This is Your Brain on Disk: The Impact of Numerical Instabilities in Neuroscience

Gregory Kiar

Biological & Biomedical Engineering, McGill University, Montréal, QC, Canada. January, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Ph.D. in Biological & Biomedical Engineering.

© Gregory Kiar 2021

Dedication

To the family, friends, teammates, conference buddies, and mentors who supported me in not only starting this journey, but keeping me sane and taken care of throughout it. I don't know where I would be otherwise, but because of all of you, I feel like I'm exactly where I'm supposed to be.

An extra thank you to Charlie for keeping me going while I dealt with a curveball to my health in wrapping this up. She made sure I was fed, loved, sane, and somehow even happy despite everything.

Acknowledgements

First and foremost, I would like to thank my supervisors Tristan Glatard and Alan Evans for all of the guidance, support, wisdom, and encouragement they have given me over the years. I came into this program without strong direction but knowing that I wanted to impact science on a larger scale than just answering a specific question that interested me. Alan played the matchmaker and connected me with Tristan, who had similar goals to my own. Ever since, I have had the pleasure of working with both of them across a range of scales: Alan providing our eye-in-the-sky view, and Tristan getting down in the weeds with me on details. With their distinct strengths and perspectives but common optimism and positive outlook, I couldn't have asked for a better pair of advisors.

I would also like to thank Pierre Bellec, Jean-Baptiste Poline, and Christine Tardif for their ongoing contributions to my work. Having each of their unique perspectives and backgrounds on my advisory committee always led us to interesting discussions and steered me towards more impactful and rigorous work. Throughout my Ph.D. I was very lucky to spend lots of time with communities of scientists and software developers, including my office mates on the CBRAIN team, Camille Maumet and Elisa Fromont who mentored me Inria, the global Brainhack community, and the countless friends and collaborators I've met through conferences and workshops. All of these groups have not only helped guide and motivate my work, but have provided fostering and safe environments to fail and grow.

Finally, I would like to thank and acknowledge the support of various organizations which have funded my research, including Healthy Brains for Healthy Lives (supported by the Canada First Research Excellence Fund), The Natural Sciences and Engineering Research Council of Canada, Mitacs (Globalink), and The Canadian Open Neuroscience Platform (supported by Brain Canada).

Abstract

my abstract is great

Resumé

mon abstract est superb

Contents

List of Figures

List of Tables

1 Introduction

In an age of big data and an ever growing landscape of –omics', the structure and function of the brain is being increasingly explored through connectomics. Brain maps, called connectomes, are constructed as networks in which areas of the brain are related to one another through properties of interest [??]. These areas, also called regions of interest or vertices, and the properties or edges connecting them are intentionally flexible, allowing the brain networks to describe a limitless range of scales and organisms. While the connectome derived from a section of mouse brain may summarize the synapses between neurons found in an Electron Miscropy image [??], the construction of a similar map from humans *in vivo* is limited to data collected through non-invasive imaging techniques such as Magnetic Resonance Imaging (MRI) [??].

Though the resolution of a typical MRI is approximately $1,000,000\times$ less precise than an Electron Microscope image [??]which shows true cellular connectivity, MRI techniques allow for the capture of entire brains without harming the subject and can be measured over time to observe network evolution. While regions would be defined in the MRI case as macro-scale brain areas (often $1mm^3$ or larger in size [??]), the edges between them can be defined as similarity in structure [??], connectivity [??], or function [??]. Exploring these networks can not only lead to an understanding of intelligent network architectures, such as which may be replicated in fields like Artificial Intelligence (AI) and Machine Learning (ML) [??], but may also inform the development of diagnostics and healthcare interventions [??].

In recent years, several large consortia such as the UK BioBank [??], Human Connectome Project [??], and Consortium of Reproducibility and Reliability [??]have made it their mission to capture and share brain imaging data from thousands of individuals which can be used to construct maps and explore the brain. However, connectomes are not an automatic result of these images. The estimation of brain networks relies on complex image processing software tools and scientific pipelines [??], including the denoising and alignment of images, tissue classification and segmentation, and modelling or relating connectivity or function across regions. The orchestration and design of these pipelines has considerable impact on the derived networks [??]and, importantly, their application [??]. In the absence of ground-truth

to evaluate the accuracy of results generated from these tools, it is essential to understand their consistency across minor perturbations (e.g. small amounts of noise). A lack of such consistency has recently become apparent across brain imaging [??], necessitating a shift in the focus of researchers with an increasing emphasis on reproducibility.

While the irreproducibility of findings may on occasion be the result of p-hacking (i.e. the modification of analyses in search for significant results), it is often due to much more innocent means such as an inability to re-execute a previous workflow [??], software errors [??], system upgrades [??], or algorithmic instability [??]. While the existence of these problems is becoming increasingly understood and accepted, their impact on the validity scientific claims or models in neuroimaging has remained largely uncharacterized.

The objective of this thesis was to understand the role that numerical instabilities play in the reproducibility of results, and develop methods around this exploration which enable higher quality and more easily reproducible claims in neuroimaging. To this end, I have:

(i) developed a software library which facilitates and records provenance for the perfectly parallel execution, re-execution, visualization, and error detection of neuroimaging pipelines and datasets;

(ii) developed and evaluated various methods for perturbing pipelines and observing the numerical instabilities inherent to structural connectome estimation;

(iii) quantified the impact of numerical instability on a set of neuroimaging analyses measuring absolute change, dataset reliability, network topology, and the robustness of a brain-phenotype relationship; and,

(iv) improved the quality and generalizability of modelling a brain-phenotype relationship through the aggregation of connectomes in a perturbation-augmented dataset.

Ultimately, I created a piece of computational infrastructure which was used to facilitate the execution of pipelines consuming millions of CPU-hours for the induction of numerical instabilities in neuroimaging pipelines. I characterized the significant impact of these instabilities in various analytic contexts, and

proposed a method for leveraging instabilities in machine learning applications which improves the generalizability of learned models. My thesis not only sheds light on an impactful issue in neuroimaging, but it presents a method for both shedding further light on the trustworthiness of scientific tools and results, and demonstrates how scientific workflows can immediately benefit from these explorations in practice.

1.1 Contributions to Original Knowledge

I have developed and contributed to several tools which increase the accessibility of deploying and evaluating neuroimaging pipelines at scale. I have demonstrated the utility of these tools to explore and characterize the stability of neuroimaging pipelines. Below, I summarize original contributions in each of these areas.

Software Contributions

Extending the Boutiques command-line descriptive framework[1], for which I am a co-maintainer, I developed Clowdr[2] to enable the rapid deployment of scientific workflows across cloud and cluster resources. This tool is publicly available and has effectively been used on the Amazon Web Services cloud, Compute Canada, XSEDE, and Dell EMC resources, orchestrating decades of compute cycles over the resources in a matter of hours. As workflows in neuroimaging often rely on prebuilt and containerized dependencies through Docker or Singularity, I created Fuzzy (https://github.com/gkiar/fuzzy), a curated collection of scientific libraries which were recompiled and instrumented to allow the stability evaluation of the contained tools. These environments use Verificarlo[3] to instrument libraries with Monte Carlo Arithmetic. The precompiled libraries include: Python, Cython, BLAS, LAPACK, Libmath. The efficacy of the perturbations induced through these tools has been demonstrated for neuroimaging applications through several experiments mentioned in the following paragraph.

Scientific Contributions

The software contributions above were developed out of necessity for the exploration of the stability of neuroimaging analyses. First, I created and demonstrated the Fuzzy environments as an effective method for inducing instabilities in neuroimaging pipelines, and situated these instabilities with respect to other forms of perturbation[4]. In this paper, I demonstrated the considerable variability present in a structural

connectome estimation pipeline solely due to numerical uncertainty. I applied this technique to study the stability of a set of typical network neuroscience experiments and characterized the effect of instabilities on each of a test-retest, network topology, and phenotypic classification setting[5]. This work illustrates the significant impact that numerical instabilities play in all levels of downstream analysis, spanning the reliability of comparisons across subjects, to the lack of reliability in individual network features, and ultimately the modelling of relationships between connectivity and phenotypic information (in this case, Body Mass Index). The final chapter of my thesis focused on the aggregation of unstable derivatives and their impact on the performance and generalizability of machine learning classifiers tasked with learning brain-phenotype relationships. The findings of this paper showed that dataset augmentation through Monte Carlo Arithmetic leads to the development of more stable and performant classifiers, and has significant implications on the development of robust neuroimaging biomarkers, as well as the potential to increase the benefit gained from additional data collection. Together, these contributions bridge the gap between numerical analysis and neuroimaging, providing novel insights into the reliability of tools, their results, and their ultimate application in understanding brain structure.

1.2 Contributions of Authors

I was responsible for the experimental design, data processing, analysis, interpretation of results, and the majority of writing for each manuscript. Specific co-author contributions to each chapter are summarized below.

Ch.I – A Serverless Tool for Platform Agnostic Computational Experiment Management

I designed and developed the tools, experiments, and figures, and wrote the majority of the manuscript. Shawn T. Brown, Tristan Glatard, and Alan C. Evans supported the design and development processes, edited the manuscript, and provided valuable feedback. Tristan Glatard and Alan C. Evans jointly supervised this project.

Ch.II – Comparing Perturbation Models for Evaluating Stability of Neuroimaging Pipelines

I was responsible for the experimental design, tool development, data processing, analysis, interpretation, and the majority of writing. All authors contributed to the revision of the manuscript. Pablo de Oliveira

Castro, Pierre Rioux, Eric Petit, and Shawn T. Brown all provided software development support. Alan C. Evans and Tristan Glatard supported the development process and jointly supervised this project.

Ch.III – Numerical Instabilities in Analytical Pipelines Lead to Large and Meaningful Variability in Brain Networks

I was responsible for the experimental design, data processing, analysis, interpretation, and the majority of writing. , Pierre Rioux, All authors contributed to the revision of the manuscript. Yohan Chatelain, Pablo de Oliveira Castro, and Eric Petit were responsible for MCA tool development and software testing. Ariel Rokem, Gaël Varoquaux, and Bratislav Misic contributed to experimental design and interpretation. Tristan Glatard contributed to experimental design, analysis, and interpretation. Tristan Glatard and Alan C. Evans were responsible for supervising and supporting all of my contributions.

Ch.IV – Data Augmentation Through Monte Carlo Arithmetic Leads to More Generalizable Classification in Connectomics

I was responsible for the experimental design, data processing, analysis, interpretation, and the majority of writing. All authors contributed to the revision of the manuscript. Yohan Chatelain and Ali Salari provided feedback on the experimental design. Tristan Glatard and Alan C. Evans contributed to experimental design, analysis, interpretation, and jointly supervised this project.

2 Background

We will first define and discuss the state of reproducibility in science across a range of domains, and explore several specific examples. Next, we will discuss the role that software reproducibility and numerical analysis play in these problems, and advances which have been made in each of these spaces. We will then approach neuroimaging, with a brief overview of the field and commonly used methods. Finally, we will explore several case studies of reproducibility in neuroimaging to demonstrate the relevance of exploring reproducibility and stability in this space.

2.1 Reproducibility

At the heart of science is the ability for researchers to build upon previous work, incrementally advancing a given field of research [??][TMP] (some subset of refs 1-6 from open science collab). Despite being accepted as a cornerstone for progress, there is not a fully unified definition around this practice, or the lack thereof in some cases [??](https://doi.org/10.3389/fninf.2017.00076, https://www.nature.com/articles/s41562-019-0629-z, https://stm.sciencemag.org/content/8/341/341ps12.full). In practice, distinct definitions have emerged to serve specific communities tackling issues of reproducibility, such as The Association for Computing Machinery which defines reproducibility as the ability to re-obtain a measure with stated quality and precision using an identical experimental setup when carried out by a different team in a different location [??](https://www.acm.org/publications/policies/artifact-review-badging). In the case of life sciences, this definition has limited use since the equivalence of samples being measured (e.g. humans in a different city) or equipment being used (e.g. equivalent equipment from two different manufacturers) is difficult or impossible to quantify. In these cases, a distinction in the level of reproducibility is often drawn with milestones corresponding to the ability to reproduce the methods used, the ability to obtain the same or equivalent results, and the ability to draw equivalent inferences from the results obtained [??](https://doi.org/10.3389/fninf.2017.00076).

In an effort to increase clarity around this topic and handfuls of overlapping definitions, there have been recent efforts to visualize the intentions and slight conceptual differences across each [??](https://www.nature.com/articl 019-0629-z) [TMP] add figure. In the paper referenced here, a distinction is made from reproducibility and

replicability. In this case, reproducibility is defined as the ability to exactly re-run the analysis using the existing data and tools, but a unique analyst. Replicability then describes the ability to successfully re-do the experiment, including new data collection, experimenters, and software, but ultimately arrive at the same claim. These definitions have practical value in the life sciences as they allow for the differentiation between a claim remaining unchanged in either the same or differing experimental configurations and equipment. These definitions, often referred to colloquially as "Peng's Reproducibility", will be the definitions referred to throughout this thesis.

In addition to the definitions of reproducibility and replicability accepted above, I will refer to an additional term in this space: re-executability, which simply refers to the ability of being able to "hit go" on the analysis subsequent to its original execution. This definition closely matches a definition of reproducibility mentioned above, but no analog exists within the Peng framework, so it is added here for clarity.

With a rich and growing space of conceptual frameworks through which re-executability, reproducibility, and replicability can be evaluated, it is perhaps self-evident that the level of trustworthiness across many disciplines has recently become a topic of interest. I will now introduce the so-called "Reproducibility Crisis", an umbrella phrase which captures this movement, and highlight its relevance in both psychological and neurological sciences, serving as motivation for work carried out as a part of this thesis.

2.1.1 The Reproducibility Crisis

The reproducibility of findings has long been a topic of concern to researchers in the life sciences [??](https://journals.plo https://www.nature.com/articles/483531a?linkId=33568136, https://www.nature.com/articles/nrd3439-c1?linkId=33568131, https://science.sciencemag.org/content/343/6168/229.full). However, several recent initiatives have brought this somewhat niche question into an area of broad concern. In 2015, the Open Science Collaboration [??](https://science.sciencemag.org/content/349/6251/aac4716) organized an attempted replication of 100 recent research papers in psychology [??]. Their result, showing that approximately two-thirds of the studies failed to replicate ([TMP] add figure), was swept up by mainstream media and proclaimed a crisis.

In an effort to characterize the so-called crisis, Nature conducted a survey of $1,500$ scientists across a

wide range of disciplines, probing their beliefs about the reproducibility of science in their field, and found that 90% of respondants indeed felt that their field was in crisis [??](https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970).

With this topic now in the public eye, studies continued to explore various causes in attempt to understand and correct the widely shared concern. Unsurprisingly, this problem was tackled from a number of different directions, even within a given discipline. In neuroscience, statistical power has long been believed to be a culprit for irreproducibility [??](https://www.nature.com/articles/nrn3475); a meta-analytic study of the power of findings in literature estimated a median power of findings at 21%. In practice, this work assumed the robustness of data and methods used throughout the studies, and performed a purely statistical evaluation.

In neuroimaging, studies began to dive deeper into these intermediate methods. A study evalauting the accuracy of significance testing frameworks within commonly used libraries for analysis of functional-MRI data found that under certain conditions false-positive rates were as high as 70% [??](https://www.pnas.org/content/113/ This... https://onlinelibrary.wiley.com/doi/full/10.1002/hbm.24603).

While methods which have been developed and used previously, such as software written to accomplish a certain task, would continue to function in an ideal world, in practice this is not the case. A study exploring published papers in computer science found that of 600 papers which were analyzed, approximately only 200 were able to weakly replicate [??](http://repeatability.cs.arizona.edu/v2/RepeatabilityTR.pdf).

- (OSF, ...)
- `https://randomascii.wordpress.com/2014/10/09/intel-underestimates-error-b`
- Batch effects in other contexts: `https://academic.oup.com/biostatistics/article/8/1/118/252073`

2.1.2 Software Reproducibility

- (Boutiques, BIDS, CBRAIN, NIDM, ...)

2.2 Stability

words, maybe?

2.2.1 Numerical Analysis and Uncertainty Quantification

- (MCA, Verificarlo, Valgrind, ...)
- https://www.nature.com/articles/s41586-020-2649-2
- http://eprints.maths.manchester.ac.uk/2763/1/paper.pdf#page3
- https://stackoverflow.com/questions/2284860/how-does-c-compute-sin-and-ot
- https://pubs.acs.org/doi/pdf/10.1021/acs.orglett.9b03216
- MULLER, Jean-Michel, BRISEBARRE, Nicolas, DE DINECHIN, Florent, et al. Handbook of floating-point arithmetic. Birkhäuser, 2018.
- Arrondi correct de fonctions mathématiques Fonctions univariées et bivariées, certification et automatisation, Christophe Lauter
- Fonctions élémentaires : algorithmes et implémentations efficaces pour l'arrondi correct en double précision, David Defour.

2.3 Neuroimaging

2.3.1 Diffusion MRI Processing

- concept
- modalities of interest: diffusion and structural
- pipelines
- strengths and weaknesses
- (DWI strengths and limitations, Network Analysis, Applications, ...)
- https://link.springer.com/article/10.1007/s00429-020-02129-z
- https://www.pnas.org/content/111/46/16574
- https://autofq.org/bibliography/diffusion-mri/

2.3.2 Uncertainty in Neuroimaging Pipelines

- (OSes, NARPS, tractograms, CoRR, One Voxel, Cluster Failure, ...)
- Bias-variance tradeoff https://ieeexplore.ieee.org/document/824819
- `https://www.frontiersin.org/articles/10.3389/fnins.2015.00018/full`
- `https://www.biorxiv.org/content/10.1101/2020.10.07.321083v1.full.pdf`

Ch.I: A Serverless Tool for Platform Agnostic Computational Experiment Management

Gregory Kiar[1], Shawn T. Brown[1], Tristan Glatard[2], Alan C. Evans[1]

[1] *Montréal Neurological Institute, McGill University, Montréal, QC, Canada*;

[2] *Department of Computer Science and Software Engineering, Concordia University, Montréal, QC, Canada*.

Published in:

Frontiers in Neuroinformatics

https://doi.org/10.3389/fninf.2019.00012

Abstract

Neuroscience has been carried into the domain of big data and high performance computing (HPC) on the backs of initiatives in data collection and an increasingly compute-intensive tools. While managing HPC experiments requires considerable technical acumen, platforms, and standards have been developed to ease this burden on scientists. While web-portals make resources widely accessible, data organizations such as the Brain Imaging Data Structure and tool description languages such as Boutiques provide researchers with a foothold to tackle these problems using their own datasets, pipelines, and environments. While these standards lower the barrier to adoption of HPC and cloud systems for neuroscience applications, they still require the consolidation of disparate domain-specific knowledge. We present Clowdr, a lightweight tool to launch experiments on HPC systems and clouds, record rich execution records, and enable the accessible sharing and re-launch of experimental summaries and results. Clowdr uniquely sits between web platforms and bare-metal applications for experiment management by preserving the flexibility of do-it-yourself solutions while providing a low barrier for developing, deploying and disseminating neuroscientific analysis.

Ch.I - 1 Introduction

The increasing adoption of distributed computing, including cloud and high-performance computing (HPC), has played a crucial role in the expansive growth of neuroscience. With an emphasis on big-data analytics, collecting large datasets such as the Consortium for Reliability and Reproducibility[1], UK-Biobank[2], and Human Connectome Project[3] is becoming increasingly popular and necessary. While these datasets provide the opportunity for unprecedented insight into human brain function, their size makes non-automated analysis impractical.

At the backbone of science is the necessity that claims are reproducible. The reproducibility of findings has entered the spotlight as a key question of interest, and has been explored extensively in psychology[4], neuroimaging[5,6], and other domains[7,8]. Computational experiments must be re-executable as a critical condition for reproducibility, and this bare minimum requirement becomes increasingly challenging with larger datasets and more complex analyses. While sharing all code and data involved may appear a compelling solution, this is often inadequate for achieving re-runnability or reproducibility of the presented findings and models[8]. When re-executable applications fail to reproduce findings, there is a gray area where the source of errors are often unknown and may be linked to misinterpretation of data, computing resources or undocumented execution details, rather than scientific meaning.

As a result, new tools and standards have emerged to aid in producing more reusable datasets and tools, and thereby more reproducible science. The Brain Imaging Data Structure (BIDS)[9] and the associated BIDS apps[10] prescribe a standard for sharing and accessing datasets, and therefore, increasing the accessibility and impact of both datasets and tools. This standard includes the definition of file organization on disk, as well as key-value pairs of metadata information in JavaScript Object Notation (JSON) files, and assigns specific meaning to command-line arguments to be used when processing these datasets. The Boutiques framework[11] provides a standard for software documentation in a machine-interpretable way, allowing the automation of tool execution and evaluation. These descriptions fully encapsulate the runtime instructions for a given tool in JSON-files, and are appropriate for a majority of command-line applications. Software containerization initiatives such as Docker[12] and Singularity[13] facilitate execution consistently across arbitrary computing environments with minimal burden on the user.

Web-platforms such as OpenNeuro[14], LONI Pipeline[15], and CBRAIN[16] simplify the analysis process further by providing an accessible way to construct neuroscience experiments on commonly used tools and uploaded-datasets. These systems deploy tools on HPC environments and record detailed execution

information so that scientists can keep accurate records and debug their workflows. Tools such as LONI's provenance manager[17], Reprozip[18], and ReCAP[19] capture system-level properties such as system resources consumed and files accessed, where tools supporting the Neuroimaging Data Model (NIDM)[20], a neuroimaging-specific provenance model based on W3C-PROV[21], capture information about the domain-specific transformations applied to the data of interest.

The initiatives enumerated above have synergistic relationships, where each solves a small but significant piece of the larger puzzle that is computational and scientific reproducibility and replicability. However, the learning curve associated with adopting each of these technologies is considerable, leveraging them in an impactful way is difficult, and certain applications may benefit from different approaches so these learning curves may have to be paid multiple times. For instance, interoperability is mainly valuable in contexts which there is a large variety of datasets or tools, and provenance may be of importance to identify the impact of an underlying system on a processing or modeling task. We present Clowdr, a lightweight tool which ties these approaches together so that researchers can minimize the learning burden and lower the barrier to develop, perform, and disseminate reproducible, interoperable and provenance-rich neuroscience experiments.

Ch.I - 2 Emergent Technologies in Reproducible Neuroscience

Conducting reproducible analyses in neuroscience requires many complementary facets, building on technologies which are commonly adopted as *de facto* standards.

Ch.I - 2.1 Data and Code Interoperability

Due in part to its simplicity and active public development community, BIDS[9] has become an increasingly prominent data organization format in neuroimaging. This standard makes use of the Nifti file format[22] and human-readable JSON files to capture both imaging and subject-specific metadata. An important benefit of this data organization is the ability to launch data processing pipelines in the form of BIDS applications[10], which expose a consistent set of instructions compatible with the data organization. Together, these complementary standards are suitable for performing a large variety of neuroimaging experiments. In contexts where tools have heterogeneous interfaces, or data organizations are custom-built for a particular context, the Boutiques[11] framework allows the rich description of a pipeline such that tool execution, validation, and exploration can be automated. These descriptors include the command-line structure to be populated as well as rich parameter descriptions and interactions, such as mutually exclusivity or

dependence, such that complicated data interactions required or forbidden by the tool can be accounted for.

Ch.I - 2.2 Software Virtualization

While virtual machines have long been used for deploying analysis pipelines with complex dependencies in heterogeneous environments, software containers have recently emerged as lighter-weight alternatives suitable for transient data processing applications. Docker[12] provides this functionality across all major host operating systems, but is often not supported by HPC centers due to security vulnerabilities[23,24]. Singularity[13] addresses the security risks of Docker, but currently only supports Linux operating systems, filling the niche of containerization on shared computing resources. A detailed comparison in the context of medical imaging can be found in[25].

Ch.I - 2.3 Workflow Engines

Custom scientific pipelines can be composed in Python with Nipype[26], Dask[27], Pegasus[28], Toil[29], or several other tools which facilitate the modular interaction of complex independent processing stages. While the underlying tasks in Nipype, Dask, and Toil are defined in Python, Pegasus uses a Domain Specific Language (DSL) for representing tasks, increasing the barrier for defining tasks but ultimately increasing their portability. While Nipype is a widely used tool in neuroimaging and has many readily-defined interfaces available for researchers, the others require non-insignificant development to describe interfaces for common neuroimaging applications such as FSL[30] or MRtrix[31]. PSOM[32] and Scipipe[33] are functionally similar to Nipype but have been developed for GNU Octave/MATLAB and Golang, respectively. Several domains have more specialized tools which accomplish similar feats in their area of interest. These include Pypet[34], Neuromanager[35], Arachne[36], and others which facilitate the automation of modeling and simulation workflows using tools such as Neuron[37]. For an in depth look at other tools in this space please refer to[34] and[35]. Each of these tools enables the construction of dependency graphs between pipeline components, and allow the deployment either to cluster scheduling software, multiple processing threads, and in some cases computing clouds. These tools primarily function through programmatic interfaces, though LONI pipeline[15], OpenMOLE[38], and Galaxy[39] provide both DSL and graphical user interfaces. Many of these tools also embed provenance capture, fault-tolerance features, and data tracking to avoid recomputations across similar executions. While each of these tools is a powerful and attractive option for creating workflows, they remain complex and potentially overkill when launching atomic single-step analyses, prebuilt pipelines, or analysis software developed in a different language than

the workflow engine of choice.

Ch.I - 2.4 Provenance

Building on the W3-PROV[21] standard for data provenance metadata put forth by the World Wide Web Consortium, NIDM[20] is a standard which represents a processing and data provenance graph specific to neuroimaging analyses. While this standard is machine-interpretable and interoperable-by-design, supporting it currently requires tight integration with analysis pipelines. In LONI pipeline, a provenance model exists which includes detailed records of data use and file lifecycle[17], which is designed to inform data consumers what types of analyses can be and have been performed with the data in question; this tool is tightly coupled with the LONI pipeline ecosystem. The ReCAP[19] project has been developed to evaluate the resource consumption of arbitrary pipelines on the cloud and can aid in cloud-instance optimization. While this tool has potential for a large impact in designing both cost effective and scalable analyses, there is considerable overhead as it manages executions through a persistent server and workflow engine. While various other libraries exist to monitor some piece of data or compute provenance, Reprozip[18] is perhaps the most exciting as it uniquely captures records of all files accessed and created throughout an execution, which allows for the creation of rich file dependency graphs. The limitation of this technique is that it requires data of interest to be written to disk, as opposed to managed in memory, which may not always be the case in some applications.

Ch.I - 2.5 Web Platforms

Increasing the portability and accessibility of launching large scale analyses, web platforms such as CBRAIN[16], LONI pipeline[15], and OpenNeuro[14] provide science-as-a-service where users can upload and process their data on distant computing resources. Additionally, these platforms provide an accessible and immediate way to share the results produced from experiments with collaborators or the public. These tools provide incredible value to the community and allow the deployment of production-level pipelines from the web, but they are not suitable for prototyping analyses or developing pipelines, and it is cumbersome to run these services on a lab's own resources. In addition, monolithic Web interfaces are only suitable for a certain type of use-cases and high-level users, while developers or computer-savvy users prefer to rely on modular command-line tools and libraries.

Ch.I - 3 The Clowdr Microtool

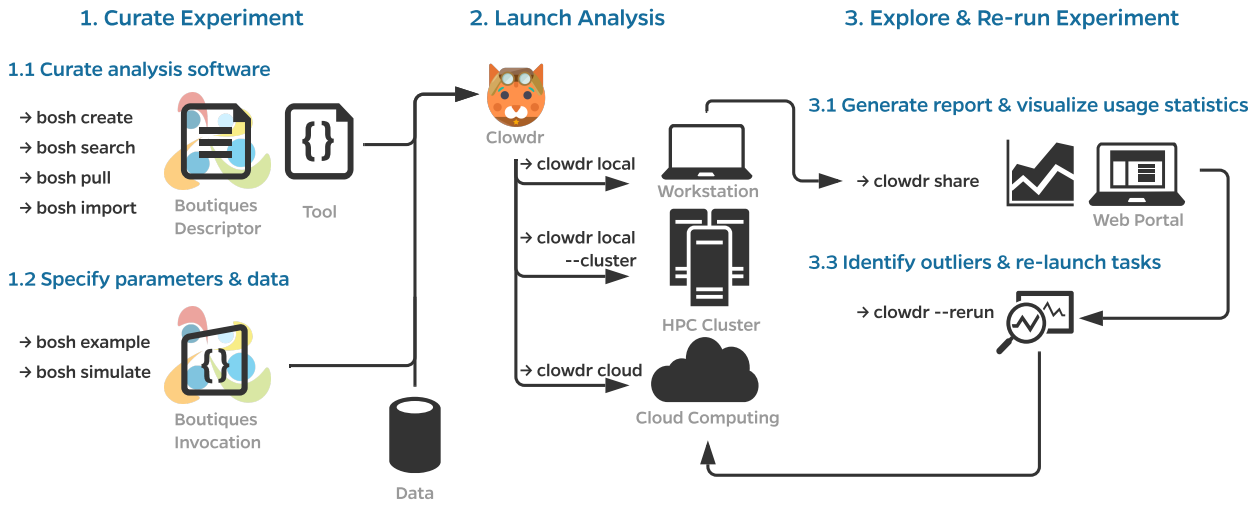

Figure Ch.I - 1. Clowdr Workflow. **(1)** Prior to launching an analysis with Clowdr, users must curate the analysis tools and their inputs. For the sake of portability, Clowdr supports both native and containerized applications described in the Boutiques format. Several tools exist in Boutiques which simplify the adoption/creation or execution of tools and are enumerated in (**1.1**,**1.2**), respectively. **(2)** Scientists can then launch their analysis with Clowdr either locally, on HPC systems, or computing clouds. Possible workflows could involve the tuning of hyperparameters locally on a subset of the dataset of interest, and ultimately deploying the analysis at scale using the same arguments, or sweeping hyperparameter values on an HPC system. **(3)** After execution, summary reports can be produced by Clowdr **(3.1)** and visualized through a custom web portal enabling filtering by both execution properties and parameters, facilitating outlier detection and comparison across executions. Identified outliers, such as failures, incomplete tasks, or those which consumed more resources than expected can be re-run through Clowdr without having to regenerate any of the information previously provided. Clowdr facilitates the development, deployment, and debugging of analyses in a closed-loop provenance-rich microservice.

While the technologies enumerated above are essential pieces toward reproducible neuroscience, they are largely isolated from one another and place a large burden on researchers who wish to adopt all of these best practices. Clowdr leverages these tools to increase the deployability, provenance capture, and shareability of experiments. In summary, Clowdr:

i is tightly based on Boutiques and is BIDS-aware, supporting both arbitrary pipelines and providing

an accessible entrypoint for neuroimaging;

ii executes both bare-metal workflows and Docker or Singularity virtualized pipelines through Boutiques on local, HPC, and cloud resources;

iii supports the parallelized batch deployment and redeployment of pipelines constructed with workflow-engines, while being agnostic to programming language and construct;

iv captures system-level provenance information (i.e., CPU and RAM usage), supports Reprozip, and internal provenance captured by arbitrary pipelines such as NIDM; and

v supports the deployment of both development- and production-level tools without an active server, and provides a web-report for exploring and sharing experiments.

A typical workflow using Clowdr is summarized in Figure Ch.I - 1. While a Clowdr experiment follows the same workflow as traditional experiments, beginning with tool and data curation through prototyping, deployment, and exploration, there are several considerable benefits provided by Clowdr over traditional approaches. In particular, Clowdr is based on the rich Boutiques framework for tool description and execution, ensuring that documentation, parameter definitions, and real-world parameter values accompany the tool at all times. Clowdr also treats all computing systems the same, from the users perspective, so transitioning from local development of analyses to at-scale systems is seamless, which minimizes errors made during this transition. Clowdr also provides a visualization portal for exploring executions and filtering either based on parameter values or runtime statistics, allowing for quality control of the execution in addition to commonly used quality control of processed derivatives themselves.

Figure Ch.I - 2 shows the execution lifecycle within Clowdr. Starting from user-provided Boutiques descriptor (B) and invocation(s) (C), and access to any required datasets, Clowdr begins by identifying a list of tasks to launch. For a new experiment, tasks are identified in one of three main ways: (1) a one:one mapping from a list of invocations, (2) a one:many mapping from a single invocation in which parameter(s) have been specified for sweeping during execution, or a BIDS-specific, and (3) one:many mapping from a single invocation for a BIDS app, which will iterate upon the participant- and session-label fields, and described in the BIDS app specification[10]. Experiments can be re-run, and determine the task-list based on whether a full, failure-only, or incomplete-only re-execution is desired. Once the task-list is determined, Clowdr creates an independent invocation which explicitly defines the arguments used in each task.

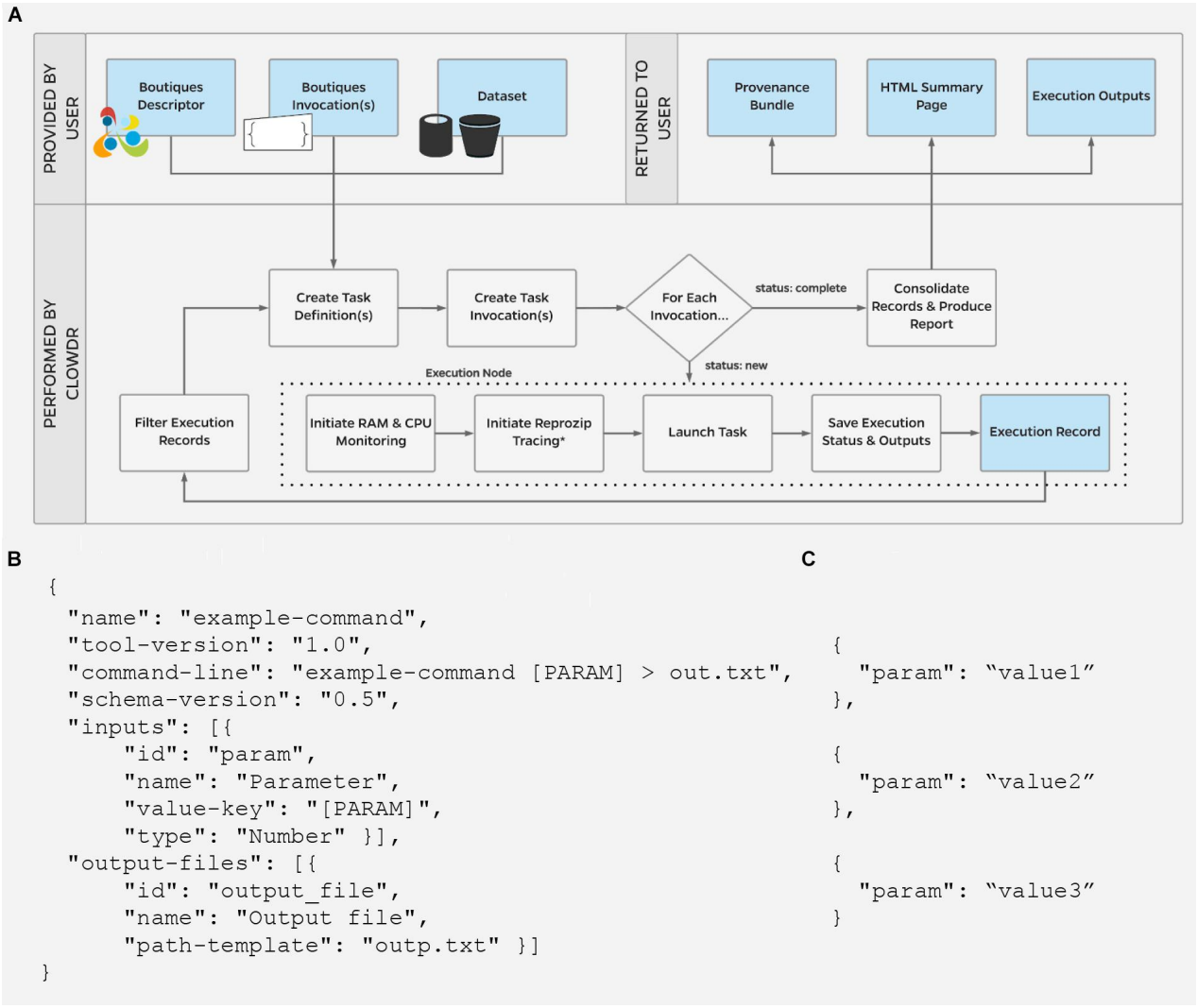

Figure Ch.I - 2. **(A)** Clowdr Data Flow. Beginning with a user-supplied tool descriptor **(B)** and parameter invocation(s) **(C)**, Clowdr identifies unique tasks to launch and wraps each with usage and log monitoring tools, to ultimately provide a rich record of execution to the user alongside the expected output products of the experiment. Clowdr ultimately produces an HTML summary for users to explore, update, filter, and share the record of their experiment. In the above schematic, blue boxes indicate data, where gray indicate processing steps. *External reprozip tracing is supported on limited infrastructures, as running virtualized environments within a trace capture requires elevated privileges which may be a security risk on some systems.

At this stage, Clowdr distributes tasks to the Cloud system or local cluster scheduler being used for deployment. Presently Clowdr supports the SLURM scheduler and Amazon Web Services (AWS) cloud through their Batch service with adoption of more platforms ongoing. Each task is launched through

a Clowdr-wrapper, which initializes CPU and RAM monitoring and triggers Reprozip tracing prior to launching the analysis itself. Reprozip tracing has limited support in conjunction with containerized analyses on HPC systems due to potential security issues. Reprozip is built upon the Linux command "ptrace," which traces processes to monitor or control them. To eliminate the potential risk of using this tool, it is common for systems to disallow the tracing of administrator-level processes. The requirement of limited administrator privileges by Singularity (during the creation of multiple user namespaces) and Docker (for interacting with the daemon) makes encapsulating these tools within the restricted ptrace scope not possible on many shared systems. For more information on the specific conditions in which these technologies can be made to interoperate please view the GitHub repository for this manuscript, linked below.

Upon completion of the analysis, Clowdr bundles the system monitored records, standard output and error, exit status, and any other information collected by either the tool itself or the Boutiques runtime engine, and concludes its execution. Once the experiment has begun, Clowdr provides the user with the Clowdr provenance directory which will be updated automatically as executions progress.

The researcher can monitor the provenance directory using the Clowdr share portal (Figure Ch.I - 3), which provides a web interface summarizing the task executions. Once the analysis concludes, the figures on this web page and the associated metadata can be saved and serve as a record of the experiment either for evaluation or dissemination alongside published results.

The Clowdr package is open-source on GitHub[41], and installable through the Python Package Index.

Ch.I - 4 Performing Experiments With Clowdr

Here, we explore an experiment in which we used Clowdr to process the Human Connectome Project (HCP)[3] dataset with a structural and functional connectome estimation pipeline, ndmg[40]. The records of this experiment, and materials and instructions that can be used to reproduce a similar analysis with Clowdr using the publicly available DS114 BIDS dataset[14] an the example BIDS application[10] can be found on Github at: `https://github.com/clowdr/clowdr-paper`. Specific packages and their versions for both experiments can be found at the end of this manuscript.

As summarized above, performing an analysis with Clowdr requires the creation of a Boutiques descriptor summarizing the pipeline of interest, an invocation containing the parameters to supply to this pipeline on execution, and curation of the data to be processed. There are several utilities in Boutiques

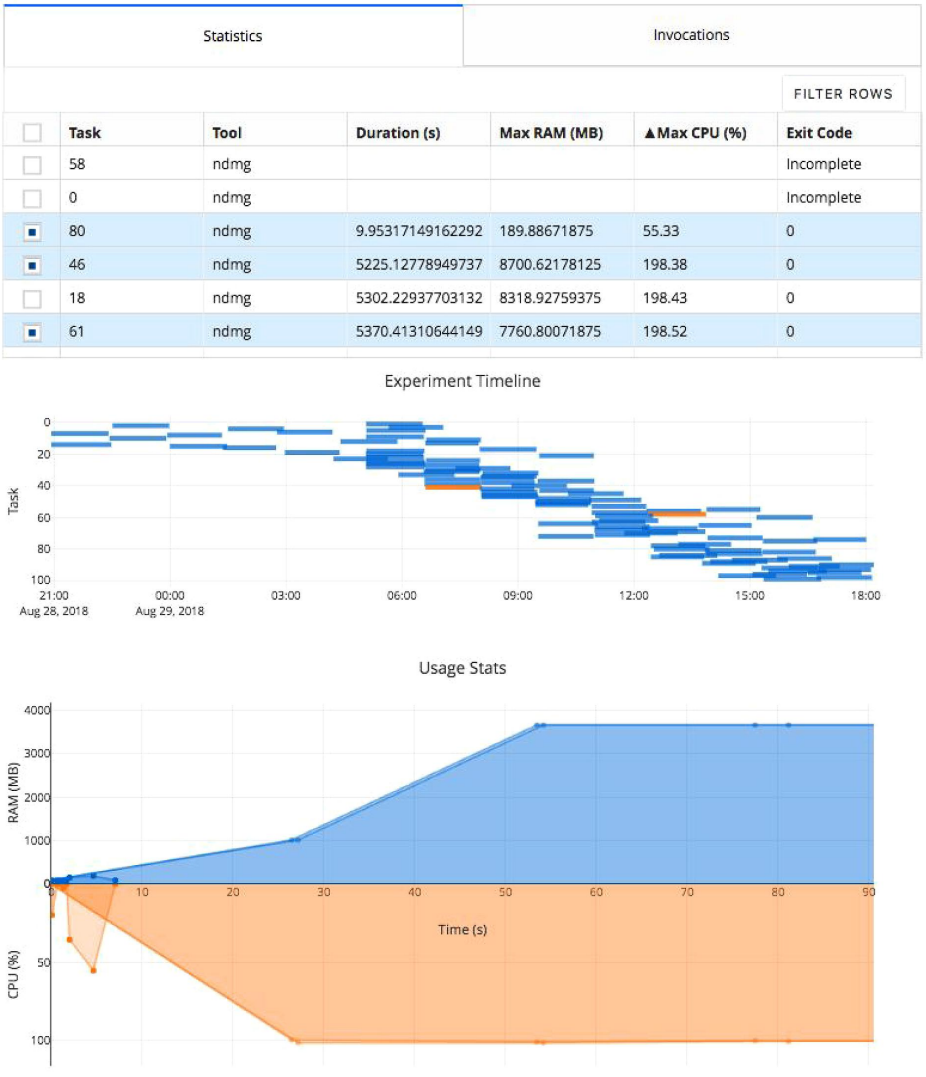

Figure Ch.I - 3. Clowdr Experiment Viewer. Experiments launched with Clowdr can be monitored and both progress and runtime statistics explored. The page is produced using Plotly Dash to produce highly interactive plots and tables, enabling rich filtering, rescaling, and exploration of executions. The table can be toggled to present summary statistics about experiment execution or invocation parameters identifying parameters used for each task in the experiment. The subsequent Gantt plot shows the timeline of executed tasks in the experiment, where those selected for visualization in the usage plot below are highlighted. The final plot in this view shows the memory and processing footprint throughout all selected tasks. Selection and filtering may be done by value in the tables or selection in the task timeline. In this example, several tasks did not complete and one appeared to exit after 10 s erroneously. The Clowdr portal enables quick identification of these outliers, and the table view can be switched to identify more information such as parameters pertaining to the executions of interest. For more information about the pipeline being executed in particular, please see[40].

which aid in this setup process, including to automatically generate a descriptor and sample inputs from a tool, and can be explored in the associated documentation[11].

Clowdr experiments can be launched locally, on cluster, or submitted to cloud resources. In each case, invocations and task definitions are created locally, and then the jobs are run serially, submitted to a cluster queue, or pushed to cloud storage and called remotely. The commands used in Clowdr to launch these commands are the local, local with the cluster switch, and cloud modes, respectively. Upon completion of each tasks, summary files created by Clowdr can be either inspected manually or consolidated and visualized in the web with the Clowdr share command (Figure Ch.I - 3).

The share tool, launchable on any computer with access to the experiment, creates a lightweight web service displaying summary statistics and invocation information from the experiment, including memory usage, task duration, launch order, and log information. The visualizations provided are filterable and sortable, enabling users to interrogate and identify outliers in their experiment, explore potential sources of failure, and effectively profile the analysis pipeline in use. The modified figures can be downloaded from this interface, serving as accessible records of execution.

In the example above, the HCP dataset has been processed using a pipeline performing image denoising, registration, model fitting, and connectivity estimation, all of which are commonly used processing steps in neuroimaging. For more information on this pipeline, please see[40].

In this experiment the table has been filtered to show several tasks which appeared spurious in their execution compared to the others. We can see that several tasks failed to complete and one appeared to terminate in significantly less time than the others. After identifying these tasks and exploring the time series' to see at what stage of processing the job failed (in this case, immediately), we can investigate parameter selections used in each and attempt the re-execution of these jobs using the local or cloud command with the rerun switch in Clowdr. Clowdr provides a layer of quality control on executions, in addition to that which is regularly performed by researchers on their datasets, which provides immediate value when identifying task failures which otherwise may be difficult to identify, especially in cases which intermediate and terminal derivatives are written to the same location, which can often be the case with transformations estimated by registration pipelines, for example.

While the share tool currently requires maintaining an active server, the plots can be exported statically and it is in the development roadmap to enable exporting the entire web page as static files, as discussed here: `https://github.com/plotly/dash/issues/266`. Since the record created by Clowdr is stored in the machine-readable and JSON format, researchers can easily extract their records and

integrate it into other interfaces that suit their application.

Ch.I - 5 Discussion

Clowdr addresses several barriers to performing reproducible neuroscience. Clowdr experiments consist of enclosed computational environments, versioned-controlled Boutiques-described tools with explicit usage parameters, rich execution history, and can be re-executed or distributed with minimal effort. Clowdr provides an accessible interface for initially running analyses locally, and translating them seamlessly to HPC environments. The rich record keeping provided with Clowdr is system-agnostic resulting in uniformly interpretable summaries of execution. As a Python library, Clowdr can be used as a module in a larger platform, or directly as a command-line tool.

Clowdr uniquely packages an executable tool summary, parameters, and results together, in a language- and tool-agnostic way, and therefore, greatly increases the transparency and shareability of experiments. Importantly, this adds clarity to experimental failures and documents the hyper-parameter tuning process of experiments, which has been historically largely undocumented in literature[42].

There are several axes upon which the value of Clowdr can be discussed. In particular: lines of code written, time spent, and the ultimate re-runnability of analyses. While these remain subjective areas for comparison, we can conceptually consider a workflow dependent on Clowdr to those constructed with traditional scripting, workflow engines (WEs), and software-as-a-service (SaaS) platforms.

Where Clowdr has been built upon tools and standards to provide users with a series of single-commands for launching and managing analyses, accomplishing a similar result with traditional scripting would take considerably more lines of code and time. Similarly, where command-line execution may be similar in complexity to tools developed with WEs, their integration within tools requires substantial development and is only practical in cases for which there is a WE written in the same language as the underlying application. SaaS platforms provide a similar type of abstraction to Clowdr, where tools are treated as black-box objects, but come with the added overhead of maintaining complex database architectures, often complex integration of tools, and primarily restrict access through web-based interfaces which leads to reduced flexibility for the user.

The clear benefit of Clowdr is in the simplicity it provides for identifying outliers or failed tasks and either re-launching specific subsets of an analysis or the entire experiment. Clowdr records and visualizes detailed logging information about executions and the specific instructions which were used, which isn't

guaranteed in either traditional scripting or WE-based applications. To replicate this feature across these systems, tools which (1) record execution instructions, (2) identify parameters used for parallelization, (3) produce summary plots, and (4) reconstruct and (5) re-execute instructions would require development.

While SaaS platforms often contain these features, an additional limitation of large platforms is that they are often designed for consumers of widely adopted tools consumers rather than tool developers. Clowdr fills the void between these types of pipeline deployments by providing a programmatic tool-independent method for managing job submission and collecting provenance across multiple architectures and enabling the rapid prototyping of analyses.

Several immediate applications of the provenance information captured by Clowdr include the benchmarking of tools, and resource optimization during the selection of cloud resources, as was done in[19]. While the value of comparing provenance records has not been demonstrated here, other studies such as[43] have demonstrated the efficacy of leveraging provenance information to identify sources of variability or instability within pipelines.

Future work includes adopting a W3C-PROV compatible format for Clowdr provenance records, increasing the machine-readability and interoperability of these records with other standards such as NIDM. Integrating the reports produced by Clowdr with a system such as Datalad would allow for record versioning and more strictly enforce the complete reporting of experiments. Clowdr will also continually be extended with greater testing and support for more HPC schedulers, clouds, and provenance capture models.

Tools and Versions

The following is a list of tools and data used in this manuscript, and their respective versions. The architecture and analysis presented for the Clowdr package corresponds to version 0.1.0. The key Python packages and specific versions tested are: boutiques (version 0.5.12), boto3 (1.7.81), botocore (1.10.81), slurmpy (0.0.7), psutil (5.4.7), pandas (0.23.4), plotly (3.1.1), and plotly dash (0.24.1), including dash-core-components (0.27.1), dash-html-components (0.11.0), dash-renderer (0.13.0), dash-table-experiments (0.6.0), and flask (0.12.2). Executions were tested locally using Docker (17.12.0-ce), and on Compute Canada's Cedar high performance cluster using Singularity (2.5.1-dist). The Docker container used for ndmg can be found on Docker hub as neurodata/m3r-release (0.0.5), which contains ndmg (0.1.0-f). The Singularity container used was pulled and dynamically created from this Docker hub endpoint. The dataset use was a subset of the HCP 1200 collection[3].

Author Contributions

GK designed and developed the tools, experiments, and figures, and wrote the majority of the manuscript. SB supported the design and development processes, and edited the manuscript and provided valuable feedback. TG provided insight and contributed to the design and development of the tools and experiments, and contributed to writing the manuscript. AE edited the manuscript and provided valuable feedback. TG and AE jointly supervised this project.

Funding

Funding for this work was provided by CFREF/HBHL (Canada First Research Excellence Fund/Healthy Brains for Healthy Lives) and the Natural Sciences and Engineering Research Council of Canada (CGSD3-519497-2018).

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgments

The authors would like to thank Pierre Rioux and Valerie Hayot-Sasson for their insight and many helpful discussions.

References

[1] X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. C. S. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos, A. Chen, B. Chen, J. Chen, X. Chen, S. J. Colcombe, W. Courtney, R. C. Craddock, A. Di Martino, H.-M. Dong, X. Fu, Q. Gong, K. J. Gorgolewski, Y. Han, Y. He, Y. He, E. Ho, A. Holmes, X.-H. Hou, J. Huckins, T. Jiang, Y. Jiang, W. Kelley, C. Kelly, M. King, S. M. LaConte, J. E. Lainhart, X. Lei, H.-J. Li, K. Li, K. Li, Q. Lin, D. Liu, J. Liu, X. Liu, Y. Liu, G. Lu, J. Lu, B. Luna, J. Luo, D. Lurie, Y. Mao, D. S. Margulies, A. R. Mayer, T. Meindl, M. E. Meyerand, W. Nan, J. A. Nielsen, D. O'Connor, D. Paulsen, V. Prabhakaran, Z. Qi, J. Qiu, C. Shao, Z. Shehzad, W. Tang, A. Villringer, H. Wang, K. Wang, D. Wei, G.-X. Wei, X.-C. Weng, X. Wu, T. Xu, N. Yang, Z. Yang, Y.-F. Zang, L. Zhang, Q. Zhang, Z. Zhang, Z. Zhang, K. Zhao, Z. Zhen, Y. Zhou, X.-T. Zhu, and M. P. Milham, "An open science resource for establishing reliability and reproducibility in functional connectomics," *Sci Data*, vol. 1, p. 140049, Dec. 2014.

[2] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, B. Liu, P. Matthews, G. Ong, J. Pell, A. Silman, A. Young, T. Sprosen, T. Peakman, and R. Collins, "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age," *PLoS Med.*, vol. 12, no. 3, p. e1001779, Mar. 2015.

[3] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. J. Behrens, E. Yacoub, K. Ugurbil, and WU-Minn HCP Consortium, "The WU-Minn human connectome project: an overview," *Neuroimage*, vol. 80, pp. 62–79, Oct. 2013.

[4] Open Science Collaboration, "PSYCHOLOGY. estimating the reproducibility of psychological science," *Science*, vol. 349, no. 6251, p. aac4716, Aug. 2015.

[5] A. Bowring, C. Maumet, and T. E. Nichols, "Exploring the impact of analysis software on task fmri results," *Human brain mapping*, vol. 40, no. 11, pp. 3362–3384, 2019.

[6] A. Eklund, T. E. Nichols, and H. Knutsson, "Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 28, pp. 7900–7905, Jul. 2016.

[7] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, vol. 533, no. 7604, pp. 452–454, May 2016.

[8] M. Miłkowski, W. M. Hensel, and M. Hohol, "Replicability or reproducibility? on the replication crisis in computational neuroscience and sharing only relevant detail," *J. Comput. Neurosci.*, Oct. 2018.

[9] K. J. Gorgolewski, T. Auer, V. D. Calhoun, R. C. Craddock, S. Das, E. P. Duff, G. Flandin, S. S. Ghosh, T. Glatard, Y. O. Halchenko, D. A. Handwerker, M. Hanke, D. Keator, X. Li, Z. Michael, C. Maumet, B. N. Nichols, T. E. Nichols, J. Pellman, J.-B. Poline, A. Rokem, G. Schaefer, V. Sochat, W. Triplett, J. A. Turner, G. Varoquaux, and R. A. Poldrack, "The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments," *Sci Data*, vol. 3, p. 160044, Jun. 2016.

[10] K. J. Gorgolewski, F. Alfaro-Almagro, T. Auer, P. Bellec, M. Capotă, M. M. Chakravarty, N. W. Churchill, A. L. Cohen, R. C. Craddock, G. A. Devenyi, and Others, "BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods," *PLoS Comput. Biol.*, vol. 13, no. 3, p. e1005209, 2017.

[11] T. Glatard, G. Kiar, T. Aumentado-Armstrong, N. Beck, P. Bellec, R. Bernard, A. Bonnet, S. T. Brown, S. Camarasu-Pop, F. Cervenansky, S. Das, R. Ferreira da Silva, G. Flandin, P. Girard, K. J. Gorgolewski, C. R. G. Guttmann, V. Hayot-Sasson, P.-O. Quirion, P. Rioux, M.-É. Rousseau, and A. C. Evans, "Boutiques: a flexible framework to integrate command-line applications in computing platforms," *Gigascience*, vol. 7, no. 5, May 2018.

[12] D. Merkel, "Docker: Lightweight linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, Mar. 2014.

[13] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLoS One*, vol. 12, no. 5, p. e0177459, May 2017.

[14] R. A. Poldrack, D. M. Barch, J. P. Mitchell, T. D. Wager, A. D. Wagner, J. T. Devlin, C. Cumba, O. Koyejo, and M. P. Milham, "Toward open sharing of task-based fMRI data: the OpenfMRI project," *Front. Neuroinform.*, vol. 7, p. 12, Jul. 2013.

[15] D. E. Rex, J. Q. Ma, and A. W. Toga, "The LONI pipeline processing environment," *Neuroimage*, vol. 19, no. 3, pp. 1033–1048, Jul. 2003.

[16] T. Sherif, P. Rioux, M.-E. Rousseau, N. Kassis, N. Beck, R. Adalat, S. Das, T. Glatard, and A. C. Evans, "CBRAIN: a web-based, distributed computing platform for collaborative neuroimaging research," *Front. Neuroinform.*, vol. 8, p. 54, May 2014.

[17] I. Dinov, K. Lozev, P. Petrosyan, Z. Liu, P. Eggert, J. Pierce, A. Zamanyan, S. Chakrapani, J. Van Horn, D. S. Parker, R. Magsipoc, K. Leung, B. Gutman, R. Woods, and A. Toga, "Neuroimaging study designs, computational analyses and data provenance using the LONI pipeline," *PLoS One*, vol. 5, no. 9, Sep. 2010.

[18] F. Chirigati, R. Rampin, D. Shasha, and J. Freire, "ReproZip: Computational reproducibility with ease," in *Proceedings of the 2016 International Conference on Management of Data*, ser. SIGMOD '16. New York, NY, USA: ACM, 2016, pp. 2085–2088.

[19] K. Hasham, K. Munir, and R. McClatchey, "Cloud infrastructure provenance collection and management to reproduce scientific workflows execution," *Future Gener. Comput. Syst.*, vol. 86, pp. 799–820, Sep. 2018.

[20] V. Sochat and B. N. Nichols, "The neuroimaging data model (NIDM) API," *Gigascience*, vol. 5, no. suppl_1, pp. 23–24, Nov. 2016.

[21] P. Missier, K. Belhajjame, and J. Cheney, "The W3C PROV family of specifications for modelling provenance metadata," in *Proceedings of the 16th International Conference on Extending Database Technology*, ser. EDBT '13. New York, NY, USA: ACM, 2013, pp. 773–776.

[22] R. W. Cox, J. Ashburner, H. Breman, K. Fissell, C. Haselgrove, C. J. Holmes, J. L. Lancaster, D. E. Rex, S. M. Smith, J. B. Woodward, and S. C. Strother, "**A (Sort of) New Image Data Format Standard: NIfTI-1**: **WE** 150," *Neuroimage*, vol. 22, p. e1440, Jun. 2004.

[23] T. Bui, "Analysis of docker security," Jan. 2015.

[24] T. Combe, A. Martin, and R. D. Pietro, "To docker or not to docker: A security perspective," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 54–62, 2016.

[25] J. Matelsky, G. Kiar, E. Johnson, C. Rivera, M. Toma, and W. Gray-Roncal, "Container-Based clinical solutions for portable and reproducible image analysis," *J. Digit. Imaging*, vol. 31, no. 3, pp. 315–320, May 2018.

[26] K. Gorgolewski, C. D. Burns, C. Madison, D. Clark, Y. O. Halchenko, M. L. Waskom, and S. S. Ghosh, "Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python," *Front. Neuroinform.*, vol. 5, p. 13, Aug. 2011.

[27] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling," in *Proceedings of the 14th Python in Science Conference*, 2015.

[28] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Future Gener. Comput. Syst.*, vol. 46, pp. 17–35, May 2015.

[29] J. Vivian, A. A. Rao, F. A. Nothaft, C. Ketchum, J. Armstrong, A. Novak, J. Pfeil, J. Narkizian, A. D. Deran, A. Musselman-Brown, H. Schmidt, P. Amstutz, B. Craft, M. Goldman, K. Rosenbloom, M. Cline, B. O'Connor, M. Hanna, C. Birger, W. J. Kent, D. A. Patterson, A. D. Joseph, J. Zhu, S. Zaranek, G. Getz, D. Haussler, and B. Paten, "Toil enables reproducible, open source, big biomedical data analyses," *Nat. Biotechnol.*, vol. 35, no. 4, pp. 314–316, Apr. 2017.

[30] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," *Neuroimage*, vol. 62, no. 2, pp. 782–790, Aug. 2012.

[31] J. D. Tournier, F. Calamante, and others, "MRtrix: diffusion tractography in crossing fiber regions," *International Journal of*, 2012.

[32] P. Bellec, S. Lavoie-Courchesne, P. Dickinson, J. P. Lerch, A. P. Zijdenbos, and A. C. Evans, "The pipeline system for octave and matlab (PSOM): a lightweight scripting framework and execution engine for scientific workflows," *Front. Neuroinform.*, vol. 6, p. 7, Apr. 2012.

[33] S. Lampa, M. Dahlö, J. Alvarsson, and O. Spjuth, "SciPipe - a workflow library for agile development of complex and dynamic bioinformatics pipelines," Oct. 2018.

[34] R. Meyer and K. Obermayer, "pypet: A python toolkit for data management of parameter explorations," *Front. Neuroinform.*, vol. 10, p. 38, Aug. 2016.

[35] D. B. Stockton and F. Santamaria, "NeuroManager: a workflow analysis based simulation management engine for computational neuroscience," *Front. Neuroinform.*, vol. 9, p. 24, Oct. 2015.

[36] S. G. Aleksin, K. Zheng, D. A. Rusakov, and L. P. Savtchenko, "ARACHNE: A neural-neuroglial network builder with remotely controlled parallel computing," *PLoS Comput. Biol.*, vol. 13, no. 3, p. e1005467, Mar. 2017.

[37] M. L. Hines and N. T. Carnevale, "NEURON: a tool for neuroscientists," *Neuroscientist*, vol. 7, no. 2, pp. 123–135, Apr. 2001.

[38] R. Reuillon, M. Leclaire, and S. Rey-Coyrehourcq, "OpenMOLE, a workflow engine specifically tailored for the distributed exploration of simulation models," *Future Gener. Comput. Syst.*, vol. 29, no. 8, pp. 1981–1990, Oct. 2013.

[39] J. Goecks, A. Nekrutenko, J. Taylor, and Galaxy Team, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biol.*, vol. 11, no. 8, p. R86, Aug. 2010.

[40] G. Kiar, E. W. Bridgeford, W. R. Gray Roncal, V. Chandrashekhar, D. Mhembere, S. Ryman, X. N. Zuo, D. S. Marguiles, R. C. Craddock, C. E. Priebe, R. Jung, V. D. Calhoun, B. Caffo, R. Burns, M. P. Milham, and J. T. Vogelstein, *A High-Throughput Pipeline Identifies Robust Connectomes But Troublesome Variability*, 2018.

[41] G. Kiar, "Clowdr: Accessible pipeline deployment and sharing," Zenodo, Mar. 2018.

[42] J. Reunanen, "Overfitting in making comparisons between variable selection methods," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1371–1382, 2003.

[43] *Numerical error propagation in the HCP structural pre-processing pipelines*, Jun. 2018.

Ch.II: Comparing Perturbation Models for Evaluating Stability of Neuroimaging Pipelines

Gregory Kiar[1], Pablo de Oliveira Castro[2], Pierre Rioux[1], Eric Petit[3], Shawn T. Brown[1], Alan C. Evans[1], Tristan Glatard[4]

[1] *Montréal Neurological Institute, McGill University, Montréal, QC, Canada*;

[2] *University of Versailles, Versailles, France*;

[3] *Exascale Computing Lab, Intel, Paris, France*;

[4] *Department of Computer Science and Software Engineering, Concordia University, Montréal, QC, Canada*.

Published in:

The International Journal of High Performance Computing Applications

https://doi.org/10.1177/1094342020926237

Abstract

With an increase in awareness regarding a troubling lack of reproducibility in analytical software tools, the degree of validity in scientific derivatives and their downstream results has become unclear. The nature of reproducibility issues may vary across domains, tools, datasets, and computational infrastructures, but numerical instabilities are thought to be a core contributor. In neuroimaging, unexpected deviations have been observed when varying operating systems, software implementations, or adding negligible quantities of noise. In the field of numerical analysis these issues have recently been explored through Monte Carlo Arithmetic, a method involving the instrumentation of floating point operations with probabilistic noise injections at a target precision. Exploring multiple simulations in this context allows the characterization of the result space for a given tool or operation. In this paper we compare various perturbation models to introduce instabilities within a typical neuroimaging pipeline, including i) targeted noise, ii) Monte Carlo Arithmetic, and iii) operating system variation, to identify the significance and quality of their impact on the resulting derivatives. We demonstrate that even low-order models in neuroimaging such as the structural connectome estimation pipeline evaluated here are sensitive to numerical instabilities, suggesting that stability is a relevant axis upon which tools are compared, alongside more traditional criteria such as biological feasibility, computational efficiency, or, when possible, accuracy. Heterogeneity was observed across participants which clearly illustrates a strong interaction between the tool and dataset being processed, requiring that the stability of a given tool be evaluated with respect to a given cohort. We identify use cases for each perturbation method tested, including quality assurance, pipeline error detection, and local sensitivity analysis, and make recommendations for the evaluation of stability in a practical and analytically-focused setting. Identifying how these relationships and recommendations scale to higher-order computational tools, distinct datasets, and their implication on biological feasibility remain exciting avenues for future work.

Ch.II - 1 Introduction

A lack of computational reproducibility[1] has become increasingly apparent in the last several years, calling into question the validity of scientific findings affected by published tools. Reproducibility issues may have numerous sources of error, including undocumented system or parametrization differences and the underlying numerical stability of algorithms and implementations employed. While containerization can mitigate the extent of machine-introduced variability, understanding the effect that these sources of error have on the encapsulated numerical algorithms remains difficult to explore. In simple cases where algorithms are differentiable or invertible, it is possible to obtain closed-form solutions for their stability. However, as software pipelines grow, containing multiple complex steps, using non-linear optimizations and non-differentiable functions, the stability of these algorithms must be explored empirically.

As neuroscience has evolved into an increasingly computational field, it has suffered from the same questions of numerical reproducibility as many other domains[2]. In particular, neuroimaging often attempts to fit alignments, segmentations, or models of the brain using few samples with variable signal to noise properties. The nature of these operations leaves them potentially vulnerable to instability when presented with minor perturbations in either the data themselves or their processing implementations. The independent evaluation of atomic pipeline components may be feasible in some cases, as was done by Skare et al. in[3]. Here, the authors computed the theoretical conditioning of various tensor models used in diffusion modeling, and compared these values to the observed variances in tensor features when fit on simulated data. While approaches like the above provide valuable insights to algorithms and their implementations independently, the impact of these stepwise instabilities within composite pipelines remains unknown. Even if one were able to evaluate each step within a pipeline, identifying the impact these instabilities may have on a result when composed together, both structurally and analytically, remains practically difficult to evaluate. Additionally, as datasets grow in size, the adoption of High Performance Computing environments becomes a necessity. These environments are highly heterogeneous in terms of hardware, operating systems, and parallelization schemes, and this heterogeneity has been shown to compound with these instabilities and impact results[4].

Various forms of instability have been observed in structural and functional magnetic resonance (MR) imaging, including across operating system versions[4], minor noise injections[5], as well as dataset or implementation of theoretically equivalent algorithms[6,7]. These approaches may have practical applications in decision making, such as deciding which tool/implementation should be used for an experiment.

However, they are relatively far removed from the underlying numerical instabilities being observed. Recent advances in numerical analysis allow for the replacement of floating point operations with Monte Carlo Arithmetic simulations[8] which inject a random zero-bias rounding error to operations for a target floating-point precision[8,9]. This method can be used for evaluating the numerical stability of tools by wrapping existing analyses[9] and providing a foothold for scientists wishing to explore the space of their pipeline's compound instabilities[10].

In this paper we explore the effect of various perturbations on a typical diffusion MR image processing pipeline through the use of i) targeted noise injections, ii) Monte Carlo Arithmetic, and iii) varying operating systems to identify the quality and severity of their impact on derived data. This evaluation will inform future work exploring the stability of these pipelines and downstream analyses dependent upon them. The processing pipeline selected for exploration is Dipy[11], a popular tool that generates structural connectivity maps (connectomes) for each participant. The pipeline accepts de-noised and co-registered images as inputs, and then performs two key processing steps: tensor fitting and tractography. We demonstrate the relative impact that each of the tested perturbation methods has on the resulting connectomes and explore the nature of where these differences emerge.

Ch.II - 2 Methods

All processing described below was run using servers provided by Compute Canada. Software pipelines were encapsulated and run using Singularity[12] version 2.6.1. Tasks were submitted, monitored, and provenance captured using Clowdr[13] version 0.1.2-1. All code for performing the experiments and creating associated figures are available on GitHub at https://github.com/gkiar/stability and https://github.com/gkiar/stability-mca, respectively.

Ch.II - 2.1 Dataset and pre-processing

The dataset used for processing is a 10-session subset of the Nathan Kline Institute Rockland Sample dataset (NKI-RS)[14]. This dataset contains high fidelity structural, functional, and diffusion MR data and is openly available for research consumption. The 10 sessions used were chosen by randomly selecting 10 participants and selecting their alphabetically-first session of data. This data was preprocessed prior to the modelling evaluated here using a standard de-noising and image alignment pipeline[15] built upon the FSL toolbox[16]. The steps in this pipeline include eddy current correction, brain extraction, tissue segmentation, and image registration. The boundary between white and gray matter was obtained by computing the

difference between a dilated version of the white matter mask and the original. Data volumes at this stage of processing are four-dimensional and variable in spatial extent (first three dimensions) with a fixed number of diffusion directions (fourth dimension), totalling approximately $100^3 \times 137$ voxels in each case.

Ch.II - 2.2 Modeling

After pre-processing the raw diffusion data using FSL, structural connectomes were generated for an 83-region cortical and sub-cortical parcellation[17] using Dipy[11]. A six-component tensor model was fit to the diffusion data residing within white matter. Seeds were generated in a $2 \times 2 \times 2$ arrangement for each voxel within the boundary mask, resulting in 8 seeds per boundary voxel. Deterministic tracing was then performed using a half-voxel step size, and streamlines shorter than 3-points in length were discarded as spurious. Once streamlines were generated they were traced through the parcellation. Edges were added to the graph corresponding to the end-points of each fiber, and were weighted by the streamline count. This pipeline was implemented in Python, including a few components in Cython, and relies on the Numpy library for a large proportion of operations. Each resulting network is a square connectivity matrix of 83×83 edges, as shown in Fig. Ch.II - 1. This pipeline was chosen as it is both common and simple relative to many alternatives.

Ch.II - 2.3 Stability Evaluation

Targeted and Monte Carlo perturbation modes were tested 100x per image. Noise was represented by percent deviation of the Frobenius norm of a resulting connectome from the corresponding reference (no noise injection). A deviation of 50% indicates that the norm of the difference between the noisy and reference networks is 50% the size of the norm of the reference graph. This is formalized below in Eq. (1):

$$\%Dev(A,B) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}-b_{ij}|^2} \Big/ \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}, \tag{1}$$

where A is the reference graph, B is the perturbed graph, and $\square_{ij}$ is an element therein at row i and column j.

The perturbation methods evaluated, presented below, are summarized in Table Ch.II - 1.

Ch.II - 2.4 Subject-Level Variation

Comparison between subjects will be used as a reference error. If the differences observed by other methods are similar in magnitude to the subject-level difference, then the validity of the processed

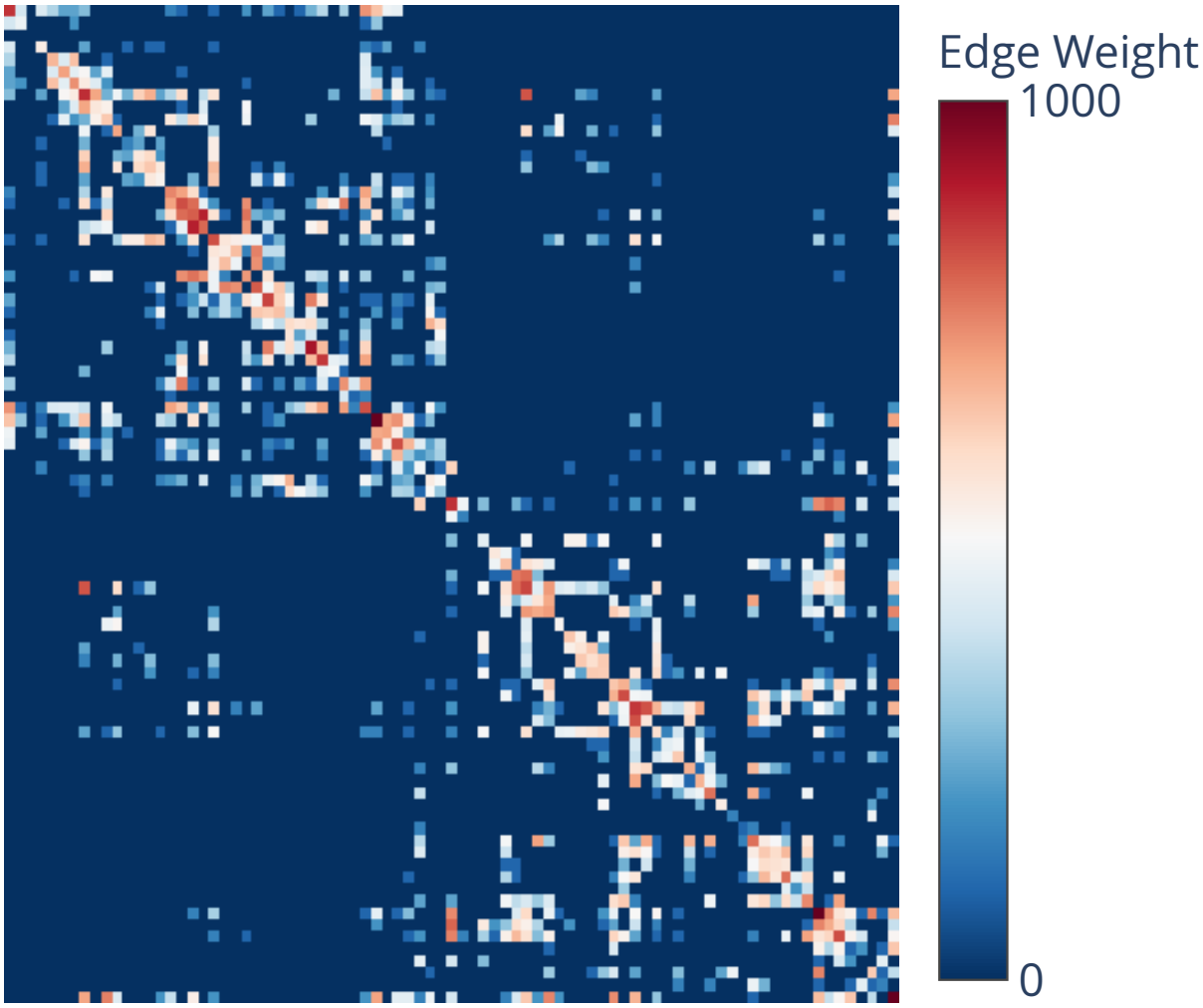

Figure Ch.II - 1. Example connectome. Each row and column corresponds to a region within the brain, and the intersection a connection between them. If no connection is found between regions, the edge strength is zero. If a streamline is found to connect two regions, the weight is incremented by 1. The resulting weights are the sum of all observed connections for every streamline traced within a brain image.

networks for use in downstream phenotypic analysis becomes questionable as subjects cannot be reliably distinguished from one another. This error is computed as the pairwise distance between all 10 subjects included in this cohort.

Ch.II - 2.5 Targeted Noise

The goal of targeted noise was to inject data perturbations sufficiently small that the resulting images would be indistinguishable from the original. This is meant to test the lower-bound of noise sensitivity for processing pipelines. The type of targeted noise used here will be referred to as 1-voxel noise and is similar to the method employed in[5]. In our case, the intensity of a single voxel in the defined range will be scaled based on a scaling factor. The voxels modified in this case were randomly generated within the mask of brain regions being modeled by the pipeline.

The two modes of 1-voxel noise injection tested here were: a) a single voxel per entire image of size (X,Y,Z,D) (approximately $100^3 \times 137$ for all images), or b) a single voxel per 3D volume of size (X,Y,Z) (approximately 100^3 for all images), and are referred to as "single" and "independent" modes, respectively. While the number of perturbed voxels in the independent case is approximately 100 times larger, the intensity of magnification was consistent as in both cases the original voxel intensities were doubled.

Ch.II - 2.6 Monte Carlo Arithmetic

Verificarlo[10] is an extension of the LLVM compiler which automatically instruments floating point operations at build-time for software written in C, C++, and Fortran. Once compiled with Verificarlo, the Monte Carlo emulation method and target precision can be set as environment variables. For all simulations a rounding error on the least significant floating point bit in the mantissa (bit 53) was introduced. The simulations were computed using the custom QUAD backend which is optimized to reduce computation time over the traditional mcalib MPFR backend leveraging GNU's multiple precision library[9]. Noise through Verificarlo can be injected as "Precision Bounded", simulating floating point cancellations, "Random Rounding", simulating only rounding errors on computation, and "MCA", which includes both of these modes. A particularity of the Random Rounding mode is that it only injects rounding noise on inexact floating-point operations (i.e. operations that have a rounding error in IEEE-754 at the target precision). Therefore, RR mode preserves the original exact operations, it is a more conservative noise simulation. We used both the RR and MCA modes of simulation.

Verificarlo was used to instrument tools in two modes we will refer to as "Python" and "Full Stack". In the Python instrumentation, the core Python libraries were recompiled with Verificarlo as well as any subsequently installed Cython libraries. In the Full Stack instrumentation, BLAS and LAPACK were also recompiled, meaning that Numpy, a dominant Python library for linear algebra, was also instrumented. The Full Stack implementation did not run successfully using the MCA mode. We suspect that some libraries

require exact floating-point operations or are sensitive to cancellation errors, so only the Random Rounding (RR) mode was able to be evaluated for the Full Stack. These instrumentations took approximately 10 hours for the authors to refine, and the images are available on DockerHub at gkiar/fuzzy-python.

Ch.II - 2.7 Operating System Variation

Operating system noise was evaluated across Alpine Linux 3.7.1 and Ubuntu 16.04. Alpine is a lightweight distribution which comes with minimal packages or libraries, and Ubuntu is a popular Linux distribution with a large user and development community. Alpine was chosen as its lightweight nature makes it an efficient choice for the packaging and distribution of libraries in containers for scientific computing, reducing the overhead of shipping code towards data sources. Ubuntu was chosen due to is high adoption and community support by major libraries. While Alpine comes with a minimal set of libraries, a core difference between these systems as noted by DistroWatch (https://distrowatch.com/) is their dependence on a different version of the Linux kernel. While numerical differences between operating systems are likely the result of compilers[18] and installed libraries, the purpose of testing across operating systems explicitly rather than combinations of specific tools is to re-create a real-world setting in which typical scientific users observe numerical differences across equivalent high-level pipelines.

Ubuntu was used as the base operating system for all simulations other than this comparison. The variability observed across operating systems was aggregated across participants and included as a reference margin of error.

Table Ch.II - 1. Description of perturbation modes

Permutation	Description
X-Subject	Pairwise comparison of sessions based on **Subject ID**.
1-voxel	Intensity value doubled for either **Single** (one voxel in entire 4D volume) or **Independent** (one voxel per 3D sub-volume) voxels.
MCA	Simulation of all floating point operations in **Python** (Python and Cython-compiled libraries).
RR	Simulation of all rounding operations in **Python** or the **Full Stack** (BLAS, and LAPACK, Python and Cython-compiled libraries).
X-OS	One of **Ubuntu 16.04** or **Alpine 3.7.1**.

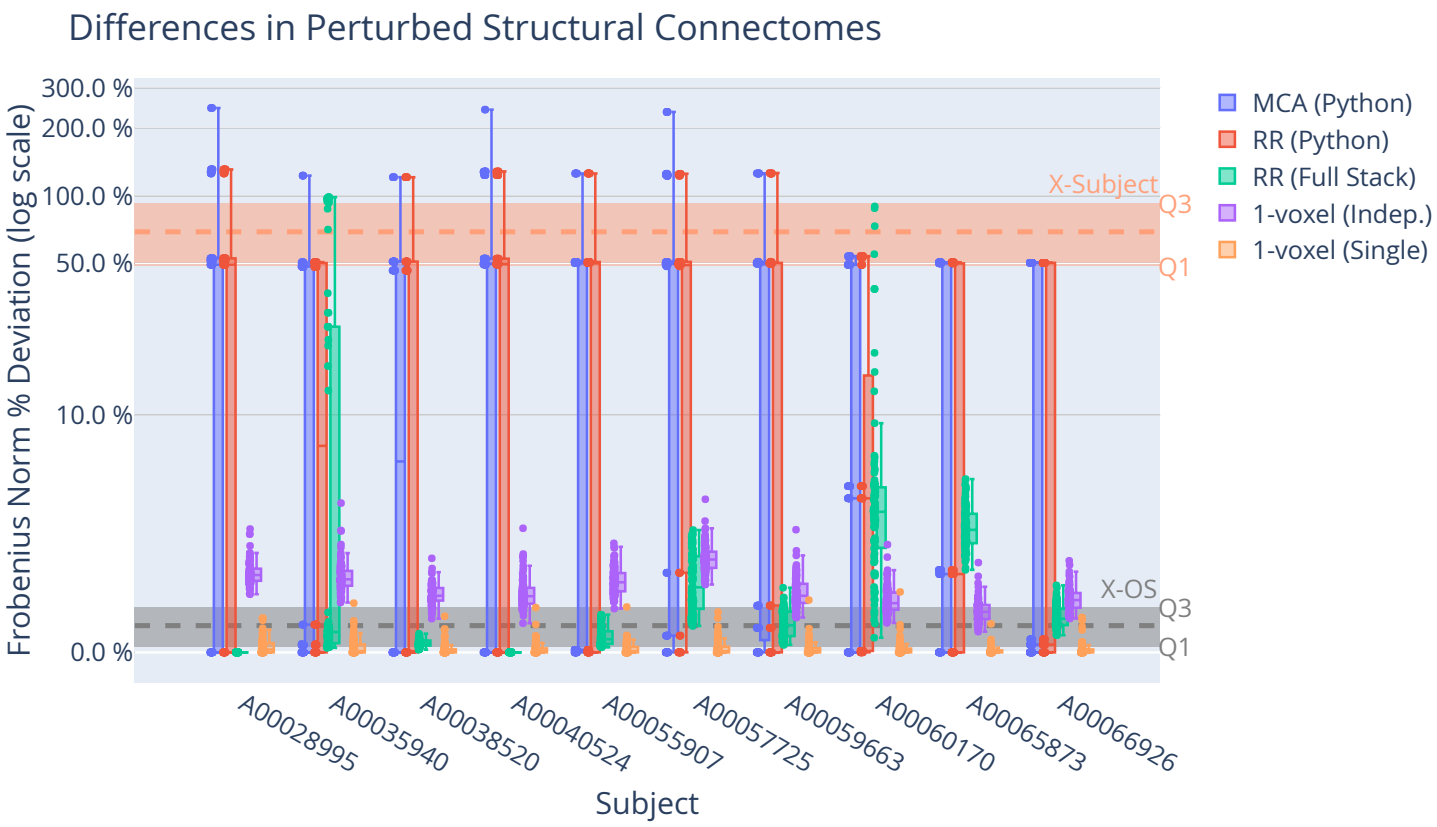

Figure Ch.II - 2. Comparison of perturbation modes. As evaluated by the percent deviation from reference in the Frobenius Norm of a resulting connectome, each of the 10 processed subjects were re-processed 100 times for each perturbation method. We see that the MCA and RR (Python) methods resulted in distinct modes for the outputs in all cases reaching extreme deviations equivalent to cross-subject variation. The RR (Full Stack) method shows high variability across subjects, and only reaching cross-subject variation in the case of 2 subjects. The 1-voxel methods result in considerably less deviation from reference, and are more consistent across subjects than the RR (Full Stack) method.

Ch.II - 2.8 Aggregation of Simulated Graphs

To structurally evaluate each simulation setting, connectomes were aggregated within setting and subject combinations. Several aggregation methods were explored to preserve various sensitivity and stability properties across the aggregated graphs. In each case, the operations are performed edge-wise, so the aggregated graph is not guaranteed to be single graph in the set of perturbed graphs. The aggregation operations are the edge-wise mean and the 0^{th} (min), 10^{th}, 50^{th} (median), 90^{th}, and 100^{th} (max) %-iles. The mean aggregate will include a non-zero weight for every edge which appears in at least one simulation, and the 0^{th} and 100^{th} %-iles will include the lowest and highest observed weight for every

edge, respectively. The 90^{th}, 50^{th}, and 10^{th} %-iles increasingly aggressively filter edges based on their prominence across simulations. The combination of percentile aggregates also enable isolation of the most spurious edges, such as by taking the difference of maximum and minimum aggregates. A volatile aggregate was created to this effect which consists of edges which are found in the maximum aggregate but not the minimum aggregate. Note that in this case, the weight for these edges is not implied and can be defined as an alternative function of the graph collection, such as mean, but as the weight does not appear when comparing binary edges, no recommendation for this weighting is made here.

Ch.II - 3 Results

All perturbation modes were applied to either the input data or post-processing pipeline described in the Section Ch.II - 2.2, and were evaluated according to Eq. (1).

Ch.II - 3.1 Perturbation Induced Differences

Fig. Ch.II - 2 shows the percentage deviation for each simulation mode on 10 subjects. Introduced perturbations show highly-variable changes in resulting connectomes across both the perturbation model and subject, ranging from no change to deviations equivalent to difference typically observed across subjects. For the 10 subjects tested, we see that the Python-instrumented MCA and RR pipelines resulted in the largest deviation from the reference connectome. In these cases we also see that the results are modal, where each subject has discrete states that may be settled in, some of which result in deviations comparable to subject-level noise. This modality is likely due to minor differences introduced at crucial branch-points which then cascaded throughout the pipeline. This hypothesis is supported by observing that the Full Stack implementation with RR perturbations shows a continuous distribution of differences that are highly variable in intensity, ranging from no deviation to subject-level in some cases for some subjects, which are explored in Section Ch.II - 3.2.

The 1-voxel independent mode unsurprisingly produces larger changes than the 1-voxel single mode. These changes are larger than or comparable to operating system variability, respectively, resulting in small deviations from the reference, and are relatively minor in comparison to the extremes observed with Monte Carlo Arithmetic. Operating system deviations are very low or even zero in some cases. In all perturbation settings we can see that there is large variability both across simulations on the same data and across subjects.

Ch.II - 3.2 Progression of Deviations in a Continuous Setting

In the case of subject A00035940, the Full Stack RR perturbations led to a continuous distribution of outputs, ranging in difference from none to subject-level from the reference. Fig. Ch.II - 3 explores the progression of these deviations by visualizing the difference-connectome for samples along various points of this distribution. In the center we show the reference connectome, and surrounding it the difference graph for a simulated sample with labelled $\%Dev$ from this reference. In this case, we can see a progression of structurally consistent deviations. In particular, edges corresponding to regions in the left hemisphere become increasingly distorted (bottom-right portion of the connectome), whereas the within-hemisphere connectivity for the right hemisphere (top-left portion) remains largely intact in all cases except the extreme difference case. We notice in all cases that the connectivity between regions is decreasing until the edges disappear entirely. While this behaviour is not consistent across all subjects, this observation suggests a peculiarity in the quality of data in this region for the subject in question. This could be due to artifacts caused by motion or other factors, ultimately reducing the stability of modeling connectivity in this region.

Ch.II - 3.3 Structural Properties of Introduced Perturbation

While the case investigated above notably showed a significant degradation of regional signal quality for Full Stack RR noise in a single subject, Fig. Ch.II - 4 explores the relative change in connectivity from the reference for each perturbation mode and subject. Edges in the presented graphs are weighted by their standard deviation across all simulations for that participant, and coloured as positive or negative deviations based on whether the mean weight for all simulations was greater or lower than the reference weight, respectively. All edges with a standard deviation of 0 across all simulations were greened out for clarity.

For the Python instrumented MCA and RR implementations, edge weight was generally inflated non-specifically for existing edges in the reference connectome for all subjects. The Full Stack RR implementation shows significant variability across subjects, where the number of affected edges ranges from none to all. In each case where there exists some deviation, intensities appear to be spatially linked, suggesting the differences may be due to variable quality in the underlying data. In this case, Monte Carlo Arithmetic may have served to shed light on poor signal-to-noise properties present within regions of the images being modelled.

For 1-voxel noise, the differences introduced across independent injections impacted a larger portion

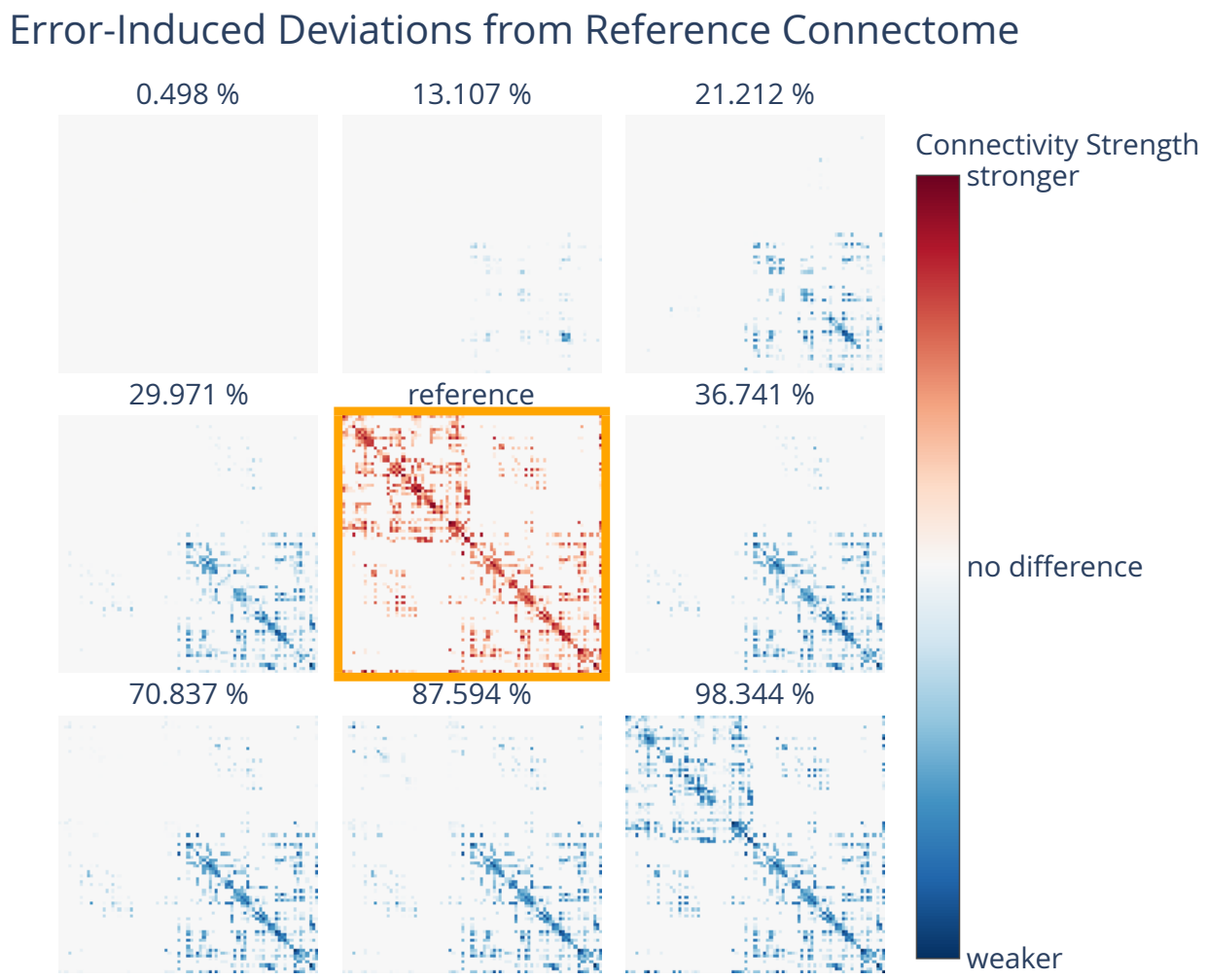

Figure Ch.II - 3. Structure of Deviations. Shown in increasing deviation from left–right and top–bottom, with the reference in the centre, are the difference connectomes observed for the RR (Full Stack) perturbations of subject A00035940. In this case, the left hemisphere (bottom-right portion of the graph) begins to degrade quickly, eventually reaching an almost complete loss in signal.

of edges than single injections, unsurprisingly. By design (i.e. injection at random locations for each simulation), the deviations appear non-specifically spatially distributed. However, 1-voxel noise could be modified to spatially constrain the location for noise injection regionally, allowing the evaluation of modelling for particular sub-structures within the images.

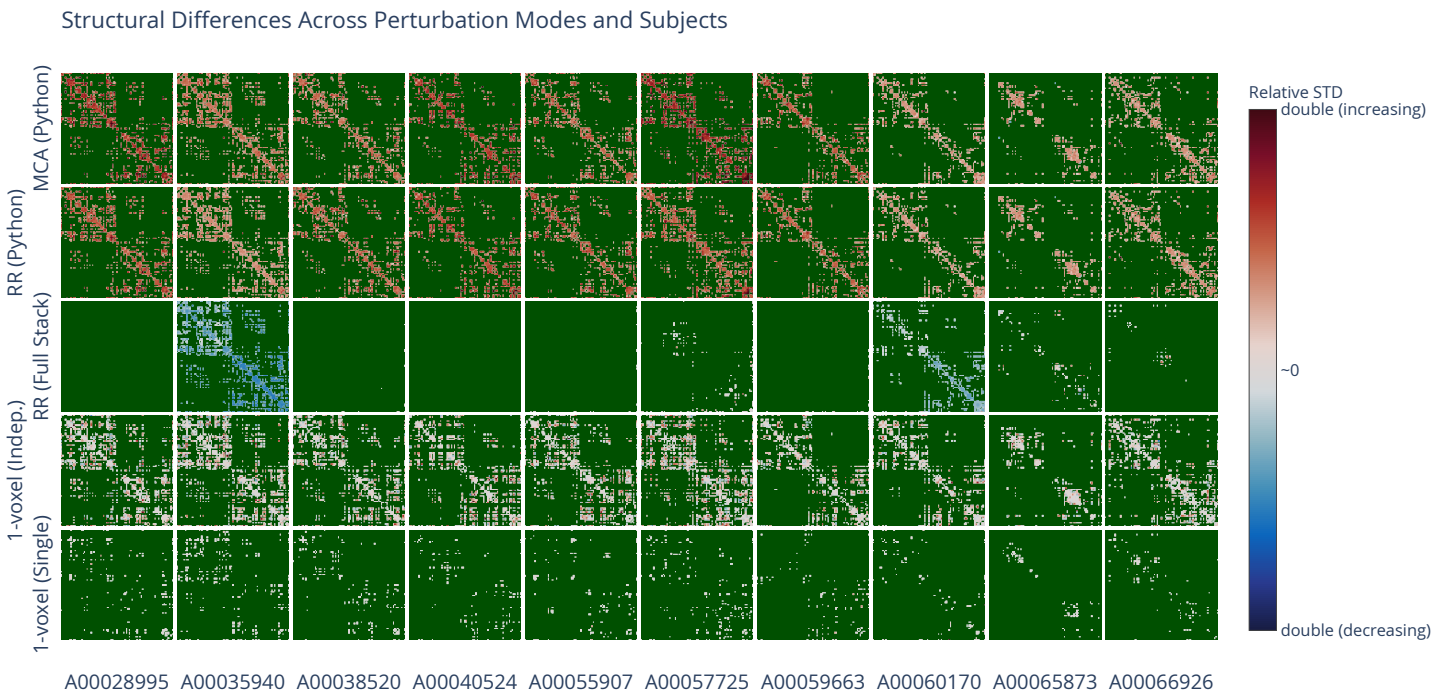

Figure Ch.II - 4. Perturbation introduced structural differences. The variance of each edge is shown relative to the reference edge strength, and coloured either red or blue based on the mean perturbed weight was higher or lower than that of the reference, respectively. Edges which experienced no variation were coloured as green to be distinct from all edges which experience any variation.

Ch.II - 3.4 Aggregation Across Simulations

For each simulation method there existed a graph nearly identical to the reference, but the variability introduced by these simulations were highly variable both in terms of the method of perturbation used and the dataset being processed. The aggregation of the simulated graphs into a consensus graph allows features of this variation to be encoded implicitly in connectomes which may be used for downstream analyses. Fig. Ch.II - 5 shows the relative percentage of added and missing edges for each setting across all subjects using a variety of such aggregation methods.

By aggregating the simulated connectomes in a variety of methods, the resulting edges would be a product of applying some filter to the set of observed edges, and succinctly represented in a single graph. While minor deviations in one edge may reduce the strength of connectivity between two strongly linked regions, the addition of a connection between two regions which were previously unconnected may be significant in one aggregation method but ignored in another. In the case of the above example, despite the strength of connectivity remaining low between the newly connected nodes many graph theoretic

measures rely on binarized graphs and may be considerably affected, such as the degree.

We notice that the 1-voxel independent (i.e. single voxel per 3D volume) method shows the most variability across each aggregation method. Where all of the MCA-derived methods perturb the pipeline non-locally, both epsilon-level methods add local noise at arbitrary locations. This distinction seems to manifest in more widely added or knocked-out edges for the 1-voxel cases, as the location of noise may have considerable impact on a multitude of nearby fibers, where MCA methods have a zero-bias noise globally, meaning all deviations from the reference are spurious and due to numerical error rather than the introduction of a systemic change that sheds light on an underlying cascading instability.

Unsurprisingly, the only aggregation method which shows considerable amount of both new and missing edges is the volatile technique, which takes edges that exist in the binary difference of 100^{th} and 0^{th} percentile graphs, eliminating all extremely stable edges from the graph (i.e. those which exist for the reference and all simulations). While the mean sparsity of the reference graphs is 0.30, meaning 30% of possible connections have non-zero weight on average, the sparsity of the volatile aggregates ranges from 0.005 to 0.130, or, the aggregates contain between 2.5% and 43.0% the number of edges as the reference graphs.

Ch.II - 3.5 Comparison of Simulation Performance

While the application of each perturbation model tested sheds light on different properties of pipeline stability, the resource consumption of these methods has significant bearing when processing data in the context of a real experiment often consisting of dozens to hundreds of subjects worth of data. In this experiment, a single unperturbed pipeline execution took approximately 20 minutes using 1 core and 6 GB of RAM. Fig. Ch.II - 6 shows the relative Time-on-CPU for a single simulation of each method tested, relative to the reference task with no instrumentation. For Monte Carlo Arithmetic instrumented executions, we expect to see a considerable increase in computation time as additional overhead is added to each floating point operation. In the case of 1-voxel noise it is expected to see a minor increase in computation time as the perturbed data volumes were generated at runtime, reducing the data redundancy on disk.

The Python MCA and RR modes show a slight increase in computation time to the reference task, whereas the Full Stack version approaches a nearly $7\times$ slowdown, on average. This discrepancy further supports the hypothesis stated above that floating point logic implemented directly in Python, without the use of Numpy or external libraries, account for a minor portion of the total floating point operations.

As Verificarlo has been shown to increase the runtime of floating point operations by approximately 100×, this result suggests that the pipeline evaluated here is largely I/O limited. In the case of 1-voxel perturbations, we see a slowdown approximately equivalent to that of the Python instrumentation, not exceeding a 2× increase. Across all executions approximately 2000 CPU hours were consumed. While this is a small workload in the context of HPC, the required resources quickly reach the order of CPU years after extrapolating to the entire NKI-RS dataset or others in neuroimaging.

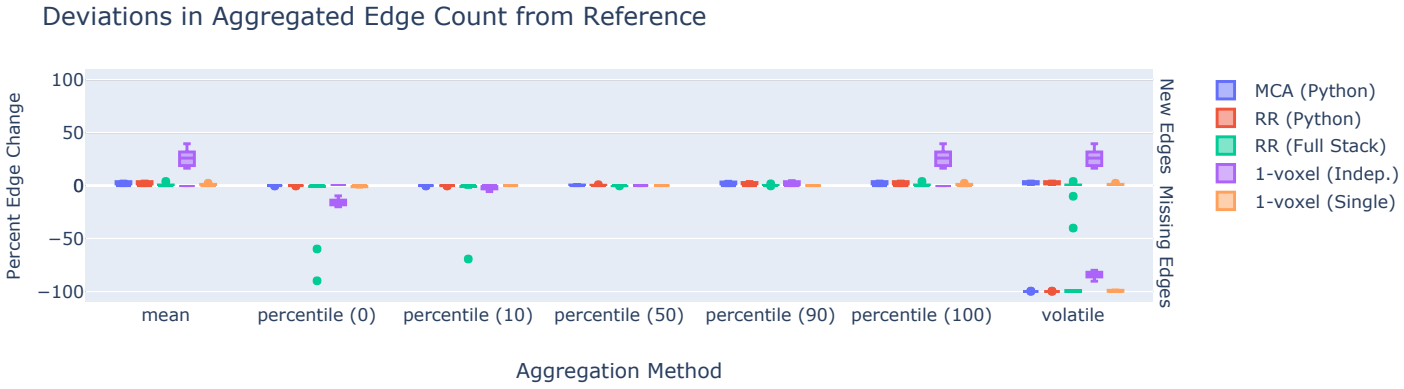

Figure Ch.II - 5. Gain and loss of edges in aggregation of simulations. The relative gain and loss of edges is shown for each aggregation method and perturbation method in terms of binary edge count. The volatile aggregation is the difference between percentile (100) and percentile (0) aggregates, and is contains all edges which do not appear in every graph. The volatile set of edges for each of MCA (Python), RR(Python), RR (Full Stack), 1-voxel (independent), and 1-voxel (single) contain 2.5%, 2.5%, 18.5%, 43.0%, and 1.7% of the number of edges found in the reference, respectively. In the worst case, 1-voxel (independent), this means that the existence of nearly half the edges in the graph fail to have consensus across the simulations.

Ch.II - 4 Discussion

We have demonstrated through the application of multiple perturbation methods how noise can be effectively injected into neuroimaging pipelines enabling the exploration and evaluation of the stability of resulting derivatives. These methods operate by either perturbing the datasets or tools used in processing, resulting in a range of structurally distinct noise profiles and distributions which may each provide value when exploring the stability of analyses. While 1-voxel noise is injected directly into the datasets prior to analysis, MCA and RR methods iteratively add significantly smaller amounts of noise to each operation

performed.

In the case of partial (Python) instrumentation with MCA and RR, distinct and considerably distinct modes emerged in all tested subjects. We hypothesize that software branching likely played a role leading to this unexpected result. As the majority of numerical analysis in Python is traditionally performed using the Numpy library, and therefore BLAS and LAPACK, it is possible that the error introduced by Python was allowed to cascade throughout the pipeline without correction, until the next Python branch point occured and this repeated, eventually growing to the often subject-level differences observed. These modes would then be the result of a small number of instrumented numerically-sensitive operations, leading to a bounded set of possible outcomes of an otherwise deterministic process. It is possible that these distinct modes could serve as upper-bounds for the deviation due to instabilities within a pipeline, and is an area for further exploration. Future work will also more closely instrument libraries with functionality that will enable the identification of crucial branch points, as this functionality is already present within Verificarlo. The identified crucial branch points could be leveraged for the re-engineering of pipelines with more stable behaviour, and potentially shed light on new "best practices".

An exciting application of MCA and RR (Python) analyses in cases where pipeline modification is not feasible is the generation of synthetic datasets. Using each mode or an aggregated collection of modes as samples in the MCA-boosted dataset could potentially increase the statistical power of analyses for datasets which may suffer from small samples, or be used to increase the robustness of derivatives by bagging the results using an appropriate averaging technique for the simulated derivatives.

While the Python instrumentation with MCA and RR resulted in derivative modes, the Full Stack instrumentation with RR produced a continuous distribution of derivatives which were often less distinct from the reference results. Extending the hypothesis posited above, this continuous set of results may be due to a law of large numbers effect emerging when performing a considerable number of small perturbations, leading to a normalized error distribution and effectively a self-correction of deviations. Future work will test this hypothesis and consider the relationship between the fraction of instrumented floating point operations and modality, as well as through the incremental profiling and evaluation of tools for the comparison of intermediate derivatives and their deviation from a reference execution. These experiments have potential to provide more insight into the origin of instabilities in scientific pipelines and identify rich optimization targets.

As the significance of RR (Full Stack) perturbation was highly variable across participants, this technique could also be used for automated quality control, flagging high-variance subjects for further

inspection or exclusion from analyses. From the top level, inspecting the regional degradation of signal across these perturbations as shown in Fig. Ch.II - 3, researchers could lead a targeted interrogation of their raw datasets to identify underlying causes of signal loss. Conversely, investigating which low-level BLAS operations contribute to the observed instabilities will allow researchers to clarify the link between ill-conditioning and so-called "bad data" directly within their pipelines. Upon characterizing this relationship it would be valuable to identify the point (if any) at which targeted N-voxel perturbations become equivalent to MCA-induced variability, bridging the Uncertainty Quantification and Numerical Analysis approaches.

The differences observed when performing 1-voxel perturbations were often comparable in magnitude to the variation introduced across Operating Systems. As OS noise is not controlled and may differ greatly

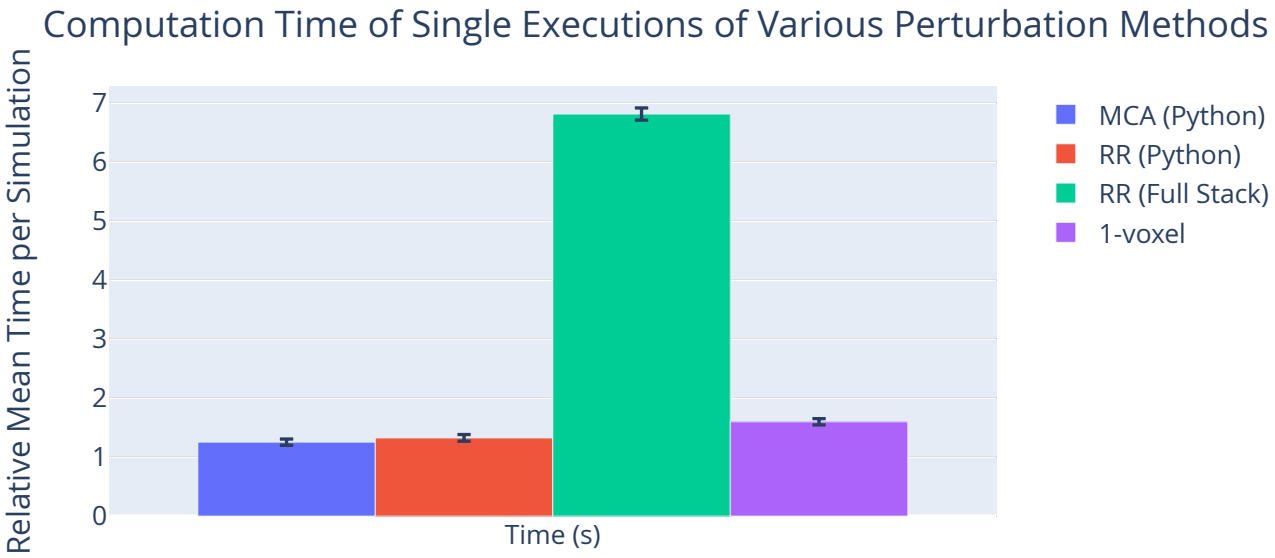

Figure Ch.II - 6. Computation time for each perturbation method. Shown in relative time to the reference execution, plotted is the average execution time for the perturbation methods. MCA and RR (Python) have a small increase in computation time per run, as few floating point operations were instrumented in these settings. The RR (Full Stack) method has nearly a 7× slowdown. In this case, all floating point operations were instrumented, but the slowdown of less than the estimated 100× would suggest that the bulk of computation time is not spent on floating point arithmetic. The 1-voxel implementations had a minor slowdown due to the regeneration of data prior to pipeline execution. In every case, the real-world slowdown is $S\times$ larger, where S is the number of simulations, in this case 100.

among distributions, package updates, etc., it is likely an insufficiently descriptive evaluation method, and should be used as a reference alongside others. The level of control made available through 1-voxel perturbations in terms of both locality and strength of noise makes it a flexible option that could potentially be used to target known areas of key importance for subsequent analyses. Due to the fact that these perturbations introduce a minor change to input images, this method could also be used for estimating global pipeline stability in a classical sense (i.e. conditioning).

While each of the perturbation modes showed distinct differences with respect to the magnitude and continuity of their induced deviations, Fig. Ch.II - 4 illustrates that the structure of these deviations was also highly variable across both perturbation method and data. This suggests different applications and use cases for each perturbation method. While MCA and RR Python implementations impact connectomes globally, these could be applied to generate synthetic datasets. Full Stack RR is highly variable with respect to dataset, suggesting possible applications in quality control, granted further work is performed to more fully understand the effect observed between this and the Python-only case. Both 1-voxel methods add noise locally, and can test the sensitivity of specific pipeline components or regions of interest to variation. Other methods, such as automatic differentiation, could also be explored as possible avenues leading towards an understanding of the end-to-end conditioning of pipelines.

In addition to generating unstable derivatives which could be looked at or analyzed independently, this type of perturbation analyses enables the aggregation of derivatives. As is summarized in Fig. Ch.II - 5, the method by which graphs or edges are aggregated can drastically change the construction of resulting graphs. While the mean and max (i.e. 100^{th} percentile) methods both retain all edges that have appeared in even a single graph, the minimum (0^{th} percentile) and other low-percentile aggregations require a stricter consensus of edges for inclusion in the final graph. A benefit of performing multiple aggregations is the composition of graphs with complex edge composition, such as the most volatile edges, as is shown in the final column of Fig. Ch.II - 5. While the binary edge count in the composite graphs varies in each of these methods, it is unclear how derived graph statistics will be affected, and that remains an exciting question for further exploration.

From a resource perspective, each of the perturbation methods evaluated requires multiple iterations to get a sense of the pipeline stability or build aggregates, here taken as 100 iterations. Though the MCA-based methods have the obvious disadvantage of extra computational overhead within each execution cycle of the pipeline, the noise-injection methods do not increase the computation time for a single pipeline execution itself but in this case added computational burden for the generation of synthetic data

dynamically, reducing the redundancy of stored images on disk. While Verificarlo has been demonstrated to account for an approximately 100× slowdown in floating point operations[10], the largest slowdown observed in this pipeline is approximately a factor of 7, as shown in Fig. Ch.II - 6. This suggests that the bulk of time on CPU for this pipeline is not spent on floating point operations, but perhaps other operations such as looping, data access, or manipulation of information belonging to other data types. While this slowdown is observed for the the Full Stack implementation, the Python-only implementation is negligibly slower than the reference execution, suggesting that even fewer of the floating point logic is directly written in Python. The slowdown in the 1-voxel setting is of a similar scale to that of the Python-only implementation, with the slowdown likely caused by the addition of 2 read and 1 write operations to the pipeline's execution (reading of simulation parameters and original image, application of simulation, and subsequent writing of perturbed image to temporary storage). Note that the figures shown in Fig. Ch.II - 6 are for a single simulation, and real relative CPU time in each case would be 100× larger for the experimental application of these methods.

The work presented here demonstrates that even low order computational models such as a 6-component tensor used in diffusion modelling are susceptible to noise. This suggests that stability is a relevant axis upon which tools should be compared, developed, or improved, alongside more commonly considered axes such as accuracy/biological feasibility or performance. The heterogeneity observed across participants clearly illustrates that stability is a property of not just the data or tools independently, but their interaction. Characterization of stability should therefore be evaluated for specific analyses and performed on a representative set of subjects for consideration in subsequent statistical testing. Additionally, identifying how this relationship scales to higher-order models is an exciting next step which will be explored. Finally, the joint application of perturbation methods with more complex post-processing bagging or signal normalization techniques may lead to the development of more numerically stable analyses while maintaining sensitivity that would be lost in traditional approaches such as smoothing.

Ch.II - 5 Conclusion

All pipeline perturbation methods showed unique non-zero output noise patterns in low-order diffusion modeling, demonstrating their viability for exploring numerical stability of pipelines in neuroimaging. MCA and RR (Python) instrumented pipelines resulted in a wide range of variability, sometimes equivalent to subject-level differences, and are recommended as possible methods to estimate the lower-bound of

stability of analyses, generation of synthetic datasets, and possible identification of Python-introduced critical branch points. RR (Full Stack) perturbations resulted in continuously distributed connectomes that were highly variable across datasets, ranging from negligible deviations to complete regional signal degradation. We provisionally recommend the use of RR (Full Stack) noise for automated quality control and identifying global pipeline stability. While 1-voxel methods result in considerably smaller maximum deviations than the MCA-based methods, they are far more flexible and enable evaluating the sensitivity of pipelines to minor local data perturbations. While the MCA-based methods are more computationally expensive than direct 1-voxel noise injections, the slowdown was found to be less significant in practice than the $100\times$ scaling factor estimated per floating point operation, presumably due to a significant portion of the pipeline computation time being spent on data management or string and integer processing rather than the constant use of floating point arithmetic.

In all cases, while tool instrumentation enables the parallelized simulation of a particular set of instructions, the aggregation of the simulated graphs is an essential component of the downstream analyses both when exploring the nature of instabilities or developing inferences upon the pipeline's derivatives. We recommend a percentile approach to aggregation, where the threshold can be adjusted based on the desired robustness of the resulting graphs. An advantage of percentile approaches is also that composite aggregates can be formed, isolating edges based on their prevalence across simulations. Further exploration of the distribution of perturbed results should be performed to conclude on the relevance of the aggregation used, as the desired aggregate should be close to the expected value of the distribution.

While both MCA and random-injection simulations are computationally expensive in that they require the evaluation of many simulations, they provide an opportunity to characterize processing modes that may emerge when analyzing either noisy datasets or unstable tools. This work also highlighted an important relationship between the noise properties of an incoming dataset and the tool, validating the need to jointly evaluate the stability of tool–dataset combinations.

Where this work demonstrates a range of numerical variation across minor changes in the quality of data or computation, it does not address the analytic impact of these deviations on downstream statistical approaches. This open question, as well as the relative impact of normalization techniques on this process, present avenues for research which will more clearly place these results in a biologically relevant context, allowing characterization of the functional impact of the observed instabilities.

Acknowledgments

This research was enabled in part by support provided by Calcul Quebec (http://www.calculquebec.ca) and Compute Canada (http://www.computecanada.ca). We would also like to thank Dell and Intel for their collaboration and contribution of computing infrastructure. The authors would also like to thank their reviewers for thoughtful and insightful comments and suggestions.

References

[1] R. D. Peng, "Reproducible research in computational science," *Science*, vol. 334, no. 6060, pp. 1226–1227, Dec. 2011.

[2] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, vol. 533, no. 7604, pp. 452–454, May 2016.

[3] S. Skare, M. Hedehus, M. E. Moseley, and T. Q. Li, "Condition number as a measure of noise performance of diffusion tensor data acquisition schemes with MRI," *Journal of Magnetic Resonance*, vol. 147, no. 2, pp. 340–352, Dec. 2000.

[4] T. Glatard, L. B. Lewis, R. Ferreira da Silva, R. Adalat, N. Beck, C. Lepage, P. Rioux, M.-E. Rousseau, T. Sherif, E. Deelman, N. Khalili-Mahani, and A. C. Evans, "Reproducibility of neuroimaging analyses across operating systems," *Frontiers in Neuroinformatics*, vol. 9, p. 12, Apr. 2015.

[5] L. B. Lewis, C. Y. Lepage, N. Khalili-Mahani, M. Omidyeganeh, S. Jeon, P. Bermudez, A. Zijdenbos, R. Vincent, R. Adalat, and A. C. Evans, "Robustness and reliability of cortical surface reconstruction in CIVET and FreeSurfer," *Annual Meeting of the Organization for Human Brain Mapping*, 2017.

[6] A. Bowring, C. Maumet, and T. E. Nichols, "Exploring the impact of analysis software on task fMRI results," *biorXiv*, Mar. 2018.

[7] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. Thompson, T. Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey, "Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration," *Neuroimage*, vol. 46, no. 3, pp. 786–802, Jul. 2009.

[8] D. S. Parker, *Monte Carlo Arithmetic: exploiting randomness in floating-point arithmetic*. University of California (Los Angeles). Computer Science Department, 1997.

[9] M. Frechtling and P. H. W. Leong, "MCALIB: Measuring sensitivity to rounding error with monte carlo programming," *ACM Transactions in Programming Language Systems*, vol. 37, no. 2, pp. 5:1–5:25, Apr. 2015.

[10] C. Denis, P. de Oliveira Castro, and E. Petit, "Verificarlo: Checking floating point accuracy through monte carlo arithmetic," 2016.

[11] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. van der Walt, M. Descoteaux, I. Nimmo-Smith, and Dipy Contributors, "Dipy, a library for the analysis of diffusion MRI data," *Frontiers in Neuroinformatics*, vol. 8, p. 8, Feb. 2014.

[12] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLoS One*, vol. 12, no. 5, p. e0177459, May 2017.

[13] G. Kiar, S. T. Brown, T. Glatard, and A. C. Evans, "A serverless tool for platform agnostic computational experiment management," *Frontiers in Neuroinformatics*, vol. 13, p. 12, Mar. 2019.

[14] K. B. Nooner, S. J. Colcombe, R. H. Tobe, M. Mennes, M. M. Benedict, A. L. Moreno, L. J. Panek, S. Brown, S. T. Zavitz, Q. Li, S. Sikka, D. Gutman, S. Bangaru, R. T. Schlachter, S. M. Kamiel, A. R. Anwar, C. M. Hinz, M. S. Kaplan, A. B. Rachlin, S. Adelsberg, B. Cheung, R. Khanuja, C. Yan, C. C. Craddock, V. Calhoun, W. Courtney, M. King, D. Wood, C. L. Cox, A. M. C. Kelly, A. Di Martino, E. Petkova, P. T. Reiss, N. Duan, D. Thomsen, B. Biswal, B. Coffey, M. J. Hoptman, D. C. Javitt, N. Pomara, J. J. Sidtis, H. S. Koplewicz, F. X. Castellanos, B. L. Leventhal, and M. P. Milham, "The NKI-Rockland sample: A model for accelerating the pace of discovery science in psychiatry," *Frontiers in Neuroscience*, vol. 6, p. 152, Oct. 2012.

[15] G. Kiar, "BIDS app - FSL diffusion preprocessing," Feb. 2019.

[16] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," *Neuroimage*, vol. 62, no. 2, pp. 782–790, Aug. 2012.

[17] L. Cammoun, X. Gigandet, D. Meskaldji, J. P. Thiran, O. Sporns, K. Q. Do, P. Maeder, R. Meuli, and P. Hagmann, "Mapping the human connectome at multiple scales with diffusion spectrum MRI," *Journal of Neuroscience Methods*, vol. 203, no. 2, pp. 386–397, Jan. 2012.

[18] G. Sawaya, M. Bentley, I. Briggs, G. Gopalakrishnan, and D. H. Ahn, "Flit: Cross-platform floating-point result-consistency tester and workload," in *2017 IEEE international symposium on workload characterization (IISWC)*. IEEE, 2017, pp. 229–238.

Numerical Instabilities in Analytical Pipelines Lead to Large and Meaningful Variability in Brain Networks

Gregory Kiar[1], Yohan Chatelain[2], Pablo de Oliveira Castro[3], Eric Petit[4], Ariel Rokem[5], Gaël Varoquaux[6], Bratislav Misic[1], Alan C. Evans[1†], Tristan Glatard[2†]

[1] *Montréal Neurological Institute, McGill University, Montréal, QC, Canada*;

[2] *Department of Computer Science and Software Engineering, Concordia University, Montréal, QC, Canada*;

[3] *Department of Computer Science, Université of Versailles, Versailles, France*;

[4] *Exascale Computing Lab, Intel, Paris, France*;

[5] *Department of Psychology and eScience Institute, University of Washington, Seattle, WA, USA*;

[6] *Parietal project-team, INRIA Saclay-ile de France, France*;

†Authors contributed equally.

Pre-print available at:

BioRxiv

https://doi.org/10.1101/2020.10.15.341495

Abstract

The analysis of brain-imaging data requires complex processing pipelines to support findings on brain function or pathologies. Recent work has shown that variability in analytical decisions can lead to substantial differences in the results, endangering the trust in conclusions[1–7]. We explored the instability of results by instrumenting a connectome estimation pipeline with Monte Carlo Arithmetic[8,9] to introduce random noise throughout. We evaluated the reliability of the connectomes, their features[10,11], and the impact on analysis[12,13]. The stability of results was found to range from perfectly stable to highly unstable. This paper highlights the potential of leveraging induced variance in estimates of brain connectivity to reduce the bias in networks alongside increasing the robustness of their applications in the classification of individual differences. We demonstrate that stability evaluations are necessary for understanding error inherent to scientific computing, and how numerical analysis can be applied to typical analytical workflows. Overall, while the extreme variability in results due to analytical instabilities could severely hamper our understanding of brain organization, it also leads to an increase in the reliability of datasets.

The modelling of brain networks, called connectomics, has shaped our understanding of the structure and function of the brain across a variety of organisms and scales over the last decade[11,14–18]. In humans, these wiring diagrams are obtained *in vivo* through Magnetic Resonance Imaging (MRI), and show promise towards identifying biomarkers of disease. This can not only improve understanding of so-called "connectopathies", such as Alzheimer's Disease and Schizophrenia, but potentially pave the way for therapeutics[19–23].

However, the analysis of brain imaging data relies on complex computational methods and software. Tools are trusted to perform everything from pre-processing tasks to downstream statistical evaluation. While these tools undoubtedly undergo rigorous evaluation on bespoke datasets, in the absence of ground-truth this is often evaluated through measures of reliability[24–27], proxy outcome statistics, or agreement with existing theory. Importantly, this means that tools are not necessarily of known or consistent quality, and it is not uncommon that equivalent experiments may lead to diverging conclusions[1,5–7]. While many scientific disciplines suffer from a lack of reproducibility[28], this was recently explored in brain imaging by a 70 team consortium which performed equivalent analyses and found widely inconsistent results[1], and it is likely that software instabilities played a role.

The present study approached evaluating reproducibility from a computational perspective in which a series of brain imaging studies were numerically perturbed such that the plausibility of results was not affected, and the biological implications of the observed instabilities were quantified. We accomplished this through the use of Monte Carlo Arithmetic (MCA)[8], a technique which enables characterization of the sensitivity of a system to small perturbations. We explored the impact of perturbations through the direct comparision of structural connectomes, the consistency of their features, and their eventual application in a neuroscience study. Finally we conclude on the consequences and opportunities afforded by the observed instabilities and make recommendations for the roles stability analyses may play towards increasing the reliability of brain imaging research.

Ch.III - 1 Graphs Vary Widely With Perturbations

Prior to exploring the analytic impact of instabilities, a direct understanding of the induced variability was required. A subset of the Nathan Kline Institute Rockland Sample (NKIRS) dataset[29] was randomly selected to contain 25 individuals with two sessions of imaging data, each of which was subsampled into two components, resulting in four collections per individual. Structural connectomes were generated with

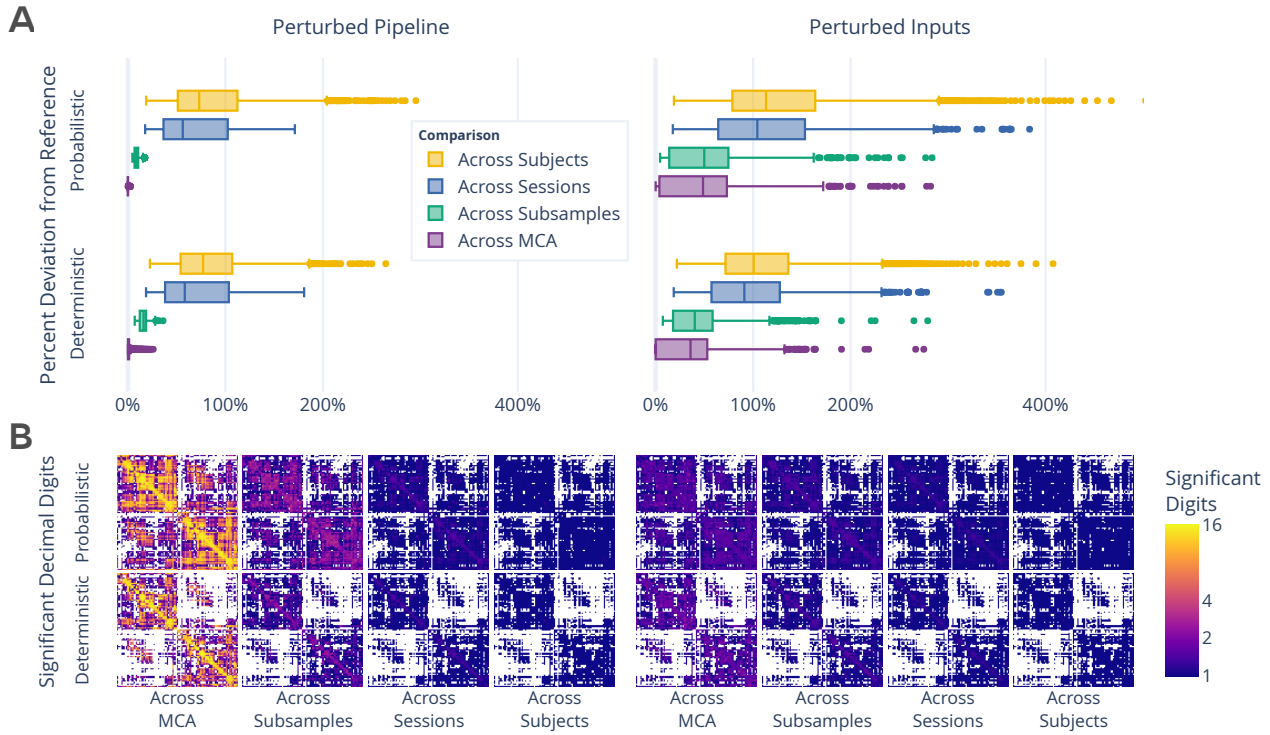

Figure Ch.III - 1. Exploration of perturbation-induced deviations from reference connectomes. (**A**) The absolute deviations, in the form of normalized percent deviation from reference, shown as the across MCA series relative to Across Subsample, Across Session, and Aross Subject variations. (**B**) The number of significant decimal digits in each set of connectomes as obtained after evaluating the effect of perturbations. In the case of 16, values can be fully relied upon, whereas in the case of 1 only the first digit of a value can be trusted. Pipeline- and input-perturbations are shown on the left and right, respectively.

canonical deterministic and probabilistic pipelines[30,31] which were instrumented with MCA, replicating computational noise at either the inputs or throughout the pipelines[4,9]. The pipelines were sampled 20 times per collection and once without perturbations, resulting in a total of $4,200$ connectomes.

The stability of connectomes was evaluated through the deviation from reference and the number of significant digits (Figure Ch.III - 1). The comparisons were grouped according to differences across simulations, subsampling of data, sessions of acquisition, or subjects. While the similarity of connectomes decreases as the collections become more distinct, connectomes generated with input perturbations show considerable variability, often reaching deviations equal to or greater than those observed across individuals or sessions (Figure Ch.III - 1A; right). This finding suggests that instabilities inherent to these pipelines may mask session or individual differences, limiting the trustworthiness of derived connectomes.

While both pipelines show similar performance, the probabilistic pipeline was more stable in the face of pipeline perturbations whereas the deterministic was more stable to input perturbations ($p < 0.0001$ for all; exploratory). The stability of correlations can be found in Supplemental Section Ch.III - S1.

The number of significant digits per edge across connectomes (Figure Ch.III - 1B) similarly decreases across groups. While the cross-MCA comparison of connectomes generated with pipeline perturbations show nearly perfect precision for many edges (approaching the maximum of 15.7 digits for 64-bit data), this evaluation uniquely shows considerable drop off in performance across data subsampling (average of < 4 digits). In addition, input perturbations show no more than an average of 3 significant digits across all groups, demonstrating a significant limitation in the reliability independent edge weights. Significance across individuals did not exceed a single digit per edge in any case, indicating that only the magnitude of edges in naively computed groupwise average connectomes can be trusted. The combination of these results with those presented in Figure Ch.III - 1A suggests that while specific edge weights are largely affected by instabilities, macro-scale network topology is stable.

Ch.III - 2 Subject-Specific Signal is Amplified While Off-Target Biases Are Reduced

Table Ch.III - 1. The impact of instabilities as evaluated through the separability of the dataset based on individual (or subject) differences, session, and subsample. The performance is reported as mean Discriminability. While a perfectly separable dataset would be represented by a score of 1.0, the chance performance, indicating minimal separability, is $1/$the number of classes. H_3 could not be tested using the reference executions due to too few possible comparisons. The alternative hypothesis, indicating significant separation, was accepted for all experiments, with $p < 0.005$.

			Reference Execution		Perturbed Pipeline		Perturbed Inputs	
Comparison	**Chance**	**Target**	Det.	Prob.	Det.	Prob.	Det.	Prob.
H_1: Across Subjects	0.04	1.0	0.64	0.65	0.82	0.82	0.77	0.75
H_2: Across Sessions	0.5	0.5	1.00	1.00	1.00	1.00	0.88	0.85
H_3: Across Subsamples	0.5	0.5			0.99	1.00	0.71	0.61

We assessed the reproducibility of the dataset through mimicking and extending a typical test-retest experiment[26] in which the similarity of samples across multiple measurements were compared to distinct samples in the dataset (Table Ch.III - 1, with additional experiments and explanation in Supplemental

Section Ch.III - S2). The ability to separate connectomes across subjects (Hypothesis 1) is an essential prerequisite for the application of brain imaging towards identifying individual differences[18]. In testing hypothesis 1, we observe that the dataset is separable with a score of 0.64 and 0.65 ($p < 0.001$; optimal score: 1.0; chance: 0.04) without any instrumentation. However, we can see that inducing instabilities through MCA improves the reliability of the dataset to over 0.75 in each case ($p < 0.001$ for all), significantly higher than without instrumentation ($p < 0.005$ for all). This result impactfully suggests the utility of perturbation methods for synthesizing robust and reliable individual estimates of connectivity, serving as a cost effective and context-agnostic method for dataset augmentation.

While the separability of individuals is essential for the identification of brain networks, it is similarly reliant on network similarity across equivalent acquisitions (Hypothesis 2). In this case, connectomes were grouped based upon session, rather than subject, and the ability to distinguish one session from another was computed within-individual and aggregated. Both the unperturbed and pipeline perturbation settings perfectly preserved differences between cross-sectional sessions with a score of 1.0 ($p < 0.005$; optimal score: 0.5; chance: 0.5), indicating a dominant session-dependent signal for all individuals despite no intended biological differences. However, while still significant relative to chance (score: 0.85 and 0.88; $p < 0.005$ for both), input perturbations lead to significantly lower separability of the dataset ($p < 0.005$ for all). This reduction of the difference between sessions of data within individuals suggests that increased variance caused by input perturbations reduces the impact of non-biological acquisition-dependent bias inherent in the brain graphs.

Though the previous sets of experiments inextricably evaluate the interaction between the dataset and tool, the use of subsampling allowed for characterizing the separability of networks sampled from within a single acquisition (Hypothesis 3). While this experiment could not be evaluated using reference executions, the executions performed with pipeline perturbations showed near perfect separation between subsamples, with scores of 0.99 and 1.0 ($p < 0.005$; optimal: 0.5; chance: 0.5). Given that there is no variability in data acquisition or preprocessing that contributes to this reliable identification of scans, the separability observed in this experiment may only be due to instability or bias inherent to the pipelines. The high variability introduced through input perturbations considerably lowered the reliability towards chance (score: 0.71 and 0.61; $p < 0.005$ for all), further supporting this as an effective method for obtaining lower-bias estimates of individual connectivity.

Across all cases, the induced perturbations showed an amplification of meaningful biological signal alongside a reduction of off-target signal. This result appears strikingly like a manifestation of the

well-known bias-variance tradeoff[32] in machine learning, a concept which observes a decrease in bias as variance is favoured by a model. In particular, this highlights that numerical perturbations can be used to not only evaluate the stability of pipelines, but that the induced variance may be leveraged for the interpretation as a robust distributions of possible results.

Ch.III - 3 Distributions of Graph Statistics Are Reliable, But Individual Statistics Are Not

Exploring the stability of topological features of connectomes is relevant for typical analyses, as low dimensional features are often more suitable than full connectomes for many analytical methods in practice[11]. A separate subset of the NKIRS dataset was randomly selected to contain a single non-subsampled session for 100 individuals, and connectomes were generated as above.

The stability of several commonly-used multivariate graph features[10] was explored in Figure Ch.III - 2. The cumulative density of the features was computed within individuals and the mean density and associated standard error were computed for across individuals (Figures Ch.III - 2A and Ch.III - 2B). There was no significant difference between the distributions for each feature across the two perturbation settings, suggesting that the topological features summarized by these multivariate features are robust across both perturbation modes.

In addition to the comparison of distributions, the stability of the first 5 moments of these features was evaluated (Figures Ch.III - 2C and Ch.III - 2D). In the face of pipeline perturbations, the feature-moments were stable with more than 10 significant digits with the exception of edge weight when using the deterministic pipeline, though the probabilistic pipeline was more stable for all comparisons ($p < 0.0001$; exploratory). In stark contrast, input perturbations led to highly unstable feature-moments (Figure Ch.III - 2D), such that none contained more than 5 significant digits of information and several contained less than a single significant digit, indicating a complete lack of reliability. This dramatic degradation in stability for individual measures strongly suggests that these features may be unreliable as individual biomarkers when derived from a single pipeline evaluation, though their reliability may be increased when studying their distributions across perturbations. A similar analysis was performed for univariate statistics and can be found in Supplemental Section Ch.III - S3.

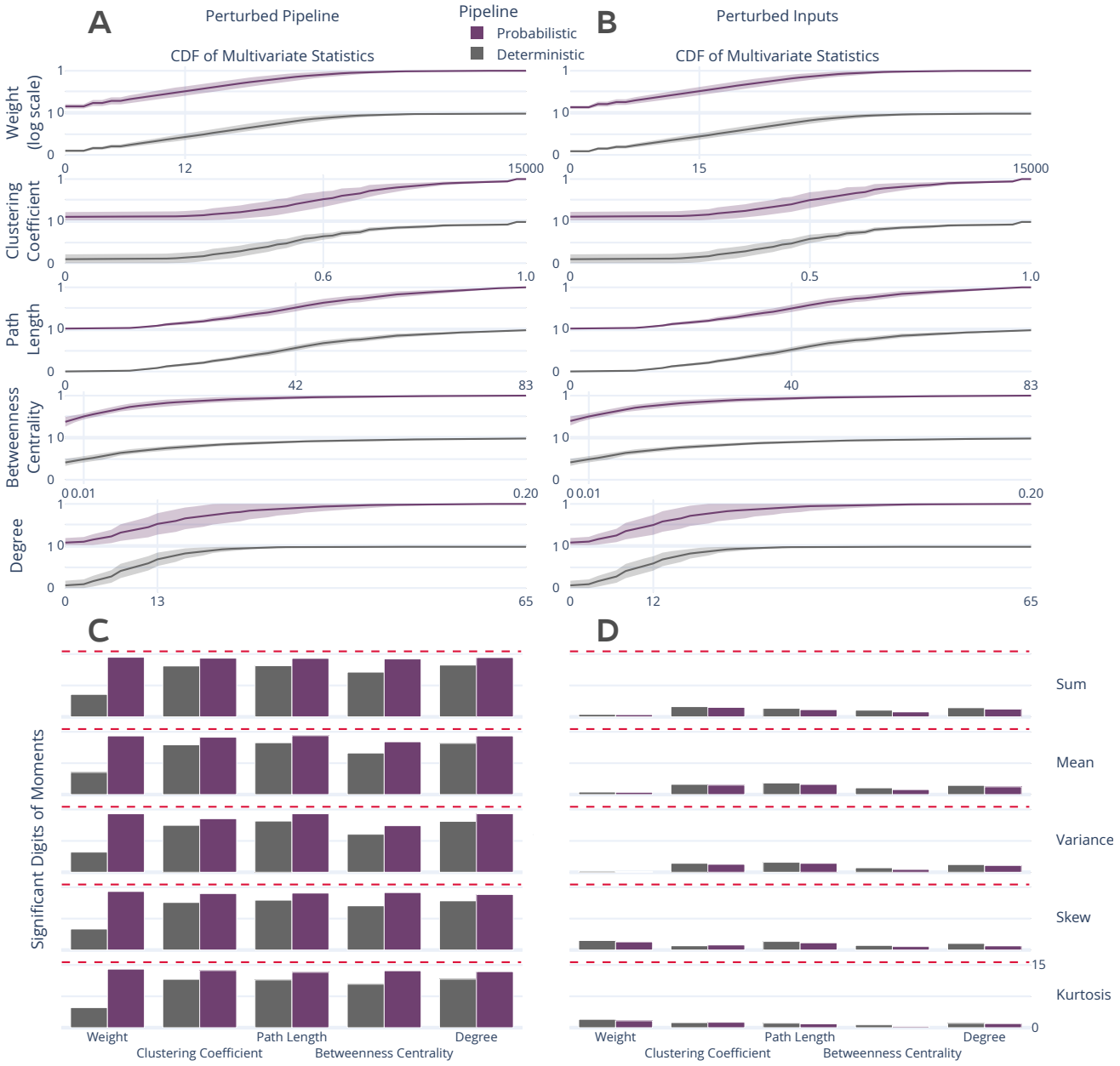

Figure Ch.III - 2. Distribution and stability assessment of multivariate graph statistics. (**A**, **B**) The cumulative distribution functions of multivariate statistics across all subjects and perturbation settings. There was no significant difference between the distributions in A and B. (**C**, **D**) The number of significant digits in the first 5 five moments of each statistic across perturbations. The dashed red line refers to the maximum possible number of significant digits.

Ch.III - 4 Uncertainty in Brain-Phenotype Relationships

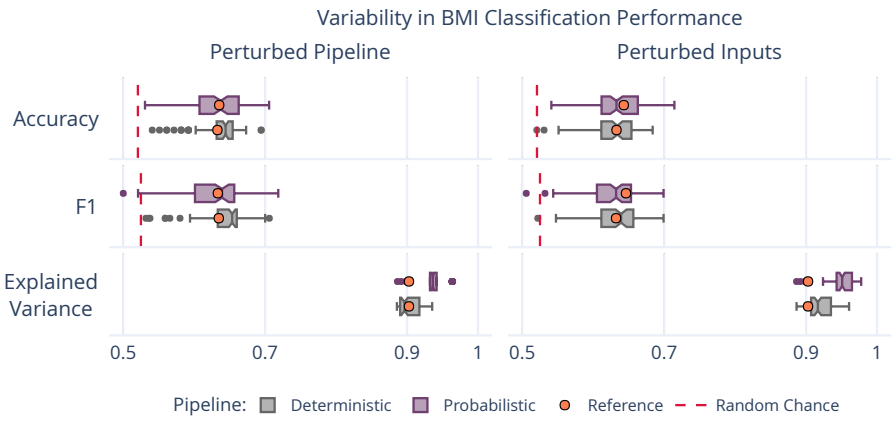

Figure Ch.III - 3. Variability in BMI classification across the sampling of an MCA-perturbed dataset. The dashed red lines indicate random-chance performance, and the orange dots show the performance using the reference executions.

While the variability of connectomes and their features was summarized above, networks are commonly used as inputs to machine learning models tasked with learning brain-phenotype relationships[18]. To explore the stability of these analyses, we modelled the relationship between high- or low- Body Mass Index (BMI) groups and brain connectivity[12,13], using standard dimensionality reduction and classification tools, and compared this to reference and random performance (Figure Ch.III - 3).

The analysis was perturbed through distinct samplings of the dataset across both pipelines and perturbation methods. The accuracy and F1 score for the perturbed models varied from 0.520 – 0.716 and 0.510 – 0.725, respectively, ranging from at or below random performance to outperforming performance on the reference dataset. This large variability illustrates a previously uncharacterized margin of uncertainty in the modelling of this relationship, and limits confidence in reported accuracy scores on singly processed datasets. The portion of explained variance in these samples ranged from 88.6% -– 97.8%, similar to the reference, suggesting that the range in performance was not due to a gain or loss of meaningful signal, but rather the reduction of bias towards specific outcome. Importantly, this finding does not suggest that modelling brain-phenotype relationships is not possible, but rather it sheds light on impactful uncertainty that must be accounted for in this process, and supports the use of ensemble modeling techniques.

Ch.III - 5 Discussion

The perturbation of structural connectome estimation pipelines with small amounts of noise, on the order of machine error, led to considerable variability in derived brain graphs. Across all analyses the stability of results ranged from nearly perfectly trustworthy (i.e. no variation) to completely unreliable (i.e. containing no trustworthy information). Given that the magnitude of introduced numerical noise is to be expected in typical settings, this finding has potentially significant implications for inferences in brain imaging as it is currently performed. In particular, this bounds the success of studying individual differences, a central objective in brain imaging[18], given that the quality of relationships between phenotypic data and brain networks will be limited by the stability of the connectomes themselves. This issue was accentuated through the crucial finding that individually derived network features were unreliable despite there being no significant difference in their aggregated distributions. This finding is not damning for the study of brain networks as a whole, but rather is strong support for the aggregation of networks, either across perturbations for an individual or across groups, over the use of individual estimates.

Underestimated False Positive Rates While the instability of brain networks was used here to demonstrate the limitations of modelling brain-phenotype relationships in the context of machine learning, this limitation extends to classical hypothesis testing, as well. Though performing individual comparisons in a hypothesis testing framework will be accompanied by reported false positive rates, the accuracy of these rates is critically dependent upon the reliability of the samples used. In reality, the true false positive rate for a test would be a combination of the reported confidence and the underlying variability in the results, a typically unknown quantity.

When performing these experiments outside of a repeated-measure context, such as that afforded here through MCA, it is impossible to empirically estimate the reliability of samples. This means that the reliability of accepted hypotheses is also unknown, regardless of the reported false positive rate. In fact, it is a virtual certainty that the true false positive rate for a given hypothesis exceeds the reported value simply as a result of numerical instabilities. This uncertainty inherent to derived data is compounded with traditional arguments limiting the trustworthiness of claims[33], and hampers the ability of researchers to evaluate the quality of results. The accompaniment of brain imaging experiments with direct evaluations of their stability, as was done here, would allow researchers to simultaneously improve the numerical stability of their analyses and accurately gauge confidence in them. The induced variability in derived brain networks may be leveraged to estimate aggregate connectomes with lower bias than any single

independent observation, leading to learned relationships that are more generalizable and ultimately more useful.

Cost-Effective Data Augmentation The evaluation of reliability in brain imaging has historically relied upon the expensive collection of repeated measurements choreographed by massive cross-institutional consortia[34,35]. The finding that perturbing experiments using MCA both increased the reliability of the dataset and decreased off-target differences across acquisitions opens the door for a promising paradigm shift. Given that MCA is data-agnostic, this technique could be used effectively in conjunction with, or in lieu of, realistic noise models to augment existing datasets. While this of course would not replace the need for repeated measurements when exploring the effect of data collection paradigm or study longitudinal progressions of development or disease, it could be used in conjunction with these efforts to increase the reliability of each distinct sample within a dataset. In contexts where repeated measurements are collected to increase the fidelity of the dataset, MCA could potentially be employed to increase the reliability of the dataset and save millions of dollars on data collection. This technique also opens the door for the characterization of reliability across axes which have been traditionally inaccessible. For instance, in the absence of a realistic noise model or simulation technique similar to MCA, the evaluation of network stability across data subsampling would not have been possible.

Shortcomings and Future Questions Given the complexity of recompiling complex software libraries, pre-processing was not perturbed in these experiments. Other work has shown that linear registration, a core piece of many elements of pre-processing such as motion correction and alignment, is sensitive to minor perturbations[7]. It is likely that the instabilities across the entire processing workflow would be compounded with one another, resulting in even greater variability. While the analyses performed in this paper evaluated a single dataset and set of pipelines, extending this work to other modalities and analyses is of interest for future projects.

This paper does not explore methodological flexibility or compare this to numerical instability. Recently, the nearly boundless space of analysis pipelines and their impact on outcomes in brain imaging has been clearly demonstrated[1]. The approach taken in these studies complement one another and explore instability at the opposite ends of the spectrum, with human variability in the construction of an analysis workflow on one end and the unavoidable error implicit in the digital representation of data on the other. It is of extreme interest to combine these approaches and explore the interaction of these scientific degrees of freedom with effects from software implementations, libraries, and parametric choices.

Finally, it is important to state explicitly that the work presented here does not invalidate analytical pipelines used in brain imaging, but merely sheds light on the fact that many studies are accompanied by an unknown degree of uncertainty due to machine-introduced errors. The presence of unknown error-bars associated with experimental findings limits the impact of results due to increased uncertainty. The desired outcome of this paper is to motivate a shift in scientific computing – both in neuroimaging and more broadly – towards a paradigm which favours the explicit evaluation of the trustworthiness of claims alongside the claims themselves.

Ch.III - 6 Methods

Ch.III - 6.1 Dataset

The Nathan Kline Institute Rockland Sample (NKI-RS)[29] dataset contains high-fidelity imaging and phenotypic data from over 1,000 individuals spread across the lifespan. A subset of this dataset was chosen for each experiment to both match sample sizes presented in the original analyses and to minimize the computational burden of performing MCA. The selected subset comprises 100 individuals ranging in age from 6 – 79 with a mean of 36.8 (original: 6 – 81, mean 37.8), 60% female (original: 60%), with 52% having a BMI over 25 (original: 54%).

Each selected individual had at least a single session of both structural T1-weighted (MPRAGE) and diffusion-weighted (DWI) MR imaging data. DWI data was acquired with 137 diffusion directions; more information regarding the acquisition of this dataset can be found in the NKI-RS data release[29].

In addition to the 100 sessions mentioned above, 25 individuals had a second session to be used in a test-retest analysis. Two additional copies of the data for these individuals were generated, including only the odd or even diffusion directions (64 + 9 B0 volumes = 73 in either case). This allowed for an extra level of stability evaluation to be performed between the levels of MCA and session-level variation.

In total, the dataset is composed of 100 downsampled sessions of data originating from 50 acquisitions and 25 individuals for in depth stability analysis, and an additional 100 sessions of full-resolution data from 100 individuals for subsequent analyses.

Ch.III - 6.2 Processing

The dataset was preprocessed using a standard FSL[36] workflow consisting of eddy-current correction and alignment. The MNI152 atlas[37] was aligned to each session of data, and the resulting transformation was applied to the DKT parcellation[38]. Downsampling the diffusion data took place after preprocessing

was performed on full-resolution sessions, ensuring that an additional confound was not introduced in this process when comparing between downsampled sessions. The preprocessing described here was performed once without MCA, and thus is not being evaluated.

Structural connectomes were generated from preprocessed data using two canonical pipelines from Dipy[30]: deterministic and probabilistic. In the deterministic pipeline, a constant solid angle model was used to estimate tensors at each voxel and streamlines were then generated using the EuDX algorithm[31]. In the probabilistic pipeline, a constrained spherical deconvolution model was fit at each voxel and streamlines were generated by iteratively sampling the resulting fiber orientation distributions. In both cases tracking occurred with 8 seeds per 3D voxel and edges were added to the graph based on the location of terminal nodes with weight determined by fiber count.

The random state of the probabilistic pipeline was fixed for all analyses. Fixing this random seed allowed for explicit attribution of observed variability to Monte Carlo simulations rather than internal state of the algorithm.

Ch.III - 6.3 Perturbations

All connectomes were generated with one reference execution where no perturbation was introduced in the processing. For all other executions, all floating point operations were instrumented with Monte Carlo Arithmetic (MCA)[8] through Verificarlo[9]. MCA simulates the distribution of errors implicit to all instrumented floating point operations (flop). This rounding is performed on a value x at precision t by:

$$inexact(x) = x + 2^{e_x - t}\xi \tag{1}$$

where e_x is the exponent value of x and ξ is a uniform random variable in the range $(-\frac{1}{2}, \frac{1}{2})$. MCA can be introduced in two places for each flop: before or after evaluation. Performing MCA on the inputs of an operation limits its precision, while performing MCA on the output of an operation highlights round-off errors that may be introduced. The former is referred to as Precision Bounding (PB) and the latter is called Random Rounding (RR).

Using MCA, the execution of a pipeline may be performed many times to produce a distribution of results. Studying the distribution of these results can then lead to insights on the stability of the instrumented tools or functions. To this end, a complete software stack was instrumented with MCA and is made available on GitHub at `https://github.com/gkiar/fuzzy`.

Both the RR and PB variants of MCA were used independently for all experiments. As was presented in[4], both the degree of instrumentation (i.e. number of affected libraries) and the perturbation mode have an effect on the distribution of observed results. For this work, the RR-MCA was applied across the bulk of the relevant libraries and is referred to as Pipeline Perturbation. In this case the bulk of numerical operations were affected by MCA.

Conversely, the case in which PB-MCA was applied across the operations in a small subset of libraries is here referred to as Input Perturbation. In this case, the inputs to operations within the instrumented libraries (namely, Python and Cython) were perturbed, resulting in less frequent, data-centric perturbations. Alongside the stated theoretical differences, Input Perturbation is considerably less computationally expensive than Pipeline Perturbation.

All perturbations targeted the least-significant-bit for all data ($t = 24$ and $t = 53$ in float32 and float64, respectively[9]). Simulations were performed 20 times for each pipeline execution. A detailed motivation for the number of simulations can be found in[39].

Ch.III - 6.4 Evaluation

The magnitude and importance of instabilities in pipelines can be considered at a number of analytical levels, namely: the induced variability of derivatives directly, the resulting downstream impact on summary statistics or features, or the ultimate change in analyses or findings. We explore the nature and severity of instabilities through each of these lenses. Unless otherwise stated, all p-values were computed using Wilcoxon signed-rank tests.

Ch.III - 6.4.1 Direct Evaluation of the Graphs

The differences between simulated graphs was measured directly through both a direct variance quantification and a comparison to other sources of variance such as individual- and session-level differences.

Quantification of Variability Graphs, in the form of adjacency matrices, were compared to one another using three metrics: normalized percent deviation, Pearson correlation, and edgewise significant digits. The normalized percent deviation measure, defined in[4], scales the norm of the difference between a simulated graph and the reference execution (that without intentional perturbation) with respect to the norm of the reference graph. The purpose of this comparison is to provide insight on the scale of differences in observed graphs relative to the original signal intensity. A Pearson correlation coefficient[40] was computed in complement to normalized percent deviation to identify the consistency of structure and not just intensity

between observed graphs.

Finally, the estimated number of significant digits, s', for each edge in the graph is calculated as:

$$s' = -log_{10}\frac{\sigma}{|\mu|} \tag{2}$$

where μ and σ are the mean and unbiased estimator of standard deviation across graphs, respectively. The upper bound on significant digits is 15.7 for 64-bit floating point data.

The percent deviation, correlation, and number of significant digits were each calculated within a single session of data, thereby removing any subject- and session-effects and providing a direct measure of the tool-introduced variability across perturbations. A distribution was formed by aggregating these individual results.

Class-based Variability Evaluation To gain a concrete understanding of the significance of observed variations we explore the separability of our results with respect to understood sources of variability, such as subject-, session-, and pipeline-level effects. This can be probed through Discriminability[26], a technique similar to ICC[24] which relies on the mean of a ranked distribution of distances between observations belonging to a defined set of classes. The discriminability statistic is formalized as follows:

$$Disc. = Pr(\|g_{ij} - g_{ij'}\| \leq \|g_{ij} - g_{i'j'}\|) \tag{3}$$

where g_{ij} is a graph belonging to class i that was measured at observation j, where $i \neq i'$ and $j \neq j'$.

Discriminability can then be read as the probability that an observation belonging to a given class will be more similar to other observations within that class than observations of a different class. It is a measure of reproducibility, and is discussed in detail in[26]. This definition allows for the exploration of deviations across arbitrarily defined classes which in practice can be any of those listed above. We combine this statistic with permutation testing to test hypotheses on whether differences between classes are statistically significant in each of these settings.

With this in mind, three hypotheses were defined. For each setting, we state the alternate hypotheses, the variable(s) which were used to determine class membership, and the remaining variables which may be sampled when obtaining multiple observations. Each hypothesis was tested independently for each pipeline and perturbation mode, and in every case where it was possible the hypotheses were tested using the reference executions alongside using MCA.

H_{A1}: Individuals are distinct from one another

Class definition: *Subject ID*

Comparisons: *Session (1 subsample)*, *Subsample (1 session)*, *MCA (1 subsample, 1 session)*

H_{A2}: Sessions within an individual are distinct

Class definition: *Session ID* | *Subject ID*

Comparisons: *Subsample*, *MCA (1 subsample)*

H_{A3}: Subsamples are distinct

Class definition: *Subsample* | *Subject ID, Session ID*

Comparisons: *MCA*

As a result, we tested 3 hypotheses across 6 MCA experiments and 3 reference experiments on 2 pipelines and 2 perturbation modes, resulting in a total of 30 distinct tests.

Ch.III - 6.4.2 Evaluating Graph-Theoretical Metrics

While connectomes may be used directly for some analyses, it is common practice to summarize them with structural measures, which can then be used as lower-dimensional proxies of connectivity in so-called graph-theoretical studies[11]. We explored the stability of several commonly-used univariate (graphwise) and multivariate (nodewise or edgewise) features. The features computed and subsequent methods for comparison in this section were selected to closely match those computed in[10].

Univariate Differences For each univariate statistic (edge count, mean clustering coefficient, global efficiency, modularity of the largest connected component, assortativity, and mean path length) a distribution of values across all perturbations within subjects was observed. A Z-score was computed for each sample with respect to the distribution of feature values within an individual, and the proportion of "classically significant" Z-scores, i.e. corresponding to $p < 0.05$, was reported and aggregated across all subjects. The number of significant digits contained within an estimate derived from a single subject were calculated and aggregated.

Multivariate Differences In the case of both nodewise (degree distribution, clustering coefficient, betweenness centrality) and edgewise (weight distribution, connection length) features, the cumulative density functions of their distributions were evaluated over a fixed range and subsequently aggregated across individuals. The number of significant digits for each moment of these distributions (sum, mean, variance, skew, and kurtosis) were calculated across observations within a sample and aggregated.

Ch.III - 6.4.3 Evaluating A Brain-Phenotype Analysis

Though each of the above approaches explores the instability of derived connectomes and their features, many modern studies employ modeling or machine-learning approaches, for instance to learn brain-phenotype relationships or identify differences across groups. We carried out one such study and explored the instability of its results with respect to the upstream variability of connectomes characterized in the previous sections. We performed the modeling task with a single sampled connectome per individual and repeated this sampling and modelling 20 times. We report the model performance for each sampling of the dataset and summarize its variance.

BMI Classification Structural changes have been linked to obesity in adolescents and adults[41]. We classified normal-weight and overweight individuals from their structural networks (using for overweight a cutoff of BMI > 25[13]). We reduced the dimensionality of the connectomes through principal component analysis (PCA), and provided the first N-components to a logistic regression classifier for predicting BMI class membership, similar to methods shown in[12,13]. The number of components was selected as the minimum set which explained $> 90\%$ of the variance when averaged across the training set for each fold within the cross validation of the original graphs; this resulted in a feature of 20 components. We trained the model using k-fold cross validation, with $k = 2, 5, 10$, and N (equivalent to leave-one-out; LOO).

Ch.III - 6.4.4 Data & Code Provenance

The unprocessed dataset is available through The Consortium of Reliability and Reproducibility (`http://fcon_1000.projects.nitrc.org/indi/enhanced/`), including both the imaging data as well as phenotypic data which may be obtained upon submission and compliance with a Data Usage Agreement. The connectomes generated through simulations have been bundled and stored permanently (`https://doi.org/10.5281/zenodo.4041549`), and are made available through The Canadian Open Neuroscience Platform (`https://portal.conp.ca/search`, search term "Kiar").

All software developed for processing or evaluation is publicly available on GitHub at `https://github.com/gkpapers/2020ImpactOfInstability`. Experiments were launched using Boutiques[42] and Clowdr[43] in Compute Canada's HPC cluster environment. MCA instrumentation was achieved through Verificarlo[9] available on Github at `https://github.com/verificarlo/verificarlo`. A set of MCA instrumented software containers is available on Github at `https://github.com/gkiar/fuzzy`.

Ch.III - 6.5 Author Contributions

GK was responsible for the experimental design, data processing, analysis, interpretation, and the majority of writing. All authors contributed to the revision of the manuscript. YC, POC, and EP were responsible for MCA tool development and software testing. AR, GV, and BM contributed to experimental design and interpretation. TG contributed to experimental design, analysis, and interpretation. TG and ACE were responsible for supervising and supporting all contributions made by GK. The authors declare no competing interests for this work. Correspondence and requests for materials should be addressed to Tristan Glatard at tristan.glatard@concordia.ca.

Ch.III - 6.6 Acknowledgments

This research was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (award no. CGSD3-519497-2018). This work was also supported in part by funding provided by Brain Canada, in partnership with Health Canada, for the Canadian Open Neuroscience Platform initiative.

References

[1] R. Botvinik-Nezer, F. Holzmeister, C. F. Camerer, A. Dreber, J. Huber, M. Johannesson, M. Kirchler, R. Iwanir, J. A. Mumford, R. A. Adcock *et al.*, "Variability in the analysis of a single neuroimaging dataset by many teams," *Nature*, pp. 1–7, 2020.

[2] C. M. Bennett, M. B. Miller, and G. L. Wolford, "Neural correlates of interspecies perspective taking in the post-mortem Atlantic salmon: An argument for multiple comparisons correction," *Neuroimage*, vol. 47, no. Suppl 1, p. S125, 2009.

[3] A. Eklund, T. E. Nichols, and H. Knutsson, "Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates," *Proceedings of the national academy of sciences*, vol. 113, no. 28, pp. 7900–7905, 2016.

[4] G. Kiar, P. de Oliveira Castro, P. Rioux, E. Petit, S. T. Brown, A. C. Evans, and T. Glatard, "Comparing perturbation models for evaluating stability of neuroimaging pipelines," *The International Journal of High Performance Computing Applications*, 2020.

[5] A. Salari, G. Kiar, L. Lewis, A. C. Evans, and T. Glatard, "File-based localization of numerical perturbations in data analysis pipelines," *arXiv preprint arXiv:2006.04684*, 2020.

[6] L. B. Lewis, C. Y. Lepage, N. Khalili-Mahani, M. Omidyeganeh, S. Jeon, P. Bermudez, A. Zijdenbos, R. Vincent, R. Adalat, and A. C. Evans, "Robustness and reliability of cortical surface reconstruction in CIVET and FreeSurfer," *Annual Meeting of the Organization for Human Brain Mapping*, 2017.

[7] T. Glatard, L. B. Lewis, R. Ferreira da Silva, R. Adalat, N. Beck, C. Lepage, P. Rioux, M.-E. Rousseau, T. Sherif, E. Deelman, N. Khalili-Mahani, and A. C. Evans, "Reproducibility of neuroimaging analyses across operating systems," *Front. Neuroinform.*, vol. 9, p. 12, Apr. 2015.

[8] D. S. Parker, *Monte Carlo Arithmetic: exploiting randomness in floating-point arithmetic*. University of California (Los Angeles). Computer Science Department, 1997.

[9] C. Denis, P. de Oliveira Castro, and E. Petit, "Verificarlo: Checking floating point accuracy through monte carlo arithmetic," *2016 IEEE 23nd Symposium on Computer Arithmetic (ARITH)*, 2016.

[10] R. F. Betzel, A. Griffa, P. Hagmann, and B. Mišić, "Distance-dependent consensus thresholds for generating group-representative structural brain networks," *Network neuroscience*, vol. 3, no. 2, pp. 475–496, 2019.

[11] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, Sep. 2010.

[12] B.-Y. Park, J. Seo, J. Yi, and H. Park, "Structural and functional brain connectivity of people with obesity and prediction of body mass index using connectivity," *PLoS One*, vol. 10, no. 11, p. e0141376, Nov. 2015.

[13] A. Gupta, E. A. Mayer, C. P. Sanmiguel, J. D. Van Horn, D. Woodworth, B. M. Ellingson, C. Fling, A. Love, K. Tillisch, and J. S. Labus, "Patterns of brain structural connectivity differentiate normal weight from overweight subjects," *Neuroimage Clin*, vol. 7, pp. 506–517, Jan. 2015.

[14] T. E. Behrens and O. Sporns, "Human connectomics," *Current opinion in neurobiology*, vol. 22, no. 1, pp. 144–153, 2012.

[15] M. Xia, Q. Lin, Y. Bi, and Y. He, "Connectomic insights into topologically centralized network edges and relevant motifs in the human brain," *Frontiers in human neuroscience*, vol. 10, p. 158, 2016.

[16] J. L. Morgan and J. W. Lichtman, "Why not connectomics?" *Nature methods*, vol. 10, no. 6, p. 494, 2013.

[17] M. P. Van den Heuvel, E. T. Bullmore, and O. Sporns, "Comparative connectomics," *Trends in cognitive sciences*, vol. 20, no. 5, pp. 345–361, 2016.

[18] J. Dubois and R. Adolphs, "Building a science of individual differences from fMRI," *Trends Cogn. Sci.*, vol. 20, no. 6, pp. 425–443, Jun. 2016.

[19] A. Fornito and E. T. Bullmore, "Connectomics: a new paradigm for understanding brain disease," *European Neuropsychopharmacology*, vol. 25, no. 5, pp. 733–748, 2015.

[20] G. Deco and M. L. Kringelbach, "Great expectations: using whole-brain computational connectomics for understanding neuropsychiatric disorders," *Neuron*, vol. 84, no. 5, pp. 892–905, 2014.

[21] T. Xie and Y. He, "Mapping the alzheimer's brain with connectomics," *Frontiers in psychiatry*, vol. 2, p. 77, 2012.

[22] M. Filippi, M. P. van den Heuvel, A. Fornito, Y. He, H. E. H. Pol, F. Agosta, G. Comi, and M. A. Rocca, "Assessment of system dysfunction in the brain through mri-based connectomics," *The Lancet Neurology*, vol. 12, no. 12, pp. 1189–1199, 2013.

[23] M. P. Van Den Heuvel and A. Fornito, "Brain networks in schizophrenia," *Neuropsychology review*, vol. 24, no. 1, pp. 32–48, 2014.

[24] J. J. Bartko, "The intraclass correlation coefficient as a measure of reliability," *Psychol. Rep.*, vol. 19, no. 1, pp. 3–11, Aug. 1966.

[25] A. M. Brandmaier, E. Wenger, N. C. Bodammer, S. Kühn, N. Raz, and U. Lindenberger, "Assessing reliability in neuroimaging research through intra-class effect decomposition (ICED)," *Elife*, vol. 7, Jul. 2018.

[26] E. W. Bridgeford, S. Wang, Z. Yang, Z. Wang, T. Xu, C. Craddock, J. Dey, G. Kiar, W. Gray-Roncal, C. Coulantoni *et al.*, "Eliminating accidental deviations to minimize generalization error: applications in connectomics and genomics," *bioRxiv*, p. 802629, 2020.

[27] G. Kiar, E. Bridgeford, W. G. Roncal, V. Chandrashekhar, and others, "A High-Throughput pipeline identifies robust connectomes but troublesome variability," *bioRxiv*, 2018.

[28] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nature*, 2016.

[29] K. B. Nooner, S. J. Colcombe, R. H. Tobe, M. Mennes *et al.*, "The NKI-Rockland sample: A model for accelerating the pace of discovery science in psychiatry," *Front. Neurosci.*, vol. 6, p. 152, Oct. 2012.

[30] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. van der Walt, M. Descoteaux, I. Nimmo-Smith, and Dipy Contributors, "Dipy, a library for the analysis of diffusion MRI data," *Front. Neuroinform.*, vol. 8, p. 8, Feb. 2014.

[31] E. Garyfallidis, M. Brett, M. M. Correia, G. B. Williams, and I. Nimmo-Smith, "QuickBundles, a method for tractography simplification," *Front. Neurosci.*, vol. 6, p. 175, Dec. 2012.

[32] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[33] J. P. Ioannidis, "Why most published research findings are false," *PLoS medicine*, vol. 2, no. 8, p. e124, 2005.

[34] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium *et al.*, "The WU-Minn human connectome project: an overview," *Neuroimage*, vol. 80, pp. 62–79, 2013.

[35] X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. C. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos *et al.*, "An open science resource for establishing reliability and reproducibility in functional connectomics," *Scientific data*, vol. 1, no. 1, pp. 1–13, 2014.

[36] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," *Neuroimage*, vol. 62, no. 2, pp. 782–790, Aug. 2012.

[37] J. L. Lancaster, D. Tordesillas-Gutiérrez, M. Martinez, F. Salinas, A. Evans, K. Zilles, J. C. Mazziotta, and P. T. Fox, "Bias between mni and talairach coordinates analyzed using the icbm-152 brain template," *Human brain mapping*, vol. 28, no. 11, pp. 1194–1205, 2007.

[38] A. Klein and J. Tourville, "101 labeled brain images and a consistent human cortical labeling protocol," *Front. Neurosci.*, vol. 6, p. 171, Dec. 2012.

[39] D. Sohier, P. De Oliveira Castro, F. Févotte, B. Lathuilière, E. Petit, and O. Jamond, "Confidence intervals for stochastic arithmetic," Jul. 2018.

[40] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*, I. Cohen, Y. Huang, J. Chen, and J. Benesty, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4.

[41] C. A. Raji, A. J. Ho, N. N. Parikshak, J. T. Becker, O. L. Lopez, L. H. Kuller, X. Hua, A. D. Leow, A. W. Toga, and P. M. Thompson, "Brain structure and obesity," *Hum. Brain Mapp.*, vol. 31, no. 3, pp. 353–364, Mar. 2010.

[42] T. Glatard, G. Kiar, T. Aumentado-Armstrong, N. Beck, P. Bellec, R. Bernard, A. Bonnet, S. T. Brown, S. Camarasu-Pop, F. Cervenansky, S. Das, R. Ferreira da Silva, G. Flandin, P. Girard, K. J. Gorgolewski, C. R. G. Guttmann, V. Hayot-Sasson,

P.-O. Quirion, P. Rioux, M.-É. Rousseau, and A. C. Evans, "Boutiques: a flexible framework to integrate command-line applications in computing platforms," *Gigascience*, vol. 7, no. 5, May 2018.

[43] G. Kiar, S. T. Brown, T. Glatard, and A. C. Evans, "A serverless tool for platform agnostic computational experiment management," *Front. Neuroinform.*, vol. 13, p. 12, Mar. 2019.

[44] H. Huang and M. Ding, "Linking functional connectivity and structural connectivity quantitatively: a comparison of methods," *Brain connectivity*, vol. 6, no. 2, pp. 99–108, 2016.

Ch.III - S1 Graph Correlation

The correlations between observed graphs (Figure Ch.III - S1) across each grouping follow the same trend to as percent deviation, as shown in Figure Ch.III - 1. However, notably different from percent deviation, there is no significant difference in the correlations between pipeline or input instrumentations. By this measure, the probabilistic pipeline is more stable in all cross-MCA and cross-directions except for the combination of input perturbation and cross-MCA ($p < 0.0001$ for all; exploratory).

The marked lack in drop-off of performance across these settings, inconsistent with the measures show in Figure Ch.III - 1 is due to the nature of the measure and the graphs. Given that structural graphs are sparse and contain considerable numbers of zero-weighted edges, the presence or absense of an edge dominated the correlation measure where it was less impactful for the others. For this reason and others[44], correlation is not a commonly used measure in the context of structural connectivity.

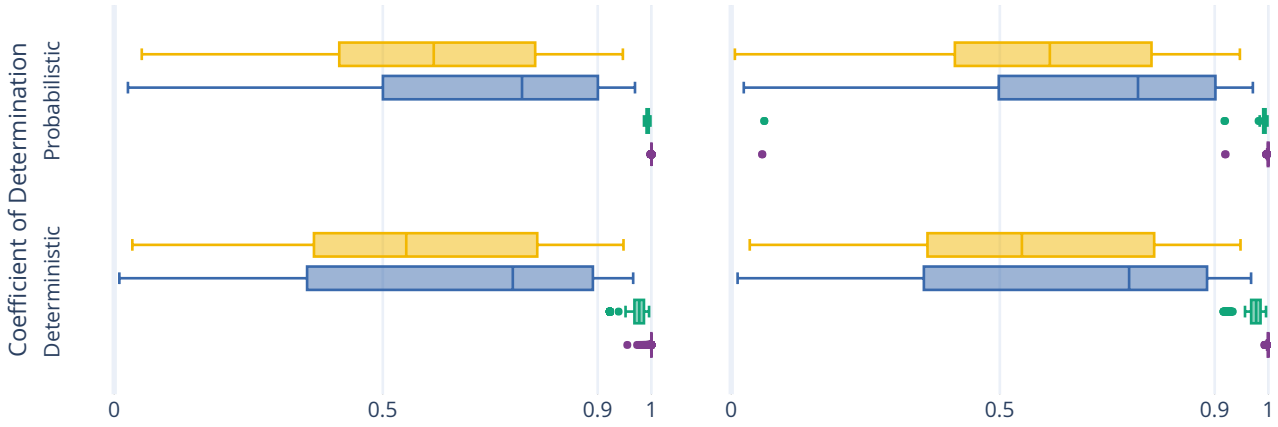

Figure Ch.III - S1. The correlation between perturbed connectomes and their reference.

Ch.III - S2 Complete Discriminability Analysis

Table Ch.III - S1. The complete results from the Discriminability analysis, with results reported as mean ± standard deviation Discriminability. As was the case in the condensed table, the alternative hypothesis, indicating significant separation across groups, was accepted for all experiments, with $p < 0.005$.

				Reference Execution		**Perturbed Pipeline**		**Perturbed Inputs**	
Exp.	Subj.	Sess.	Samp.	Det.	Prob.	Det.	Prob.	Det.	Prob.
1.1	All	All	1	0.64 ± 0.00	0.65 ± 0.00	0.82 ± 0.00	0.82 ± 0.00	0.77 ± 0.00	0.75 ± 0.00
1.2	All	1	All	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.93 ± 0.02	0.90 ± 0.02
1.3	All	1	1			1.00 ± 0.00	1.00 ± 0.00	0.94 ± 0.02	0.90 ± 0.02
2.4	1	All	All	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.88 ± 0.12	0.85 ± 0.12
2.5	1	All	1			1.00 ± 0.00	1.00 ± 0.00	0.89 ± 0.11	0.84 ± 0.12
3.6	1	1	All			0.99 ± 0.03	1.00 ± 0.00	0.71 ± 0.07	0.61 ± 0.05

The complete discriminability analysis includes comparisons across more axes of variability than the condensed version. The reduction in the main body was such that only axes which would be relevant for a typical analysis were presented. Here, each of Hypothesis 1, testing the difference across subjects, and 2, testing the difference across sessions, were accompanied with additional comparisons to those shown in the main body.

Subject Variation Alongside experiment 1.1, that which mimicked a typical test-retest scenario, experiments 1.2 and 1.3 could be considered a test-retest with a handicap, given a single aqcuisition per individual was compared either across subsamples or simulations, respectively. For this reason, it is unsurprising that the dataset achieved considerably higher discriminability scores.

Session Variation Similar to subject variation, the session variation was also modelled across either both or a single subsample. In both of these cases the performance was similar, and the finding that input perturbation reduced the off-target signal was consistent.

Ch.III - S3 Univariate Graph Statistics

Figure Ch.III - S2 explores the stability of univariate graph-theoretical metrics computed from the perturbed graphs, including modularity, global efficiency, assortativity, average path length, and edge count. When aggregated across individuals and perturbations, the distributions of these statistics (Figures Ch.III - S2A and Ch.III - S22B) showed no significant differences between perturbation methods for either deterministic or probabilistic pipelines.

However, when quantifying the stability of these measures across connectomes derived from a single session of data, the two perturbation methods show considerable differences. The number of significant digits in univariate statistics for Pipeline Perturbation instrumented connectome generation exceeded 11 digits for all measures except modularity, which contained more than 4 significant digits of information (Figure Ch.III - S2C). When detecting outliers from the distributions of observed statistics for a given session, the false positive rate (using a threshold of $p = 0.05$) was approximately 2% for all statistics with the exception of modularity which again was less stable with an approximately 10% false positive rate. The probabilistic pipeline is significantly more stable than the deterministic pipeline ($p < 0.0001$; exploratory) for all features except modularity. When similarly evaluating these features from connectomes generated in the input perturbation setting, no statistic was stable with more than 3 significant digits or a false positive rate lower than nearly 6% (Figure Ch.III - S2D). The deterministic pipeline was more stable than the probabilistic pipeline in this setting ($p < 0.0001$; exploratory).

Two notable differences between the two perturbation methods are, first, the uniformity in the stability of the statistics, and second, the dramatic decline in stability of individual statistics in the input perturbation setting despite the consistency in the overall distribution of values. It is unclear at present if the discrepancy between the stability of modularity in the pipeline perturbation context versus the other statistics suggests the implementation of this measure is the source of instability or if it is implicit to the measure itself. The dramatic decline in the stability of features derived from input perturbed graphs despite no difference in their overall distribution both shows that while individual estimates may be unstable the comparison between aggregates or groups may be considered much more reliable; this finding is consistent with that presented for multivariate statistics.

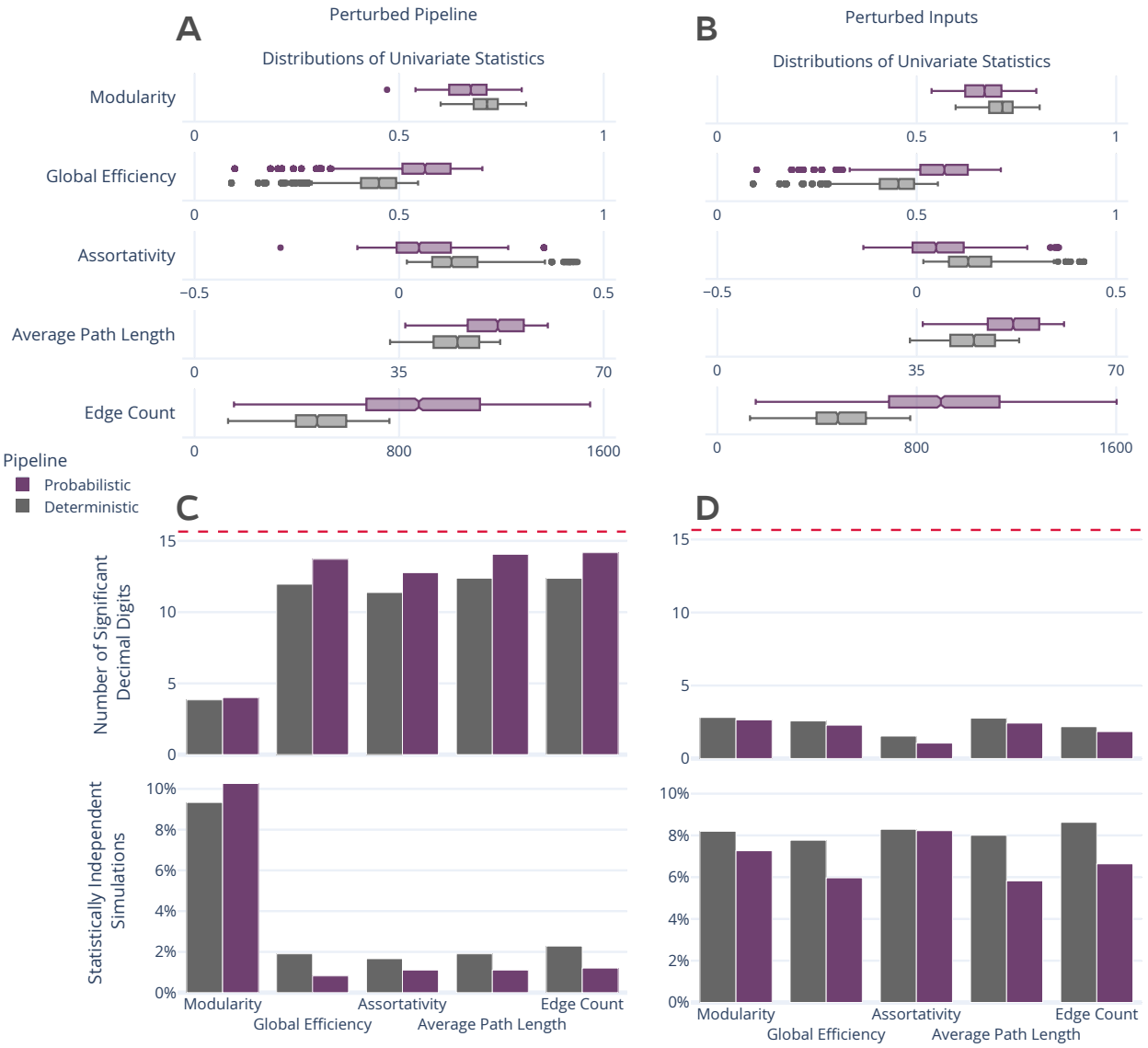

Figure Ch.III - S2. Distribution and stability assessment of univariate graph statistics. (**A**, **B**) The distributions of each computed univariate statistic across all subjects and perturbations for Pipeline and Input settings, respectively. There was no significant difference between the distributions in A and B. (**C**, **D**; top) The number of significant decimal digits in each statistic across perturbations, averaged across individuals. The dashed red line refers to the maximum possible number of significant digits. (**C**, **D**; bottom) The percentage of connectomes which were deemed significantly different ($p < 0.05$) from the others obtained for an individual.

Data Augmentation Through Monte Carlo Arithmetic Leads to More Generalizable Classification in Connectomics

Gregory Kiar[1], Yohan Chatelain[2], Ali Salari[2], Alan C. Evans[1], Tristan Glatard[2]

[1] *Montréal Neurological Institute, McGill University, Montréal, QC, Canada*;

[2] *Department of Computer Science and Software Engineering, Concordia University, Montréal, QC, Canada*.

Pre-print available at:

BioRxiv

https://doi.org/10.1101/2020.10.15.341495

Abstract

Machine learning models are commonly applied to human brain imaging datasets in an effort to associate function or structure with behaviour, health, or other individual phenotypes. Such models often rely on low-dimensional maps generated by complex processing pipelines. However, the numerical instabilities inherent to pipelines limit the fidelity of these maps and introduce computational bias. Monte Carlo Arithmetic, a technique for introducing controlled amounts of numerical noise, was used to perturb a structural connectome estimation pipeline, ultimately producing a range of plausible networks for each sample. The variability in the perturbed networks was captured in an augmented dataset, which was then used for an age classification task. We found that resampling brain networks across a series of such numerically perturbed outcomes led to improved performance in all tested classifiers, preprocessing strategies, and dimensionality reduction techniques. Importantly, we find that this benefit does not hinge on a large number of perturbations, suggesting that even minimally perturbing a dataset adds meaningful variance which can be captured in the subsequently designed models.

Ch.IV - 1 Introduction

The application of machine learning has become commonplace for the identification and characterization of individual biomarkers in neuroimaging[1]. Models have been applied to discriminate between measures of brain structure or function based upon phenotypic variables related to disease[2–4], development[5], or other axes of potential consequence[6,7].

These models often build representations upon processed imaging data, in which 3D or 4D images have been transformed into estimates of structure[8], function[9], or connectivity[10]. However, there is a lack of reliability in these estimates, including variation across analysis team[11], software library[12], operating system[13], and instability in the face of numerical noise[14]. This uncertainty limits the ability of models to learn generalizable relationships among data, and leads to biased predictors. Traditionally, this bias has been reduced through the collection and application of repeated-measurement datasets[15,16], though this requires considerable resources and is not feasible in the context of all clinical populations.

Perturbation methods which inject small amounts of noise through the execution of a pipeline, such as Monte Carlo Arithmetic (MCA)[17,18], have recently been used to induce instabilities in structural connectome estimation software[19]. Importantly, this technique produces a range of equally plausible results, where no single observation is more or less valid than the others – including those which were left unperturbed. While sampling from a set of perturbed connectomes may have an impact on learning brain-phenotype relationships[14], there remains potential for leveraging the distribution of perturbed results to augment datasets in lieu of increasing sample sizes or performing repeated measurements.

Using an existing collection of MCA-perturbed structural connectomes[20], we trained classifiers on networks sampled from the distribution of results and evaluated their performance relative to using only the unperturbed networks. We evaluate several techniques for resampling the networks, and compare classifiers through their validation performance, the performance on an out-of-sample test set, and the generalizability of their performance across the two. We demonstrate the efficacy of using MCA as a method for dataset augmentation which leads to more robust and generalizable models of brain-phenotype relationships.

Ch.IV - 2 Materials & Methods

The objective of this study was to evaluate the impact of aggregating collections of unstable brain networks towards learning robust brain-phenotype relationships. We sampled and aggregated simulated networks

within individuals to learn relationships between brain connectivity and individual phenotypes, in this case age, and compared this to baseline performance on this task. We compared aggregation strategies with respect to baseline validation performance, performance out-of-sample, and generalizability. The experimental workflow is visualized in Figure Ch.IV - 1.

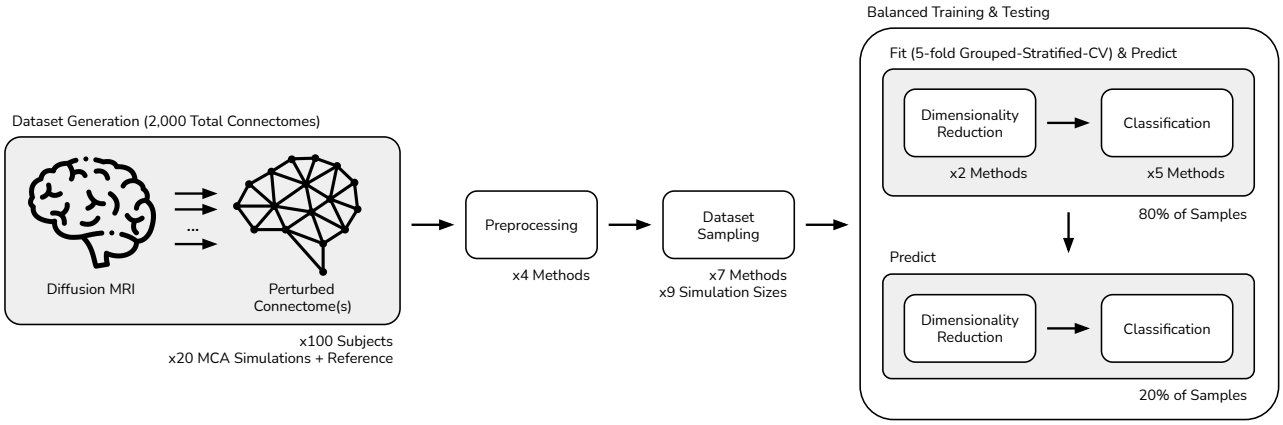

Figure Ch.IV - 1. Experiment workflow. MCA-simulated connectomes were previously generated for 100 subjects, 20 times each. The resulting dataset of $2,000$ connectomes were independently preprocessed using one of 4 techniques. The dataset was then sampled according to one of 7 dataset sampling strategies and using one of 9 possible number of MCA simulations per subject. The dataset was split into balanced training and testing sets, and fed into the models. The models consisted of one of 2 dimensionality reduction techniques prior to classification using one of 5 classifier types. The models were fit and made predictions on the training set prior to making predictions on the test set.

All developed software and analysis resources for this project have been made available through GitHub at `https://github.com/gkpapers/2020AggregateMCA`.

Ch.IV - 2.1 Dataset

An existing dataset containing Monte Carlo Arithmetic (MCA) perturbed structural human brain networks was used for these experiments[20]. The perturbations introduced for the generation of brain networks in this dataset were at the level of machine-error, simulating expected error over a typical pipeline execution. This dataset contains a single session of data from 100 individuals ($100 \times 1 \times 1$). The derived brain networks were generated with a probabilistic structural connectome estimation pipeline[21] using a fixed random seed, and minimal Monte Carlo Arithmetic (MCA) perturbations were added to all Python-implemented operations throughout the pipeline[17,18]. Each sample was simulated 20 times, resulting in $2,000$ unique

graphs. Further information on the processing and curation of this dataset can be found here[14].

This collection enabled the exploration of subsampling and aggregation methods in a typical learning context for neuroimaging[22,23]. Exploring the relationship between the number of simulations and performance further allows for MCA-enabled resampling to be evaluated as a method of dataset augmentation.

As the target for classification, individual-level phenotypic data strongly implicated in brain connectivity was desired. Participant age, which has consistently been shown to have a considerable impact on brain connectivity[24–27], was selected and turned into a binary target by dividing participants into adult (> 18) and non-adult groups (68% adult).

Ch.IV - 2.2 Preprocessing

Prior to being used for this task, the brain networks were represented as symmetric 83×83 adjacency matrices, sampled upon the Desikan-Killiany-Tourville[28] anatomical parcellation. To reduce redundancy in the data, all edges belonging to the upper-triangle of these matrices were preserved and vectorized, resulting in a feature vector of $3,486$ edges per sample. All samples were preprocessed using one of four standard techniques:

Raw The raw streamline count edge-weight intensities were used as originally calculated.

Log Transform The log10 of edge weights was taken, and edges with 0 weight prior to the transform were reset to 0.

Rank Transform The edges were ranked based on their intensity, with the largest edge having the maximum value. Ties were settled by averaging the rank, and all ranks were finally min-max scaled between 0 and 1.

Z-Score The edge weights were z-scored to have a mean intensity of 0 and unit variance.

Ch.IV - 2.3 Machine Learning Pipelines

The preprocessed connectomes were fed into pipelines consisting of two steps: dimensionality reduction and classification. Dimensionality reduction was applied using one of two methods:

Principal Component Analysis The connectomes were projected into the 20 dimensions of highest variance. The number of components was chosen to capture approximately 90% of the variance present within the dataset.

Feature Agglomeration The number of features in each connectome were reduced by combining edges according to maximum similarity/minimum variance using agglomerative clustering[29]. The number of resulting features was 20, to be consistent with the number of dimensions present after PCA, above.

After dimensionality reduction, samples were fed into one of five distinct classifiers as implemented through scikit learn[30]:

Support Vector Machine The model was fit using a radial basis function (RBF) kernel, L2 penalty, and a balanced regularization parameter to account for uneven class membership.

Logistic Regression A linear solver was used due to the relatively small dataset size. L2 regularization and balanced class weights were used, as above.

K-Nearest Neighbour Class membership was determined using an L2 distance and the nearest 10% of samples, scaling with the number of samples used for training.

Random Forest 100 decision trees were fit using balanced class weights, each splitting the dataset according to a maximum of 4 features per node (corresponding to the rounded square root of 20 total features).

AdaBoost A maximum of 50 decision trees were fit sequentially such that sample weights were iteratively adjusted to prioritize performance on previously incorrectly-classified samples, consistent with[31].

The hyperparameters for all models were refined from their default values to be appropriate for a small and imbalanced dataset. The performance for all pipeline combinations of preprocessing methods, dimensionality reduction techniques, and models using the reference (i.e. unperturbed) executions in the dataset ranged from an F1 score of 0.64–0.875 with a mean of 0.806; this evaluation was performed on a consistent held-out test set which was used for all experiments, as described in a following section. This set of models was chosen as it includes i) well understood standard techniques, ii) both parametric and non-parametric methods, iii) both ensemble and non-ensemble methods, and iv) models which have been commonly deployed for the classification of neuroimaging datasets[2–4,6,7,24,32,33].

Ch.IV - 2.4 Dataset Sampling

A chief purpose of this manuscript involves the comparison of various forms of aggregation across equivalently-simulated pipeline outputs. Accordingly, the dataset was resampled after preprocessing but prior to dimensionality reduction and classifiers were trained, evaluated, and combined according to the

following procedures:

Reference Networks generated without any MCA perturbations were selected for input to the models, serving as a benchmark.

Truncate The number of significant digits[17] per-edge was calculated using all simulated networks, and the edge weights in the reference graph were truncated to the specified number of digits. Importantly, this is the only method used which deliberately squashes the variance observed across simulations.

Jackknife The datasets were repeatedly sampled such that a single randomly chosen observation of each unique network was selected (i.e. derived from the same input datum). This resampling was performed 100 times, resulting in the total number of resamplings being $5\times$ larger than the number of unique observations per network, ensuring a broad and overlapping sampling of the datasets.

Median The edgewise median of all observations of the same network were used as the samples for training and evaluation.

Mean Similar to the above, the edgewise mean of all observations for each network were computed and used as input data to the classifiers in both collections.

Consensus A distance-dependent average network[34] was computed across all observations of each network. This data-aware aggregation method, developed for structural brain network analysis, preserves network properties often distorted when computing mean or median networks.

Mega-analysis All observations of each network were used simultaneously for classification, increasing the effective sample size. Samples were organized such that all observations of the same network only appeared within a single fold for training and evaluation, ensuring double-dipping was avoided.

Meta-analysis Individual classifiers trained across jackknife dataset resamplings, above, were treated as independent models and aggregated into an ensemble classifier. The ensemble was fit using a logistic regression classifier across the outputs of the jackknifed classifiers to learn a relationship between the predicted and true class labels.

The robustness and possible benefit of each subsampling approach was measured by evaluation on a subset of all MCA simulations, including 9 distinct numbers of simulations, ranging from 2 to 20 simulations per sample. Combining the dataset sampling methods, the set of simulations, preprocessing

strategies, dimensionality reduction techniques, and classifier models, there were 2,520 perturbed models trained and evaluated next to 40 reference models.

Ch.IV - 2.5 Training & Evaluation

Prior to training models on the brain networks, 20% of subjects were excluded from each dataset for use as an out-of-sample test dataset for all experiments. With the remaining 80% of subjects, cross validation was performed following a stratified grouped k-fold approach ($k = 5$). In this approach, samples were divided into training and validation sets such that the target variable was proportionally represented on each side of the fold (stratified), conditional upon all observations from the same individual, relevant for the mega-analysis dataset sampling method, falling upon the same side of the fold (grouped). This resulted in 5 fold-trained classifiers per configuration, each trained on 64% of the samples and validated on 16%, prior to each being tested on the remaining 20% of held-out samples. All random processes used in determining the splits used the same seed to remove the effect of random sampling.

Classifiers were primarily evaluated on both the validation and test (out-of-sample) sets using F1 score, a standard measure for evaluating classification performance. The generalizability of predictions was defined as:

$$G = 1 - |F1_{test} - F1_{validation}| \tag{1}$$

where a score of 1 (maximum) indicates the equivalent performance across both the validation and test sets, and a lower score (minimum of 0) indicates inconsistent performance. The absolute change in performance was used in Eq. 1, resulting in a score which penalizes spurious over-performance similarly to under-performance. This is a desired attribute of the measure as all inconsistency, whether due to chance or model fit, is undesirable when applying a classifier out-of-sample.

Differences in F1 score and generalizability for perturbed experiments with respect to their reference were used to measure the change in performance between for each dataset sampling technique, and statistical comparisons were made through Wilcoxon Signed-Rank tests.

Ch.IV - 3 Results

The figures and findings presented in this section represent a summary of the complete experiment table which consists of performance measures and metadata for all 2,560 models tested. The complete

performance table alongside the table of significant differences, are made available through the GitHub repository.

Ch.IV - 3.1 Data Resampling Improves Classification

The overall performance of each subsampling method is summarized in Table Ch.IV - 1. The change in performance was measured in both cases as a change in F1 score on the validation set, the change in F1 score on the test set, and the change in overall generalizability, a measure which summarizes the similarity between validation and test performance for a given model.

Table Ch.IV - 1. Statistically significant change in performance. Red values indicate significant decline in performance, black values indicate improvement, and empty cells indicate no change. A single star represents $p < 0.05$, and each additional star is an additional order of magnitude of significance.

Dataset Sampling	Validation	Test	Generalizability
Truncate	**		
Jackknife	**	**	
Mean		***	
Median		***	
Consensus		***	*
Mega-Analysis	*	*	***
Meta-Analysis	**	***	*

Across all classifier types it was found that consensus, mega-, and meta-analytic approaches outperformed other dataset resampling methods, as each of these methods led to improved testing performance and generalizability. The only method which did not improve performance at all was the truncation resampling approach. This method was distinct from the others in that the variance observed across simulations was used to estimate and squash variance in the reference network, whether other approaches captured the variance. The finding that truncation hurts performance importantly suggests that the variability across the observed networks is biologically meaningful and contains signal.

The change in performance for each model and dataset sampling technique is shown in Figure Ch.IV - 2. The support vector machine and logistic regression models improve across each of the three measures for a variety of dataset sampling techniques, suggesting that the addition of the MCA-perturbed samples improves the training, testing, and overall generalizability of these classifiers.

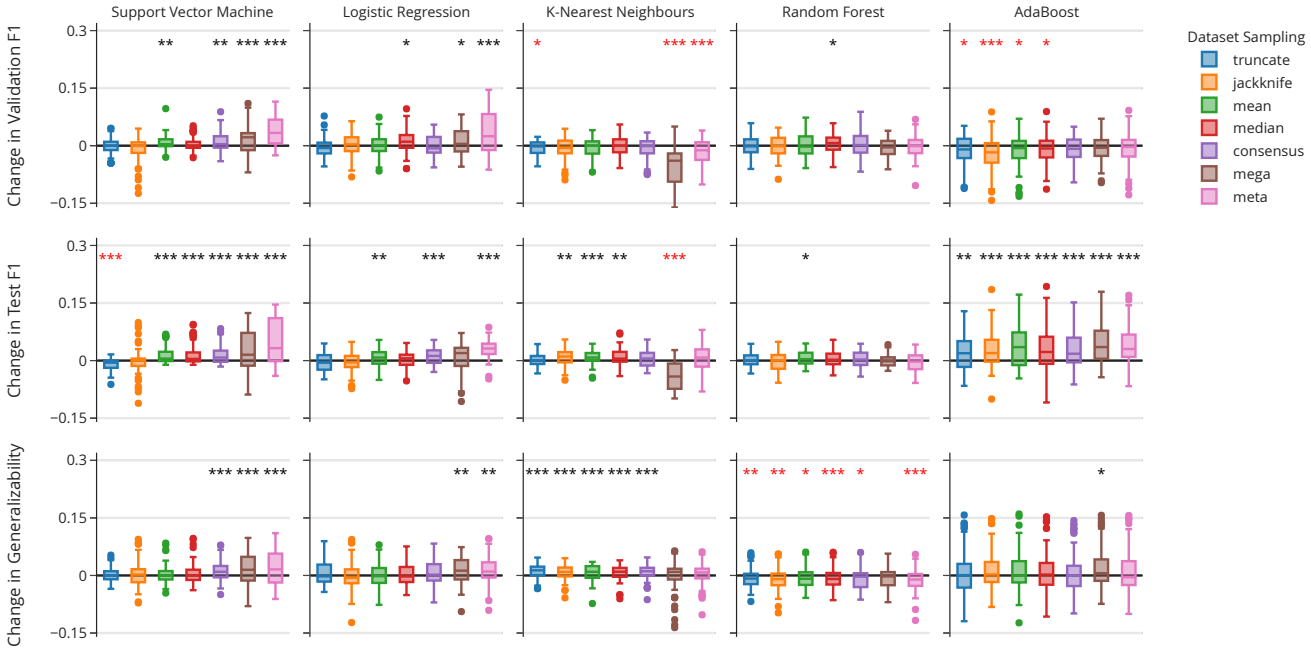

Figure Ch.IV - 2. Relative change in classifier performance with respect to classifier type and dataset sampling strategies as measured by change in F1 score on the validation set (top) or test set (middle), as well as the generalizability of performance (bottom). Each star annotation indicates an order of magnitude of statistically significant change, beginning at 0.05 for one star and decreasing from there, with those in black or red indicating an increase or decrease due to resampling, respectively.

Distinctly, k-nearest neighbours (KNN) and AdaBoost classifiers experienced minimal change in validation and often saw their performance decline. However, the improvement of these classifiers on the test set suggests that resampling reduced overfitting in these classifiers. In the case of KNN, this translates to improved generalizability, while in the case of AdaBoost generalizability was largely unchanged, suggesting that the model went from underperforming to overperforming after dataset resampling. The unique decline in performance when using the mega-analytic resampling technique on KNN classifier is suggestive of poor hyper-parameterization, as there is a strong relationship between the performance and the ratio of the number of samples in the dataset to the k parameter of the model. At present this parameter was scaled linearly with the number of MCA simulations used, however, it is both possible that an improved scaling function exists or that the model performance degrades with large sample sizes making it a poor model choice given this resampling technique.

The random forest classifiers uniquely did not see a significant change in validation or testing performance across the majority of resampling techniques. However, these classifiers did experience a significant decrease in the generalizability of their performance, meaning that there was a larger discrepancy between training and testing performance in many cases. This distinction from the other models is possibly due to the fact that random forest is a simple ensemble technique which takes advantage of training many independent classifiers and samples them to assign final class predictions, allowing this approach to form more generalizable predictions, and thus the addition of more data does not significantly improve performance further. While AdaBoost is also an ensemble method, the iterative training of models with increasing emphasis on difficult samples allows for the added variance in those samples to play an increasingly central role in the construction of class relationships, and thus more directly takes advantage of the added variability in samples.

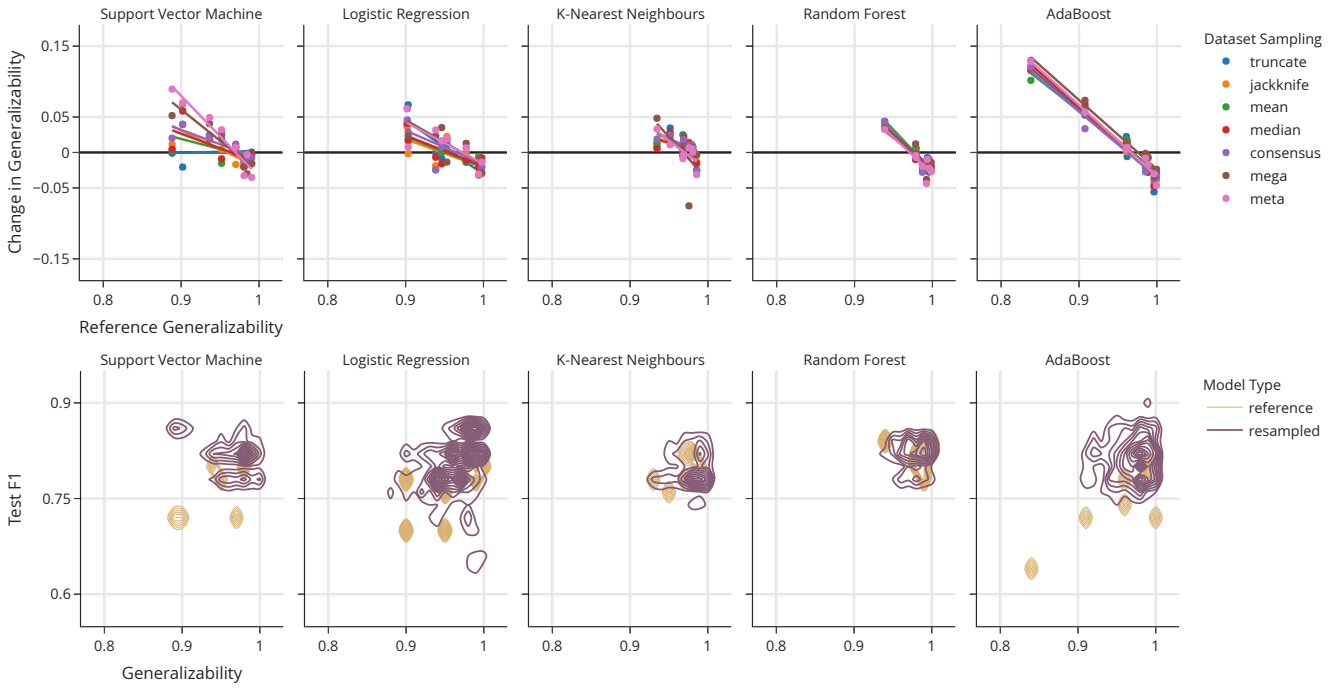

Figure Ch.IV - 3. Relationship between generalizability and resampling. Top: change in the generalizability of classifiers with respect to the reference generalizability. Each data point represents the mean change in generalizability for all models using the same preprocessing and dimensionality reduction techniques for a given classifier and dataset sampling strategy. Bottom: contour density distributions of generalizability and F1 scores across all models for both reference and resampled training.

While certain combinations of preprocessing, dimensionality reduction, and classifiers performed

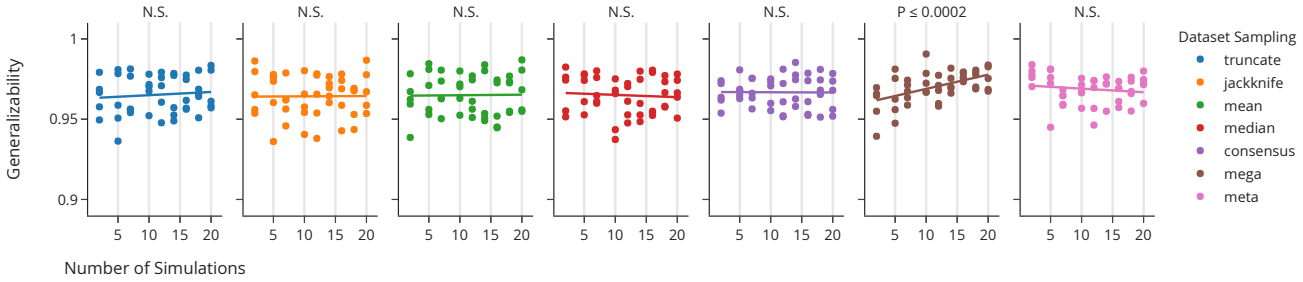

Figure Ch.IV - 4. The generalizability of classifiers using each dataset sampling technique with respect to the number of MCA simulations. Each number of simulations was sampled a single time, to avoid artificial skewing of the dataset due to the inclusion of "higher" or "lower" quality samples; a single drawing of each split mimics a true perturbation experiment context.

more harmoniously than others, there was no significant relationship between the performance of any single resampling method and preprocessing or dimensionality reduction technique. Overall, the above results show that dataset augmentation through MCA-perturbed pipeline outputs may be an effective way to improve the performance and generalizability of non-ensemble classifiers tasked with modeling brain-phenotype relationships, both within and out of sample, especially when variance is captured rather than removed.

Ch.IV - 3.2 Resampling Leads to Consistent Performance

To better characterize the the benefit of resampling, the relationship between the magnitude of improvement and the baseline performance of the classifier were further explored (Figure Ch.IV - 3). We found that the increase in the generalizability of a classifier was inversely related to the baseline generalizability (Figure Ch.IV - 3; top). In other words, the less generalizable a classifier was originally, the more its generalizability improved (significant at $p < 0.05$ for all dataset sampling strategies and classifier other than KNN). There were several situations in which the generalizability of models were noted to decrease, however, though this only occurred for models with high generalizability scores (all > 0.935). Importantly, the relative change in generalizability shifts scores towards a single "mode", suggesting a less biased estimate of the true generalizability of performance on the task, and mitigating both under- and over-performance due to chance.

When exploring the relationship between F1 and generalizability (Figure Ch.IV - 3; bottom), it becomes apparent that even for the majority of models which may not have improved performance along

both axes, either their generalizability or F1 score is improved. While an ideal classifier would reside in the top-right of the shown plots, the dataset resampling techniques consistently shift the distributions in this direction and often improve classifiers along one or both of these axes. Importantly, the variance in performance across both measures is significantly decreased, suggesting that resampling leads to more reliable and reproducible classifiers.

Ch.IV - 3.3 Number of Simulations is Largely Unimpactful

While we previously noted an increase in classifier performance due to perturbation-enabled dataset resampling, it was important to explore the relationship between the number of simulated samples and performance (Figure Ch.IV - 4). There was no relationship between the number of independent simulations and performance, as measured by either F1 or generalizability, for all dataset resampling techniques other than mega-analysis. In the case of the mega-analytic approach, however, there was a significant positive relationship between the number of samples used and the generalizability of performance, though there remained no increase in F1 score. The mega-analysis approach is the only approach which changes the number of samples being provided directly to the classifiers, thus mimics an increase in sample size for traditional experiments. While outlying samples may play a small role in many of the proposed forms of resampling, or non-existent in the median case, the mega analytic approach treats all simulations with equal importance as unique samples in the dataset. In this case, the relationship we observe is consistent to what one might expect when increasing the number of samples.

Ch.IV - 4 Discussion

The numerical perturbation of analytic pipelines provides a unique, data-agnostic, and computationally unintrusive method for dataset augmentation. Using a technique such as MCA, samples can be simulated across an array of controlled executions and used to enrich datasets with a range of plausible results per sample. We demonstrate that this method of dataset augmentation can be used to improve the training, testing, and generalizability of classifiers.

Through the training and evaluation of $2,560$ models combining varying strategies for preprocessing, dimensionality reduction, classifier, and resampling, we found consistent improvement across all measured axes. Interestingly, while there was a statistically significant improvement when using many dataset resampling techniques, there was no significant improvement in the performance, and in fact a reduction, using the truncation resampling method as is shown in Table Ch.IV - 1. This result importantly

demonstrates that the added variability in results obtained through MCA is meaningful and signal-rich itself, and an important determination of performance is the inclusion of this variability.

While the non-ensemble methods benefited most obviously from the dataset resampling strategies, where both F1 and generalizability were often improved, the results presented in Figure Ch.IV - 3 demonstrate that variability in performance across both of these axes was reduced across all classifier configurations. While a reduction in the variability of performance is desirable in itself, this figure also illustrates that the performance of resulting models converges around the more performant models in the case of all classifiers. The reduction in variability also results in models which differed less significantly when looking across classifier types.

Although performance was improved by the integration of MCA simulated samples, performance was not significantly related to the number of simulations used in any case other than the mega-analytic resampling strategy. The independence of performance and number of simulations is encouraging, as a key limitation for using Monte Carlo methods is the often extreme computational overhead. The ability to use a small number of simulations and achieve equivalent performance through the majority of resampling techniques allows for significantly better performance without added data collection and only a theoretical doubling the sample processing time. The benefit of increasing the number of simulations in the mega-analytic case could be considered an analog to increasing the sample size of an experiment. While the range of simulations used here demonstrated a consistent improvement in generalizability, there will be a plateau in performance, either at a perfect score or, more likely, before this is reached. Further work is required for characterizing the relationship between the performance of mega-analytic resampling and the number of simulations, though it is likely that this relationship will be domain-specific and dependent on other experimental design variables such as the number of features per sample.

While our study shows that classifiers with poorer baseline performance benefit more from augmentation, an important limitation of this is the operating point to which that claim remains true. For example, it is unlikely that the trend observed here for a task with a mean reference performance of 0.81 would hold across models operating with reference performance near chance or near perfect. Characterizing the behaviour of this technique across a range of classification contexts and performances would shed light on whether this technique could be applied globally or if it is limited to making “good” models better.

It is a well understood problem that small sample sizes lead to uncertainty in modeling[35]. This is generally planned for in one of two ways: the collection of vast datasets, as is the case in the UK-BioBank which strives to collect samples from half a million individuals[15], or the collection of repeated

measurements from the selected samples, as is the case in the Consortium of Reliability and Reproducibility which orchestrates multiple centres and scanners, each collecting multiple acquisitions[16]. In either case, the additional data collection by these initiatives is both financially and temporally expensive and leads to unintended confounding effects associated with time of day[36], weather[37], or other off-target variables that may be poorly described in the resulting dataset[38].

While the results presented here provide strong evidence in favour of dataset augmentation through numerical perturbations, the improvement from these methods has not been directly compared to additional data acquisitions in this experiment due to the limited sample size of the available perturbed dataset[20]. Previous studies exploring the effect of sample size on neuroimaging classification tasks have shown that variability in performance decreases with sample size[39], where a doubling of sample size from 100 to 200 approximately corresponded to halving the uncertainty in performance[35]. However, this decrease in variability is often accompanied by a decrease in out of sample performance in practice[40]. A meta-analysis across 69 studies showed that increasing sample size was negatively related to out-of-sample performance[41], where accuracy was noted to decline by approximately 5% in a similar doubling from 100 to 200 samples, suggesting that a major contribution of increasing sample size in neuroimaging is a reduction in overfitting which must occur prior to a possible boost in performance. Our finding that MCA-based dataset augmentation reduced overfitting and improved upon baseline performance is encouraging, and suggests that models trained using such perturbed datasets may benefit more from increased data collection.

A common issue in many machine learning contexts is the unbalanced nature of datasets. When using a nearest-neighbour classifier, for instance, a dramatic difference in the membership of each group could have significant impact on model hyper-parameters and performance. In contexts where balanced sampling is not possible, such as when considering a rare clinical population, perturbation-augmented datasets could be applied for realistic upsampling of data. In this case, a mega-analytic aggregation strategy could be used in which more simulations would be performed for members of the under-represented class, similar to the balancing of weights applicable to some models. This application is particularly important, as upsampling is often challenging in biological contexts where realistic simulation models are sparse.

Ch.IV - 5 Conclusion

This work demonstrates the benefit of augmenting datasets through numerical perturbations. We present an approach which leverages the numerical instability inherent to pipelines for creating more accurate and generalizable classifiers. While the approach and results demonstrated here were specifically relevant in the context of brain imaging, the data-agnostic method for inducing perturbations and off-the-shelf machine learning techniques used suggest that this approach may be widely applicable across domains. This work uniquely shows that numerical uncertainty is an asset which can be harnessed to increase the signal captured from datasets and lead to more robust learned relationships.

Data & Code Availability

The perturbed connectomes were publicly available in a data resource previously produced and made available by the authors[20]. They can be found persistently at https://doi.org/10.5281/zenodo.4041549, and are made available through The Canadian Open Neuroscience Platform (https://portal.conp.ca/search, search term "Kiar"). All software developed for processing or evaluation is publicly available on GitHub at https://github.com/gkpapers/2020AggregateMCA. Experiments were launched on Compute Canada's HPC cluster environment.

Author Contributions

GK was responsible for the experimental design, data processing, analysis, interpretation, and the majority of writing. All authors contributed to the revision of the manuscript. TG and ACE contributed to experimental design, analysis, interpretation. The authors declare no competing interests for this work. Correspondence and requests for materials should be addressed to Gregory Kiar at gregory.kiar@mail.mcgill.ca.

Acknowledgments

This research was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) (award no. CGSD3-519497-2018). This work was also supported in part by funding provided by Brain Canada, in partnership with Health Canada, for the Canadian Open Neuroscience Platform initiative.

References

[1] C.-W. Woo, L. J. Chang, M. A. Lindquist, and T. D. Wager, "Building better biomarkers: brain models in translational neuroimaging," *Nature neuroscience*, vol. 20, no. 3, p. 365, 2017.

[2] N. A. Crossley, A. Mechelli, J. Scott, F. Carletti, P. T. Fox, P. McGuire, and E. T. Bullmore, "The hubs of the human connectome are generally implicated in the anatomy of brain disorders," *Brain*, vol. 137, no. Pt 8, pp. 2382–2395, Aug. 2014.

[3] S. Payabvash, E. M. Palacios, J. P. Owen, M. B. Wang, T. Tavassoli, M. Gerdes, A. Brandes-Aitken, D. Cuneo, E. J. Marco, and P. Mukherjee, "White matter connectome edge density in children with autism spectrum disorders: Potential imaging biomarkers using Machine-Learning models," *Brain Connect.*, vol. 9, no. 2, pp. 209–220, Mar. 2019.

[4] E. Tolan and Z. Isik, "Graph theory based classification of brain connectivity network for autism spectrum disorder," in *Bioinformatics and Biomedical Engineering*. Springer International Publishing, 2018, pp. 520–530.

[5] M. Zhang, C. Desrosiers, Y. Guo, B. Khundrakpam, N. Al-Sharif, G. Kiar, P. Valdes-Sosa, J.-B. Poline, and A. Evans, "Brain status modeling with non-negative projective dictionary learning," *Neuroimage*, p. 116226, Oct. 2019.

[6] X. Zhu, X. Du, M. Kerich, F. W. Lohoff, and R. Momenan, "Random forest based classification of alcohol dependence patients and healthy controls using resting state MRI," *Neurosci. Lett.*, vol. 676, pp. 27–33, May 2018.

[7] B.-Y. Park, J. Seo, J. Yi, and H. Park, "Structural and functional brain connectivity of people with obesity and prediction of body mass index using connectivity," *PLoS One*, vol. 10, no. 11, p. e0141376, Nov. 2015.

[8] B. S. Wade, S. H. Joshi, B. A. Gutman, and P. M. Thompson, "Machine learning on high dimensional shape data from subcortical brain surfaces: A comparison of feature selection and classification methods," *Pattern Recognition*, vol. 63, pp. 731–739, 2017.

[9] S. Weis, K. R. Patil, F. Hoffstaedter, A. Nostro, B. T. Yeo, and S. B. Eickhoff, "Sex classification by resting state brain connectivity," *Cerebral cortex*, vol. 30, no. 2, pp. 824–835, 2020.

[10] B. C. Munsell, C.-Y. Wee, S. S. Keller, B. Weber, C. Elger, L. A. T. da Silva, T. Nesland, M. Styner, D. Shen, and L. Bonilha, "Evaluation of machine learning algorithms for treatment outcome prediction in patients with epilepsy based on structural connectome data," *Neuroimage*, vol. 118, pp. 219–230, 2015.

[11] R. Botvinik-Nezer, F. Holzmeister, C. F. Camerer, A. Dreber, J. Huber, M. Johannesson, M. Kirchler, R. Iwanir, J. A. Mumford, R. A. Adcock *et al.*, "Variability in the analysis of a single neuroimaging dataset by many teams," *Nature*, pp. 1–7, 2020.

[12] A. Bowring, C. Maumet, and T. E. Nichols, "Exploring the impact of analysis software on task fMRI results," *Human brain mapping*, vol. 40, no. 11, pp. 3362–3384, 2019.

[13] A. Salari, G. Kiar, L. Lewis, A. C. Evans, and T. Glatard, "File-based localization of numerical perturbations in data analysis pipelines," *GigaScience*, vol. 9, no. 12, 12 2020, giaa106.

[14] G. Kiar, Y. Chatelain, P. de Oliveira Castro, E. Petit, and others, "Numerical instabilities in analytical pipelines lead to large and meaningful variability in brain networks," *bioRxiv*, 2020.

[15] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray *et al.*, "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age," *Plos med*, vol. 12, no. 3, p. e1001779, 2015.

[16] X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. C. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos *et al.*, "An open science resource for establishing reliability and reproducibility in functional connectomics," *Scientific data*, vol. 1, no. 1, pp. 1–13, 2014.

[17] D. S. Parker, *Monte Carlo Arithmetic: exploiting randomness in floating-point arithmetic*. University of California (Los Angeles). Computer Science Department, 1997.

[18] C. Denis, P. de Oliveira Castro, and E. Petit, "Verificarlo: Checking floating point accuracy through Monte Carlo Arithmetic," *2016 IEEE 23nd Symposium on Computer Arithmetic (ARITH)*, 2016.

[19] G. Kiar, P. de Oliveira Castro, P. Rioux, E. Petit, S. T. Brown, A. C. Evans, and T. Glatard, "Comparing perturbation models for evaluating stability of neuroimaging pipelines," *The International Journal of High Performance Computing Applications*, 2020.

[20] G. Kiar, "Numerically perturbed structural connectomes from 100 individuals in the NKI Rockland dataset," Apr. 2020.

[21] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. van der Walt, M. Descoteaux, I. Nimmo-Smith, and Dipy Contributors, "Dipy, a library for the analysis of diffusion MRI data," *Front. Neuroinform.*, vol. 8, p. 8, Feb. 2014.

[22] S. I. Dimitriadis, M. Drakesmith, S. Bells, G. D. Parker, D. E. Linden, and D. K. Jones, "Improving the reliability of network metrics in structural brain networks by integrating different network weighting strategies into a single graph," 2017.

[23] C. R. Buchanan, C. R. Pernet, K. J. Gorgolewski, A. J. Storkey, and M. E. Bastin, "Test–retest reliability of structural brain networks from diffusion MRI," *Neuroimage*, vol. 86, pp. 231–243, Feb. 2014.

[24] T. B. Meier, A. S. Desphande, S. Vergun, V. A. Nair, J. Song, B. B. Biswal, M. E. Meyerand, R. M. Birn, and V. Prabhakaran, "Support vector machine classification and characterization of age-related reorganization of functional brain networks," *Neuroimage*, vol. 60, no. 1, pp. 601–613, Mar. 2012.

[25] K. Wu, Y. Taki, K. Sato, S. Kinomura, R. Goto, K. Okada, R. Kawashima, Y. He, A. C. Evans, and H. Fukuda, "Age-related changes in topological organization of structural brain networks in healthy individuals," *Hum. Brain Mapp.*, vol. 33, no. 3, pp. 552–568, Mar. 2012.

[26] S. Y. Bookheimer, D. H. Salat, M. Terpstra, B. M. Ances, D. M. Barch, R. L. Buckner, G. C. Burgess, S. W. Curtiss, M. Diaz-Santos, J. S. Elam, B. Fischl, D. N. Greve, H. A. Hagy, M. P. Harms, O. M. Hatch, T. Hedden, C. Hodge, K. C. Japardi, T. P. Kuhn, T. K. Ly, S. M. Smith, L. H. Somerville, K. Uğurbil, A. van der Kouwe, D. Van Essen, R. P. Woods, and E. Yacoub, "The lifespan Human Connectome Project in aging: An overview," *Neuroimage*, vol. 185, pp. 335–348, Jan. 2019.

[27] T. Zhao, M. Cao, H. Niu, X.-N. Zuo, A. Evans, Y. He, Q. Dong, and N. Shu, "Age-related changes in the topological organization of the white matter structural connectome across the human lifespan," *Hum. Brain Mapp.*, vol. 36, no. 10, pp. 3777–3792, 2015.

[28] A. Klein and J. Tourville, "101 labeled brain images and a consistent human cortical labeling protocol," *Front. Neurosci.*, vol. 6, p. 171, Dec. 2012.

[29] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and Others, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[31] Y. Freund and R. E. Schapire, "A Decision-Theoretic generalization of On-Line learning and an application to boosting," *J. Comput. System Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[32] B. Tunç, B. Solmaz, D. Parker, T. D. Satterthwaite, M. A. Elliott, M. E. Calkins, K. Ruparel, R. E. Gur, R. C. Gur, and R. Verma, "Establishing a link between sex-related differences in the structural connectome and behaviour," *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, vol. 371, no. 1688, p. 20150111, Feb. 2016.

[33] D. R. Nayak, R. Dash, and B. Majhi, "Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with Random Forests," *Neurocomputing*, vol. 177, pp. 188–197, Feb. 2016.

[34] R. F. Betzel, A. Griffa, P. Hagmann, and B. Misic, "Distance-dependent consistency thresholds for generating group-representative structural brain networks," *bioRxiv*, 2018.

[35] G. Varoquaux, "Cross-validation failure: small sample sizes lead to large error bars," *Neuroimage*, vol. 180, pp. 68–77, 2018.

[36] G. Vandewalle, S. N. Archer, C. Wuillaume, E. Balteau, C. Degueldre, A. Luxen, P. Maquet, and D.-J. Dijk, "Functional magnetic resonance imaging-assessed brain responses during an executive task depend on interaction of sleep homeostasis, circadian phase, and per3 genotype," *Journal of Neuroscience*, vol. 29, no. 25, pp. 7948–7956, 2009.

[37] X. Di, M. Wolfer, S. Kühn, Z. Zhang, and B. B. Biswal, "Estimations of the weather effects on brain functions using functional mri–a cautionary tale," *bioRxiv*, p. 646695, 2019.

[38] L. Chaddock, K. I. Erickson, R. S. Prakash, J. S. Kim, M. W. Voss, M. VanPatter, M. B. Pontifex, L. B. Raine, A. Konkel, C. H. Hillman *et al.*, "A neuroimaging investigation of the association between aerobic fitness, hippocampal volume, and memory performance in preadolescent children," *Brain research*, vol. 1358, pp. 172–183, 2010.

[39] C. Chu, A.-L. Hsu, K.-H. Chou, P. Bandettini, C. Lin, A. D. N. Initiative *et al.*, "Does feature selection improve classification accuracy? impact of sample size and feature selection on classification using anatomical magnetic resonance images," *Neuroimage*, vol. 60, no. 1, pp. 59–70, 2012.

[40] H. G. Schnack and R. S. Kahn, "Detecting neuroimaging biomarkers for psychiatric disorders: sample size matters," *Frontiers in psychiatry*, vol. 7, p. 50, 2016.

[41] A. A. Pulini, W. T. Kerr, S. K. Loo, and A. Lenartowicz, "Classification accuracy of neuroimaging biomarkers in attention-deficit/hyperactivity disorder: Effects of sample size and circular analysis," *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, vol. 4, no. 2, pp. 108–120, 2019.

3 Discussion

- Comparing and contrasting community variability
- https://www.biorxiv.org/content/10.1101/2020.10.07.321083v1
- NARPS
- Expanding insights to other modalities and tool domains

4 Conclusion & Summary

words

5 References

words

References

[1] T. Glatard, G. Kiar, T. Aumentado-Armstrong, N. Beck, P. Bellec, R. Bernard, A. Bonnet, S. T. Brown, S. Camarasu-Pop, F. Cervenansky, S. Das, R. Ferreira da Silva, G. Flandin, P. Girard, K. J. Gorgolewski, C. R. G. Guttmann, V. Hayot-Sasson, P.-O. Quirion, P. Rioux, M.-É. Rousseau, and A. C. Evans, "Boutiques: a flexible framework to integrate command-line applications in computing platforms," *Gigascience*, vol. 7, no. 5, May 2018.

[2] G. Kiar, S. T. Brown, T. Glatard, and A. C. Evans, "A serverless tool for platform agnostic computational experiment management," *Front. Neuroinform.*, vol. 13, p. 12, Mar. 2019.

[3] C. Denis, P. de Oliveira Castro, and E. Petit, "Verificarlo: Checking floating point accuracy through monte carlo arithmetic," *2016 IEEE 23nd Symposium on Computer Arithmetic (ARITH)*, 2016.

[4] G. Kiar, P. de Oliveira Castro, P. Rioux, E. Petit, S. T. Brown, A. C. Evans, and T. Glatard, "Comparing perturbation models for evaluating stability of neuroimaging pipelines," *The International Journal of High Performance Computing Applications*, 2020.

[5] G. Kiar, Y. Chatelain, P. de Oliveira Castro, E. Petit, and others, "Numerical instabilities in analytical pipelines lead to large and meaningful variability in brain networks," *bioRxiv*, 2020.