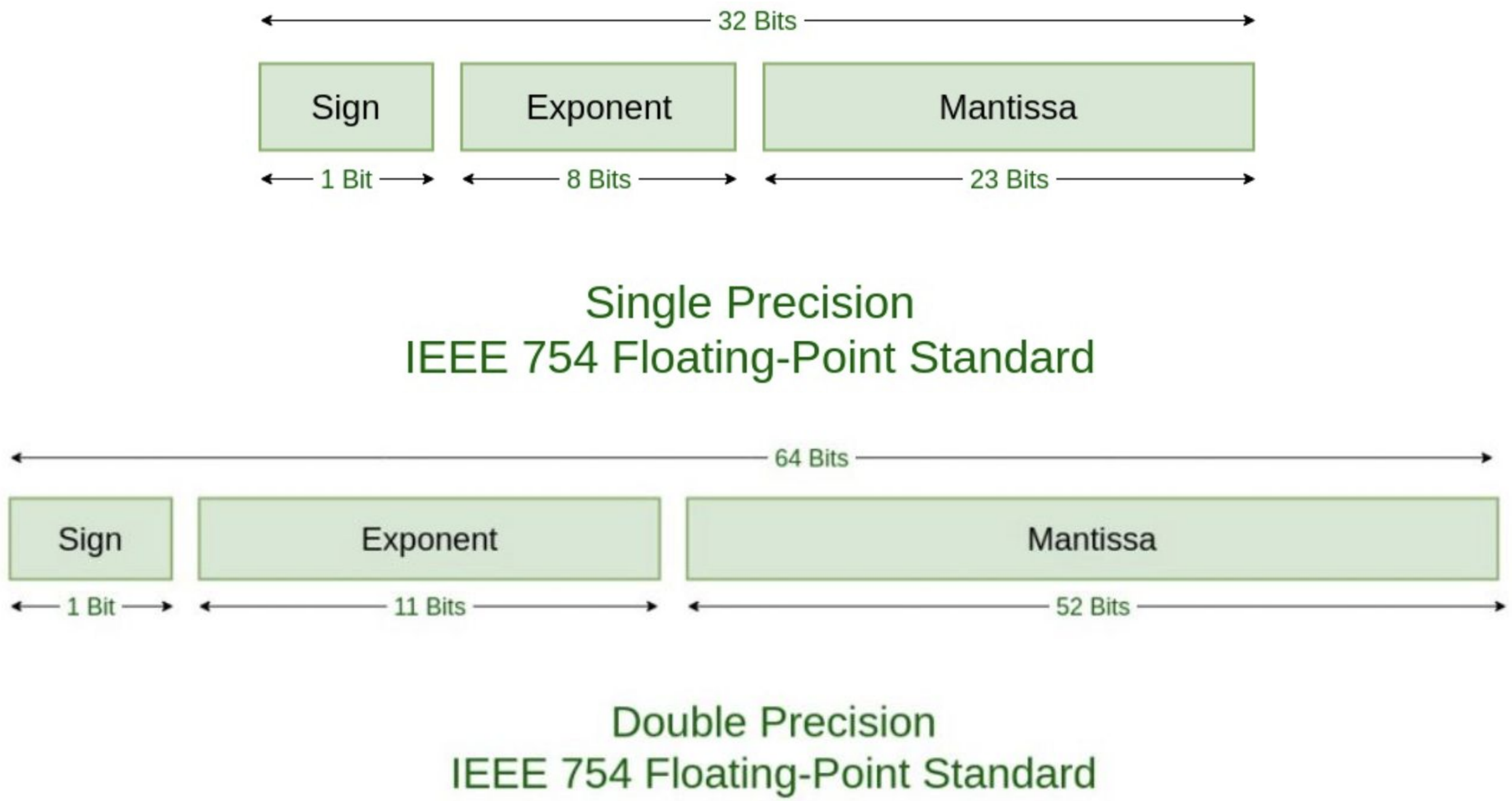
In [3]:
```
N = 1e3
print(count_to_one(N))
```

```
(0.001, 1.0, 1.0000000000000007)
```

… 🤷‍♂️ 🤦‍♀️

https://github.com/gkiar/fuzzy/blob/master/fuzzy/tutorial/

Floating Point Data are Finite



https://www.geeksforgeeks.org/ieee-standard-754-floating-point-numbers/

Floating Point Arithmetic is Inexact

E.g. addition is non associative

<u>Let's say we have 4 decimal digits of precision:</u>

$$(\ 2000. \oplus -1998. \ ) \oplus 1.333 = 2.000 \oplus 1.333 = 3.333$$

$$2000. \oplus (\ -1998. \oplus 1.333 \ ) = 2000. \oplus -1997. = 3.000$$

What is the right answer?

(inspired by Parker et al., 1997)

Floating Point Arithmetic is Inexact

E.g. addition is non associative

Catastrophic Cancellation

$$(\ 2000. \oplus -1998.\ ) \oplus 1.333 = 2.000 \oplus 1.333 = 3.333$$

$$2000. \oplus (\ -1998. \oplus 1.333\ ) = 2000. \oplus -1997. = 3.000$$

(inspired by Parker et al., 1997)

Floating Point Arithmetic is Inexact

E.g. addition is non associative

$$(\;2000. \oplus -1998.\;) \oplus 1.333 = 2.000 \oplus 1.333 = 3.333$$

$$2000. \oplus (\;-1998. \oplus 1.333\;) = 2000. \oplus -1997. = 3.000$$

Round-off Error

(inspired by Parker et al., 1997)

Monte Carlo Arithmetic (MCA)

Inexact FP quantities become random variables

$$\tilde{x} \quad = \quad \mathsf{inexact}\,(x,\, s,\, \xi) \quad = \quad x \;+\; 2^{e-s}\,\xi \qquad \text{where } e \text{ is the order of magnitude of } x$$

$$t\_\mathsf{digit\_precision}\,(x) \quad = \quad \begin{cases} x & \text{if } x \text{ can be expressed exactly with } t \text{ digits} \\ \mathsf{inexact}\,(x,\, t,\, \xi) & \text{otherwise.} \end{cases}$$

(Parker et al., 1997)

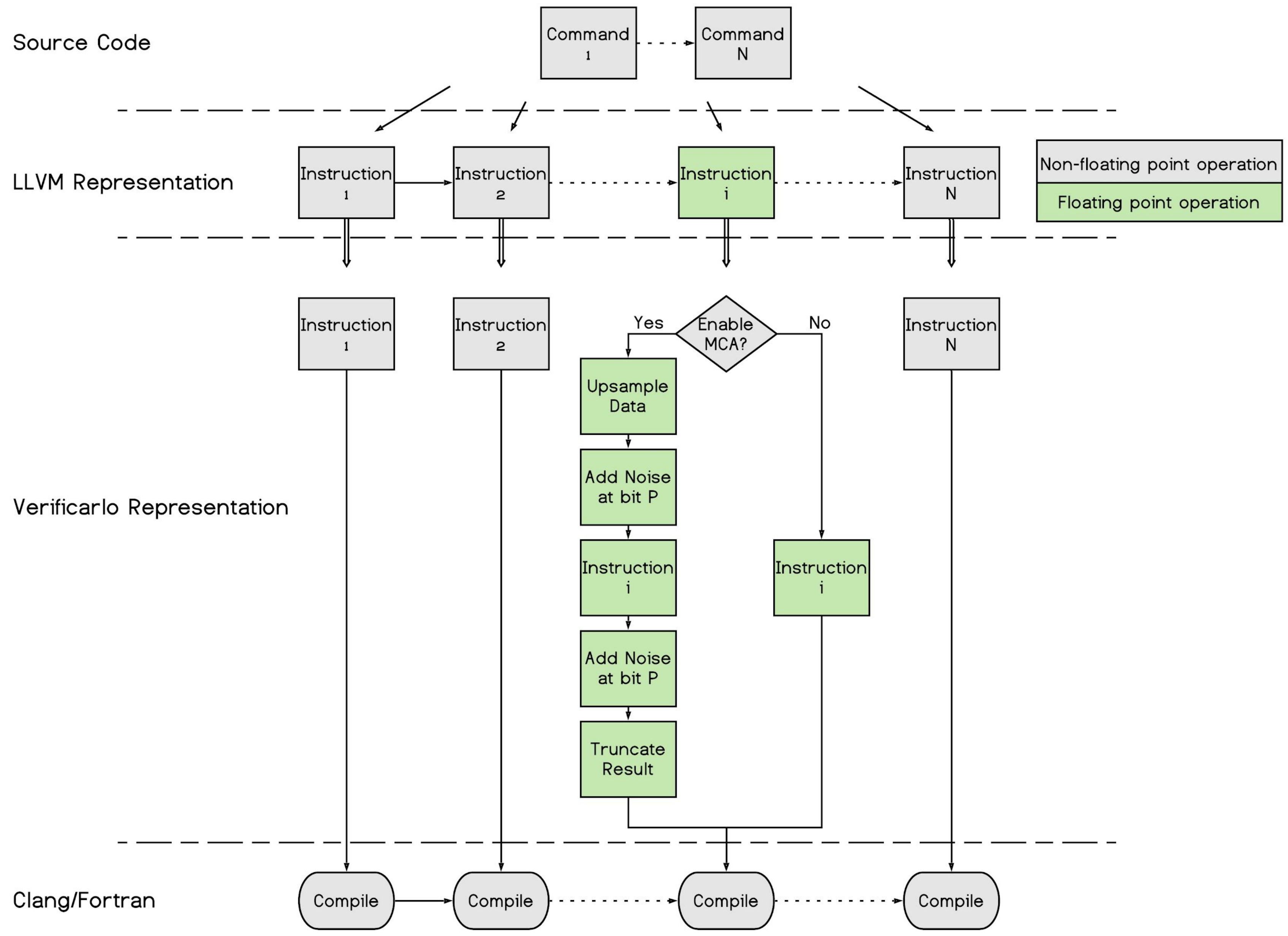
Setup

Compile C/C++/Fortran lib with Verificarlo

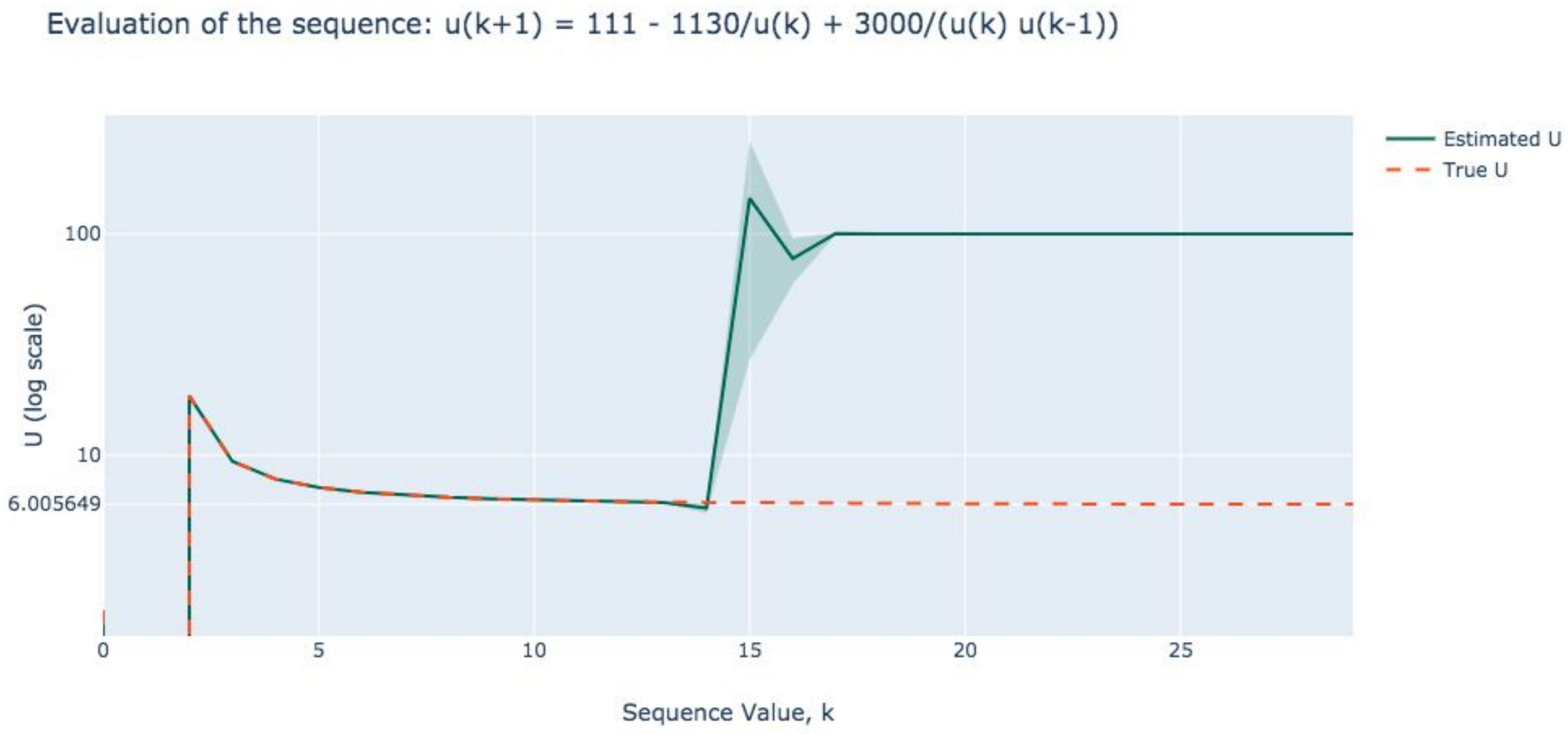
Instrumentation

1. if floating point operation:
2. {float, double} -> {double, quad}
3. (PB) simulate unrounding
4. perform operation
5. (RR) simulate rounding
6. {double, quad} -> {float, double}
7. endif

MCA


Source Code
Command 1
Command N
LLVM Representation
Instruction 1
Instruction 2
Instruction i
Instruction N
Non-floating point operation
Floating point operation
Verificarlo Representation
Instruction 1
Instruction 2
Yes
Enable MCA?
No
Instruction N
Upsample Data
Add Noise at bit P
Instruction i
Instruction i
Add Noise at bit P
Truncate Result
Clang/Fortran
Compile
Compile
Compile
Compile

MCA can help identify failure points

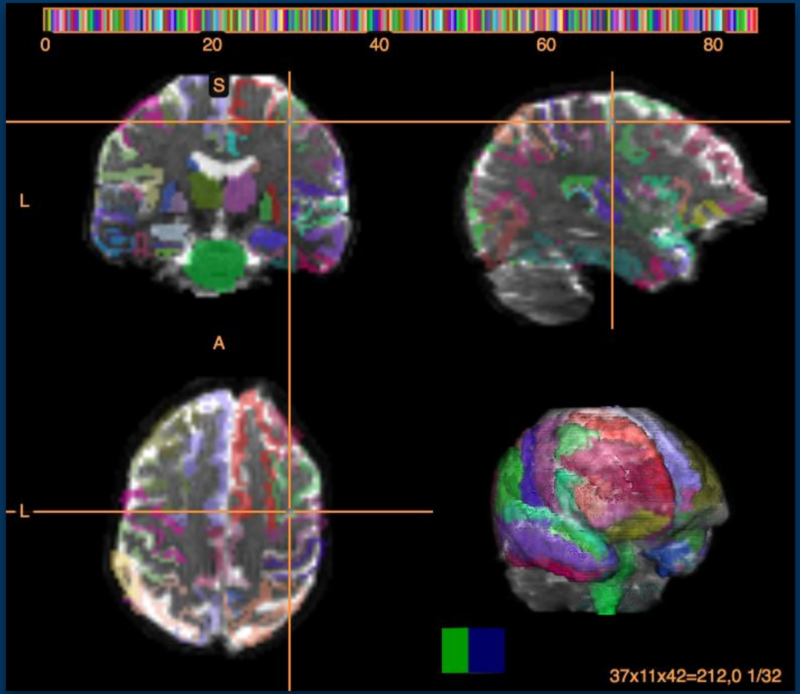


Objective:

Perturb a neuroimaging pipeline and observe induced instabilities

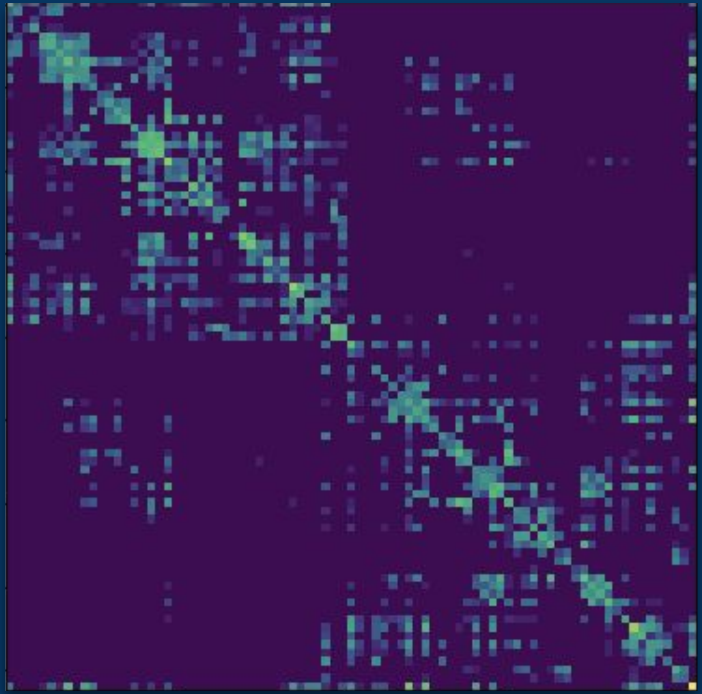
Data in Structural Connectomics

Diffusion MRI Volumes + Labels



(X, Y, Z, D) ~ O(10M voxels)

Connectivity Matrix



(83, 83) = 3403 edges

Perturbation Models

MCA

Instrument all FLOPs with…

- Full MCA
- RR only (less aggressive)

1-voxel noise

Double image intensity in…

- Single (1 vox / (X, Y, Z, D))
- Independent (1 vox / (X, Y, Z))

We Instrumented…

- Python
- Cython libs
- Numpy
- BLAS
- Lapack
- Libmath

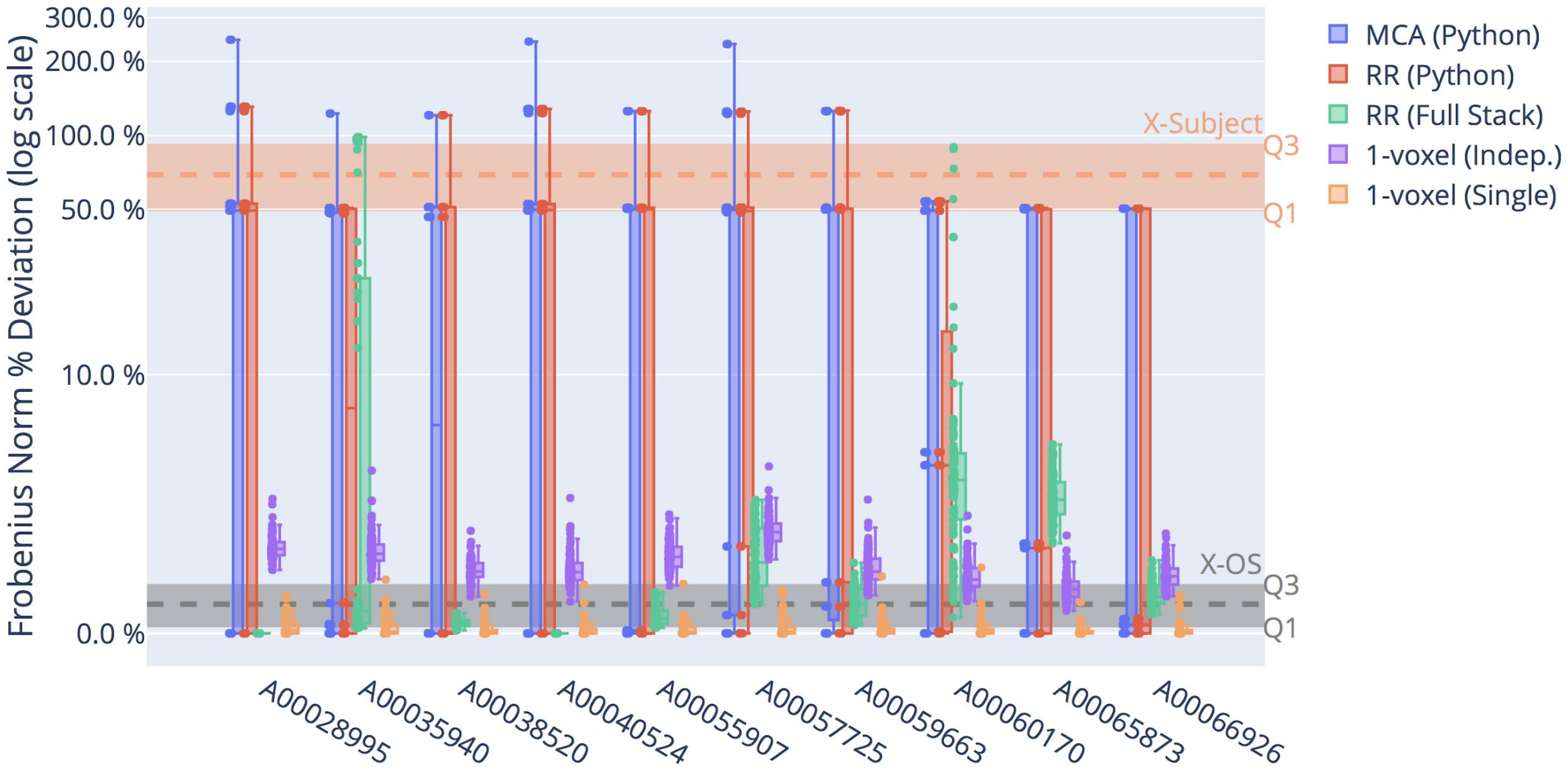
https://hub.docker.com/gkiar/fuzzy/

We ran our Pipeline with…

- MCA (Python/Cython only)
- RR (Python/Cython only)
- RR (Full Stack)
- 1-Voxel noise (per 3D volume)
- 1-Voxel noise (per 4D volume)

… 100 x each, for 10 subjects

Differences in Perturbed Structural Connectomes

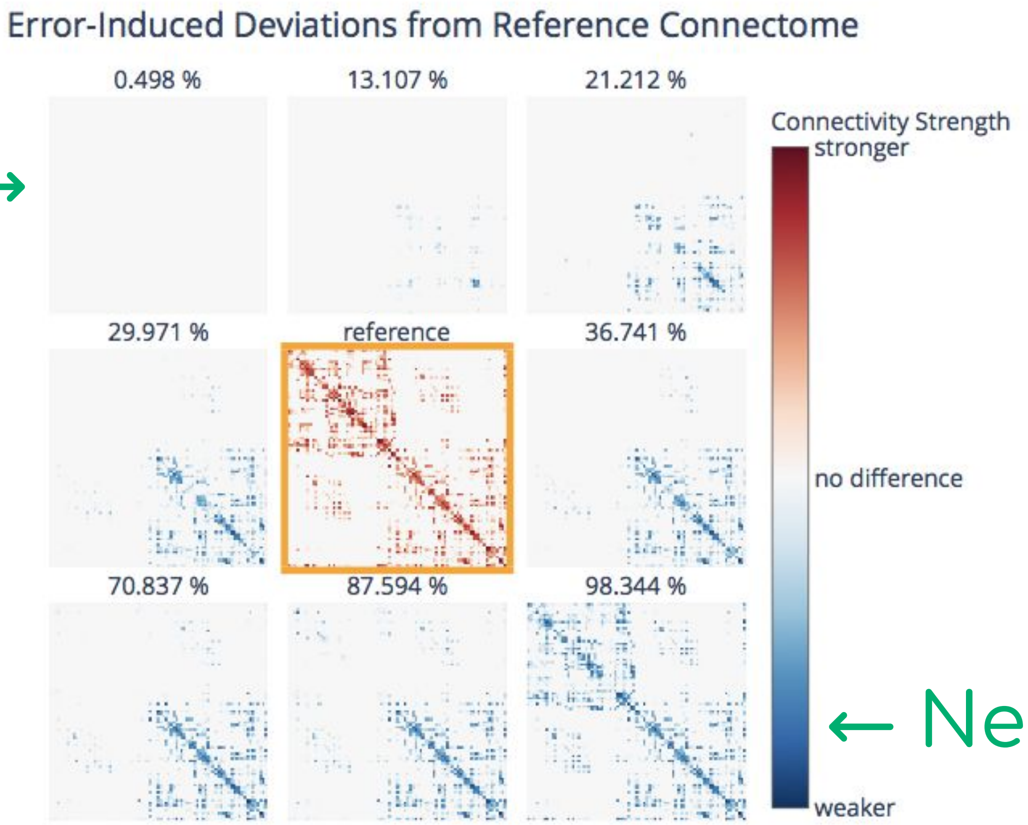


Takeaway #1:

A deterministic pipeline shows instabilities comparable to individual-level variation

What do these changes look like?

No deviation →

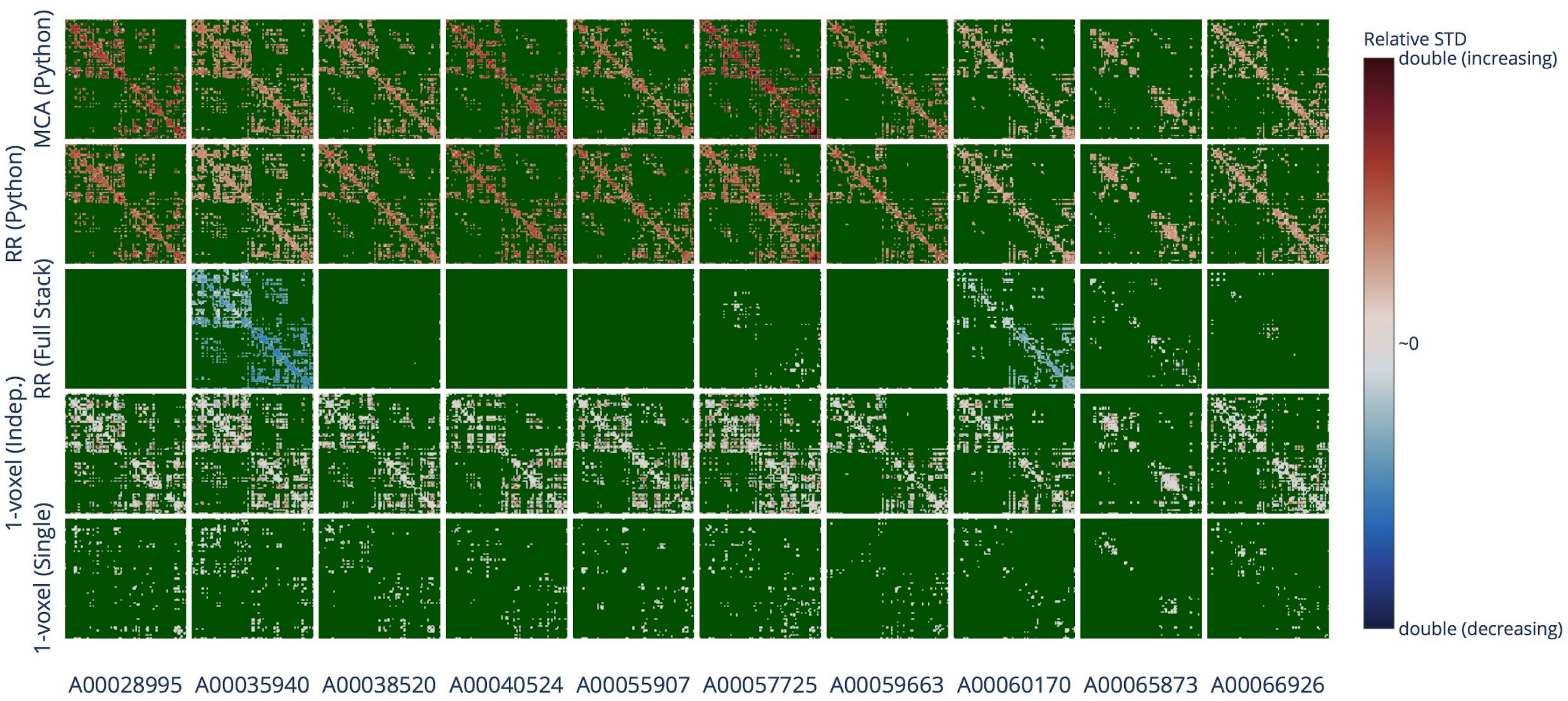


← Near full deviation

Takeaway #2:

The perturbation-induced instabilities are, at least in some cases, structured

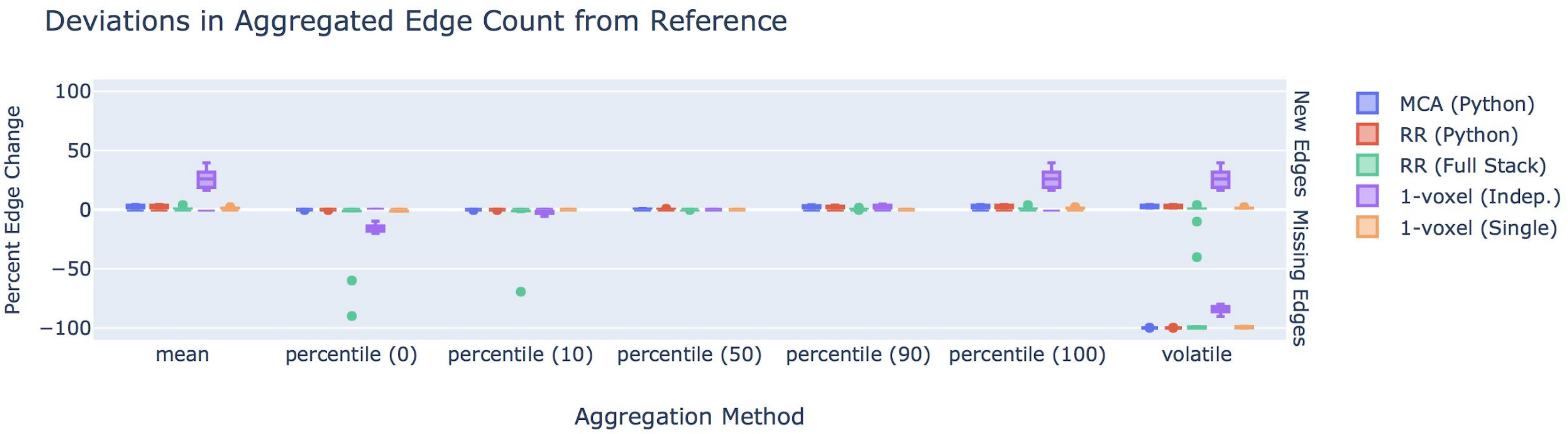
Structural Differences Across Perturbation Modes and Subjects



Takeaway #3:

Instabilities have a bi-directional effect and are highly data-dependent

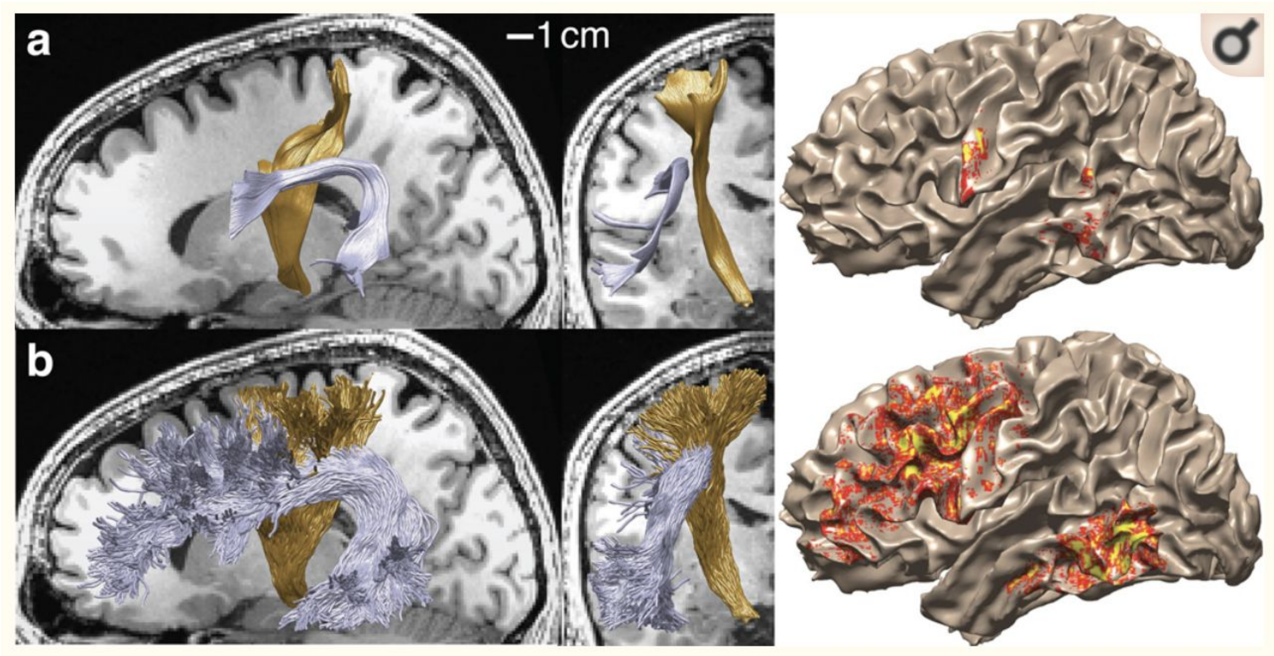
Can aggregation help?



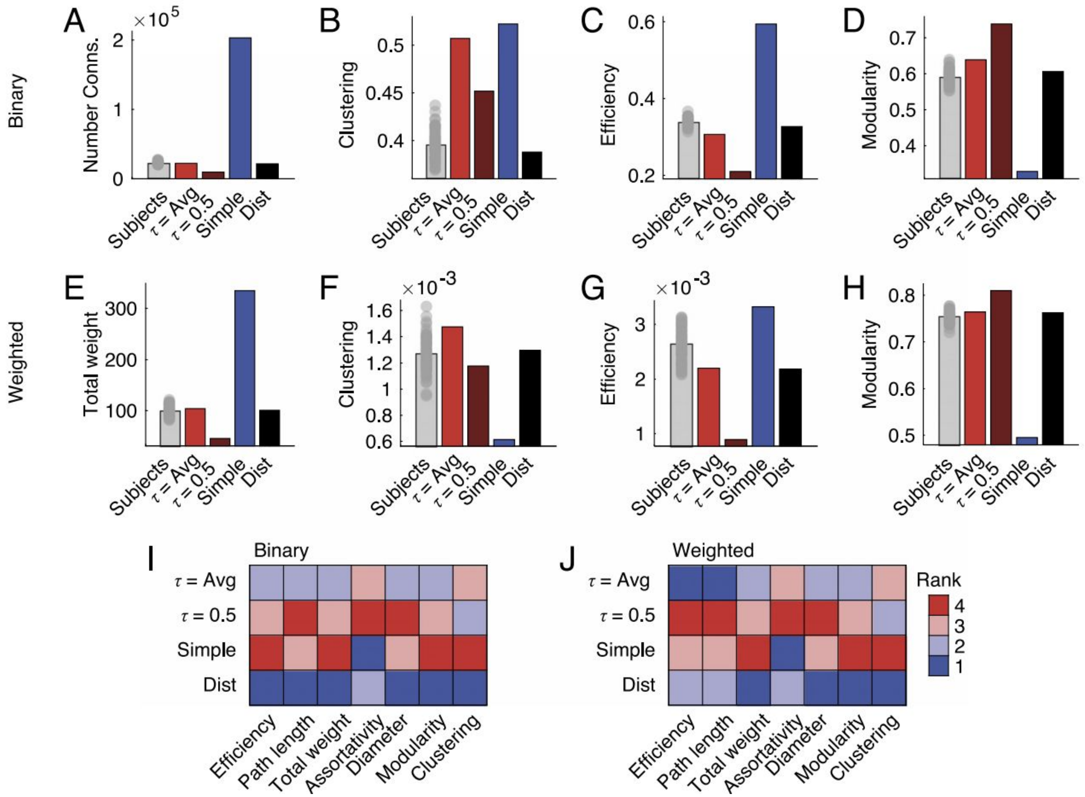
Takeaway #4:

Aggregation may create stable derivatives alongside estimates of their variance

Evaluation of "Validity" Still Needed



(Pestilli, 2015)

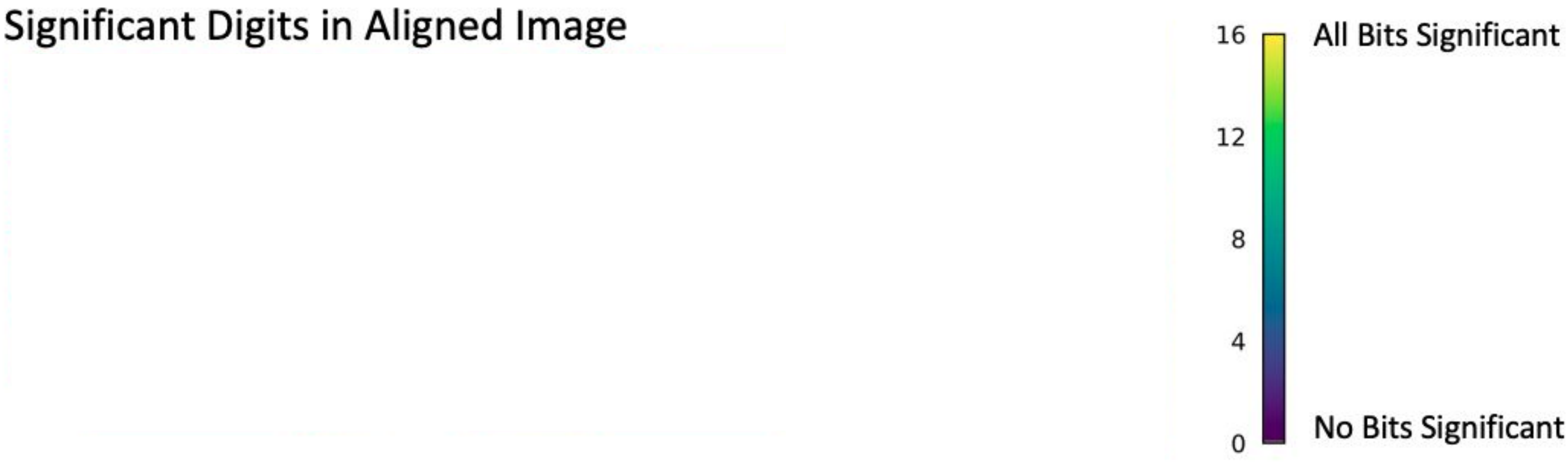


(Betzel, 2019)

And one more example...

(that does impact your work regardless of modality)

Linear Registration with FLIRT



Linear Registration with FLIRT

perfect world v1

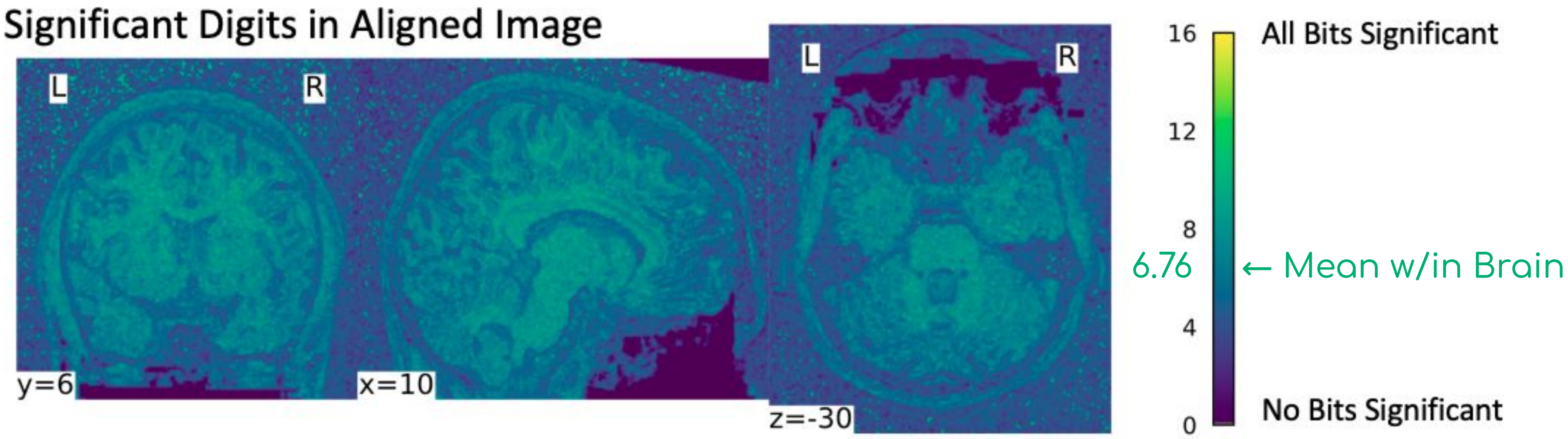
Linear Registration with FLIRT

perfect world v2

Significant Digits in Aligned Image

Linear Registration with FLIRT

real world

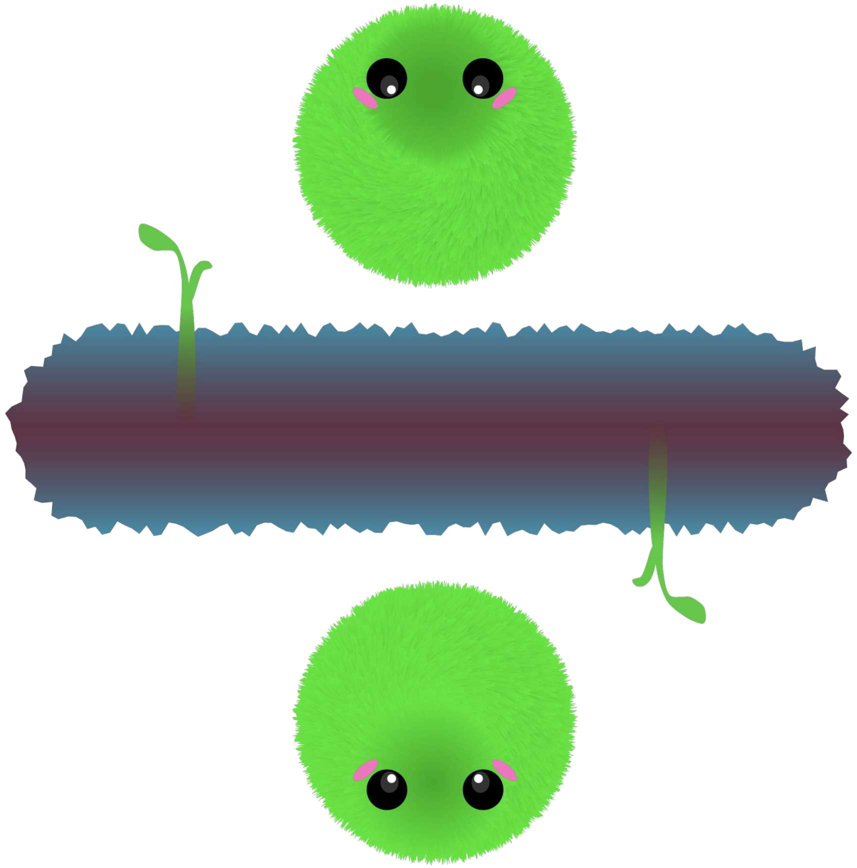


Before you ask...

- Yes, non-linear registration is coming soon
  (it just takes 90 hrs per run with MCA...)

- Yes, we can do this for "your" tool

https://github.com/gkiar/fuzzy



Base — 2 min 8 sec

✓	# 68.1	AMD64	Verificarlo	2 min 8 sec

Environments (level 1) — 21 min 42 sec

✓	# 68.2	AMD64	Blas & Lapack	7 min
✓	# 68.3	AMD64	Python	21 min 42 sec
✓	# 68.4	AMD64	Libmath	1 min 29 sec

Environments (level 2) — 21 min 51 sec

✓	# 68.5	AMD64	Python & Numpy	21 min 51 sec

Applications — 8 min 17 sec

✓	# 68.6	AMD64	FSL	5 min 20 sec
✓	# 68.7	AMD64	AFNI	8 min 17 sec
✓	# 68.8	AMD64	AFQ (Dipy)	3 min 43 sec

Summary

- Minor perturbations can induce severe instabilities in neuroimaging pipelines
- Stability analyses inseparably evaluate tool-dataset pairs, rather than either in isolation
- Jointly exploring analytical impact is in progress

All code mentioned in this presentation is publicly available on GitHub.

Thanks!

Find me @

gkiar

g_kiar

greg.kiar@mail.mcgill.ca

Acknowledgements

Fondation Brain Canada Foundation

HEALTHY BRAINS FOR HEALTHY LIVES

NSERC CRSNG

CANADA FIRST RESEARCH EXCELLENCE FUND

APOGÉE CANADA FONDS D'EXCELLENCE EN RECHERCHE

Mitacs Globalink

Inria inventeurs du monde numérique

...

Questions?