Developing and Validating Open Source Tools for Advanced Neuroimaging Research

Taylor Salo
Florida International University, tsalo006@fiu.edu

Follow this and additional works at: https://digitalcommons.fiu.edu/etd

Part of the Cognitive Neuroscience Commons

Recommended Citation
https://digitalcommons.fiu.edu/etd/5010

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.
DEVELOPING AND VALIDATING OPEN SOURCE TOOLS FOR ADVANCED NEUROIMAGING RESEARCH

A dissertation submitted in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
COGNITIVE NEUROSCIENCE

by
Taylor Salo

2022
To: Dean Michael R. Heithaus  
College of Arts, Sciences, & Education  

This dissertation, written by Taylor Salo, and entitled Developing and Validating Open Source Tools for Advanced Neuroimaging Research, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Matthew Sutherland

Aaron Mattfeld

Erica Musser

Robert Laird

Angela R. Laird, Major Professor

Date of Defense: June 27, 2022

The dissertation of Taylor Salo is approved.

Dean Michael R. Heithaus  
College of Arts, Sciences, & Education

Andrés G. Gil  
Vice President for Research and Economic Development and Dean of the University Graduate School  

Florida International University, 2022
DEDICATION

I dedicate this dissertation to my mother, Jannine Salo.
ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Angela Laird, for supporting and encouraging the research interests of all of her trainees, myself included. Dr. Laird has fostered a culture of collaborative, reproducible research, both within her lab and in the broader cognitive neuroscience community.

I also must thank my colleagues within the Neuroinformatics and Brain Connectivity lab, as well as the faculty and my fellow students in FIU’s Cognitive Neuroscience program. The program has always emphasized team-based research and a friendly atmosphere, which have been invaluable throughout my graduate training.

Finally, I would like to thank all of the contributors to each of the open source libraries I have maintained; not just my fellow maintainers, but also everyone who has contributed in small ways over the years, as well as the users who have been kind enough to report bugs and provide feedback. Research software is developed over the course of years, and relies on the work of maintainers, contributors, and users. I sincerely hope that the team-based nature of this work is reflected in my dissertation.
Almost all scientific research relies on software. This is particularly true for research that uses neuroimaging technologies, such as functional magnetic resonance imaging (fMRI). These technologies generate massive amounts of data per participant, which must be processed and analyzed using specialized software. A large portion of these tools are developed by teams of researchers, rather than trained software developers. In this kind of ecosystem, where the majority of software creators are scientists, rather than trained programmers, it becomes more important than ever to rely on community-based development, which may explain why most of this software is open source. It is in the development of this kind of research-oriented, open source software that I have focused much of my graduate training, as is reflected in this dissertation.

One software package I have helped to develop and maintain is tedana, a Python library for denoising multi-echo fMRI data. In chapter 2, I describe this library in a short, published software paper.

Another library I maintain as the primary developer is NiMARE, a Python library for performing neuroimaging meta-analyses and derivative analyses, such as automated annotation and functional decoding. In chapter 3, I present NiMARE in a hybrid software paper with embedded tutorial code exhibiting the functionality
of the library. This paper is currently hosted as a Jupyter book that combines narrative content and code snippets that can be executed online.

In addition to research software development, I have focused my graduate work on performing reproducible, open fMRI research. To that end, chapter 4 is a replication and extension of a recent paper on multi-echo fMRI denoising methods Power et al. (2018a). This replication was organized as a registered report, in which the introduction and methods were submitted for peer review before the analyses were performed.

Finally, chapter 5 is a conclusion to the dissertation, in which I reflect on the work I have done and the skills I have developed throughout my training.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. TE-dependent analysis of multi-echo fMRI with tedana</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Summary</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Statement of Need</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Figures</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Acknowledgements</td>
<td>10</td>
</tr>
<tr>
<td>3.1 Summary</td>
<td>11</td>
</tr>
<tr>
<td>3.2 Introduction</td>
<td>13</td>
</tr>
<tr>
<td>3.3 NiMARE Overview</td>
<td>14</td>
</tr>
<tr>
<td>3.3.1 Application Programming Interface</td>
<td>15</td>
</tr>
<tr>
<td>3.3.2 Package Organization</td>
<td>16</td>
</tr>
<tr>
<td>3.3.3 Dependencies</td>
<td>19</td>
</tr>
<tr>
<td>3.4 Download the Data</td>
<td>19</td>
</tr>
<tr>
<td>3.5 External Meta-Analytic Resources</td>
<td>19</td>
</tr>
<tr>
<td>3.5.1 BrainMap</td>
<td>20</td>
</tr>
<tr>
<td>3.5.2 Neurosynth</td>
<td>22</td>
</tr>
<tr>
<td>3.5.3 NeuroQuery</td>
<td>25</td>
</tr>
<tr>
<td>3.5.4 NeuroVault</td>
<td>27</td>
</tr>
<tr>
<td>3.6 Coordinate-Based Meta-Analysis</td>
<td>27</td>
</tr>
<tr>
<td>3.6.1 CBMA kernels</td>
<td>29</td>
</tr>
<tr>
<td>3.6.2 Multilevel kernel density analysis</td>
<td>32</td>
</tr>
<tr>
<td>3.6.3 Kernel density analysis</td>
<td>37</td>
</tr>
<tr>
<td>3.6.4 Activation likelihood estimation</td>
<td>37</td>
</tr>
<tr>
<td>3.6.5 Specific coactivation likelihood estimation</td>
<td>38</td>
</tr>
<tr>
<td>3.6.6 MKDA Chi-Squared Analysis</td>
<td>39</td>
</tr>
<tr>
<td>3.6.7 Comparing algorithms</td>
<td>39</td>
</tr>
<tr>
<td>3.7 Image-Based Meta-Analysis</td>
<td>41</td>
</tr>
<tr>
<td>3.7.1 Transforming images</td>
<td>44</td>
</tr>
<tr>
<td>3.7.2 Comparing algorithms</td>
<td>46</td>
</tr>
<tr>
<td>3.8 Multiple Comparisons Correction</td>
<td>48</td>
</tr>
<tr>
<td>3.9 Derivative Analyses</td>
<td>51</td>
</tr>
<tr>
<td>3.10 Meta-Analytic Subtraction Analysis</td>
<td>52</td>
</tr>
<tr>
<td>3.11 Meta-Analytic Coactivation Modeling</td>
<td>54</td>
</tr>
<tr>
<td>3.12 Automated Annotation</td>
<td>58</td>
</tr>
<tr>
<td>3.12.1 N-gram term extraction</td>
<td>59</td>
</tr>
<tr>
<td>3.12.2 Cognitive Atlas term extraction and hierarchical expansion</td>
<td>60</td>
</tr>
<tr>
<td>3.12.3 Latent Dirichlet allocation</td>
<td>63</td>
</tr>
</tbody>
</table>
3.12.4 Generalized correspondence latent Dirichlet allocation .................. 66
3.13 Meta-Analytic Functional Decoding ............................................. 69
3.13.1 Decoding continuous inputs .................................................... 70
3.13.2 Decoding discrete inputs ....................................................... 73
3.14 Future Directions ................................................................. 80
3.14.1 Integration with external databases .......................................... 80
3.14.2 Seed-based D-Mapping ........................................................... 81
3.14.3 Model-based CBMA ............................................................... 82
3.14.4 Additional automated annotation methods ................................. 82
3.15 Summary ................................................................................. 83
3.16 Acknowledgements .................................................................... 84

4.1 Summary ..................................................................................... 85
4.2 Introduction ................................................................................. 85
4.3 Experiment 1: Independent Replication of Respiration Analyses .......... 92
4.3.1 Materials & Methods ............................................................... 92
4.3.2 Results .................................................................................... 114
4.3.3 Discussion ............................................................................. 133
4.4 Experiment 2: Replication and Extension of Methods for Removing Brain-Wide Signals ................................................................. 137
4.4.1 Materials & Methods ............................................................... 137
4.4.2 Results .................................................................................... 149
4.4.3 Discussion ............................................................................. 162
4.5 Limitations and Future Directions .................................................. 165
4.5.1 Limitations to the original paper .............................................. 165
4.5.2 Limitations to this replication .................................................. 166
4.5.3 Deviations from the original experiments ................................... 167
4.6 Code availability ........................................................................... 169
4.7 Author Contributions .................................................................... 169

5. Conclusion .................................................................................... 171
A. Appendices .................................................................................. 188
VITA ................................................................................................. 202
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Summaries of modules in NiMARE.</td>
<td>18</td>
</tr>
<tr>
<td>3.2 Alternate forms in the Cognitive Atlas</td>
<td>61</td>
</tr>
<tr>
<td>3.3 Results from an LDA topic model</td>
<td>65</td>
</tr>
<tr>
<td>3.4 Results from a GCLDA topic model.</td>
<td>67</td>
</tr>
<tr>
<td>3.5 Results of correlation-based decoding.</td>
<td>73</td>
</tr>
<tr>
<td>3.6 Results of ROI association-based decoding.</td>
<td>75</td>
</tr>
<tr>
<td>3.7 BrainMap chi-squared decoding results.</td>
<td>77</td>
</tr>
<tr>
<td>3.8 Neurosynth chi-squared decoding results.</td>
<td>79</td>
</tr>
<tr>
<td>4.1 Common abbreviations</td>
<td>91</td>
</tr>
<tr>
<td>4.2 Results from experiment 1 power analyses.</td>
<td>93</td>
</tr>
<tr>
<td>4.3 Demographics for samples.</td>
<td>95</td>
</tr>
<tr>
<td>4.4 Scan parameters from original and replication datasets.</td>
<td>95</td>
</tr>
<tr>
<td>4.5 Experiment 1, Analysis Group 3.</td>
<td>119</td>
</tr>
<tr>
<td>4.6 Experiment 1, Analysis Group 4.</td>
<td>121</td>
</tr>
<tr>
<td>4.7 Experiment 1, Analysis Group 5.</td>
<td>129</td>
</tr>
<tr>
<td>4.8 Results from experiment 2 power analyses.</td>
<td>140</td>
</tr>
<tr>
<td>4.9 Reconstruction of distance-dependence analyses from the original publication</td>
<td>158</td>
</tr>
<tr>
<td>4.10 Results of Experiment 2, Analysis Group 5</td>
<td>160</td>
</tr>
<tr>
<td>4.11 Distance-dependence analyses of the Cambridge dataset</td>
<td>162</td>
</tr>
<tr>
<td>A.1 Distance-dependence analyses of the CamCAN dataset</td>
<td>196</td>
</tr>
<tr>
<td>A.2 Distance-dependence analyses of the DuPre dataset</td>
<td>197</td>
</tr>
<tr>
<td>A.3 Distance-dependence analyses of the all three datasets, with a regression.</td>
<td>198</td>
</tr>
<tr>
<td>A.4 Distance-dependence analyses of the Cambridge dataset, with a regression</td>
<td>199</td>
</tr>
<tr>
<td>A.5 Distance-dependence analyses of the CamCAN dataset, with a regression</td>
<td>200</td>
</tr>
</tbody>
</table>
A.6 Distance-dependence analyses of the DuPre dataset, with a regression
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 An interactive report generated by <em>tedana</em>.</td>
<td>9</td>
</tr>
<tr>
<td>3.1 A graphical representation of tools and methods implemented in NiMARE.</td>
<td>12</td>
</tr>
<tr>
<td>3.2 A schematic figure of <em>Datasets</em>, <em>Estimators</em>, <em>Transformers</em>, and <em>MetaResults</em> in NiMARE.</td>
<td>16</td>
</tr>
<tr>
<td>3.3 A flowchart of the typical workflow for coordinate-based meta-analyses in NiMARE.</td>
<td>29</td>
</tr>
<tr>
<td>3.4 Modeled activation maps produced by kernel transformation.</td>
<td>31</td>
</tr>
<tr>
<td>3.5 Thresholded results from MKDA Density, KDA, ALE, and SCALE meta-analyses.</td>
<td>41</td>
</tr>
<tr>
<td>3.6 Image-based meta-analysis results</td>
<td>48</td>
</tr>
<tr>
<td>3.7 Multiple comparisons-corrected MKDA results.</td>
<td>51</td>
</tr>
<tr>
<td>3.8 Subtraction analysis results.</td>
<td>54</td>
</tr>
<tr>
<td>3.9 Region of interest masks for MACMs.</td>
<td>56</td>
</tr>
<tr>
<td>3.10 Results from MACMs.</td>
<td>58</td>
</tr>
<tr>
<td>3.11 The effect of hierarchical expansion on Cognitive Atlas term counts from abstracts in Neurosynth’s first 500 papers.</td>
<td>63</td>
</tr>
<tr>
<td>3.12 Topic weight maps for the first ten topics in the GCLDA model.</td>
<td>69</td>
</tr>
<tr>
<td>3.13 The unthresholoded statistical map that will be used for continuous decoding.</td>
<td>71</td>
</tr>
<tr>
<td>3.14 The amygdala region of interest mask that will be used for discrete decoding.</td>
<td>74</td>
</tr>
<tr>
<td>4.1 A schematic of the analyses performed in this replication, along with the dataset(s) used for each.</td>
<td>90</td>
</tr>
<tr>
<td>4.2 Flowchart of basic processing steps used to generate target datasets for Aim 1 analyses.</td>
<td>99</td>
</tr>
<tr>
<td>4.3 Experiment 1, Analysis Group 1, Part A.</td>
<td>114</td>
</tr>
<tr>
<td>4.4 Experiment 1, Analysis Group 1, Part B.</td>
<td>115</td>
</tr>
<tr>
<td>4.5 Experiment 1, Analysis Group 2.</td>
<td>117</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

In this dissertation, I consolidate results from a wide range of projects on which I have worked throughout my graduate career. These projects vary in domain and design, but all reflect my scientific interests: transparency, reproducibility, interactivity, and tool development.

My field, cognitive neuroscience, is much like most other modern scientific areas; it depends heavily on software. Analyzing functional magnetic resonance imaging (fMRI) data, in particular, requires specialized software throughout the scientific process, from experiment inception to manuscript preparation. Additionally, even the code used to call this specialized software is necessary to reproduce a given study- in addition to the study’s data, of course. As with many other fields, cognitive neuroscience has struggled with reproducibility- an issue which has come to the forefront in recent years. In order to improve reproducibility within the field, researchers need access to data, study-specific code, and general-purpose software. It is in this intersection between reproducibility and software development where I have chosen to focus my work.

My research can broadly be divided into two tracts: reproducible neuroimaging research and development of software for reproducible neuroimaging research. In terms of performing my own reproducible research, I wish to conduct science in a manner which allows others to build directly from my work. To me, this means that other scientists must be able to use my data, my code, and my findings to replicate or extend my experiments. My other interest is in developing tools by which other scientists can conduct research in a reproducible and open manner. To this end, I have contributed to a number of open projects, and am a core contributor or maintainer on several others.
Most fMRI researchers are not software developers. In fact, even the software developers who work on fMRI-related tools are not typically computer scientists. They are often fMRI researchers who learn to program in order to do better science. On the one hand, this means that these software developers (myself included) have a lot of domain knowledge about the tools we need to develop. On the other hand, we often have to learn software development skills by actively contributing to projects (i.e., as we go along, rather than as predeveloped skills). As such, it is especially important that we ensure our code is openly accessible and that we employ best practices in developing our software, so that users can help identify bugs or improve performance as needed.

In addition to a lack of formal training, research software often suffers from a lack of funding. Funding agencies often fail to fund the development and, perhaps more importantly, maintenance of scientific software. As a result, software developed by individuals (especially graduate students) is often abandoned after the individuals leave their current lab, or academia as a whole. It is thus crucial to develop strong communities around software, so that the burden of maintenance can be spread across many contributors. Building and participating in such teams has been an important part of my work.

Working on the software and standards that facilitate reproducible research has become my primary interest over the course of my graduate education, and the chapters of my dissertation reflect this. The skills I have developed include collaborative, open source software development, community-building, and expertise with fMRI data preprocessing, analysis, and meta-analysis. The work I have done reflects my training to become a ”research software engineer” Baxter et al. (2012), specifically focused on neuroimaging research.
The second chapter in this dissertation is a brief software paper, recently published in the Journal of Open Source Software, describing the tedana library; a Python package that I have been helping to maintain and develop for the past four years. Scientific software is often undercited, because it is typically used "under the hood" in research, and there are rarely references available to cite in publications. Software papers, like the tedana paper, briefly describe the software that has been implemented and are peer reviewed, much like "standard" research articles, but the focus of the peer review process is on the actual software and its documentation, rather than the associated manuscript. This type of paper provides an invaluable mechanism by which scientists who contribute to open source software can receive recognition for their work, in the form of citations.

In the third chapter, I present NiMARE, an open source library for neuroimaging meta-analysis for which I have been the primary maintainer since its inception, in 2018. Much like the first chapter, this manuscript is a type of software paper; however, unlike the first chapter, this chapter is formatted as an interactive tutorial, built with the Jupyter book tool. The result, which has been published as a preprint by the NeuroLibre service and submitted to the Aperture journal, is a browseable, online resource containing both narrative text and code which can be executed online, allowing users to directly execute examples within the paper, as well as manipulate and change the examples for their own purposes. This chapter acts as both a teaching tool for neuroimaging meta-analysis and a demonstration of NiMARE’s capabilities for prospective users.

The fourth chapter is a replication and extension of an impactful study on the appropriate denoising strategy for multi-echo fMRI data— the type of data that can be denoised with tedana. In this replication, I was able to pursue another of my research interests: open and reproducible science. This paper was prepared as
a registered report, a type of manuscript in which the introduction and methods sections are peer reviewed and accepted in principle before data are analyzed, in the hopes of combatting unintentional questionable research practices on the part of the experimenter and publication bias on the part of the journal. Additionally, this paper uses publicly available datasets, which ensures that replication will be possible by independent researchers in the future.
CHAPTER 2
TE-DEPENDENT ANALYSIS OF MULTI-ECHO FMRI WITH TEDANA

2.1 Summary

Functional magnetic resonance imaging (fMRI) is a popular method for in vivo neuroimaging. Modern fMRI sequences are often weighted towards the blood oxygen level dependent (BOLD) signal, which is closely linked to neuronal activity (Logothetis, 2002). This weighting is achieved by tuning several parameters to increase the BOLD-weighted signal contrast. One such parameter is “TE,” or echo time. TE is the amount of time elapsed between when protons are excited (the MRI signal source) and measured. Although the total measured signal magnitude decays with echo time, BOLD sensitivity increases (Silvennoinen et al., 2003). The optimal TE maximizes the BOLD signal weighting based on a number of factors, including several MRI scanner parameters (e.g., field strength), imaged tissue composition (e.g., grey vs. white matter), and proximity to air-tissue boundaries.

Even as optimal TE values vary by brain region, most whole-brain fMRI scans are “single-echo,” where signal is collected at a fixed TE everywhere in the brain. This TE value is often based on either a value that is best on average across all brain regions or an optimised value for a specific region of interest (Stöcker et al., 2006; Peters et al., 2007). Generally, these choices reflect a tradeoff between BOLD weighting, overall signal-to-noise ratio (SNR), and signal loss due to magnetic susceptibility artifacts. Further, for any TE with BOLD signal there is also susceptibility to contamination from noise sources including head motion, respiration, and cardiac pulsation (Chang and Glover, 2009; Power et al., 2018b; Murphy et al., 2013; Caballero-Gaudes and Reynolds, 2017).
Rather than collect data at a single TE, an alternative approach is to collect multiple TEs (that is, multiple echos) for each time point. This approach, also known as multi-echo fMRI, has several benefits, including allowing researchers to estimate each voxel’s $T_2^*$ value, combining echos (Posse et al., 1999), recovering signal in regions typically not sampled at longer echo times (Kundu et al., 2013a), and improving activation and connectivity mapping (Gonzalez-Castillo et al., 2016; Caballero-Gaudes et al., 2019; Lynch et al., 2020) even in real time fMRI (Heunis et al., 2020a). In addition, artifactual non-$T_2^*$ changes (known as $S_0$ in this context) may be identified and removed by leveraging the relationship between BOLD contrast and $T_2^*$ obtained with multi-echo fMRI (Kundu et al., 2012a). Strategies to perform this efficiently and robustly are in active development.

Continuing these efforts, we present *tedana* (TE-Dependent ANAlysis) as an open-source Python package for processing and denoising multi-echo fMRI data. *tedana* implements two approaches to multi-echo preprocessing: (1) estimating a $T_2^*$ map and using these values to generate a weighted sum of individual echos, and (2) using echo-time dependent information in analysis and denoising (Kundu et al., 2012a).

### 2.2 Statement of Need

To date, multi-echo fMRI has not been widely adopted within the neuroimaging community. This is likely due to two constraints: (1) until recently, the lack of available multi-echo fMRI acquisition protocols, and (2) the lack of software for processing multi-echo fMRI data in a way that integrates with existing platforms, such as AFNI (Cox, 1996), SPM (Penny et al., 2011), FSL (Jenkinson et al., 2012), and fMRIPrep (Esteban et al., 2020).
tedana helps to address these gaps both as a software tool and as a community of practice. We have tightly scoped tedana processing to focus on those portions of the fMRI analysis workflow which are multi-echo specific in order to maximize their compatibility with other community tools. The primary interfaces for users are (1) a t2smap workflow, which estimates voxel-wise T2* and S0 and combines data across echos to increase temporal SNR, and (2) a full tedana workflow, which performs the same steps as the t2smap workflow and additionally performs ICA-based denoising to remove components exhibiting noise-like signal decay patterns across echos (Kundu et al., 2012a). The tedana workflow additionally generates interactive HTML reports through which users may visually inspect their denoising results and evaluate each component’s classification. An example report is presented in 2.1.

The limited focus and modularity of each workflow allows for easy integration into existing fMRI processing platforms. Individual modules also allow researchers to flexibly perform T2*/S0 estimation, combination across echos, decomposition with PCA or ICA, and component selection outside of a specific workflow call. As a community of practice, tedana serves as a resource for researchers looking to learn more about multi-echo fMRI, from theory to collection to analysis. To specifically increase the availability of multi-echo protocols, tedana’s documentation (available at https://tedana.readthedocs.io) consolidates acquisition guidelines for multi-echo sequences across a variety of field strengths and scanner vendors, as well as general recommendations for balancing relevant trade-offs in fMRI acquisition parameter choices. It further serves to consolidate community knowledge, including guides explaining the underlying principles of multi-echo fMRI and information on publicly available multi-echo datasets and general recommendations for balancing relevant trade-offs in sequence development.
Although *tedana* is still in alpha release, it has already been incorporated into fMRIPrep and is supported by AFNI. *tedana* has additionally been used in a number of publications and conference presentations (Lynch et al., 2020; Moia et al., 2021, 2020; Asyraff et al., 2020; Cohen et al., 2021). We further hope that *tedana* will serve as a testing bed for new multi-echo related methods. To this end, we have developed a detailed contributing process and explicit project governance to encourage a healthy community and encourage other multi-echo research groups to contribute.

*tedana* is installable via PyPi (**pip install tedana**) and contains extensive documentation (https://tedana.readthedocs.io) to orient researchers to multi-echo fMRI acquisition and processing.
2.3 Figures

Figure 2.1 An interactive report generated by tedana. Example reports can be accessed at: https://me-ica.github.io/tedana-ohbm-2020/
2.4 Acknowledgements

We would like to thank the Mozilla Open Leaders program, and the NIMH intramural research program, including the Section on Functional Imaging Methods and the Statistical and Scientific Computing Core, which have all provided funding or resources for tedana development.

Funding for ARL, KLB, and TS was provided by NIH R01-DA041353 and NIH U01-DA041156. Funding for KJW was provided through The Alan Turing Institute under the EPSRC grant EP/N510129/1.
3.1 Summary

We present NiMARE (Neuroimaging Meta-Analysis Research Environment), a Python library for neuroimaging meta-analyses and meta-analysis-related analyses. NiMARE is an open source, collaboratively-developed package that implements a range of meta-analytic algorithms, including coordinate- and image-based meta-analyses, automated annotation, functional decoding, and meta-analytic coactivation modeling. By consolidating meta-analytic methods under a common library and syntax, NiMARE makes it straightforward for users to employ the appropriate approach for a given analysis. In this paper, we describe NiMARE’s architecture and the methods implemented in the library. Additionally, we provide example code and results for each of the available tools in the library.
Figure 3.1 A graphical representation of tools and methods implemented in NiMARE. This diagram outlines six of the most common use-cases for NiMARE. (A) Coordinate-based meta-analysis (CBMA) is performed by creating a NiMARE Dataset with coordinate information stored in the Dataset.coordinates attribute, which is then used in a CBMA Estimator. This produces a MetaResult object with statistical maps, which can then be used in a Corrector object for multiple comparisons correction. Once the Corrector has been fitted, it will produce a corrected version of the MetaResult object, containing updated statistical maps. (B) Image-based meta-analysis (IBMA) operates similarly to CBMA, except that IBMA Estimators use statistical maps stored in the Dataset.images attribute. (C) Meta-analytic coactivation modeling (MACM) uses a region of interest to select coordinate-based studies within a Dataset, after which the standard CBMA workflow is performed. (D) Automated annotation infers labels from textual (and sometimes other) data associated with the Dataset, as stored in the Dataset.texts attribute. The annotation functions produce labels which may be integrated into the Dataset as the Dataset.annotations attribute. (E) Functional decoding of continuous statistical maps operates similarly to discrete decoding, in that the input Dataset must have both coordinates and annotations attributes. The Dataset, along with an unthresholded statistical map to decode, is provided to the Decoder object, which then outputs measures of similarity or associativeness with each label. (F) Functional decoding of discrete inputs applies a selection criterion to a Dataset with both coordinates and annotations attributes, using a Decoder object. The decoding algorithm will output measures of similarity or associativeness with each label in the annotations.
3.2 Introduction

We introduce NiMARE (Neuroimaging Meta-Analysis Research Environment), a Python package for analyzing meta-analytic neuroimaging data. NiMARE is a new library developed as a component in a burgeoning open-source meta-analytic ecosystem for neuroimaging data, which currently includes Neurosynth, NeuroVault, NeuroQuery, and PyMARE.

While several libraries already exist for neuroimaging meta-analysis, these libraries are generally algorithm-specific, and are provided in a range of very different user interfaces, languages, and licenses. This variability may prevent meta-analysts from using the most appropriate algorithm for a given analysis. Further, having multiple meta-analysis algorithms available in one library facilitates direct comparisons of methods. With NiMARE, we consolidate meta-analytic algorithms from a range of libraries and publications, and provide a common Python syntax and well documented application program interfaces. Additionally, NiMARE is a collaboratively-developed open source package, enabling researchers to contribute new methods not included in the current version.

In this paper, we describe NiMARE's aims, architecture and the functionality it supports—including tools for database extraction, automated annotation, meta-analysis, meta-analytic coactivation modeling, and functional decoding. The text is accompanied by extensive code samples and results (also available online in the form of Python scripts; https://github.com/NBCLab/nimare-paper with additional documentation in https://github.com/neurodatascience/meta-analysis_notebook), ensuring that users can follow along interactively.
3.3 NiMARE Overview

NiMARE is designed to be modular and object-oriented, with an interface that mimics popular Python libraries, including scikit-learn and nilearn. This standardized interface allows users to employ a wide range of meta-analytic algorithms without having to familiarize themselves with the idiosyncrasies of algorithm-specific tools. This lets users use whatever method is most appropriate for a given research question with minimal mental overhead from switching methods. Additionally, NiMARE emphasizes citability, with references in the documentation and citable boilerplate text that can be copied directly into manuscripts, in order to ensure that the original algorithm developers are appropriately recognized.

NiMARE works with Python versions 3.6 and higher, and can easily be installed with pip. Its source code is housed and version controlled in a GitHub repository at https://github.com/neurostuff/NiMARE.

NiMARE is under continued active development, and we anticipate that the user-facing API (application programming interface) may change over time. Our emphasis in this paper is thus primarily on reviewing the functionality implemented in the package and illustrating the general interface, and not on providing a detailed and static user guide that will be found within the package documentation.

Tools in NiMARE are organized into several modules, including nimare.meta, nimare.correct, nimare.annotate, nimare.decode, and nimare.workflows. In addition to these primary modules, there are several secondary modules for data wrangling and internal helper functions, including nimare.io, nimare.dataset, nimare.extract, nimare.stats, nimare.utils, and nimare.base. These modules are summarized in Section 3.3.1, as well as in Table 3.1.
3.3.1 Application Programming Interface

One of the principal goals of NiMARE is to implement a range of methods with a set of shared interfaces, to enable users to employ the most appropriate algorithm for a given question without introducing a steep learning curve. This approach is modeled on the widely-used scikit-learn package (Pedregosa et al., 2011; Buitinck et al., 2013), which implements a large number of machine learning algorithms - all with simple, consistent interfaces. Regardless of the algorithm employed, data should be in the same format and the same class methods should be called to fit and/or generate predictions from the model.

To this end, we have adopted an object-oriented approach to NiMARE’s core API that organizes tools based on the type of inputs and outputs they operate over. The key data structure is the Dataset class, which stores a range of neuroimaging data amenable to various forms of meta-analysis. There are two main types of tools that operate on a Dataset class. Transformer classes, as their name suggests, perform some transformation on a Dataset - i.e., they take a Dataset instance as input, and return a modified version of that Dataset instance as output (for example, with newly generated maps stored within the object). Estimator classes apply a meta-analytic algorithm to a Dataset and return a set of statistical images stored in a MetaResult container class. The key methods supported by each of these base classes, as well as the main arguments to those methods, are consistent throughout the hierarchy (e.g., all Transformer classes must implement a transform() method), minimizing the learning curve and ensuring a high degree of predictability for users.
3.3.2 Package Organization

At present, the package is organized into 14 distinct modules. `nimare.dataset` defines the `Dataset` class. `nimare.meta` includes `Estimators` for coordinate- and

![Figure 3.2 A schematic figure of Datasets, Estimators, Transformers, and MetaResults in NiMARE.](image-url)
image-based meta-analysis methods. `nimare.results` defines the `MetaResult` class, which stores statistical maps produced by meta-analyses. `nimare.correct` implements `Corrector` classes for family-wise error (FWE) and false discovery rate (FDR) multiple comparisons correction. `nimare.annotate` implements a range of automated annotation methods, including latent Dirichlet allocation (LDA) and generalized correspondence latent Dirichlet allocation (GCLDA). `nimare.decode` implements a number of meta-analytic functional decoding and encoding algorithms. `nimare.io` provides functions for converting alternative meta-analytic dataset structure, such as Sleuth text files or Neurosynth datasets, to NiMARE format. `nimare.transforms` implements a range of spatial and data type transformations, including a function to generate new images in the `Dataset` from existing image types. `nimare.extract` provides methods for fetching datasets and models across the internet. `nimare.generate` includes functions for generating data for internal testing and validation. `nimare.base` defines a number of base classes used throughout the rest of the package. Finally, `nimare.stats` and `nimare.utils` are modules for statistical and generic utility functions, respectively. These modules are summarized in Table 3.1.
Table 3.1 **Summaries of modules in NiMARE.**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset</td>
<td>This module stores the <code>Dataset</code> class.</td>
</tr>
<tr>
<td>meta</td>
<td>This module contains <code>Estimators</code> for image- and coordinate-based meta-analysis algorithms, as well as <code>KernelTransformers</code>, which are used in conjunction with coordinate-based methods.</td>
</tr>
<tr>
<td>results</td>
<td>This module stores the <code>MetaResult</code> class, which in turn is used to manage statistical maps produced by meta-analytic algorithms.</td>
</tr>
<tr>
<td>correct</td>
<td>This module contains classes for multiple comparisons correction, including <code>FWECorrector</code> (family-wise error rate correction) and <code>FDRCorrector</code> (false discovery rate correction).</td>
</tr>
<tr>
<td>annotate</td>
<td>This module includes a range of tools for automated annotation of studies.</td>
</tr>
<tr>
<td>decode</td>
<td>This module includes a number of methods for functional characterization analysis, also known as functional decoding.</td>
</tr>
<tr>
<td>io</td>
<td>This module contains functions for converting common file types, such as Neurosynth- or Sleuth-format files, into NiMARE-compatible formats.</td>
</tr>
<tr>
<td>transforms</td>
<td>This module contains classes and functions for converting between common data types, such as different image types.</td>
</tr>
<tr>
<td>extract</td>
<td>This module contains functions for downloading external resources, such as Neurosynth and the Cognitive Atlas.</td>
</tr>
<tr>
<td>stats</td>
<td>This module contains miscellaneous statistical methods.</td>
</tr>
<tr>
<td>generate</td>
<td>This module contains functions for generating test data.</td>
</tr>
<tr>
<td>utils</td>
<td>This module contains miscellaneous utility functions.</td>
</tr>
<tr>
<td>workflows</td>
<td>This module contains a number of common workflows that can be run from the command line. All of the workflow functions generate boilerplate text that can be included in manuscript methods sections.</td>
</tr>
<tr>
<td>base</td>
<td>This module defines a number of base classes.</td>
</tr>
</tbody>
</table>
3.3.3 Dependencies

NiMARE depends on the standard SciPy stack, as well as a small number of widely-used packages. Dependencies from the SciPy stack include `scipy` (Virtanen et al., 2020), `numpy` (Walt et al., 2011; Harris et al., 2020), `pandas` (McKinney, 2010), and `scikit-learn` (Pedregosa et al., 2011; Buitinck et al., 2013). Additional requirements include `fuzzywuzzy`, `nibabel` (Brett et al., 2020), `nilearn` (Abraham et al., 2014), `statsmodels` (Seabold and Perktold, 2010), and `tqdm` (da Costa-Luis et al., 2020).

3.4 Download the Data

```python
# First, import the necessary modules and functions
import os
from pprint import pprint

from repo2data.repo2data import Repo2Data

# Install the data if running locally, or points to cached data
# if running on neurolibre
DATA_REQ_FILE = os.path.abspath("../binder/data_requirement.json")
repo2data = Repo2Data(DATA_REQ_FILE)
data_path = repo2data.install()
data_path = os.path.join(data_path[0], "data")

# We will also create a directory in which to save files that
# are generated within the book.
out_dir = os.path.abspath("../outputs/")
os.makedirs(out_dir, exist_ok=True)
```

3.5 External Meta-Analytic Resources

Large-scale meta-analytic databases have made systematic meta-analyses of the neuroimaging literature possible. These databases combine results from neuroimaging
studies, whether represented as coordinates of peak activations or unthresholded statistical images, with important study metadata, such as information about the samples acquired, stimuli used, analyses performed, and mental constructs putatively manipulated. The two most popular coordinate-based meta-analytic databases are BrainMap and Neurosynth, while the most popular image-based database is NeuroVault.

The studies archived in these databases may be either manually or automatically annotated—often with reference to a formal ontology or controlled vocabulary. Ontologies for cognitive neuroscience define what mental states or processes are postulated to be manipulated or measured in experiments, and may also include details of said experiments (e.g., the cognitive tasks employed), relationships between concepts (e.g., verbal working memory is a kind of working memory), and various other metadata that can be standardized and represented in a machine-readable form (Poldrack and Yarkoni, 2016; Poldrack, 2010; Turner and Laird, 2012). Some of these ontologies are very well-defined, such as expert-generated taxonomies designed specifically to describe only certain aspects of experiments and the relationships between elements within the taxonomy, while others are more loosely defined, in some cases simply building a vocabulary based on which terms are commonly used in cognitive neuroscience articles.

3.5.1 BrainMap

BrainMap (Fox et al., 2005; Fox and Lancaster, 2002; Laird et al., 2005b) relies on expert annotators to label individual comparisons within studies according to its internally developed ontology, the BrainMap Taxonomy (Fox et al., 2005). While this approach is likely to be less noisy than an automated annotation method using
article text or imaging results to predict content, it is also subject to a number of limitations. First, there are simply not enough annotators to keep up with the ever-expanding literature. Second, any development of the underlying ontology has the potential to leave the database outdated. For example, if a new label is added to the BrainMap Taxonomy, then each study in the full BrainMap database needs to be evaluated for that label before that label can be properly integrated into the database. Finally, a manually annotated database like BrainMap will be biased by which subdomains within the literature are annotated. While outside contributors can add and annotate studies to the database, the main source of annotations has been researchers associated with the BrainMap project.

While BrainMap is a semi-closed resource (i.e., a collaboration agreement is required to access the full database), registered users may search the database using the Sleuth search tool, in order to collect samples for meta-analyses. Sleuth can export these study collections as text files with coordinates. NiMARE provides a function to import data from Sleuth text files into the NiMARE Dataset format.

The function `convert_sleuth_to_dataset` can be used to convert text files exported from Sleuth into NiMARE Datasets. Here, we convert two files from a previous publication by NiMARE contributors (Yanes et al., 2018) into two separate Datasets.

```python
from nimare import io

sleuth_dset1 = io.convert_sleuth_to_dataset(
    os.path.join(
        data_path,
        "contrast-CannabisMinusControl_" \
        "space-talairach_sleuth.txt",
    )
)
sleuth_dset2 = io.convert_sleuth_to_dataset(
    os.path.join(
        data_path,
        "contrast-ControlMinusCannabis_"
    )
)
```
3.5.2 Neurosynth

Neurosynth (Yarkoni et al., 2011) uses a combination of web scraping and text mining to automatically harvest neuroimaging studies from the literature and to annotate them based on term frequency within article abstracts. As a consequence of its relatively crude automated approach, Neurosynth has its own set of limitations. First, Neurosynth is unable to delineate individual comparisons within studies, and consequently uses the entire paper as its unit of measurement, unlike BrainMap. This risks conflating directly contrasted comparisons (e.g., AïB and BïA), as well as comparisons which have no relation to one another. Second, coordinate extraction and annotation are noisy. Third, annotations automatically performed by Neurosynth are also subject to error, although the reasons behind this are more nuanced and will be discussed later in this paper. Given Neurosynth’s limitations, we recommend that it be used for casual, exploratory meta-analyses rather than for publication-quality analyses. Nevertheless, while individual meta-analyses should not be published from Neurosynth, many derivative analyses have been performed and published (e.g., Chang et al. (2013); de la Vega et al. (2016, 2018); Poldrack et al. (2012)). As evidence of its utility, Neurosynth has been used to define a priori
regions of interest (e.g., Josipovic (2014); Zeidman et al. (2012); Wager et al. (2013)) or perform meta-analytic functional decoding (e.g., Chen et al. (2018); Pantelis et al. (2015); Tambini et al. (2017)) in many first-order (rather than meta-analytic) fMRI studies.

Here, we show code that would download the Neurosynth database from where it is stored (https://github.com/neurosynth/neurosynth-data) and convert it to a NiMARE Dataset using fetch_neurosynth, for the first step, and convert_neurosynth_to_dataset, for the second.

```python
from nimare import extract

# Download the desired version of Neurosynth from GitHub.
files = extract.fetch_neurosynth(
    data_dir=data_path,
    version="7",
    source="abstract",
    vocab="terms",
    overwrite=False,
)
pprint(files)
neurosynth_db = files[0]

[[
    'coordinates': 'data-neurosynth_version-7_coordinates.tsv.gz',
    'features': [{
        'features': (n
            'data-neurosynth_version-7_vocab-terms_
            'source-abstract_type-tfidf_features.npz'
        ),
        'vocabulary': (n
            'data-neurosynth_version-7_vocab-terms_
            'vocabulary.txt'
        ),
    }],
    'metadata': 'data-neurosynth_version-7_metadata.tsv.gz',
]]

# Convert the files to a Dataset.
# This may take a while (~10 minutes)
```
Many of the methods in NiMARE can be very time-consuming or memory-intensive. Therefore, for the sake of ensuring that the analyses in this article may be reproduced by as many people as possible, we will use a reduced version of the Neurosynth Dataset, only containing the first 500 studies, for those methods which may not run easily on the full database.

In addition to a large corpus of coordinates, Neurosynth provides term frequencies derived from article abstracts that can be used as annotations.

One additional benefit to Neurosynth is that it has made available the coordinates for a large number of studies for which the study abstracts are also readily available. This has made the Neurosynth database a common resource upon which
to build other automated ontologies. Data-driven ontologies which have been developed using the Neurosynth database include the generalized correspondence latent Dirichlet allocation (GCLDA) (Rubin et al., 2017) topic model and Deep Boltzmann machines (Monti et al., 2016).

### 3.5.3 NeuroQuery

A related resource is **NeuroQuery** (Dockès et al., 2020). NeuroQuery is an online service for large-scale predictive meta-analysis. Unlike Neurosynth, which performs statistical inference and produces statistical maps, NeuroQuery is a supervised learning model and produces a prediction of the brain areas most likely to contain activations. These maps predict locations where studies investigating a given area (determined by the text prompt) are likely to produce activations, but they cannot be used in the same manner as statistical maps from a standard coordinate-based meta-analysis. In addition to this predictive meta-analytic tool, NeuroQuery also provides a new database of coordinates, text annotations, and metadata via an automated extraction approach that improves on Neurosynth’s original methods.

While NiMARE does not currently include an interface to NeuroQuery’s predictive meta-analytic method, there are functions for downloading the NeuroQuery database and converting it to NiMARE format, much like Neurosynth. The function for downloading the NeuroQuery database is `fetch_neuroquery`. A related function, `convert_neurosynth_to_dataset`, converts the downloaded files to a `Dataset`. We are able to use the same function for converting the database to a `Dataset` for NeuroQuery as Neurosynth because both databases store their data in the same structure.

```python
# Download the desired version of NeuroQuery from GitHub.
files = extract.fetch_neuroquery(
```
data_dir=data_path,
version="1",
source="combined",
vocab="neuroquery6308",
type="tfidf",
overwrite=False,
)
pprint(files)
neuroquery_db = files[0]

[

    {'coordinates': 'data-neuroquery_version-1_coordinates.tsv.gz',
     'features': [{
        'features': (  
            'data-neuroquery_version-1_vocab-neuroquery6308_'
            'source-combined_type-tfidf_features.npz'
        ),
        'vocabulary': (  
            'data-neuroquery_version-1_vocab-neuroquery6308_'
            'vocabulary.txt'
        ),
    },
    ],
    'metadata': 'data-neuroquery_version-1_metadata.tsv.gz',
}
]

# Convert the files to a Dataset.
# This may take a while (~10 minutes)
neuroquery_dset = io.convert_neurosynth_to_dataset(
    coordinates_file=neuroquery_db["coordinates"],
    metadata_file=neuroquery_db["metadata"],
    annotations_files=neuroquery_db["features"],
)
print(neuroquery_dset)

# Save the Dataset for later use.
neuroquery_dset.save(
    os.path.join(out_dir, "neuroquery_dataset.pkl.gz")
)

Dataset(13459 experiments, space='mni152_2mm')
3.5.4 NeuroVault

**NeuroVault** (Gorgolewski et al., 2015) is a public repository of user-uploaded, whole-brain, unthresholded brain maps. Users may associate their image collections with publications, and can annotate individual maps with labels from the Cognitive Atlas, which is the ontology of choice for NeuroVault. NiMARE includes a function, `convert_neurovault_to_dataset`, with which users can search for images in NeuroVault, download those images, and convert them into a `Dataset` object.

3.6 Coordinate-Based Meta-Analysis

```python
import matplotlib.pyplot as plt
import numpy as np
from nilearn import plotting
from nimare import dataset

# Now, load the Datasets we will use in this chapter
sleuth_dset1 = dataset.Dataset.load(
    os.path.join(data_path, "sleuth_dset1.pkl.gz")
)
sleuth_dset2 = dataset.Dataset.load(
    os.path.join(data_path, "sleuth_dset2.pkl.gz")
)
neurosynth_dset = dataset.Dataset.load(
    os.path.join(data_path, "neurosynth_dataset.pkl.gz")
)
```

Coordinate-based meta-analysis (CBMA) is currently the most popular method for neuroimaging meta-analysis, given that the majority of fMRI papers currently report their findings as peaks of statistically significant clusters in standard space and do not release unthresholded statistical maps. These peaks indicate where significant results were found in the brain, and thus do not reflect an effect size estimate for each hypothesis test (i.e., each voxel) as one would expect for a typical meta-analysis. As such, standard methods for effect size-based meta-analysis cannot be
applied. Over the past two decades, a number of algorithms have been developed to determine whether peaks converge across experiments in order to identify locations of consistent or specific activation associated with a given hypothesis (Samartsidis et al., 2017; Müller et al., 2018).

Kernel-based methods evaluate convergence of coordinates across studies by first convolving foci with a spatial kernel to produce study-specific modeled activation maps, then combining those modeled activation maps into a sample-wise map, which is compared to a null distribution to evaluate voxel-wise statistical significance. Additionally, for each of the following approaches, except for SCALE, voxel- or cluster-level multiple comparisons correction may be performed using Monte Carlo simulations or false discovery rate (FDR) (Laird et al., 2005a) correction. Basic multiple-comparisons correction methods (e.g., Bonferroni correction) are also supported.
3.6.1 CBMA kernels

CBMA kernels are available as KernelTransformers in the nimare.meta.kernel module. There are three standard kernels that are currently available: MKDAKernel, KDAKernel, and ALEKernel. Each class may be configured with certain parameters when a new object is initialized. For example, MKDAKernel accepts an \( r \) parameter, which determines the radius of the spheres that will be created around each peak.
coordinate. **ALEKernel** automatically uses the sample size associated with each experiment in the Dataset to determine the appropriate full-width-at-half-maximum of its Gaussian distribution, as described in Eickhoff et al. (2012a); however, users may provide a constant `sample_size` or `fwhm` parameter when sample size information is not available within the Dataset metadata.

Here we show how these three kernels can be applied to the same Dataset.

```python
from nimare.meta import kernel

mkda_kernel = kernel.MKDAKernel(r=10)
mkda_ma_maps = mkda_kernel.transform(sleuth_dset1)
kda_kernel = kernel.KDAKernel(r=10)
kda_ma_maps = kda_kernel.transform(sleuth_dset1)
ale_kernel = kernel.ALEKernel(sample_size=20)
ale_ma_maps = ale_kernel.transform(sleuth_dset1)

# Generate figure
study_idx = 10  # a study with overlapping kernels
max_value = np.max(kda_ma_maps[study_idx].get_fdata()) + 1

ma_maps = {
    "MKDA Kernel": mkda_ma_maps[study_idx],
    "KDA Kernel": kda_ma_maps[study_idx],
    "ALE Kernel": ale_ma_maps[study_idx],
}

fig, axes = plt.subplots(nrows=3, figsize=(6, 6))
for i_meta, (name, img) in enumerate(ma_maps.items()):
    if "ALE" in name:
        vmax = None
    else:
        vmax = max_value
    display = plotting.plot_stat_map(img,
                                       annotate=False,
                                       axes=axes[i_meta],
                                       cmap="Reds",
                                       cut_coords=[5, 0, 29],
                                       draw_cross=False,
                                       figure=fig,
                                       vmax=vmax,
        )
```

In addition to being able to generate modeled activation images, KernelTransformers can generate an updated Dataset with the modeled activation images stored in its images attribute, as shown below.

```python
from nimare import dataset, meta
```
neurosynth_dset_first500 = dataset.Dataset.load(
    os.path.join(
        data_path, "neurosynth_dataset_first500.pkl.gz"
    )
)

# Specify where images for this Dataset should be located
target_folder = os.path.join(out_dir, "neurosynth_dataset_maps")
os.makedirs(target_folder, exist_ok=True)
neurosynth_dset_first500.update_path(target_folder)

# Initialize a kernel transformer to use
kern = meta.kernel.MKDAKernel(memory_limit="500mb")

# Run the kernel transformer with return_type set to "dataset"
# to return an updated Dataset with the MA maps stored as files
# within its "images" attribute.
neurosynth_dset_first500 = kern.transform(
    neurosynth_dset_first500, return_type="dataset"
)
neurosynth_dset_first500.save(
    os.path.join(
        out_dir, 
        "neurosynth_dataset_first500_with_mkda_ma.pkl.gz",
    ),
)

3.6.2 Multilevel kernel density analysis

Multilevel kernel density analysis (MKDA) (Wager et al., 2007) is a kernel-based method that convolves each peak from each study with a binary sphere of a set radius. These peak-specific binary maps are then combined into study-specific maps by taking the maximum value for each voxel. Study-specific maps are then averaged across the meta-analytic sample. This averaging is generally weighted by studies' sample sizes, although other covariates may be included, such as weights based on the type of inference (random or fixed effects) employed in the study's analysis. An arbitrary threshold is generally employed to zero-out voxels with very low values, and then a Monte Carlo procedure is used to assess statistical significance, either at the voxel or cluster level.
In NiMARE, the MKDA meta-analyses can be performed with the `MKDADensity` class. This class, like most other CBMA classes in NiMARE, accepts a `null_method` parameter, which determines how voxel-wise (uncorrected) statistical significance is calculated.

The `null_method` parameter allows two options: "approximate" or "montecarlo." The "approximate" option builds a histogram-based null distribution of summary-statistic values, which can then be used to determine the associated p-value for observed summary-statistic values (i.e., the values in the meta-analytic map). The "montecarlo" option builds a null distribution of summary-statistic values by randomly shuffling the coordinates the `Dataset` many times, and computing the summary-statistic values for each permutation. In general, the "montecarlo" method is slightly more accurate when there are enough permutations, while the "approximate" method is much faster.

Please note that fitting the CBMA `Estimator` to a `Dataset` will produce p-value, z-statistic, and summary-statistic maps, but these are not corrected for multiple comparisons.

When performing a meta-analysis with the goal of statistical inference, you will want to perform multiple comparisons correction with NiMARE’s `Corrector` classes. Please see the multiple comparisons correction section for more information.

Here we perform an MKDADensity meta-analysis on one of the Sleuth-based Datasets. We will use the "approximate" null method for speed.

```python
from nimare.meta.cbma import mkda
mkdad_meta = mkda.MKDADensity(null_method="approximate")
mkdad_results = mkdad_meta.fit(sleuth_dset1)
```
The MetaResult class

Fitting an Estimator to a Dataset produces a MetaResult object. The MetaResult class is a light container holding the different statistical maps produced by the Estimator.

```
print(mkdad_results)

<nimare.results.MetaResult object at 0x7f886f845890>

This result is also retained as an attribute in the Estimator.

print(mkdad_meta.results)

<nimare.results.MetaResult object at 0x7f886f845890>

The maps attribute is a dictionary containing statistical map names and associated numpy arrays.

```
pprint(mkdad_results.maps)

{'p': array([1., 1., 1., ..., 1., 1., 1.]),
 'stat': array([0., 0., 0., ..., 0., 0., 0.]),
 'z': array([0., 0., 0., ..., 0., 0., 0.])}

These arrays can be transformed into image-like objects using the masker attribute. We can also use the get_map method to get that image object.

```
mkdad_img = mkdad_results.get_map("z", return_type="image")
print(mkdad_img)
<class 'nibabel.nifti1.Nifti1Image'>
data shape (91, 109, 91)
affine:
[[ -2.  0.  0.  90.]
 [ 0.  2.  0. -126.]
 [ 0.  0.  2. -72.]
 [ 0.  0.  0.  1.]]
metadata:
<class 'nibabel.nifti1.Nifti1Header'> object, endian='<'
sizeof_hdr : 348
data_type   : b''
db_name     : b''
extents     : 0
session_error : 0
regular     : b''
dim_info    : 0
dim         : [ 3 91 109 91 1 1 1 1]
intent_p1   : 0.0
intent_p2   : 0.0
intent_p3   : 0.0
intent_code : none
datatype    : float64
bitpix      : 64
slice_start : 0
pixdim      : [-1. 2. 2. 2. 1. 1. 1. 1.]
vox_offset  : 0.0
scl_slope   : nan
scl_inter   : nan
slice_end   : 0
slice_code  : unknown
xyzt_units  : 0
cal_max     : 0.0
cal_min     : 0.0
slice_duration : 0.0
toffset    : 0.0
glmax      : 0
glmin      : 0
descrip     : b''
aux_file    : b''
qform_code  : unknown
sform_code  : aligned
quatern_b   : 0.0
We can save the statistical maps to an output directory as gzipped nifti files, with a prefix. Here, we will save all of the statistical maps with the MKDADensity prefix.

```python
mkdad_results.save_maps(
    output_dir=out_dir, prefix="MKDADensity"
)
```

We will also save the `Estimator` itself, which we will reuse when we get to multiple comparisons correction.

```python
mkdad_meta.save(os.path.join(out_dir, "MKDADensity.pkl.gz"))
```

Since this is a kernel-based algorithm, the kernel transformer is an optional input to the meta-analytic estimator, and can be controlled in a more fine-grained manner.

```python
# These two approaches (initializing the kernel ahead of time
# or providing the arguments with the kernel__ prefix) are
# equivalent.
mkda_kernel = kernel.MKDAKernel(r=2)
mkdad_meta = mkda.MKDADensity(kernel_transformer=mkda_kernel)
mkdad_meta = mkda.MKDADensity(
    kernel_transformer=kernel.MKDAKernel, kernel__r=2
)
```

# A completely different kernel could even be provided,
# although this is not recommended and should only be used for
3.6.3 Kernel density analysis

Kernel density analysis (KDA) (Wager et al., 2003, 2004) is a precursor algorithm that has been replaced in the field by MKDA. For the sake of completeness, NiMARE also includes a KDA estimator that implements the older KDA algorithm for comparison purposes. The interface is virtually identical, but since there are few if any legitimate uses of KDA (which models studies as fixed rather than random effects), we do not discuss the algorithm further here.

```python
kda_meta = mkda.KDA(null_method="approximate")
kda_results = kda_meta.fit(sleuth_dset1)

# Retain the z-statistic map for later use
kda_img = kda_results.get_map("z", return_type="image")
```

3.6.4 Activation likelihood estimation

Activation likelihood estimation (ALE) (Eickhoff et al., 2012b; Turkeltaub et al., 2012, 2002) assesses convergence of peaks across studies by first generating a modeled activation map for each study, in which each of the experiment’s peaks is convolved with a 3D Gaussian distribution determined by the experiment’s sample size, and then by combining these modeled activation maps across studies into an ALE map, which is compared to an empirical null distribution to assess voxel-wise statistical significance.

```python
from nimare.meta.cbma import ale
```
ale_meta = ale.ALE()
ale_results = ale_meta.fit(sleuth_dset1)

# Retain the z-statistic map for later use
ale_img = ale_results.get_map("z", return_type="image")

### 3.6.5 Specific coactivation likelihood estimation

**Specific coactivation likelihood estimation** (SCALE) (Langner et al., 2014) is an extension of the ALE algorithm developed for meta-analytic coactivation modeling (MACM) analyses. Rather than comparing convergence of foci within the sample to a null distribution derived under the assumption of spatial randomness within the brain, SCALE assesses whether the convergence at each voxel is greater than in the general literature. Each voxel in the brain is assigned a null distribution determined based on the base rate of activation for that voxel across an existing coordinate-based meta-analytic database. This approach allows for the generation of a statistical map for the sample, but no methods for multiple comparisons correction have yet been developed. While this method was developed to support analysis of joint activation or "coactivation" patterns, it is generic and can be applied to any CBMA; see Section 3.9.

```
# Here we use the coordinates from Neurosynth as our measure of
# coordinate base-rates, because we do not have access to the
# full BrainMap database.
# However, one assumption of SCALE is that the Dataset being
# analyzed comes from the same source as the database you use
# for calculating base-rates.
xyz = neurosynth_dset.coordinates["x", "y", "z"][].values
scale_meta = ale.SCALE(n_iters=2500, xyz=xyz, memory_limit=None)
scale_results = scale_meta.fit(sleuth_dset1)

# Retain the z-statistic map for later use
scale_img = scale_results.get_map("z", return_type="image")
```
3.6.6 MKDA Chi-Squared Analysis

An alternative to the density-based approaches (i.e., MKDA, KDA, ALE, and SCALE) is the MKDA Chi-squared extension (Wager et al., 2007). Though still a kernel-based method in which foci are convolved with a binary sphere and combined within studies, this approach uses voxel-wise chi-squared tests to assess both consistency (i.e., higher convergence of foci within the meta-analytic sample than expected by chance) and specificity (i.e., higher convergence of foci within the meta-analytic sample than detected in an unrelated dataset) of activation. Such an analysis also requires access to a reference meta-analytic sample or database of studies. For example, to perform a chi-squared analysis of working memory studies, the researcher will also need a comprehensive set of studies which did not manipulate working memory—ideally one that is matched with the working memory study set on all relevant attributes except the involvement of working memory.

```python
mkdac_meta = mkda.MKDAChi2()
mkdac_results = mkdac_meta.fit(sleuth_dset1, sleuth_dset2)

# Retain the specificity analysis's z-statistic map for later use
mkdac_img = mkdac_results.get_map(
    "z_desc-specificity", return_type="image"
)
```

3.6.7 Comparing algorithms

Here we load the z-statistic map from each of the CBMA Estimators we’ve used throughout this chapter and plot them all side by side.

```python
meta_results = {
    "MKDA Density": mkdad_img,
    "MKDA Chi-Squared": mkdac_img,
    "KDA": kda_img,
    "ALE": ale_img,
}```
"SCALE": scale_img,
}
order = [
    ["MKDA Density", "ALE"],
    ["MKDA Chi-Squared", "SCALE"],
    ["KDA", None]
]

fig, axes = plt.subplots(figsize=(12, 6), nrows=3, ncols=2)
for i_row, row_names in enumerate(order):
    for j_col, name in enumerate(row_names):
        if not name:
            axes[i_row, j_col].axis("off")
            continue

        img = meta_results[name]
        if name == "MKDA Chi-Squared":
            cmap = "RdBu_r"
        else:
            cmap = "Reds"

        display = plotting.plot_stat_map(
            img,
            annotate=False,
            axes=axes[i_row, j_col],
            cmap=cmap,
            cut_coords=[5, -15, 10],
            draw_cross=False,
            figure=fig,
        )
        axes[i_row, j_col].set_title(name)

        colorbar = display._cbar
colorbar_ticks = colorbar.get_ticks()
        if colorbar_ticks[0] < 0:
            new_ticks = [
                colorbar_ticks[0],
                0,
                colorbar_ticks[-1],
            ]
        else:
            new_ticks = [colorbar_ticks[0], colorbar_ticks[-1]]
        colorbar.set_ticks(new_ticks, update_ticks=True)

40
A number of other coordinate-based meta-analysis algorithms exist which are not yet implemented in NiMARE. We describe these algorithms briefly in Section 3.14.

### 3.7 Image-Based Meta-Analysis

Image-based meta-analysis (IBMA) methods perform a meta-analysis directly on brain images (either whole-brain or partial) rather than on extracted peaks. On paper, IBMA is superior to CBMA in virtually all respects, as the availability of analysis-level parameter and variance estimates at all analyzed voxels allows researchers to use the full complement of standard meta-analysis techniques, instead of having to resort to kernel-based or other methods that require additional spatial assumptions. In principle, given a set of maps that contains no missing values (i.e., where there are $k$ valid pairs of parameter and variance estimates at each voxel),
one can simply conduct a voxel-wise version of any standard meta-analysis or meta-regression method commonly used in other biomedical or social science fields.

In practice, the utility of IBMA methods has historically been quite limited, as unthresholded statistical maps have been unavailable for the vast majority of neuroimaging studies. However, the introduction and rapid adoption of NeuroVault (Gorgolewski et al., 2015), a database for unthresholded statistical images, has made image-based meta-analysis increasingly viable. Although coverage of the literature remains limited, and IBMAs of maps drawn from the NeuroVault database are likely to omit at least some (and in some cases most) relevant studies due to limited meta-data, we believe the time is ripe for researchers to start including both CBMAs and IBMAs in published meta-analyses, with the aspirational goal of eventually transitioning exclusively to the latter. To this end, NiMARE supports a range of different IBMA methods, including a number of estimators of the gold standard mixed-effects meta-regression model, as well as several alternative estimators suitable for use when some of the traditional inputs are unavailable.

NiMARE’s IBMA Estimators are light wrappers around classes from PyMARE, a library for standard (i.e., non-neuroimaging) meta-analyses developed by the same team as NiMARE.

In the optimal situation, meta-analysts have access to both contrast (i.e., parameter estimate) maps and their associated standard error maps for a number of studies. With these data, researchers can fit the traditional random-effects meta-regression model using one of several methods that vary in the way they estimate the between-study variance ($\tau^2$). Currently supported estimators include the DerSimonian-Laird method (DerSimonian and Laird, 1986), the Hedges method (Hedges et al., 1985), and maximum-likelihood (ML) and restricted maximum-likelihood (REML) approaches. NiMARE can also perform fixed-effects meta-
regression via weighted least-squares, though there are few IBMA scenarios where a
fixed-effects analysis would be indicated. It is worth noting that the non-likelihood-
based estimators (i.e., DerSimonian-Laird and Hedges) have a closed-form solution,
and are implemented in an extremely efficient way in NiMARE (i.e., computation
is performed on all voxels in parallel). However, these estimators also produce more
biased estimates under typical conditions (e.g., when sample sizes are very small),
implying a tradeoff from the user’s perspective.

Alternatively, when users only have access to contrast maps and associated sam-
ple sizes, they can use the supported Sample Size-Based Likelihood estimator, which assumes that within-study variance is constant across studies, and uses
maximum-likelihood or restricted maximum-likelihood to estimate between-study
variance, as described in Sangnawakij et al. (2019). When users have access only to
contrast maps, they can use the Permutated OLS estimator, which uses ordinary
least squares and employs a max-type permutation scheme for family-wise error cor-
rection (Freedman and Lane, 1983; Anderson and Robinson, 2001) that has been
validated on neuroimaging data (Winkler et al., 2014) and relies on the nilearn
library.

Finally, when users only have access to z-score maps, they can use either the
Fisher’s (Fisher, 1925) or the Stouffer’s (Riley et al., 1949) estimators. When
sample size information is available, users may incorporate that information into
the Stouffer’s method, via the method described in Zaykin (2011).

Given the paucity of image-based meta-analytic datasets, we have included the
tools to build a Dataset from a NeuroVault collection of 21 pain studies, originally
described in Maumet and Nichols (2016).

```python
from nimare import dataset, extract, utils
dset_dir = extract.download_nidm_pain(
```
    data_dir=data_path, overwrite=False
)

dset_file = os.path.join(
    utils.get_resource_path(), "nidm_pain_dset.json"
)

img_dset = dataset.Dataset(dset_file)

# Point the Dataset toward the images we've downloaded
img_dset.update_path(dset_dir)


3.7.1 Transforming images

Researchers may share their statistical maps in many forms, some of which are direct transformations of one another. For example, researchers may share test statistic maps with z-statistics or t-statistics, and, as long as we know the degrees of freedom associated with the t-test, we can convert between the two easily. To that end, NiMARE includes a class, ImageTransformer, which will calculate target image types from available ones, as long as the available images are compatible with said transformation.

Here, we use ImageTransformer to calculate z-statistic and variance maps for all studies with compatible images. This allows us to apply more image-based meta-analysis algorithms to the Dataset.

    from nimare import transforms

    img_transformer = transforms.ImageTransformer(
        target=['z', 'varcope'], overwrite=False
    )

    img_dset = img_transformer.transform(img_dset)

    Now that we have filled in as many gaps in the Dataset as possible, we can start running meta-analyses. We will start with a DerSimonian-Laird meta-analysis (DerSimonianLaird).
from nimare import meta

dsl_meta = meta.ibma.DerSimonianLaird(resample=True)
dsl_results = dsl_meta.fit(img_dset)

# Retain the z-statistic map for later use
dsl_img = dsl_results.get_map("z", return_type="image")

Now we will apply other available IBMA Estimators to the same Dataset, and save their results to files for comparison.

# Stouffer's
stouffers_meta = meta.ibma.Stouffers(
    use_sample_size=False, resample=True
)
stouffers_results = stouffers_meta.fit(img_dset)
stouffers_img = stouffers_results.get_map(
    "z", return_type="image"
)

# Stouffer's with weighting based on sample size
wstouffers_meta = meta.ibma.Stouffers(
    use_sample_size=True, resample=True
)
wstouffers_results = wstouffers_meta.fit(img_dset)
wstouffers_img = wstouffers_results.get_map(
    "z", return_type="image"
)

# Fisher's
fishers_meta = meta.ibma.Fishers(resample=True)
fishers_results = fishers_meta.fit(img_dset)
fishers_img = fishers_results.get_map("z", return_type="image")

# Permuted Ordinary Least Squares
ols_meta = meta.ibma.PermutedOLS(resample=True)
ols_results = ols_meta.fit(img_dset)
ols_img = ols_results.get_map("z", return_type="image")

# Weighted Least Squares
wls_meta = meta.ibma.WeightedLeastSquares(resample=True)
wls_results = wls_meta.fit(img_dset)
wls_img = wls_results.get_map("z", return_type="image")

# Hedges'
hedges_meta = meta.ibma.Hedges(resample=True)
hedges_results = hedges_meta.fit(img_dset)
hedges_img = hedges_results.get_map("z", return_type="image")

# Use atlas for likelihood-based estimators
from nilearn import datasets, image, input_data

atlas = datasets.fetch_atlas_harvard_oxford(
    "cort-maxprob-thr25-2mm"
)

# nilearn's NiftiLabelsMasker cannot handle NaNs at the moment,
# and some of the NIDM-Results packs' beta images have NaNs at
# the edge of the brain.
# So, we will create a reduced version of the atlas for this
# analysis.

nan_mask = image.math_img(
    "~np.any(np.isnan(img), axis=3)",
    img=img_dset.images["beta"].tolist(),
)

nanmasked_atlas = image.math_img(
    "mask * atlas", mask=nan_mask, atlas=atlas["maps"]
)

masker = input_data.NiftiLabelsMasker(nanmasked_atlas)

# Variance-Based Likelihood

vbl_meta = meta.ibma.VarianceBasedLikelihood(
    method="reml", mask=masker, resample=True
)

vbl_results = vbl_meta.fit(img_dset)

vbl_img = vbl_results.get_map("z", return_type="image")

# Sample Size-Based Likelihood

ssbl_meta = meta.ibma.SampleSizeBasedLikelihood(
    method="reml", mask=masker, resample=True
)

ssbl_results = ssbl_meta.fit(img_dset)

ssbl_img = ssbl_results.get_map("z", return_type="image")

3.7.2 Comparing algorithms

Here we load the z-statistic map from each of the IBMA Estimators we’ve used throughout this chapter and plot them all side by side.

meta_results = {
    "DerSimonian-Laird": dsl_img,
    "Stouffer's": stouffers_img,
    "Weighted Stouffer's": wstouffers_img,
}
"Fisher's": fishers_img,
"Ordinary Least Squares": ols_img,
"Weighted Least Squares": wls_img,
"Hedges'": hedges_img,
"Variance-Based Likelihood": vbl_img,
"Sample Size-Based Likelihood": ssbl_img,

```python
order = [
    ["Fisher's", "Stouffer's", "Weighted Stouffer's"],
    ["DerSimonian-Laird", "Hedges'", "Weighted Least Squares"],
    ["Ordinary Least Squares",
     "Variance-Based Likelihood",
     "Sample Size-Based Likelihood"],
]

fig, axes = plt.subplots(figsize=(18, 6), nrows=3, ncols=3)
for i_row, row_names in enumerate(order):
    for j_col, name in enumerate(row_names):
        file_ = meta_results[name]
        display = plotting.plot_stat_map(
            file_,
            annotate=False,
            axes=axes[i_row, j_col],
            cmap="RdBu_r",
            cut_coords=[5, -15, 10],
            draw_cross=False,
            figure=fig,
        )
        axes[i_row, j_col].set_title(name)

colorbar = display._cbar
colorbar_ticks = colorbar.get_ticks()
if colorbar_ticks[0] < 0:
    new_ticks = [
        colorbar_ticks[0],
        0,
        colorbar_ticks[-1],
    ]
else:
    new_ticks = [colorbar_ticks[0], colorbar_ticks[-1]]
    colorbar.set_ticks(new_ticks, update_ticks=True)
```

47
Figure 3.6 **Image-based meta-analysis results**  These results come from IBMAs without multiple comparisons correction. Additionally, note that the likelihood-based meta-analyses are run on atlases instead of voxelwise.

### 3.8 Multiple Comparisons Correction

In NiMARE, multiple comparisons correction is separated from each CBMA and IBMA Estimator, so that any number of relevant correction methods can be applied after the Estimator has been fit to the Dataset. Some correction options, such as the montecarlo option for FWE correction, are designed to work specifically with a given Estimator (and are indeed implemented within the Estimator class, and only called by the Corrector).

Correctors are divided into two subclasses: FWECorrectors, which correct based on family-wise error rate, and FDRCorrectors, which correct based on false discovery rate.

All Correctors are initialized with a number of parameters, including the correction method that will be used. After that, you can use the transform method on a MetaResult object produced by a CBMA or IBMA Estimator to apply the correction method. This will return an updated MetaResult object, with both the statistical maps from the original MetaResult, as well as new, corrected maps.
Here we will apply both FWE and FDR correction to results from a MKDADensity meta-analysis, performed back in Section 3.6.2.

```python
from nimare import meta, correct

mkdad_meta = meta.cbma.mkda.MKDADensity.load(
    os.path.join(data_path, "MKDADensity.pkl.gz"))

mc_corrector = correct.FWECorrector(
    method="montecarlo", n_iters=5000, n_cores=4)
mc_results = mc_corrector.transform(mkdad_meta.results)
mc_results.save_maps(
    output_dir=out_dir, prefix="MKDADensity_FWE")

fdr_corrector = correct.FDRCorrector(method="indep")
fdr_results = fdr_corrector.transform(mkdad_meta.results)
```

Statistical maps saved by NiMARE MetaResults automatically follow a naming convention based loosely on the Brain Imaging Data Standard (BIDS).

Let’s take a look at the files created by the FWECorrector.

```python
from glob import glob

fwe_maps = sorted(glob(os.path.join(
    out_dir, "MKDADensity_FWE*.nii.gz")))
fwe_maps = [os.path.basename(fwe_map) for fwe_map in fwe_maps]
print("\n".join(fwe_maps))
```

```
MKDADensity_FWE_logp_desc-mass_level-cluster_corr-FWE_method-montecarlo.nii.gz
MKDADensity_FWE_logp_desc-size_level-cluster_corr-FWE_method-montecarlo.nii.gz
MKDADensity_FWE_logp_level-voxel_corr-FWE_method-montecarlo.nii.gz
MKDADensity_FWE_p.nii.gz
MKDADensity_FWE_stat.nii.gz
MKDADensity_FWE_z.nii.gz
```
If you ignore the prefix, which was specified in the call to `MetaResult.save()` maps, the maps all have a common naming convention. The maps from the original meta-analysis (before multiple comparisons correction) are simply named according to the values contained in the map (e.g., $z$, `stat`, `p`).

Maps generated by the correction method, however, use a series of key-value pairs to indicate how they were generated. The `corr` key indicates whether FWE or FDR correction was applied. The `method` key reflects the correction method employed, which was defined by the `method` parameter used to create the `Corrector`. The `level` key simply indicates if the map was corrected at the voxel or cluster level. Finally, the `desc` key reflects any necessary description that goes beyond what is already covered by the other entities.

```python
meta_results = {
    "Cluster-level Monte Carlo": mc_results.get_map(
        "z_desc-size_level-cluster_corr-FWE_method-montecarlo",
        return_type="image",
    ),
    "Independent FDR": fdr_results.get_map(
        "z_corr-FDR_method-indep",
        return_type="image",
    ),
}
```

```python
fig, axes = plt.subplots(figsize=(6, 4), nrows=2)
for i_meta, (name, file_) in enumerate(meta_results.items()):
    display = plotting.plot_stat_map(
        file_,
        annotate=False,
        axes=axes[i_meta],
        draw_cross=False,
        cmap="Reds",
```
3.9 Derivative Analyses

Meta-analytic databases and algorithms may be employed for derivative analyses, including subtraction analysis, meta-analytic coactivation modeling (MACM), meta-analytic clustering, coactivation-based parcellation (CBP), meta-analytic independent component analysis (meta-ICA), semantic model development, and meta-
analytic functional decoding. In this part, we describe the derivative analyses implemented in NiMARE and include examples of use cases.

### 3.10 Meta-Analytic Subtraction Analysis

```python
# Load the Datasets we will use in this section
sleuth_dset1 = dataset.Dataset.load(
    os.path.join(data_path, "sleuth_dset1.pkl.gz")
)
sleuth_dset2 = dataset.Dataset.load(
    os.path.join(data_path, "sleuth_dset2.pkl.gz")
)
```

Subtraction analysis refers to the voxel-wise comparison of two meta-analytic samples. In image-based meta-analysis, comparisons between groups of maps can generally be accomplished within the standard meta-regression framework (i.e., by adding a covariate that codes for group membership). However, coordinate-based subtraction analysis requires special extensions for CBMA algorithms.

Subtraction analysis to compare the results of two ALE meta-analyses was originally implemented by Laird et al. (2005a) and later extended by Eickhoff et al. (2012b). In this approach, two groups of experiments (A and B) are compared using a group assignment randomization procedure in which voxel-wise null distributions are generated by randomly reassigning experiments between the two groups and calculating ALE-difference scores for each permutation. Real ALE-difference scores (i.e., the ALE values for one group minus the ALE values for the other) are compared against these null distributions to determine voxel-wise significance. In the original implementation of the algorithm, this procedure is performed separately for a group A > B contrast and a group B > A contrast, where each contrast is limited to voxels that were significant in the first group’s original meta-analysis.
In NiMARE, we use an adapted version of the subtraction analysis method in ALESubtraction. The NiMARE implementation analyzes all voxels, rather than only those that show a significant effect of A alone or B alone as in the original implementation.

```python
kern = meta.kernel.ALEKernel()
sub_meta = meta.cbma.ale.ALESubtraction(
    kernel_transformer=kern, n_iters=1000
)
sub_results = sub_meta.fit(sleuth_dset1, sleuth_dset2)
```

```python
fig, ax = plt.subplots(figsize=(6, 2))
display = plotting.plot_stat_map(
    sub_results.get_map("z_desc-group1MinusGroup2", return_type=\"image\"),
    annotate=False,
    axes=ax,
    cmap="RdBu_r",
    cut_coords=[0, 0, 0],
    draw_cross=False,
    figure=fig,
)
ax.set_title("ALE Subtraction")

colorbar = display._cbar
colorbar_ticks = colorbar.get_ticks()
if colorbar_ticks[0] < 0:
    new_ticks = [colorbar_ticks[0], 0, colorbar_ticks[-1]]
else:
    new_ticks = [colorbar_ticks[0], colorbar_ticks[-1]]
colorbar.set_ticks(new_ticks, update_ticks=True)
```
Alternatively, MKDA Chi-squared analysis is inherently a subtraction analysis method, in that it compares foci from two groups of studies. Generally, one of these groups is a sample of interest, while the other is a meta-analytic database (minus the studies in the sample). With this setup, meta-analysts can infer whether there is greater convergence of foci in a voxel as compared to the baseline across the field (as estimated with the meta-analytic database), much like SCALE. However, if the database is replaced with a second sample of interest, the analysis ends up comparing convergence between the two groups.

3.11 Meta-Analytic Coactivation Modeling

```python
# Load the Dataset we use in this section
eurosynth_dset = dataset.Dataset.load(os.path.join(data_path, "neurosynth_dataset.pkl.gz"))
```

Meta-analytic coactivation modeling (MACM) (Laird et al., 2009; Robinson et al., 2010; Eickhoff et al., 2010), also known as meta-analytic connectivity modeling, uses meta-analytic data to measure co-occurrence of activations between brain regions providing evidence of functional connectivity of brain regions across tasks.
In coordinate-based MACM, whole-brain studies within the database are selected based on whether or not they report at least one peak in a region of interest specified for the analysis. These studies are then subjected to a meta-analysis, often comparing the selected studies to those remaining in the database. In this way, the significance of each voxel in the analysis corresponds to whether there is greater convergence of foci at the voxel among studies which also report foci in the region of interest than those which do not.

MACM results have historically been accorded a similar interpretation to task-related functional connectivity (e.g., Hok et al. (2015); Kellermann et al. (2013)), although this approach is quite removed from functional connectivity analyses of task fMRI data (e.g., beta-series correlations, psychophysiological interactions, or even seed-to-voxel functional connectivity analyses on task data). Nevertheless, MACM analyses do show high correspondence with resting-state functional connectivity (Reid et al., 2017). MACM has been used to characterize the task-based functional coactivation of the cerebellum (Riedel et al., 2015), lateral prefrontal cortex (Reid et al., 2016), fusiform gyrus (Caspers et al., 2014), and several other brain regions.

Within NiMARE, MACMs can be performed by selecting studies in a Dataset based on the presence of activation within a target mask or coordinate-centered sphere.

In this section, we will perform two MACMs- one with a target mask and one with a coordinate-centered sphere. For the former, we use Dataset.get_studies_by_mask. For the latter, we use Dataset.get_studies_by_coordinate.

```python
# Create Dataset only containing studies with peaks within the
# amygdala mask
amyg_mask = os.path.join(data_path, "amygdala_roi.nii.gz")
amygdala_ids = neurosynth_dset.get_studies_by_mask(amyg_mask)
dset_amyg = neurosynth_dset.slice(amygdala_ids)

# Create Dataset only containing studies with peaks within the
```
The amygdala dataset includes more than 1300 studies. Running a meta-analysis on such a large dataset may require more than 4 GB of RAM, which is NeuroLibre’s limit. Therefore, we will further reduce the Dataset to its first 500 studies, in order to run the meta-analysis successfully on NeuroLibre’s server. For publication-quality analyses, we would recommend using the entire Dataset.

Figure 3.9 Region of interest masks for MACMs. The masks are for (1) a target mask-based MACM and (2) a coordinate-based MACM.
Once the Dataset has been reduced to studies with coordinates within the mask or sphere requested, any of the supported CBMA Estimators can be run.

```python
meta_amyg = meta.cbma.ale.ALE(kernel__sample_size=20)
results_amyg = meta_amyg.fit(dset_amygdala)

meta_sphere = meta.cbma.ale.ALE(kernel__sample_size=20)
results_sphere = meta_sphere.fit(dset_sphere)

meta_results = {
    "Amygdala ALE MACM": results_amyg.get_map(
        "z", return_type="image"
    ),
    "Sphere ALE MACM": results_sphere.get_map(
        "z", return_type="image"
    ),
}

fig, axes = plt.subplots(figsize=(6, 4), nrows=2)
for i_meta, (name, file_) in enumerate(meta_results.items()):
    display = plotting.plot_stat_map(
        file_,
        annotate=False,
        axes=axes[i_meta],
        cmap="Reds",
        cut_coords=[24, -2, -20],
        draw_cross=False,
        figure=fig,
    )
    axes[i_meta].set_title(name)

    colorbar = display._cbar
    colorbar_ticks = colorbar.get_ticks()
    if colorbar_ticks[0] < 0:
        new_ticks = [colorbar_ticks[0], 0, colorbar_ticks[-1]]
    else:
        new_ticks = [colorbar_ticks[0], colorbar_ticks[-1]]
    colorbar.set_ticks(new_ticks, update_ticks=True)
```
Figure 3.10 **Results from MACMs.** Unthresholded z-statistic maps for (1) the target mask-based MACM and (2) the coordinate-based MACM.

### 3.12 Automated Annotation

As mentioned in the discussion of BrainMap (Section 3.5.1), manually annotating studies in a meta-analytic database can be a time-consuming and labor-intensive process. To facilitate more efficient (albeit lower-quality) annotation, NiMARE supports a number of automated annotation approaches. These include Section 3.12.1), Section 3.12.2, Section 3.12.3, and Section 3.12.4.

NiMARE users may download abstracts from PubMed as long as study identifiers in the **Dataset** correspond to PubMed IDs (as in Neurosynth and NeuroQuery). Abstracts are much more easily accessible than full article text, so most annotation methods in NiMARE rely on them.

Below, we use the function `download_abstracts` to download abstracts for the Neurosynth **Dataset**. This will attempt to extract metadata about each study in
the Dataset from PubMed, and then add the abstract available on Pubmed to the Dataset’s texts attribute, under a new column names ”abstract”.

```python
import pandas as pd
from nimare import extract

# First, load a Dataset without abstracts
neurosynth_dset_first_500 = dataset.Dataset.load(
    os.path.join(
        data_path,
        "neurosynth_dataset_first500.pkl.gz",
    )
)

# Now, download the abstracts using your email address
neurosynth_dset_first_500 = extract.downloadAbstracts(
    neurosynth_dset_first_500,
    email="example@email.com",
)

# Finally, save the Dataset with abstracts to a pkl.gz file
neurosynth_dset_first_500.save(
    os.path.join(
        data_path,
        "neurosynth_dataset_first500_with_abstracts.pkl.gz",
    ),
)
```

3.12.1 N-gram term extraction

N-gram term extraction refers to the vectorization of text into contiguous sets of words that can be counted as individual tokens. The upper limit on the number of words in these tokens is set by the user.

NiMARE has the function generate_counts to extract n-grams from text. This method produces either term counts or term frequency- inverse document frequency (tf-idf) values for each of the studies in a Dataset.

```python
from nimare import annotate
```
counts_df = annotate.text.generate_counts(
    neurosynth_dset_first_500.texts,
    text_column="abstract",
    tfidf=False,
    min_df=10,
    max_df=0.95,
)

This term count DataFrame will be used later, to train a GCLDA model.

3.12.2 Cognitive Atlas term extraction and hierarchical expansion

Cognitive Atlas term extraction leverages the structured nature of the Cognitive Atlas in order to extract counts for individual terms and their synonyms in the ontology, as well as to apply hierarchical expansion to these counts based on the relationships specified between terms. This method produces both basic term counts and expanded term counts based on the weights applied to different relationship types present in the ontology.

First, we must use download_cognitive_atlas to download the current version of the Cognitive Atlas ontology. This includes both information about individual terms in the ontology and asserted relationships between those terms.

NiMARE will automatically attempt to extrapolate likely alternate forms of each term in the ontology, in order to make extraction easier. For an example, see Table 3.2.

cogatlas = extract.download_cognitive_atlas(
    data_dir=data_path, overwrite=False
)
id_df = pd.read_csv(cogatlas["ids"])
rel_df = pd.read_csv(cogatlas["relationships"])
cogat_counts_df, rep_text_df = annotate.cogat.extract_cogat(

Table 3.2 **Alternate forms in the Cognitive Atlas** An example of alternate forms characterized by the Cognitive Atlas and extrapolated by NiMARE. Certain alternate forms (i.e., synonyms) are specified within the Cognitive Atlas, while others are inferred automatically by NiMARE according to certain rules (e.g., removing parentheses).

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>trm_4f244ad7dcde7</td>
<td>dot motion task</td>
<td>random-dot motion task</td>
</tr>
<tr>
<td>trm_4f244ad7dcde7</td>
<td>dot motion task</td>
<td>dot motion task</td>
</tr>
<tr>
<td>trm_4f244ad7dcde7</td>
<td>dot motion task</td>
<td>dot-motion task</td>
</tr>
<tr>
<td>trm_4f244ad7dcde7</td>
<td>dot motion task</td>
<td>moving-dot task</td>
</tr>
<tr>
<td>trm_4f244ad7dcde7</td>
<td>dot motion task</td>
<td>rdm task</td>
</tr>
</tbody>
</table>

# Define a weighting scheme.
# In this scheme, observed terms will also count toward any
# hypernyms (isKindOf), holonyms (isPartOf),
# and parent categories (inCategory) as well.
weights = {"isKindOf": 1, "isPartOf": 1, "inCategory": 1}
expanded_df = annotate.cogat.expand_counts(cogat_counts_df, rel_df, weights)

# Sort by total count and reduce for better visualization
series = expanded_df.sum(axis=0)
series = series.sort_values(ascending=False)
series = series[series > 0]
columns = series.index.tolist()

# Raw counts
fig, axes = plt.subplots(figsize=(16, 16), nrows=2, sharex=True)
pos = axes[0].imshow(cogat_counts_df[columns].values, aspect="auto",
                          vmin=0,
                          vmax=10,
                      )
fig.colorbar(pos, ax=axes[0])
axes[0].set_title("Counts Before Expansion", fontsize=20)
axes[0].yaxis.set_visible(False)
axes[0].xaxis.set_visible(False)
axes[0].set_ylabel("Study", fontsize=16)
axes[0].set_xlabel("Cognitive Atlas Term", fontsize=16)

# Expanded counts
pos = axes[1].imshow(
    expanded_df[columns].values,
    aspect="auto",
    vmin=0,
    vmax=10,
)
fig.colorbar(pos, ax=axes[1])
axes[1].set_title("Counts After Expansion", fontsize=20)
axes[1].yaxis.set_visible(False)
axes[1].xaxis.set_visible(False)
axes[1].set_ylabel("Study", fontsize=16)
axes[1].set_xlabel("Cognitive Atlas Term", fontsize=16)

fig.tight_layout()
Figure 3.11 The effect of hierarchical expansion on Cognitive Atlas term counts from abstracts in Neurosynth’s first 500 papers. There are too many terms and studies to show individual labels, but the effect of the expansion can be appreciated based on the patterns.

3.12.3 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) (Blei et al., 2003) was originally combined with meta-analytic neuroimaging data in Poldrack et al. (2012). LDA is a generative topic model which, for a text corpus, builds probability distributions across documents and words. In LDA, each document is considered a mixture of topics. This works under the assumption that each document was constructed by first
randomly selecting a topic based on the document’s probability distribution across topics, and then randomly selecting a word from that topic based on the topic’s probability distribution across words. While this is not a useful generative model for producing documents, LDA is able to discern cohesive topics of related words. Poldrack et al. (2012) were able to apply LDA to full texts from neuroimaging articles in order to develop cognitive neuroscience-related topics and to run topic-wise meta-analyses. This method produces two sets of probability distributions: (1) the probability of a word given topic and (2) the probability of a topic given article.

NiMARE’s LDAModel is a light wrapper around scikit-learn’s LDA implementation.

Here, we train an LDA model (LDAModel) on the first 500 studies of the Neurosynth Dataset, with 50 topics in the model.

```python
lda_model = annotate.lda.LDAModel(
    n_topics=50, max_iter=1000, text_column="abstract"
)

# Fit the model
lda_model.fit(neurosynth_dset_first_500)
```

The most important products of training the LDAModel object is its distributions_ attribute. LDAModel.distributions_ is a dictionary containing arrays and DataFrames created from training the model. We are particularly interested in the p_topic_g_word_df distribution, which is a pandas DataFrame in which each row corresponds to a topic and each column corresponds to a term (n-gram) extracted from the Dataset’s texts. The cells contain weights indicating the probability distribution across terms for each topic.

Additionally, the LDAModel updates the Dataset’s annotations attribute, by adding columns corresponding to each of the topics in the model. Each study in the
Dataset thus receives a weight for each topic, which can be used to select studies for topic-based meta-analyses or functional decoding.

Let’s take a look at the results of the model training. First, we will reorganize the DataFrame a bit to show the top ten terms for each of the first ten topics.

```python
lda_df = lda_model.distributions_['p_topic_g_word_df'].T
column_names = {c: f'Topic {c}' for c in lda_df.columns}
lda_df = lda_df.rename(columns=column_names)
temp_df = lda_df.copy()
lda_df = pd.DataFrame(
    columns=lda_df.columns, index=np.arange(10)
)
lda_df.index.name = 'Term'
for col in lda_df.columns:
    top_ten_terms = temp_df.sort_values(by=col, ascending=False).index.tolist()[:10]
    lda_df.loc[:, col] = top_ten_terms

lda_df = lda_df[lda_df.columns[:3]]
```

Table 3.3 Results from an LDA topic model The top ten terms for each of the first three topics in the trained LDA model.

<table>
<thead>
<tr>
<th>Term</th>
<th>Topic LDA50_1</th>
<th>Topic LDA50_2</th>
<th>Topic LDA50_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>cortex</td>
<td>attention</td>
<td>thirst</td>
</tr>
<tr>
<td>1</td>
<td>antipsychotic</td>
<td>intraparietal</td>
<td>anterior</td>
</tr>
<tr>
<td>2</td>
<td>frontal</td>
<td>sulcus</td>
<td>parietal</td>
</tr>
<tr>
<td>3</td>
<td>functional</td>
<td>spatial</td>
<td>cingulate</td>
</tr>
<tr>
<td>4</td>
<td>typical</td>
<td>intraparietal sulcus</td>
<td>cortex</td>
</tr>
<tr>
<td>5</td>
<td>antipsychotics</td>
<td>cortex</td>
<td>cerebral</td>
</tr>
<tr>
<td>6</td>
<td>posterior parietal</td>
<td>attentional</td>
<td>frontal</td>
</tr>
<tr>
<td>7</td>
<td>working memory</td>
<td>sensory</td>
<td>genesis</td>
</tr>
<tr>
<td>8</td>
<td>prefrontal</td>
<td>premotor</td>
<td>group</td>
</tr>
<tr>
<td>9</td>
<td>parietal</td>
<td>visual</td>
<td>network</td>
</tr>
</tbody>
</table>
3.12.4 Generalized correspondence latent Dirichlet allocation

Generalized correspondence latent Dirichlet allocation (GCLDA) is a recently-developed algorithm that trains topics on both article abstracts and coordinates (Rubin et al., 2017). GCLDA assumes that topics within the fMRI literature can also be localized to brain regions, in this case modeled as three-dimensional Gaussian distributions. These spatial distributions can also be restricted to pairs of Gaussians that are symmetric across brain hemispheres. This method produces two sets of probability distributions: the probability of a word given topic (GCLDAModel.p_word_g_topic_) and the probability of a voxel given topic (GCLDAModel.p_voxel_g_topic_).

Here we train a GCLDA model (GCLDAModel) on the first 500 studies of the Neurosynth Dataset. The model will include 50 topics, in which the spatial distribution for each topic will be defined as having two Gaussian distributions that are symmetrically localized across the longitudinal fissure.

```python
gclda_model = annotate.gclda.GCLDAModel(
    counts_df,
    neurosynth_dset_first_500.coordinates,
    n_regions=2,
    n_topics=50,
    symmetric=True,
    mask=neurosynth_dset_first_500.masker.mask_img,
)
gclda_model.fit(n_iters=2500, loglikely_freq=500)
```

The GCLDAModel retains the relevant probability distributions in the form of numpy arrays, rather than pandas DataFrames. However, for the topic-term weights (p_word_g_topic_), the data are more interpretable as a DataFrame, so we will
create one. We will also reorganize the raw DataFrame to show the top ten terms for each of the first ten topics.

```python
gclda_arr = gclda_model.p_word_g_topic_  
gclda_vocab = gclda_model.vocabulary  
topic_names = [  
    f"Topic {str(i).zfill(3)}" for i in  
    range(gclda_arr.shape[1])  
]  
gclda_df = pd.DataFrame(  
    index=gclda_vocab, columns=topic_names, data=gclda_arr  
)  
temp_df = gclda_df.copy()  
gclda_df = pd.DataFrame(  
    columns=gclda_df.columns, index=np.arange(10)  
)  
gclda_df.index.name = "Term"  
for col in temp_df.columns:  
    top_ten_terms = temp_df.sort_values(  
        by=col, ascending=False  
    ).index.tolist()[10]  
    gclda_df.loc[:, col] = top_ten_terms  
gclda_df = gclda_df[gclda_df.columns[:3]]
```

Table 3.4 Results from a GCLDA topic model. The top ten terms for each of the first three topics in the trained GCLDA model.

<table>
<thead>
<tr>
<th>Term</th>
<th>Topic 000</th>
<th>Topic 001</th>
<th>Topic 002</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>amygdala</td>
<td>visual</td>
<td>motion</td>
</tr>
<tr>
<td>1</td>
<td>faces</td>
<td>cortex</td>
<td>temporal</td>
</tr>
<tr>
<td>2</td>
<td>neutral</td>
<td>functional</td>
<td>mt</td>
</tr>
<tr>
<td>3</td>
<td>emotional</td>
<td>cortical</td>
<td>visual</td>
</tr>
<tr>
<td>4</td>
<td>functional</td>
<td>magnetic resonance</td>
<td>sulcus</td>
</tr>
<tr>
<td>5</td>
<td>stimuli</td>
<td>visual cortex</td>
<td>occipital</td>
</tr>
<tr>
<td>6</td>
<td>emotion</td>
<td>magnetic</td>
<td>dorsal</td>
</tr>
<tr>
<td>7</td>
<td>response</td>
<td>stimulation</td>
<td>human</td>
</tr>
<tr>
<td>8</td>
<td>functional</td>
<td>resonance</td>
<td>sensitive</td>
</tr>
<tr>
<td>9</td>
<td>responses</td>
<td>spatial</td>
<td>perception</td>
</tr>
</tbody>
</table>

We also want to see how the topic-voxel weights render on the brain, so we will simply unmask the p_voxel_g_topic_ array with the Dataset’s masker.
fig, axes = plt.subplots(nrows=5, ncols=2, figsize=(12, 10))

topic_img_4d = \
    neurosynth_dset_first_500.masker.inverse_transform(
        gclda_model.p_voxel_g_topic_.T
    )

# Plot first ten topics
topic_counter = 0
for i_row in range(5):
    for j_col in range(2):
        topic_img = image.index_img(
            topic_img_4d,
            index=topic_counter,
        )
        display = plotting.plot_stat_map(
            topic_img,
            annotate=False,
            cmap="Reds",
            draw_cross=False,
            figure=fig,
            axes=axes[i_row, j_col],
        )
        axes[i_row, j_col].set_title(
            f"Topic {str(topic_counter).zfill(3)}"
        )
        topic_counter += 1

colorbar = display._cbar
colorbar_ticks = colorbar.get_ticks()
if colorbar_ticks[0] < 0:
    new_ticks = [
        colorbar_ticks[0],
        0,
        colorbar_ticks[-1],
    ]
else:
    new_ticks = [colorbar_ticks[0], colorbar_ticks[-1]]
colorbar.set_ticks(new_ticks, update_ticks=True)
3.13 Meta-Analytic Functional Decoding

Functional decoding performed with meta-analytic data, refers to methods which attempt to predict mental states from neuroimaging data using a large-scale meta-analytic database (Smith et al., 2009). Such analyses may also be referred to as "informal reverse inference" (Poldrack, 2011), "functional characterization analysis" (Bzdok et al., 2013a; Cieslik et al., 2013; Rottschy et al., 2013), "open-ended decoding" (Rubin et al., 2017), or simply "functional decoding" (Amft et al., 2015; Bzdok et al., 2013b; Nickl-Jockschat et al., 2015). While the terminology is far from standardized, we will refer to this method as **meta-analytic functional decoding** in order to distinguish it from alternative methods like multivariate decoding and model-based decoding (Poldrack, 2011). Meta-analytic functional decoding is
often used in conjunction with MACM, meta-analytic clustering, meta-analytic parcellation, and meta-ICA, in order to characterize resulting brain regions, clusters, or components. Meta-analytic functional decoding models have also been extended for the purpose of meta-analytic functional encoding, wherein text is used to generate statistical images (Dockès et al., 2018; Nunes, 2018; Rubin et al., 2017).

Four common approaches are correlation-based decoding, dot-product decoding, weight-sum decoding, and chi-square decoding. We will first discuss continuous decoding methods (i.e., correlation and dot-product), followed by discrete decoding methods (weight-sum and chi-square).

3.13.1 Decoding continuous inputs

When decoding unthresholded statistical maps (such as Figure 3.13), the most common approaches are to simply correlate the input map with maps from the database, or to compute the dot product between the two maps. In Neurosynth, meta-analyses are performed for each label (i.e., term or topic) in the database and then the input image is correlated with the resulting unthresholded statistical map from each meta-analysis. Performing statistical inference on the resulting correlations is not straightforward, however, as voxels display strong spatial correlations, and the true degrees of freedom are consequently unknown (and likely far smaller than the nominal number of voxels). In order to interpret the results of this decoding approach, users typically select some arbitrary number of top correlation coefficients ahead of time, and use the associated labels to describe the input map. However, such results should be interpreted with great caution.
Figure 3.13  The unthresholded statistical map that will be used for continuous decoding.

This approach can also be applied to an image-based database like NeuroVault, either by correlating input data with meta-analyzed statistical maps, or by deriving distributions of correlation coefficients by grouping statistical maps in the database according to label. Using these distributions, it is possible to statistically compare labels in order to assess label significance. NiMARE includes methods for both correlation-based decoding and correlation distribution-based decoding, although the correlation-based decoding is better established and should be preferred over the correlation distribution-based decoding. As such, we will only show the CorrelationDecoder here.

CorrelationDecoder currently runs very slowly. We strongly recommend running it on a subset of labels within the Dataset. It is also quite memory-intensive.

In this example, we have only run the decoder using features appearing in >10% and <90% of the first 500 studies in the Dataset. The top ten terms from the decoding analysis are presented in Table 3.5.

```python
neurosynth_dset = dataset.Dataset.load(
    os.path.join(data_path, "neurosynth_dataset.pkl.gz")
```
kern = meta.kernel.MKDAKernel(memory_limit="500mb")
neurosynth_dset_first500 = dataset.Dataset.load(
    os.path.join(
        data_path,
        "neurosynth_dataset_first500_with_mkda_ma.pkl.gz",
    ),
)

# Collect features for decoding
# We use any features that appear in >10% of studies and <90%.
id_cols = ["id", "study_id", "contrast_id"]
frequency_threshold = 0.001
cols = neurosynth_dset_first500.annotations.columns
cols = [c for c in cols if c not in id_cols]
df = neurosynth_dset_first500.annotations.copy()[cols]
n_studies = df.shape[0]
feature_counts = (df >= frequency_threshold).sum(axis=0)
target_features = feature_counts.between(n_studies * 0.1, n_studies * 0.9)
target_features = target_features[target_features]
target_features = target_features.index.values
print(f"{len(target_features)} features selected.", flush=True)

continuous_map = os.path.join(data_path, "map_to_decode.nii.gz")
amygdala_roi = os.path.join(data_path, "amygdala_roi.nii.gz")
amygdala_ids = neurosynth_dset_first500.get_studies_by_mask(
    amygdala_roi
)

# Begin correlation-based decoding
from nimare import decode
corr_decoder = decode.continuous.CorrelationDecoder(
    frequency_threshold=0.001,
    meta_estimator=meta.MKDADensity(
        kernel_transformer=kern,
        memory_limit=None,
    ),
    target_image="z",
    features=target_features,
    memory_limit="500mb",
)
corr_decoder.fit(neurosynth_dset_first500)
corr_df = corr_decoder.transform(continuous_map)
corr_df = corr_df.reindex(
    corr_df["r"].abs().sort_values(ascending=False).index
)
Table 3.5 **Results of correlation-based decoding.** The top ten terms, sorted by absolute correlation coefficient, from the correlation decoding method.

<table>
<thead>
<tr>
<th>feature</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>cingulate</td>
<td>0.448665</td>
</tr>
<tr>
<td>anterior cingulate</td>
<td>0.415396</td>
</tr>
<tr>
<td>anterior</td>
<td>0.391049</td>
</tr>
<tr>
<td>auditory</td>
<td>0.343746</td>
</tr>
<tr>
<td>conditions</td>
<td>0.338773</td>
</tr>
<tr>
<td>cortices</td>
<td>0.331308</td>
</tr>
<tr>
<td>information</td>
<td>0.330473</td>
</tr>
<tr>
<td>role</td>
<td>0.326022</td>
</tr>
<tr>
<td>level</td>
<td>0.322515</td>
</tr>
<tr>
<td>healthy</td>
<td>0.319750</td>
</tr>
</tbody>
</table>

### 3.13.2 Decoding discrete inputs

Decoding regions of interest (ROIs) requires a different approach than decoding unthresholded statistical maps. One simple approach, used by GCLDA and implemented in the function `gclda_decode_roi`, simply sums the $P(\text{topic}|\text{voxel})$ distribution across all voxels in the ROI in order to produce a value associated with each topic for the ROI. These **weight sum** values are arbitrarily scaled and cannot be compared across ROIs. We will not show this method because of its simplicity and the fact that it can only currently be applied to a GCLDA model.

Before we dig into the other decoding methods are are available, let’s take a look at the ROI we want to decode.
One method which relies on correlations, much like the continuous correlation decoder, is the **ROI association** decoding method (`ROIAssociationDecoder`), originally implemented in the Neurosynth Python library. In this method, each study with coordinates in the dataset is convolved with a kernel transformer to produce a modeled activation map. The resulting modeled activation maps are then masked with a region of interest (i.e., the target of the decoding), and the values are averaged within the ROI. These averaged modeled activation values are then correlated with the term weights for all labels in the dataset. This decoding method produces a single correlation coefficient for each of the dataset’s labels.

Because the `ROIAssociationDecoder` generates modeled activation maps for all of the experiments in the `Dataset`, we will only fit this decoder to the first 500 studies.

```python
assoc_decoder = decode.discrete.ROIAssociationDecoder(
    amygdala_roi,
    kernel_transformer=kern,
    u=0.05,
    correction="fdr_bh",
)```
assoc_decoder.fit(neurosynth_dset_first500)
assoc_df = assoc_decoder.transform()

assoc_df = assoc_df.reindex(
    assoc_df["r"].abs().sort_values(ascending=False).index
)
assoc_df = assoc_df.iloc[:10]

Table 3.6 **Results of ROI association-based decoding.** The top ten terms, sorted by absolute correlation coefficient, from the ROI association decoding method.

<table>
<thead>
<tr>
<th>feature</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>amygdala</td>
<td>0.627186</td>
</tr>
<tr>
<td>fear</td>
<td>0.334112</td>
</tr>
<tr>
<td>reinforcement</td>
<td>0.329516</td>
</tr>
<tr>
<td>neutral faces</td>
<td>0.320801</td>
</tr>
<tr>
<td>appraisal</td>
<td>0.317214</td>
</tr>
<tr>
<td>conditioning</td>
<td>0.286165</td>
</tr>
<tr>
<td>age sex</td>
<td>0.273753</td>
</tr>
<tr>
<td>neutral</td>
<td>0.272666</td>
</tr>
<tr>
<td>amygdala response</td>
<td>0.267777</td>
</tr>
<tr>
<td>olfactory</td>
<td>0.265687</td>
</tr>
</tbody>
</table>

A more theoretically driven approach to ROI decoding is to use **chi-square-based** methods. The two methods which use chi-squared tests are the BrainMap decoding method and an adaptation of Neurosynth’s meta-analysis method.

In both chi-square-based methods, studies are first selected from a coordinate-based database according to some criterion. For example, if decoding a region of interest, users might select studies reporting at least one coordinate within 5 mm of the ROI. Metadata (such as ontological labels) for this subset of studies are then compared to those of the remaining, unselected portion of the database in a confusion matrix. For each label in the ontology, studies are divided into four groups: selected and label-positive (SS+L+), selected and label-negative (SS+L-), unselected and label-positive (SS-L+), and unselected and label-negative (SS-L-).
Each method then compares these groups in order to evaluate both consistency and specificity of the relationship between the selection criteria and each label, which are evaluated in terms of both statistical significance and effect size.

**BrainMap method**

The BrainMap discrete decoding method, implemented in BrainMapDecoder, compares the distributions of studies with each label within the sample against those in a larger database while accounting for the number of foci from each study. Broadly speaking, this method assumes that the selection criterion is associated with one peak per study, which means that it is likely only appropriate for selection criteria based around foci, such as regions of interest. One common analysis, meta-analytic clustering, involves dividing studies within a database into meta-analytic groupings based on the spatial similarity of their modeled activation maps (i.e., study-wise pseudo-statistical maps produced by convolving coordinates with a kernel). The resulting sets of studies are often functionally decoded in order to build a functional profile associated with each meta-analytic grouping. While these groupings are defined as subsets of the database, they are not selected based on the location of an individual peak, and so weighting based on the number of foci would be inappropriate.

This decoding method produces four outputs for each label. First, the distribution of studies in the sample with the label are compared to the distributions of other labels within the sample. This consistency analysis produces both a measure of statistical significance (i.e., a p-value) and a measure of effect size (i.e., the likelihood of being selected given the presence of the label). Next, the studies in the sample are compared to the studies in the rest of the database. This specificity analysis produces a p-value and an effect size measure of the posterior probability
of having the label given selection into the sample. A detailed algorithm description is presented in Appendix A.

```python
brainmap_decoder = decode.discrete.BrainMapDecoder(
    frequency_threshold=0.001,
    u=0.05,
    correction="fdr_bh",
)
brainmap_decoder.fit(neurosynth_dset)
brainmap_df = brainmap_decoder.transform(amygdala_ids)

brainmap_df = brainmap_df.reindex(
    brainmap_df['probReverse'].abs().sort_values(ascending=False).index
)
brainmap_df = brainmap_df.iloc[:10]
```

<table>
<thead>
<tr>
<th>Term</th>
<th>pFw</th>
<th>zFw</th>
<th>lFw</th>
<th>pRv</th>
<th>zRv</th>
<th>probRv</th>
</tr>
</thead>
<tbody>
<tr>
<td>magnetic</td>
<td>1.00</td>
<td>0.00</td>
<td>1.63</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>magnetic resonance</td>
<td>1.00</td>
<td>0.00</td>
<td>1.57</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>resonance</td>
<td>1.00</td>
<td>0.00</td>
<td>1.56</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>amygdala</td>
<td>0.00</td>
<td>4.99</td>
<td>6.41</td>
<td>0.00</td>
<td>8.75</td>
<td>0.01</td>
</tr>
<tr>
<td>functional magnetic</td>
<td>1.00</td>
<td>0.00</td>
<td>1.66</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>using</td>
<td>1.00</td>
<td>0.00</td>
<td>1.21</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>response</td>
<td>1.00</td>
<td>0.00</td>
<td>1.90</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>stimuli</td>
<td>1.00</td>
<td>0.00</td>
<td>2.09</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>human</td>
<td>1.00</td>
<td>0.00</td>
<td>2.13</td>
<td>1.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>neutral</td>
<td>0.01</td>
<td>2.77</td>
<td>5.50</td>
<td>0.00</td>
<td>4.69</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 3.7 BrainMap chi-squared decoding results. The top ten terms, sorted by reverse-inference posterior probability, from the BrainMap chi-squared decoding method. For the sake of space, all values were rounded to the nearest hundredth, and the column names were abbreviated as well. The abbreviated column names are: pFw, forward-inference p-value; zFw, forward-inference z-statistic; lFw, forward-inference likelihood value; pRv, reverse-inference p-value; zRv, reverse-inference z-statistic; and probRv, reverse-inference posterior probability.

**Neurosynth method**

The implementation of the MKDA Chi-squared meta-analysis method used by Neurosynth is quite similar to BrainMap’s method for decoding, if applied to
annotations instead of modeled activation values. This method, implemented in NeurosynthDecoder, compares the distributions of studies with each label within the sample against those in a larger database, but, unlike the BrainMap method, does not take foci into account. For this reason, the Neurosynth method would likely be more appropriate for selection criteria not based on regions of interest (e.g., for characterizing meta-analytic groupings from a meta-analytic clustering analysis). However, the Neurosynth method requires user-provided information that BrainMap does not. Namely, in order to estimate probabilities for the consistency and specificity analyses with Bayes’ Theorem, the Neurosynth method requires a prior probability of a given label. Typically, a value of 0.5 is used (i.e., the estimated probability that an individual is undergoing a given mental process described by a label, barring any evidence from neuroimaging data, is predicted to be 50%). This is, admittedly, a poor prediction, which means that probabilities estimated based on this prior are not likely to be accurate, though they may still serve as useful estimates of effect size for the analysis.

Like the BrainMap method, this method produces four outputs for each label. For the consistency analysis, this method produces both a p-value and a conditional probability of selection given the presence of the label and the prior probability of having the label. For the specificity analysis, the Neurosynth method produces both a p-value and a posterior probability of presence of the label given selection and the prior probability of having the label. A detailed algorithm description is presented in Appendix A.

```python
eurosynth_decoder = decode.discrete.NeurosynthDecoder(
    frequency_threshold=0.001,
    u=0.05,
    correction="fdr_bh",
)
neurosynth_decoder.fit(neurosynth_dset)
neurosynth_df = neurosynth_decoder.transform(amygdala_ids)
```
Table 3.8 **Neurosynth chi-squared decoding results.** The top ten terms, sorted by reverse-inference posterior probability, from the Neurosynth chi-squared decoding method. For the sake of space, all values were rounded to the nearest hundredth, and the column names were abbreviated as well. The abbreviated column names are: pFw, forward-inference p-value; zFw, forward-inference z-statistic; probFw, forward-inference posterior probability; pRv, reverse-inference p-value; zRv, reverse-inference z-statistic; and probRv, reverse-inference posterior probability.

<table>
<thead>
<tr>
<th>Term</th>
<th>pFw</th>
<th>zFw</th>
<th>probFw</th>
<th>pRv</th>
<th>zRv</th>
<th>probRv</th>
</tr>
</thead>
<tbody>
<tr>
<td>neutral faces</td>
<td>0.00</td>
<td>6.88</td>
<td>0.02</td>
<td>0.00</td>
<td>11.59</td>
<td>0.97</td>
</tr>
<tr>
<td>conditioning</td>
<td>0.00</td>
<td>8.30</td>
<td>0.02</td>
<td>0.00</td>
<td>11.90</td>
<td>0.97</td>
</tr>
<tr>
<td>olfactory</td>
<td>0.01</td>
<td>2.44</td>
<td>0.02</td>
<td>0.00</td>
<td>7.03</td>
<td>0.96</td>
</tr>
<tr>
<td>conditioned</td>
<td>0.00</td>
<td>3.95</td>
<td>0.02</td>
<td>0.00</td>
<td>7.31</td>
<td>0.96</td>
</tr>
<tr>
<td>reinforcement</td>
<td>0.00</td>
<td>3.95</td>
<td>0.02</td>
<td>0.00</td>
<td>7.03</td>
<td>0.95</td>
</tr>
<tr>
<td>unpleasant</td>
<td>0.00</td>
<td>3.95</td>
<td>0.02</td>
<td>0.00</td>
<td>6.92</td>
<td>0.95</td>
</tr>
<tr>
<td>amygdala</td>
<td>0.00</td>
<td>23.63</td>
<td>0.01</td>
<td>0.00</td>
<td>8.75</td>
<td>0.95</td>
</tr>
<tr>
<td>amygdala response</td>
<td>0.01</td>
<td>2.44</td>
<td>0.01</td>
<td>0.00</td>
<td>5.25</td>
<td>0.94</td>
</tr>
<tr>
<td>differentiated</td>
<td>0.01</td>
<td>2.44</td>
<td>0.01</td>
<td>0.00</td>
<td>5.25</td>
<td>0.94</td>
</tr>
<tr>
<td>pleasant</td>
<td>0.00</td>
<td>2.44</td>
<td>0.01</td>
<td>0.00</td>
<td>5.02</td>
<td>0.94</td>
</tr>
</tbody>
</table>

In both methods, the database acts as an estimate of the underlying distribution of labels in the real world, such that the probability of having a peak in an ROI given the presence of the label might be interpreted as the probability of a brain activating a specific brain region given that the individual is experiencing a given mental state. This is a very poor interpretation, given that any database of neuroimaging results will be skewed more toward the interests of the field than the distribution of mental states or processes experienced by humans, which is why decoding must be interpreted with extreme caution. It is important not to place too much emphasis on the results of functional decoding analyses, although they are very useful in
that they can provide a quantitative estimate behind the kinds of interpretations generally included in discussion sections that are normally only backed by informal literature searches or prior knowledge.

The meta-analytic functional decoding methods in NiMARE provide a very rudimentary approach for open-ended decoding (i.e., decoding across a very large range of mental states) that can be used with resources like NeuroVault. However, standard classification methods have also been applied to datasets from NeuroVault (e.g., Varoquaux et al. (2018)), although these methods do not fall under NiMARE’s scope.

### 3.14 Future Directions

NiMARE’s mission statement encompasses a range of tools that have not yet been implemented in the package. In the future, we plan to incorporate a number of additional methods. Here we briefly describe several of these tools.

#### 3.14.1 Integration with external databases

A resource which may ultimately be integrated with Neurosynth is Brainspell. Brainspell is a port of the Neurosynth database in which users may manually annotate the automatically extracted study information. The goal of Brainspell is to crowdsource annotation through both expert and nonexpert annotators, which would address the primary weaknesses of BrainMap (i.e., slow growth) and Neurosynth (i.e., noise in data extraction and annotation). Annotations in Brainspell may use labels from the Cognitive Paradigm Ontology (CogPO) (Turner and Laird, 2012), an ontology adapted from the BrainMap Taxonomy, or from the Cognitive Atlas (Poldrack et al., 2011), a collaboratively generated ontology built by contri-
butions from experts across the field of cognitive science. Users may also correct
the coordinates extracted by Neurosynth, which may suffer from extraction errors,
and may add important metadata like the number of subjects associated with each
comparison in each study.

Brainspell has suffered from low growth, which is why its annotations have not
been integrated back into Neurosynth, but a new frontend tool for Brainspell, geared
toward meta-analysts, has been developed called metaCurious. MetaCurious facili-
tates neuroimaging meta-analyses by allowing users to iteratively perform literature
searches and to annotate rejected articles with reasons for exclusion. In addition
to these features, metaCurious users can annotate studies with the same labels and
metadata as Brainspell, but with the features geared toward meta-analysts site usage
is expected to exceed that of Brainspell proper.

While NiMARE does not natively include tools for interacting with Brainspell or
metaCurious, there are plans to support NiMARE-format exports in both services.

3.14.2 Seed-based D-Mapping

Seed-based d-mapping (SDM) (Radua et al., 2012), previously known as signed
differential mapping, is a relatively recently-developed approach designed to in-
corporate both peak-specific effect size estimates and unthresholded images, when
available. In SDM, foci are convolved with an anisotropic kernel which, unlike the
Gaussian and spherical kernels employed in ALE and MKDA, respectively, accounts
for tissue type to provide more empirically realistic spatial models of the clusters
from the original studies. The SDM algorithm is not yet supported in NiMARE,
given the difficulty in implementing an algorithm without access to code.
3.14.3 Model-based CBMA

Model-based algorithms, a recent alternative to kernel-based approaches, model foci from studies as the products of stochastic models sampling some underlying distribution. Some of these methods include the Bayesian hierarchical independent cluster process model (BHICP) (Kang et al., 2011), the Bayesian spatially adaptive binary regression model (SBR) (Yue et al., 2012), the hierarchical Poisson/Gamma random field model (HPGRF/BHPGM) (Kang et al., 2014), the spatial Bayesian latent factor regression model (SBLFRM) (Montagna et al., 2018), and the random effects log Gaussian Cox process model (RFX-LGCP) (Samartsidis et al., 2019).

Although these methods are much more computationally intensive than kernel-based algorithms, they provide information that kernel-based methods cannot, such as spatial confidence intervals, effect size estimate confidence intervals, and the facilitation of reverse inference. A more thorough description of the relative strengths of model-based algorithms is presented in Samartsidis et al. (2017), but these benefits, at the cost of computational efficiency, have led the authors to recommend kernel-based methods for exploratory analysis and model-based methods for confirmatory analysis.

NiMARE does not currently implement any model-based CBMA algorithms, although there are plans to include at least one in the future.

3.14.4 Additional automated annotation methods

Several papers have used article text to automatically annotate meta-analytic databases with a range of methods. Alhazmi et al. (2018) used a combination of correspondence analysis and clustering to identify subdomains in the cognitive neuroscience literature from Neurosynth text. Monti et al. (2016) generated word and
document embeddings in vector space from Neurosynth abstracts using deep Boltzmann machines, which allowed them to cluster words based on semantic similarity or to describe Neurosynth articles in terms of these word clusters. Nunes (2018) used article abstracts from Neurosynth to represent documents as dense vectors as well. These document vectors were then used in conjunction with corresponding coordinates to cluster words into categories, essentially annotating Neurosynth articles according to a new “ontology” based on both abstract text and coordinates.

Meta-analytic databases may also be used in conjunction with existing ontologies in order to redefine mental states or to refine the ontology. For example, Yeo et al. (2016) used the Author-Topic model to identify connections between Paradigm Classes (i.e., tasks) and Behavioral Domains (i.e., mental states) from the BrainMap Taxonomy using the BrainMap database. Other examples include using meta-analytic clustering, combined with functional decoding, to identify groups of terms/labels that co-occur in neuroimaging data, in order to determine if the divisions currently employed in existing ontologies accurately reflect how mental states are separated in the mind (e.g., Laird et al. (2015); Riedel et al. (2018); Bottenhorn et al. (2019)).

3.15 Summary

The advent of open, large-scale databases of neuroimaging results, whether full, un-thresholded statistical maps or simple coordinates, has allowed for the development of a wide variety of methods for performing fMRI meta-analyses and related analyses. These methods are often (but not always) released as tools for the community to use, written in a range of languages and with highly variable interfaces. As a consequence, it is difficult for meta-analysts to keep abreast of the current literature
and to employ whatever method is most appropriate to address a given question. NiMARE provides a centralized repository for these tools, which will make it easier for researchers to keep track of new methods, and also provides said tools with extensive documentation and a standardized programmatic interface, which will allow researchers to use whatever tool is most appropriate for their research, without unnecessarily steep learning curves.

Given that NiMARE is open source and collaboratively developed on GitHub, methodologists may contribute their own meta-analytic algorithms directly, or interested third parties may implement these algorithms using papers or external tools as a basis for understanding the methods.

3.16 Acknowledgements

We would like to thank Yifan Yu and Jérôme Dockès, who provided feedback on the manuscript.

This work was partially funded by the National Institutes of Health (NIH) NIH-NIBIB P41 EB019936 (ReproNim), NIH-NIMH R01 MH083320 (CANDIShare), and NIH RF1 MH120021 (NIDM), the National Institute Of Mental Health under Award Number R01MH096906 (Neurosynth), as well as the Canada First Research Excellence Fund, awarded to McGill University for the Healthy Brains for Healthy Lives initiative and the Brain Canada Foundation with support from Health Canada.
4.1 Summary

Multi-echo denoising is a processing technique that leverages the manner in which fMRI signal decays over time in order to separate blood oxygenation-level dependent (BOLD) signals from non-BOLD noise. While this method is quite useful for improving fMRI signal-to-noise with limited spatial and temporal resolution costs, there are forms of noise that exhibit the same signal decay characteristics as neuronal signal. A recent publication by Power et al. (2018a) found that multi-echo denoising is unable to identify BOLD-based, spatially-diffuse noise caused by respiration-related motion, and further proposed global signal regression as a useful method for removing this kind of noise. In this manuscript, we replicate and extend the findings of Power et al. (2018) across a range of open-access multi-echo fMRI datasets in order to reproduce the original paper’s results, as well as to determine if global signal regression is an appropriate technique for dealing with the forms of noise that multi-echo denoising cannot.

4.2 Introduction

Multi-echo fMRI provides superior signal-to-noise compared to single-echo fMRI, but so far has not been widely adopted by the neuroimaging community. This is largely due to a lack of resources for multi-echo data, including available sequences, established protocols, public datasets, and processing software. Here we focus pri-
mainly on processing software. Of the three most widely used software packages (FSL, AfNI, and SPM), only AfNI includes multi-echo-specific processing options, in the form of meica.py (https://github.com/ME-ICA/me-ica), which is a workflow for multi-echo denoising that employs signal decomposition methods to identify and remove components that exhibit characteristics of non-BOLD noise. This is changing, with the development of a community-driven package for the tedana pipeline (Kundu et al., 2012b, 2013b; DuPre et al., 2018), as well as integration of multi-echo fMRI preprocessing methods into the fMRIPrep pipeline (Esteban et al., 2018).

Multi-echo denoising works by leveraging signal decay features of BOLD and non-BOLD signal over echo times. While signal decay can best be characterized according to a multi-compartment model (Speck et al., 2001; Havlicek et al., 2017; Kang et al., 2018), it is generally approximated with a monoexponential decay model (Kundu et al., 2017). This monoexponential decay curve is determined by two variables: $S_0$ and $T_2^*$, as shown in Equation 4.1, where $S_{TE_n}$ is the BOLD signal measured at a given echo time ($TE$):

$$S_{TE} = S_0 e^{-\frac{TE}{T_2^*}}$$

(4.1)

Changes in BOLD signal are reflected primarily in $T_2^*$, while non-BOLD signal fluctuations primarily impact $S_0$. Tedana uses multiple echoes to estimate voxelwise $S_0$ and $T_2^*$ across all timepoints in the fMRI run, resulting in average $S_0$ and $T_2^*$ maps. Independent component analysis (ICA) can be then used to decompose the runs into components. Although ICA will not perfectly delineate fMRI data into BOLD-based or non-BOLD-based components, each component will generally be driven primarily by either $S_0$ (i.e., non-BOLD) or $T_2^*$ (i.e., BOLD) fluctuations and can be classified as such by the denoising algorithm. Components classified as non-
BOLD (i.e., non-neural) can then be regressed from the run to produce a denoised version of the dataset, while specifically retaining BOLD-related signal.

In this registered report, we replicate and extend a recent paper (Power et al., 2018a) evaluating the ability of multi-echo preprocessing pipelines to isolate and remove motion-related noise from BOLD data. In the original paper, the authors found that standard multi-echo denoising steps are able to remove spatially localized, non-BOLD, motion-related noise, but are unable to remove spatially diffuse, BOLD-based noise ostensibly associated with respiration and respiration-related motion. They instead found that techniques for removing global artifacts, such as global signal regression (GSR), Go decomposition (GODEC), or robust principal components analysis, are necessary to eliminate motion-related artifacts caused by this noise from multi-echo data.

There is strong evidence in the literature supporting Power et al.’s findings that some physiological noise (characterized in the paper as respiration-related, spatially diffuse motion) is BOLD-based (Tong et al., 2019; Petridou et al., 2009). Multi-echo denoising exploits the manner in which fluctuations in $T_2^*$ and $S_0$ scale with echo time in order to separate BOLD from non-BOLD. While this makes multi-echo denoising good at removing noise that is definitively non-BOLD, it is naturally unable to separate noise that is $T_2^*$-based, but non-neural - such as respiratory variations - from neuronal BOLD signal. This motivates the use of additional denoising methods to deal with $T_2^*$-based noise, but the primary recommendation from Power et al. (2018a), GSR, is widely considered controversial (Liu et al., 2017; Murphy and Fox, 2017), as it may introduce spurious negative correlations (Colenbier et al., 2020) and necessarily combines neuronal and non-neuronal BOLD signal in its nuisance regressor (Tong et al., 2019; Uddin, 2020).
Tong et al. (2019) characterize global signal as largely composed of systemic low-frequency oscillations. These oscillations spread across the brain at hemodynamic speeds, moving along the vasculature. As such, the same oscillation impacts different parts of the brain at different times, in which case averaging signal across the brain (i.e., global signal) blurs these lagged signals over time. Additionally, global signal conflates stationary neuronal signal with these systemic low-frequency oscillations (sLFOs), which are essentially low-frequency BOLD noise waveforms traveling across the vasculature. Given the nature of global signal as a combination of neuronal and non-neuronal signal that potentially blurs a nonstationary signal over time, it is necessary to examine GSR not just as a method to remove spatially diffuse noise, but also as one that preserves neuronal signal in denoised data.

Our aims for this study were twofold:

1. We wished to perform an independent replication of the physiology-related findings from Power et al. (2018a), in order to determine if respiration is the source of the motion-related signal that persists after multi-echo denoising.

2. We sought to extend this work with additional methods for removing spatially diffuse noise, in order to evaluate the ability of the proposed methods to eliminate motion-dependent patterns in functional connectivity in a common preprocessing framework.

To accomplish these aims, we used three independent, publicly-available multi-echo fMRI datasets, representing almost the entirety of such data at the time we developed our analysis plan. For the first aim, we analyzed a dataset with resting-state fMRI and physiological data from 31 healthy, adult participants (DuPre et al., 2016), which will be referred to as the DuPre dataset. For the second aim, we combined the DuPre dataset with two more datasets: one dataset used by Power et al.,
which includes resting-state fMRI, without physiological data, from 89 healthy, adult participants (the Cambridge dataset) (Kundu et al., 2013b), and another dataset with film-viewing fMRI, without physiological data, from 649 adult participants (the CamCAN dataset) (Taylor et al., 2017; Shafto et al., 2014). Before replicating analyses in Power et al. (2018a), we preprocessed these three datasets using fMRIPrep in line with current best practices. We extended the analyses by Power et al. (2018a) to produce additional products of the multi-echo denoising pipeline and evaluate additional methods for removing respiration-related artifacts, including dynamic global signal regression and minimum image regression. Dynamic global signal regression (dGSR) uses rapid time delay analysis to remove time-lagged physiological noise (Tong et al., 2019; Erdoğan et al., 2016) and may be better able to remove slow-moving, BOLD-based noise compared to standard GSR. Minimum image regression (MIR; see description in 4.3.1) was added to tedana as the preferred method for removing global artifacts by a subset of the authors of the original Power paper, but has not yet been evaluated in the same manner as GODEC, GSR, and aCompCor.

The two aims and the datasets used for each are outlined in Figure 4.1.
Figure 4.1 A schematic of the analyses performed in this replication, along with the dataset(s) used for each.

This manuscript includes a large number of abbreviations. These abbreviations are collected in Table 4.1 to aid the reader.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOLD</td>
<td>Blood-oxygen-level-dependent</td>
</tr>
<tr>
<td>CSF</td>
<td>Cerebrospinal fluid</td>
</tr>
<tr>
<td>dGSR</td>
<td>Dynamic global signal regression. The method implemented by the rapidtide toolbox.</td>
</tr>
<tr>
<td>FD</td>
<td>Framewise displacement</td>
</tr>
<tr>
<td>FIT</td>
<td>Volume-wise estimation of $T_2^*$ and $S_0$</td>
</tr>
<tr>
<td>fMRI</td>
<td>Functional magnetic resonance imaging</td>
</tr>
<tr>
<td>GODEC</td>
<td>Go decomposition</td>
</tr>
<tr>
<td>GSR</td>
<td>Global signal regression</td>
</tr>
<tr>
<td>HCP</td>
<td>Human Connectome Project</td>
</tr>
<tr>
<td>MEDN</td>
<td>Multi-echo denoised</td>
</tr>
<tr>
<td>MEHK</td>
<td>Multi-echo high Kappa</td>
</tr>
<tr>
<td>MEICA</td>
<td>Multi-echo independent components analysis. Used in this manuscript to refer to the full ICA-based denoising pipeline implemented in tedana.</td>
</tr>
<tr>
<td>MIR</td>
<td>Minimum image regression. A novel denoising method implemented in MEICA by Dr. Prantik Kundu and evaluated in this manuscript.</td>
</tr>
<tr>
<td>OC</td>
<td>Optimally combined</td>
</tr>
<tr>
<td>QC:RSFC</td>
<td>Quality control-resting state functional connectivity</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of interest</td>
</tr>
<tr>
<td>RPV</td>
<td>Respiratory pattern variability</td>
</tr>
<tr>
<td>RRF</td>
<td>Respiratory response function</td>
</tr>
<tr>
<td>RV</td>
<td>Respiratory variance</td>
</tr>
<tr>
<td>RVT</td>
<td>Respiratory-volume-per-time</td>
</tr>
<tr>
<td>sLFO</td>
<td>Systemic low-frequency oscillation</td>
</tr>
<tr>
<td>STD</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>TE_{30}</td>
<td>Echo time closest to 30ms</td>
</tr>
<tr>
<td>tedana</td>
<td>TE-Dependent ANAlysis</td>
</tr>
<tr>
<td>TEDICA</td>
<td>TE-Dependent independent components analysis. Used in this manuscript to refer specifically to the ICA step in the overall MEICA workflow.</td>
</tr>
<tr>
<td>TR</td>
<td>Repetition time</td>
</tr>
<tr>
<td>WM</td>
<td>White matter</td>
</tr>
</tbody>
</table>
4.3 Experiment 1: Independent Replication of Respiration Analyses

4.3.1 Materials & Methods

The first experiment attempted to replicate and extend analyses from the original paper relating to physiological traces and multi-echo processing. In Power et al. (2018a), a resting state fMRI dataset of 12 participants with physiological recordings was used to suggest that multi-echo processing is unable to remove noise specifically associated with respiration-related motion and changes in BOLD signal.

In the original paper, eleven analyses probed the relationship between physiological trace data, motion parameters, and global signal in denoised and un-denoised data. Each of these analyses was a bivariate correlation between participant-level summary measures of physiological trace variability and participant-level summary measures of variability of global signal in fMRI data (either before or after denoising) or summary measures of overall motion. Brief descriptions of these analyses are provided in Table 4.2, with more detailed information available under the “Analyses” section for this experiment. While variability of global signal is not necessarily an indication of noise, given that global signal is composed of both non-neuronal noise and neuronal signal (Uddin, 2020), these analyses were used to investigate the relationship between global signal and physiological noise.
Table 4.2 **Results from experiment 1 power analyses.** The six columns represent the name of the analysis, the family (or group of analyses) to which the analysis was assigned, the original analysis’ correlation coefficient, the original analysis’ sample size, the replication analysis’ sample size, and estimated statistical power for the replication analysis, respectively.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Family</th>
<th>Correlation Coefficient</th>
<th>Original N</th>
<th>Replication N</th>
<th>Statistical Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPV correlated with SD of global fMRI signal from TE2 data</td>
<td>1</td>
<td>0.69</td>
<td>12</td>
<td>31</td>
<td>0.98</td>
</tr>
<tr>
<td>RPV correlated with SD of global fMRI signal from FIT R₂⁺ data</td>
<td>1</td>
<td>0.64</td>
<td>12</td>
<td>31</td>
<td>0.94</td>
</tr>
<tr>
<td>RPV correlated with SD of global fMRI signal from MEICA-denoised data</td>
<td>1</td>
<td>0.59</td>
<td>12</td>
<td>31</td>
<td>0.88</td>
</tr>
<tr>
<td>RPV correlated with SD of global fMRI signal from MEICA-denoised +</td>
<td>1</td>
<td>0.04</td>
<td>12</td>
<td>31</td>
<td>0.0077</td>
</tr>
<tr>
<td>GODEC data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPV correlated with SD of global fMRI signal from MEICA-denoised +</td>
<td>1</td>
<td>0.64</td>
<td>12</td>
<td>31</td>
<td>0.94</td>
</tr>
<tr>
<td>nuisance regression (motion, motion derivatives, mean e4 white, mean e2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ventricle) data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPV correlated with SD of global fMRI signal from MEICA-denoised +</td>
<td>1</td>
<td>0.51</td>
<td>12</td>
<td>31</td>
<td>0.71</td>
</tr>
<tr>
<td>RVT-Reg data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPV correlated with SD of global fMRI signal from MEICA-denoised +</td>
<td>1</td>
<td>0.52</td>
<td>12</td>
<td>31</td>
<td>0.73</td>
</tr>
<tr>
<td>RV-Reg data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HRV correlated with SD of global fMRI signal from TE2 data</td>
<td>2</td>
<td>0.07</td>
<td>12</td>
<td>31</td>
<td>0.024</td>
</tr>
<tr>
<td>HRV correlated with SD of global fMRI signal from FIT R₂⁺ data</td>
<td>2</td>
<td>0.03</td>
<td>12</td>
<td>31</td>
<td>0.017</td>
</tr>
<tr>
<td>HRV correlated with SD of global fMRI signal from MEICA-denoised data</td>
<td>2</td>
<td>-0.16</td>
<td>12</td>
<td>31</td>
<td>0.062</td>
</tr>
<tr>
<td>RPV correlated with mean FD</td>
<td>3</td>
<td>0.73</td>
<td>12</td>
<td>31</td>
<td>1</td>
</tr>
</tbody>
</table>
In addition to replicating the original analyses, in this experiment we evaluated the novel metric developed in the original paper as a measure of omnibus respiration activity, \textit{respiratory pattern variability}, given that it was used to conclude that respiration-related noise is preserved even after regressing out respiration from the data. The methods used to remove spatially diffuse noise after multi-echo denoising were also extended to include dynamic global signal regression and minimum image regression, two approaches which have informally been used with multi-echo data, but which have yet to be evaluated in comprehensive experiments.

\section*{Data}

The dataset used for this experiment was obtained from OpenNeuro (\url{https://openneuro.org}) (Poldrack and Gorgolewski, 2017). The dataset (OpenNeuro accession number ds000210 v2) (DuPre et al., 2016), which we refer to as the DuPre dataset, was not used in the original Power paper.

In the DuPre dataset, data for 31 participants were acquired using a 3-Tesla GE Discovery MR750 MRI scanner. T1-weighted structural MRI data were collected (176 slices; repetition time, TR=7.7ms; echo time, TE=3.4ms; flip angle, FA=7°; field of view, FOV=256x256mm; matrix size=256x256; voxel size=1x1x1mm). In addition to task data (not used in this study), one run of eyes-open rest multi-echo fMRI data was collected with three echoes (46 slices in an interleaved ascending order; repetition time, TR=3000ms; echo time, TE=13.7, 30, and 47ms; flip angle, FA=83°; field of view, FOV=216x216mm; matrix size=72x72; voxel size=3x3x3mm, 2.5x in-slice acceleration with sensitivity encoding). Each run was 10:12 minutes in length, during which 204 functional volumes were acquired. Dicoms were converted to NIfTI-1 format prior to processing. Physiological data were acquired simultaneously with the functional runs. Specifically, respiration and pulse were acquired
using respiratory belt and pulse oximeter, respectively, at 40 Hertz for 15 participants and 50 Hertz for 16 participants. Demographic and acquisition information for the DuPre dataset (as well as all other datasets used in this replication) are provided in Tables 4.3 and 4.4, respectively.

Table 4.3 Demographics for samples. One CamCAN participant does not have any demographic information. The DuPre dataset includes gender, rather than sex; we have chosen to treat gender as sex for the sake of simplicity. No demographic information is available for the NA or Cambridge datasets. *The NA dataset was used in the original paper, but was unavailable for this replication.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NA*</th>
<th>DuPre</th>
<th>Cambridge</th>
<th>CamCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (before QC)</td>
<td>12</td>
<td>31</td>
<td>89</td>
<td>649</td>
</tr>
<tr>
<td>Female/Male</td>
<td>–</td>
<td>16/15</td>
<td>–</td>
<td>329/319</td>
</tr>
<tr>
<td>Mean Age (STD)</td>
<td>–</td>
<td>22.39 (3.25)</td>
<td>–</td>
<td>54.81 (18.56)</td>
</tr>
<tr>
<td>Age Range</td>
<td>–</td>
<td>18 - 31</td>
<td>–</td>
<td>18.5 - 88.92</td>
</tr>
<tr>
<td>Handedness (R/L/A)</td>
<td>–</td>
<td>31/0/0</td>
<td>–</td>
<td>588/50/10</td>
</tr>
</tbody>
</table>

Table 4.4 Scan parameters from original and replication datasets. *The NA dataset was used in the original study, but was not available for the replication.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>NA*</th>
<th>DuPre</th>
<th>Cambridge</th>
<th>CamCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>N (before QC)</td>
<td>12</td>
<td>31</td>
<td>89</td>
<td>649</td>
</tr>
<tr>
<td># TEs</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>TEs (ms)</td>
<td>12, 24.5, 37</td>
<td>13.7, 30, 47</td>
<td>12, 28, 44, 60</td>
<td>9.4, 21.23, 33.06, 44.89, 56.72</td>
</tr>
<tr>
<td>TR (s)</td>
<td>2.5</td>
<td>3</td>
<td>2.47</td>
<td>2.47</td>
</tr>
<tr>
<td># TRs</td>
<td>300</td>
<td>204</td>
<td>239</td>
<td>193</td>
</tr>
<tr>
<td>Flip angle</td>
<td>–</td>
<td>83</td>
<td>78</td>
<td>78</td>
</tr>
<tr>
<td>In-plane acceleration factor</td>
<td>–</td>
<td>2.5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Multiband acceleration factor</td>
<td>–</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Voxel size (mm)</td>
<td>3x3x3</td>
<td>3x3x3</td>
<td>3.75x3.75x4.4</td>
<td>3x3x4.44</td>
</tr>
<tr>
<td>Physio</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Task</td>
<td>Rest</td>
<td>Rest</td>
<td>Rest</td>
<td>Film</td>
</tr>
<tr>
<td>Original Paper</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Experiment 1</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
This section was (in part) generated automatically using pybids (Yarkoni et al., 2019).

The DuPre dataset is used as a substitute for one of the two datasets used in the original Power paper. The original dataset, which we will refer to as the NA dataset, included both multi-echo fMRI data and corresponding physiological recordings, but was unavailable for this replication. The NA dataset includes 12 participants, each with two runs of resting-state fMRI (repetition time, TR=2.5s; echo time, TE=12, 24.5, and 37ms; voxel size=3x3x3mm). Each run was 6:15 minutes in length, during which 300 functional volumes were acquired. The fMRI data were accompanied with concurrent respiration and pulse oximeter data. Demographic and acquisition information for the NA dataset are provided in Tables 4.3 and 4.4, respectively.

**Statistical Power Analyses**

Statistical power analyses for the parametric analyses from the original paper were performed in R with the pwr package (Champely et al., 2018). These analyses were used to estimate the statistical power for each of the replication analyses that were performed in this experiment, given the sample available. Eleven analyses, all bivariate correlations, were evaluated. Analyses were first divided into three families based on the hypotheses tested, which were used for Bonferroni alpha corrections from a base alpha of 0.05.

Results from the power analyses are summarized in Table 4.2. Of the 11 parametric analyses using the 12-participant dataset with physiological data in the original paper, four were not statistically significant in the original paper and, as expected, the replication sample was not sufficiently powered to detect these small effects with 31 participants. From the remaining seven analyses, four were sufficiently powered in the replication sample, while three were not (1-B \( \hat{\beta} = 0.7 \)). However, in
all of these analyses, the original 12-participant sample would not be well powered to detect any effects, so it is likely that, if the original results are not false positives, then they are at least inflated estimates of effect size.

The 31-participant replication sample reflected an improvement in power over the sample from the original paper, although it was not likely to be sufficiently powered to accurately estimate the “true” effect sizes. Nevertheless, we expected that, even if the results of these analyses in the replication were adversely affected by insufficient power, the results would vary randomly due to sampling effects, which would provide important information regarding the true effect sizes when compared to the results of the original paper. Moreover, collecting a large enough sample of multi-echo fMRI data with physiological recording (an uncommon combination in cognitive neuroscience research) would be prohibitively expensive for the sake of a replication, so the use of the (admittedly limited) public dataset represented the best possible option for replication of the original paper.

**Data Censoring & Exclusion**

Data were subjected to quality control for the structural and functional MRI data, as well as the physiological trace data. To the first end, MRIQC (Esteban et al., 2017) was run on the neuroimaging data. The resulting MRIQC visual reports for both structural and functional scans were inspected for visually-identifiable artifacts, including ringing due to motion, ghosting, and extreme signal leakage across slices. Runs exhibiting any of these artifacts within the visual report were excluded from further preprocessing, multi-echo denoising, and all analyses. Physiological traces were subjected to automatic peak detection, and the resulting time courses were visually inspected to remove any spurious events.
Null distributions were derived within each dataset for two MRIQC image quality metrics: *ghosting-to-signal-ratio* and *foreground-background energy ratio*. Runs that are outliers in directions indicating poor data quality (upper for ghosting-to-signal ratio and lower for foreground-background energy ratio) were excluded from further analyses. Outliers were defined as values more than three standard deviations from the mean value in each dataset. These metrics were selected because they appeared to index scanner noise rather than motion-related noise. Given that the focus of these analyses was on motion-related artifact, motion was not used directly as a quality control metric for data exclusion.

One participant, sub-18, was excluded from further analyses in experiment 1 based on this automated quality control procedure. Additionally, sub-16 was found to have only partial physiological data, so their data were also excluded from analyses using physiological data. Finally, we detected several periods of bad signal in the cardiac data of both sub-11 and sub-26, so they were excluded from analyses using the cardiac data (i.e., experiment 1, analysis group 4).

**Data Processing**

The basic processing and target datasets for this Aim are outlined in Figure 4.2.
Figure 4.2 Flowchart of basic processing steps used to generate target datasets for Aim 1 analyses.

**fMRI and structural data: preprocessing**  Initial preprocessing of fMRI data was performed with fMRIPrep (Esteban et al., 2018).

fMRIPrep does not save echo-level data in the output directory, so these data were collected after slice timing correction and motion correction, from the working directory, renamed to match BIDS Derivatives convention, and copied into the output directory. Additionally, transforms used to warp native-space data into standard space were recovered from the working directory and copied over to the output directory.
fMRIPrep estimates the number of non-steady state volumes separately for each run. As such, after fMRIPrep preprocessing, but prior to multi-echo denoising, volumes from each run identified as non-steady state were removed and relevant regressors were time-shifted accordingly.

**Mask creation** Subsequent to fMRIPrep preprocessing, Freesurfer parcellations of the T1-weighted volumes were downsampled to 3mm isotropic voxels (the resolution of the processed functional data) using nearest neighbor interpolation. From these resampled parcellations, gray matter masks composed of the cortical ribbon, cerebellum, and subcortical nuclei were generated. Next, white matter (WM) and cerebrospinal fluid (CSF) nuisance masks were created from the high resolution (1mm\(^3\)) Freesurfer parcellations. The high resolution Freesurfer white matter parcellation mask was eroded using 0, 2, and 4 erosion cycles before resampling to functional resolution to produce three WM masks: ero0, ero2, and ero4. These WM masks were then subtracted from one another to create white matter nuisance masks for analysis: WM\(_{ero02}\), WM\(_{ero24}\), and WM\(_{ero4}\). The high resolution CSF parcellation mask was eroded using 0 and 2 erosion cycles before resampling to produce two CSF masks: ero0 and ero2. These CSF masks were then subtracted to create CSF nuisance masks for analysis: CSF\(_{ero02}\) and CSF\(_{ero2}\).

**Physiological trace processing** Respiratory-volume-per-time (RVT) was calculated using the phys2denoise toolbox ([https://github.com/physiopy/phys2denoise](https://github.com/physiopy/phys2denoise)). For our analyses, the resulting RVT time series were shifted forward 5, 10, 15, and 20 seconds, resulting in five RVT regressors, all of which were resampled to match the repetition time of the fMRI data. The resulting five regressors were also convolved with the respiratory response function (RRF), producing a
total of 10 regressors. Finally, each of the regressors were mean-centered, detrended, and z-scored.

Respiratory variance (RV) was calculated as the rolling standard deviation of the respiratory trace with a window of 6 data points (0.12 seconds). The resulting RV time series was shifted forward and backward 3 seconds, resulting in three regressors. Each of the regressors was resampled to a 3 second interval (matching the sampling rate of the fMRI data) and convolved with the RRF, resulting in six regressors. Finally, each of the regressors was mean-centered, detrended, and z-scored.

Respiratory pattern variability (RPV), a novel measure introduced in the original paper, was calculated by first z-scoring the respiratory belt recording, then calculating the upper root-mean-square envelope of the data using a window equivalent to 10 seconds of recording (i.e., 500 data points for 50 Hertz data), and finally taking the standard deviation of the envelope.

While the tools with which the physiological regressors were generated differ from the original paper, the formulae and additional processing steps are the same.

**fMRI data: processing to leverage multi-echo decay properties**  
Multi-echo independent component analysis (MEICA) was applied to the preprocessed fMRI data. MEICA produces two primary outputs for each run: the optimally combined (OC) dataset and the multi-echo denoised (MEDN) dataset. The discarded noise components from the multi-echo denoising procedure (MEDN-NOISE) were obtained by subtracting the MEDN time series from the OC time series. Each output was independently processed and analyzed at each of the remaining steps of the data processing pipeline.

The denoising pipeline works as follows. First, an adaptive mask is created from the input data, in which the number of echoes with sufficient signal are identified.
for each voxel. Given that signal decays with increasing echo time, the values correspond to ascending numbers of echoes (e.g., an adaptive mask value of 3 means that the first three echoes have good data). This adaptive mask identifies which echoes are used for subsequent steps in the pipeline. The next step is to fit a monoexponential decay model using non-linear curve optimization in order to estimate voxel-wise $T_2^*$ and $S_0$. The $T_2^*$ map is then used to optimally combine data across echoes, in which each voxel's optimally combined value corresponds to the weighted average across echoes predicting what that voxel's signal would be if it had been acquired at its true $T_2^*$. The optimally combined data are then decomposed with principal component analysis and low-variance components are removed before the data are reconstructed from the high-variance components. These dimensionally reduced optimally combined data are then subjected to TE-dependent independent component analysis (TEDICA). TEDICA involves applying independent components analysis to the dimensionally reduced data to generate a mixing matrix (components x time) and component weights (voxels x components). The mixing matrix is used to estimate component weights for each of the original echo-specific runs using linear regression. The distribution of weights across echoes is then used to fit TE-dependence and TE-independence models, generating pairs of component-specific model-fit maps. The TE-dependence model reflects the extent to which fluctuations in $T_2^*$ drive the component, while the TE-independence model reflects $S_0$ fluctuations. The average value within each brain-masked model-fit map is the overall model-fit value for the component (Kappa for the TE-dependence [$T_2^*$] model and Rho for the TE-independence [$S_0$] model). Components are identified as being high-Kappa (BOLD-like), high-Rho (noise), or low explained variance and unclear fit across models (ignored). The component selection step uses several metrics in addition to Kappa and Rho, although these are the primary metrics. In denoising
the data, the high-Rho noise components are discarded by regressing them out of
the optimally combined data, while the high-Kappa and low-variance components
are retained along with any unmodeled variability.

Additionally, in order to evaluate whether global signals are TE-dependent (and
thus pervade T_{2*} time series), we performed voxel- and volume-wise estimation of the
monoexponential decay model, in a method referred to as FIT. The FIT procedure
produces two outputs for each run: the FIT S_{0} time series (FIT-S_{0}) and the FIT T_{2*}
time series (FIT-R_{2}). By fitting the decay model to each volume separately, FIT
produces much noisier estimates of S_{0} and T_{2*}, which is why it is not recommended
for general use.

Both MEICA and FIT are implemented in tedana.

In addition to derivatives from multi-echo denoising, preprocessed data from the
echo time closest to 30ms was retained as a surrogate for single-echo data, and is
referred to as TE_{30}. This echo time was selected for the data because this is the
most common echo time for fMRI with a 3 Tesla scanner.

**fMRI data: nuisance regressions** Subsequent to multi-echo-specific denoising,
standard denoising steps were applied to remove global forms of noise. Specifically,
three models were applied to the FIT-R_{2} and MEDN datasets. In the first model,
the mean compartment signal from the deepest white matter (WM_{ero4}) and ven-
tricle (CSF_{ero2}) masks were included in a nuisance regression, along with the six
realignment parameters and their first temporal derivatives. This produced a total
of 14 regressors for the standard nuisance model. In the second model, RVT was
lagged five times at five second intervals (0s, 5s, 10s, 15s, 20s) and convolved with
the RRF at each lag. This resulted in ten RVT-based regressors, along with the
six realignment parameters and their first temporal derivatives, producing a total
of 22 regressors for the RVT nuisance model. In the third model, RV was lagged back 3 seconds and forward 3 seconds, and each regressor was convolved with the RRF. This resulted in six RV-based regressors, along with the 12 motion parameters, producing a total of 16 regressors for the RV nuisance model.

fMRI data: approaches for separating widespread signals from sparse signals

Go decomposition (GODEC), global signal regression (GSR), anatomical CompCor (aCompCor), dynamic global signal regression (dGSR), and minimum image regression (MIR) were performed on the multi-echo denoised data. GODEC, GSR, and aCompCor were employed in the original paper, while dGSR and MIR were included here as additional methods.

GODEC was performed subsequent to the tedana pipeline, using code excised from the original ME-ICA repository. Standard global signal regression (GSR) was implemented according to the description from the original Power publication with custom code. In this approach, signal was extracted from the cortical ribbon mask and averaged. Mean and linear trends from this mean cortical signal regressor and the data being denoised were then removed, after which the mean-centered, detrended cortical signal regressor was regressed out of the mean-centered, detrended voxel-wise data. aCompCor was performed with custom code by first extracting time series from the deepest white matter mask (WM$_{ero4}$) and then subjecting the WM data to principal component analysis. All code is made available as described in the Code availability statement, described in Section 4.6, below. The five components explaining the most variance were retained and included as regressors. dGSR was performed using the rapidtide library (Tong et al., 2019).

Finally, MIR was performed as part of the tedana pipeline. As MIR has not been used in any peer-reviewed publications at this point, we describe this method in more
detail than the other methods in the section below. While this section describes the method in detail for the first time, we recommend that anyone using the method cite Kundu et al. (2013b), as Dr. Prantik Kundu is the original developer of the method.

**Minimum image regression 4.3.1** Let $O$ be the matrix of optimally combined (OC) data, of shape $v \times t$, where $v$ is the number of voxels in the brain mask and $t$ is the number of timepoints in the scan, and let $M$ be the mixing matrix from the TEDICA decomposition, of shape $c \times t$, where $c$ is the number of components. Define the following:

$$
W = \{1, 2, 3, ..., c\}
$$

$$
N \in \mathbb{N}^k \ s.t \ 1 \leq k \leq c, N \subseteq W \tag{4.2}
$$

$$
A \in \mathbb{N}^l \ s.t \ 1 \leq l \leq k, A \subseteq N
$$

Where $W$ is the set of indices of all components in $M$, $N$ is the set of indices of all non-ignored components (i.e., all accepted or BOLD-like, and rejected or non-BOLD components) in $M$, and $A$ is the set of indices of all accepted (i.e., BOLD-like) components in $M$.

First, the voxel-wise means ($\overline{O} \in \mathbb{R}^v$) and standard deviations ($\sigma_O \in \mathbb{R}^v$) of the OC data are computed over time. The OC data are then z-normalized over time ($O_z \in \mathbb{R}^{v \times t}$) and the normalized OC matrix ($O_z$) is regressed on the TEDICA mixing matrix ($M \in \mathbb{R}^{c \times t}$) to construct component-wise parameter estimate maps ($B \in \mathbb{R}^{v \times c}$).

$$
O_z = BM + \epsilon, \ \epsilon \in \mathbb{R}^{v \times t} \tag{4.3}
$$

Next, $N$ is used to select rows from the mixing matrix $M$ and columns from the parameter estimate matrix $B$ that correspond to non-ignored (i.e., accepted and
rejected) components, forming reduced matrices $M_N$ and $B_N$. The normalized time series matrix for the combined ignored components and variance left unexplained by the TEDICA model is then computed by subtracting the scalar product of the non-ignored beta weight and mixing matrices from the normalized OC data time series ($O_z$). The result is referred to as the normalized residuals time series matrix ($R \in \mathbb{R}^{v \times t}$).

$$R = O_z - B_N M_N, \quad B_N \in \mathbb{R}^{v \times |N|}, \quad M_N \in \mathbb{R}^{|N| \times t} \quad (4.4)$$

We can likewise construct the normalized time series of BOLD-like components ($P \in \mathbb{R}^{v \times t}$) by multiplying similarly reduced parameter estimate and mixing matrices composed of only the columns and rows, respectively, that are associated with the accepted components indexed in $A$. The resulting time series matrix is similar to the time series matrix referred to elsewhere in the manuscript as multi-echo high-Kappa (MEHK), with the exception that the component time series have been normalized prior to reconstruction.

$$P = B_A M_A, \quad B_A \in \mathbb{R}^{v \times |A|}, \quad M_A \in \mathbb{R}^{|A| \times t} \quad (4.5)$$

The map of the T1-like effect ($m \in \mathbb{R}^{v}$) is constructed by taking the minimum across timepoints from the normalized MEHK time series ($P$) and then mean-centering across brain voxels. Let $J = \{1, ..., t\}$ denote the indices of the columns of matrix $P$, and let $p_{ij}$ denote the value of the element $P[i, j]$.

$$q_i = \min_{j \in J} p_{ij} \quad \forall i = 1, ..., v \quad (4.6)$$

$$m = q - \bar{q}, \quad q \in \mathbb{R}^{v} \quad (4.7)$$
The standardized OC time series matrix ($O_z$) is regressed on the T1-like effect map ($m$) to estimate the volume-wise global signal time series ($g \in \mathbb{R}^t$).

\[ O_z = m \otimes g + \epsilon, \quad \epsilon \in \mathbb{R}^{v \times t} \]  \hspace{1cm} (4.8)

Where $\otimes$ is the outer product.

The normalized BOLD time series matrix ($P$) is then regressed on this global signal time series ($g$) in order to estimate a global signal map ($s \in \mathbb{R}^v$) and the normalized BOLD time series matrix without the T1-like effect ($E \in \mathbb{R}^{v \times t}$).

\[ P = g \otimes s + E \]  \hspace{1cm} (4.9)

The time series matrix of BOLD-like components without the T1-like effect ($MEHK+MIR, H \in \mathbb{R}^{v \times t}$), scaled to match the original OC time series matrix, is constructed by multiplying each column of $E$ by the vector $\sigma_O$.

\[ H = E \odot \begin{pmatrix} \sigma_{O1} & \cdots & \sigma_{O1} \\ \vdots & \ddots & \vdots \\ \sigma_{Ov} & \cdots & \sigma_{Ov} \end{pmatrix} \]  \hspace{1cm} (4.10)

Where $\odot$ is the Hadamard product for element-wise multiplication of two matrices.

The MEICA-denoised time series without the T1-like effect ($MEDN+MIR, D \in \mathbb{R}^{v \times t}$) is constructed by adding the residuals time series ($R$) to the normalized BOLD time series ($E$), multiplying each column of the result by the vector $\sigma_O$, and adding back in the voxel-wise mean of the OC time series ($\bar{O}$).
The MEDN+MIR time series is typically retained for further analysis, although in this manuscript both MEDN+MIR and MEHK+MIR were retained.

Analyses

Analysis Group 1: Validation of RPV metric  Respiratory pattern variability (RPV) was introduced by the authors of the original paper as a summary metric of within-participant respiration. In order to validate this novel metric against other metrics of respiration, RPV was correlated with mean respiratory variance (RV), as well as with mean respiratory volume per unit time (RVT), across participants. Within-participant pairwise correlations were computed between the upper envelope used to calculate RPV with RV and RVT. These two distributions of correlation coefficients were z-transformed to enforce normality and were assessed for statistical significance using one-sample t-tests.

Analysis Group 2: Characterizing the relationship between respiration and head motion  The original paper showed that respiratory pattern variability was statistically significantly and positively correlated with mean framewise displacement. Statistical power analysis of the original finding indicated that the DuPre dataset would be sufficiently powered to detect the same effect.

As an extension to the original paper, the z-transformed correlation coefficients for within-participant correlations between respiratory volume per unit time (RVT) and framewise displacement, as well as between respiratory variance (RV) and frame-
wise displacement, were plotted in scatter plots and assessed with one-sample t-tests. These additional analyses evaluated whether respiration and head motion are related at the time series level. The distribution of z-transformed correlation coefficients between RVT and FD were predicted to be statistically significantly greater than zero, as was the distribution of coefficients between RV and FD.

Analysis Group 3: Characterizing the relationship between respiration and global BOLD signal with and without denoising  Respiratory pattern variability was correlated with the standard deviation of the mean cortical signal from each of the following inputs: TE_{30} (the echo time closest to 30ms, after pre-processing), FIT-R_2, MEDN, MEDN+GODEC, MEDN+Nuis-Reg, MEDN+RVT-Reg, and MEDN+RV-Reg. In addition to the inputs used in the original paper, correlations were performed with the inputs MEDN+aCompCor, MEDN+dGSR, MEDN+MIR, and MEDN+GSR as an extension.

Per the findings of the original paper, RPV was predicted to be statistically significantly and positively correlated with STD of global signal from TE_{30}, FIT-R_2, MEDN, MEDN+Nuis-Reg, MEDN+RVT-Reg, and MEDN+RV-Reg. Power analyses (Table 4.2) indicated that the DuPre dataset was sufficiently large to detect all of these predicted effects, except for MEDN+RVT-Reg (1-B = 0.71) and MEDN+RV-Reg (1-B = 0.73). RPV was predicted to not be statistically significantly correlated with STD of global signal from MEDN+GODEC, as in the original paper. We also predicted that the extended derivatives (MEDN+aCompCor, MEDN+dGSR, MEDN+MIR, and MEDN+GSR) would not be statistically significantly correlated with RPV.

Additionally, deep breaths were visually identified from the respiratory trace for each participant and the mean cortical signal from 30 seconds before until 40
seconds after the breath was extracted for each deep breath and plotted together in a line plot, for each of the following outputs: OC, MEDN, and MEDN+GODEC. In addition to the inputs used in the original paper, plots were generated for TE$_{30}$, FIT-R$_2$, MEDN+Nuis-Reg, MEDN+RVT-Reg, MEDN+RV-Reg, MEDN+aCompCor, MEDN+dGSR, MEDN+MIR, and MEDN+GSR.

**Analysis Group 4: Characterizing the relationship between heart rate and global BOLD signal with and without denoising**  
Heart rate variability was correlated with the standard deviation of the mean cortical signal from each of the following inputs: TE$_{30}$ (the echo time closest to 30ms, after preprocessing), FIT-R$_2$, and MEDN. HRV was predicted to not be statistically significantly correlated with STD of global signal from any of the inputs, per the findings of the original paper.

**Analysis Group 5: Characterizing the relationship between respiration and functional connectivity with and without denoising**  
Quality control: resting-state functional connectivity (QC:RSFC) and high-low motion analyses from the original paper were performed using respiration instead of motion, with RPV, mean RV, and mean RVT values as metrics of respiration, in order to evaluate the impact of respiration on resting state functional connectivity. In these analyses, elevated values relative to random expectations for ROIs 35mm apart indicate a mix of local and global effects of respiration on functional connectivity values, in that higher respiration elevates functional connectivity estimates at a rate above chance. Elevated values for the difference between the smoothing curve at 35 and 100mm index local effects of respiration specifically, in that the impact of respiration on functional connectivity increases with distance.
All tests were evaluated with a one-sided approach, in which higher values at 35mm are more significant, as are larger differences of 35mm - 100mm. Given that there is no consistent threshold for “high” respiration in the literature, the scrubbing analysis (description in Scrubbing analysis, below) was not applied here.

Regions of interest for the analyses were constructed from a set of 264 coordinates in MNI space from the Power atlas (Power et al., 2011). Each coordinate was convolved with a sphere with a 5mm radius.

**QC:RSFC respiration analysis** For each participant, the correlation between the mean time series of each pair of ROIs was computed, along with the distance between the ROIs. Across participants, the z-transformed correlation coefficients for each pair of ROIs were correlated with RPV, mean RV, or mean RVT values. The resulting correlation coefficient is referred to as the QC:RSFC r-value, where QC is defined as RPV, mean RV, or mean RVT. This resulted in a QC:RSFC r-value and a distance value (in mm) for each pair of ROIs. A smoothing curve (i.e., a moving average of 1000 data points) was computed across the QC:RSFC r-values over sorted distances. For example, the first point in the smoothing curve would be the mean QC:RSFC r-value for the first 1000 pairs of ROIs with the shortest distances between them.

10,000 permutations were performed in which the QC metric was permuted across participants, QC:RSFC correlations were performed, and the smoothing curve was calculated. The rank of the real smoothing curve at 35 mm compared to the permuted smoothing curves is interpreted as a p-value indexing general dependence on motion. The rank of the difference from 100 mm to 35 mm is interpreted as a p-value indexing distance-dependence.
**High-low respiration analysis**  For each participant, the correlation between the mean time series of each pair of ROIs was computed, as was the physical distance between the ROIs. Participants were divided into high and low respiratory variability groups using a median split of RPV, mean RV, and mean RVT. For each pair of ROIs, the difference in functional connectivity values was computed between the high and low respiration groups. This resulted in a difference value (in delta-r) and a distance value (in mm) for each pair of ROIs. A smoothing curve was then computed for these difference values over distance in the same manner as the QC:RSFC analysis.

10,000 permutations were performed in which the assignment of high vs. low respiration group is randomized across participants, delta-r values were calculated, and the smoothing curve was generated. The rank of the real smoothing curve at 35 mm compared to the permuted smoothing curves is interpreted as a p-value indexing general dependence on motion. The rank of the difference from 100 mm to 35 mm is interpreted as a p-value indexing distance-dependence.

**Analysis Group 6: Does TEDANA retain global BOLD signal in BOLD ICA components?**  In order to determine if TEDICA components retain global signal associated with motion or respiration, time series for ICA components were retained and analyzed, as in the original paper.

Carpet plots with associated line plots for motion and physiological traces were generated for each participant from the ICA time series, separated by classification (retained BOLD-like, discarded non-BOLD-like, and retained indeterminate). These carpet plots were visualized in conjunction with carpet plots of the optimally combined data, in order to identify whether vertical bands (i.e., changes in global signal) present in the optimally combined data were also present in the ICA compo-
nents. Additionally, as an extension to the original paper, these vertical bands were visually compared against large changes in the physiological trace time series. It was predicted that vertical bands present in the OC data would be mostly noticeable in BOLD-like components, and that those bands would align with large movements, although there were no hypotheses regarding vertical bands aligned with physiological traces. It is important to note that, while carpet plots are useful for artifact detection, (1) they are an inherently qualitative tool and (2) banding in carpet plots naturally reflects global signal changes, so methods that remove global signal will also reduce banding, enforcing the incorrect assumption that any global signal is inherently noise. As such, findings based on carpet plots should be interpreted carefully.

Component time series were also correlated with the mean cortical signal from the OC dataset. The percentage and number of BOLD-like and non-BOLD-like components correlated with the cortical signal at $r > 0.5$ and $r > 0.3$ were recorded across participants. Additionally, the average correlation coefficient for BOLD-like and non-BOLD-like components with the mean cortical signal was calculated for each participant, and the distributions of correlation coefficients were compared to zero and to one another with t-tests, after transforming the coefficients to z-values. The distribution of z-transformed correlation coefficients for BOLD-like components was predicted to be statistically significantly greater than zero, while the distribution for non-BOLD-like components was not predicted to be statistically significantly different from zero. The distribution of BOLD-like components was predicted to be statistically significantly greater than the distribution of non-BOLD-like components.

Finally, the mean cortical signal of the MEDN dataset was correlated with the mean cortical signal of the OC dataset for each participant, and the distribution of
z-transformed coefficients was compared to zero with a one-sample t-test in order to determine if global signal is preserved from data before denoising to data after denoising. The distribution of coefficients was predicted to be statistically significantly higher than zero.

4.3.2 Results

Analysis Group 1: Validation of RPV metric

The first two analyses of this analysis group were defined as a family, for which Bonferroni correction was applied. As such, the alpha value for the first two analyses was $0.05/2 = 0.025$.

Using this alpha value and a one-tailed test of significance, RPV and mean RV were not found to be statistically significantly correlated, $r(27) = -0.08$, $p = 0.661$. RPV and mean RVT were also not found to be statistically significantly correlated, $r(27) = 0.09$, $p = 0.324$. The results are presented in Figure 4.3.

![Figure 4.3](image-url)

Figure 4.3 **Experiment 1, Analysis Group 1, Part A.** Correlations between RPV and mean RV, as well as RPV and mean RVT.
The second pair of analyses in this analysis group were also defined as a family, with an associated alpha value of 0.025.

Correlations between the upper envelope used to calculate RPV and RV (M[Z] = 0.563, SD[Z] = 0.147) were significantly higher than zero, t(28) = 20.269, p < 0.001. Correlations between the upper envelope used to calculate RPV and RVT (M[Z] = 0.403, SD[Z] = 0.185) were also significantly higher than zero, t(28) = 11.493, p < 0.001. The distributions of the Fisher’s z-transformed correlation coefficients between RPV and RV, and RPV and RVT, are presented in Figure 4.4.

Given that RPV is defined as the standard deviation of the upper envelope, while RV and RVT were averaged before they were compared to RPV, the significant correlation between

![Figure 4.4](image_url)

**Figure 4.4** **Experiment 1, Analysis Group 1, Part B.** Distributions of Fisher’s z-transformed correlation coefficients between RPV and RV, and RPV and RVT.
Analysis Group 2: Characterizing the relationship between respiration and head motion

As in the power analyses, all three of the analyses in this analysis group were defined as a family, to which Bonferroni correction was applied. The associated alpha value for these analyses was $0.05/3 = 0.017$.

RPV and mean FD were found to be positively and statistically significantly correlated, $r(27) = 0.64$, $p < 0.001$. Correlations between RVT and FD ($M[Z] = 0.033$, $SD[Z] = 0.185$) were not significantly higher than zero, $t(28) = 0.940$, $p = 0.178$. Correlations between RV and FD ($M[Z] = 0.172$, $SD[Z] = 0.243$) were significantly higher than zero, $t(28) = 3.745$, $p < 0.001$.

The results are shown in Figure 4.5.
Figure 4.5 **Experiment 1, Analysis Group 2.** (A) A scatter plot of mean framewise displacement and respiratory pattern variability across participants in the DuPre dataset. (B) The distribution of Fisher’s z-transformed correlation coefficients between framewise displacement and respiratory volume-per-time. (C) The distribution of Fisher’s z-transformed correlation coefficients between framewise displacement and respiratory variance.
Analysis Group 3: Characterizing the relationship between respiration and global BOLD signal with and without denoising

The eleven analyses in this analysis group were defined as a family, to which Bonferroni correction was applied. The associated alpha value for these analyses was 0.05/11, 0.0045.

RPV and standard deviation of mean cortical signal of TE30 data were found to be positively and statistically significantly correlated, $r(27) = 0.56$, $p = 0.001$. RPV and standard deviation of mean cortical signal of MEDN data were found to be statistically significantly correlated, $r(27) = 0.52$, $p = 0.002$. RPV and standard deviation of mean cortical signal of MEDN+MIR data were found to be statistically significantly correlated, $r(27) = 0.64$, $p < 0.001$. RPV and standard deviation of mean cortical signal of FIT-R2 data were found to be statistically significantly correlated, $r(27) = 0.53$, $p = 0.001$. RPV and standard deviation of mean cortical signal of MEDN+GODEC (sparse) data were not found to be statistically significantly correlated, $r(27) = 0.02$, $p = 0.467$. RPV and standard deviation of mean cortical signal of MEDN+dGSR data were found to be statistically significantly correlated, $r(27) = 0.60$, $p < 0.001$. RPV and standard deviation of mean cortical signal of MEDN+aCompCor data were not found to be statistically significantly correlated, $r(27) = 0.03$, $p = 0.446$. RPV and standard deviation of mean cortical signal of MEDN+GSR data were not found to be statistically significantly correlated, $r(27) = 0.14$, $p = 0.241$. RPV and standard deviation of mean cortical signal of MEDN+Nuis-Reg data were found to be statistically significantly correlated, $r(27) = 0.55$, $p = 0.001$. RPV and standard deviation of mean cortical signal of MEDN+RV-Reg data were not found to be statistically significantly correlated, $r(27) = 0.47$, $p = 0.005$. RPV and standard deviation of mean cortical signal of MEDN+RVT-Reg data were found to be statistically significantly correlated, $r(27)$
= 0.52, p = 0.002. The results of these analyses and the corresponding coefficients from the original paper are presented in Table 4.5.

Table 4.5 **Experiment 1, Analysis Group 3.** Coefficients of correlations between RPV and standard deviation of global signal from different denoising derivatives. Analyses with p < 0.05 are bolded.

<table>
<thead>
<tr>
<th>Derivative</th>
<th>Power et al. (2018)</th>
<th>Replication</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE30</td>
<td><strong>0.69</strong></td>
<td><strong>0.56</strong></td>
</tr>
<tr>
<td>MEDN</td>
<td><strong>0.59</strong></td>
<td><strong>0.52</strong></td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>n/a</td>
<td><strong>0.64</strong></td>
</tr>
<tr>
<td>FIT-R2</td>
<td><strong>0.64</strong></td>
<td><strong>0.53</strong></td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>-0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>n/a</td>
<td><strong>0.60</strong></td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>n/a</td>
<td>0.03</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>n/a</td>
<td>0.14</td>
</tr>
<tr>
<td>MEDN+Nuis-Reg</td>
<td><strong>0.64</strong></td>
<td><strong>0.55</strong></td>
</tr>
<tr>
<td>MEDN+RV-Reg</td>
<td><strong>0.52</strong></td>
<td><strong>0.47</strong></td>
</tr>
<tr>
<td>MEDN+RVT-Reg</td>
<td><strong>0.51</strong></td>
<td><strong>0.52</strong></td>
</tr>
</tbody>
</table>

The results for TE30, MEDN, FIT-R2, and MEDN+GODEC are presented in Figure 4.6. Results for the other comparisons are presented in the supplement.
Figure 4.6 **Experiment 1, Analysis Group 3.** (A) A scatter plot of respiratory pattern variability and standard deviation of the mean cortical signal of the TE30 data, across participants, in the DuPre dataset. (B) A scatter plot of respiratory pattern variability and standard deviation of the mean cortical signal of the FIT-R2 data, across participants, in the DuPre dataset. (C) A scatter plot of respiratory pattern variability and standard deviation of the mean cortical signal of the MEDN data, across participants, in the DuPre dataset. (D) A scatter plot of respiratory pattern variability and standard deviation of the mean cortical signal of the MEDN+GODEC data, across participants, in the DuPre dataset.
Analysis Group 4: Characterizing the relationship between heart rate and global BOLD signal with and without denoising

The three analyses in this analysis group were defined as a family, to which Bonferroni correction was applied. The associated alpha value for these analyses was 0.05/3, 0.017.

HRV and standard deviation of mean cortical TE30 signal were not found to be statistically significantly correlated, $r(25) = 0.34, p = 0.041$. HRV and standard deviation of mean cortical FIT-R2 signal were not found to be statistically significantly correlated, $r(25) = 0.30, p = 0.065$. HRV and standard deviation of mean cortical MEDN signal were not found to be statistically significantly correlated, $r(25) = 0.26, p = 0.097$.

Table 4.6 Experiment 1, Analysis Group 4. Coefficients of correlations between HRV and standard deviation of global signal from different denoising derivatives. Analyses with $p < 0.05$ are bolded.

<table>
<thead>
<tr>
<th>Derivative</th>
<th>Power et al. (2018)</th>
<th>Replication</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE30</td>
<td>0.07</td>
<td>0.34</td>
</tr>
<tr>
<td>MEDN</td>
<td>-0.16</td>
<td>0.26</td>
</tr>
<tr>
<td>FIT-R2</td>
<td>0.03</td>
<td>0.30</td>
</tr>
</tbody>
</table>

The results are shown in Figure 4.7.
Figure 4.7 **Experiment 1, Analysis Group 4.** (A) A scatter plot of heart rate variability and standard deviation of the mean cortical signal of the TE30 data, across participants, in the DuPre dataset. (B) A scatter plot of heart rate variability and standard deviation of the mean cortical signal of the FIT-R2 data, across participants, in the DuPre dataset. (C) A scatter plot of heart rate variability and standard deviation of the mean cortical signal of the MEDN data, across participants, in the DuPre dataset.

**Analysis Group 5: Characterizing the relationship between respiration and functional connectivity with and without denoising**

**Distance-dependent RPV-related artifact analyses** For the QC:RSFC analysis of the OC data, both the intercept (rank 9873/10000, p = 0.0127) and the slope (rank 9746/10000, p = 0.0254) were significant. For the high-low respiration analysis of the OC data, both the intercept (rank 9926/10000, p = 0.0074) and the slope (rank 9570/10000, p = 0.0430) were significant. These results indicate that optimally combined data exhibit increased correlations among participants with higher RPV values, and that the increase in correlations increases with ROI-to-ROI distance.

For the QC:RSFC analysis of the MEDN data, the intercept was significant (rank 9989/10000, p = 0.0011); the slope was not significant (rank 8215/10000, p = 0.1785). For the high-low respiration analysis of the MEDN data, the intercept was significant (rank 9796/10000, p = 0.0204); the slope was not significant (rank
7106/10000, p = 0.2894). Taken together, these results indicate that the variance retained by tedana’s denoising workflow exhibits generally increased correlations among participants with higher RPV values, but that pattern does not increase with distance.

For the QC:RSFC analysis of the MEDN Noise data, neither the intercept (rank 9357/10000, p = 0.0643), nor the slope (rank 8918/10000, p = 0.1082) was significant. For the high-low respiration analysis of the MEDN Noise data, both the intercept (rank 9784/10000, p = 0.0216) and the slope (rank 9603/10000, p = 0.0397) were significant. Given that the QC:RSFC and high-low respiration analyses produced conflicting results, interpreting their results is complicated. However, taken together, these results indicate that the variance removed by tedana’s denoising workflow exhibits somewhat increased correlations among participants with higher RPV values, and that pattern increases with distance.

For the QC:RSFC analysis of the MEDN+Nuis-Reg data, the intercept was significant (rank 9992/10000, p = 0.0008), but the slope was not (rank 3296/10000, p = 0.6704). For the high-low respiration analysis of the MEDN+Nuis-Reg data, the intercept was significant (rank 9803/10000, p = 0.0197), but the slope was not (rank 6828/10000, p = 0.3172). These results indicate that typical nuisance regression does not eliminate the relationship between RPV and ROI-to-ROI correlations, but it does eliminate the distance-dependence of that relationship.

For the QC:RSFC analysis of the MEDN+RV-Reg data, the intercept was significant (rank 9950/10000, p = 0.0050), but the slope was not (rank 8729/10000, p = 0.1271). For the high-low respiration analysis of the MEDN+RV-Reg data, neither the intercept (rank 9479/10000, p = 0.0521) nor the slope (rank 8752/10000, p = 0.1248) was significant. These results indicate that regressing respiratory variance out of the denoised data does not eliminate the relationship between RPV
and ROI-to-ROI correlations, but it does eliminate the distance-dependence of that relationship.

For the QC:RSFC analysis of the MEDN+RVT-Reg data, the intercept was significant (rank 9981/10000, p = 0.0019), but the slope was not (rank 5386/10000, p = 0.4614). For the high-low respiration analysis of the MEDN+RVT-Reg data, the intercept was significant (rank 9564/10000, p = 0.0436), but the slope was not (rank 8004/10000, p = 0.1996). These results indicate that regressing respiratory volume-per-time out of the denoised data does not eliminate the relationship between RPV and ROI-to-ROI correlations, but it does eliminate the distance-dependence of that relationship.

For the QC:RSFC analysis of the MEDN+GODEC data, neither the intercept (rank 4613/10000, p = 0.5387) nor the slope (rank 5155/10000, p = 0.4845) was significant. For the high-low respiration analysis of the MEDN+GODEC data, neither the intercept (rank 3094/10000, p = 0.6906) nor the slope (rank 4737/10000, p = 0.5263) was significant. These results indicate that GODEC eliminates the relationship between RPV and ROI-to-ROI correlations, as well as the effect of distance on that relationship.

For the QC:RSFC analysis of the MEDN+GSR data, both the intercept (rank 9745/10000, p = 0.0255) and the slope (rank 9825/10000, p = 0.0175) were significant. For the high-low respiration analysis of the MEDN+GSR data, neither the intercept (rank 6091/10000, p = 0.3909) nor the slope (rank 6612/10000, p = 0.3388) was significant. Given that the QC:RSFC and high-low respiration analyses produced conflicting results, interpreting their results is complicated. However, taken together, these results indicate that global signal regression partially reduces the relationship between RPV and functional connectivity, and the distance between ROIs still impacts this relationship.
For the QC:RSFC analysis of the MEDN+MIR data, the intercept was significant (rank 9933/10000, p = 0.0067), but the slope was not (rank 8029/10000, p = 0.1971). For the high-low respiration analysis of the MEDN+MIR data, neither the intercept (rank 8895/10000, p = 0.1105) nor the slope (rank 6032/10000, p = 0.3968) was significant. These results indicate that minimum image regression does not eliminate the relationship between RPV and ROI-to-ROI correlations, but it does eliminate the distance-dependence of that relationship.

For the QC:RSFC analysis of the MEDN+aCompCor data, neither the intercept (rank 6390/10000, p = 0.3610) nor the slope (rank 8433/10000, p = 0.1567) was significant. For the high-low respiration analysis of the MEDN+aCompCor data, neither the intercept (rank 3902/10000, p = 0.6098) nor the slope (rank 5504/10000, p = 0.4496) was significant. These results indicate that anatomical CompCor eliminates the relationship between RPV and ROI-to-ROI correlations, as well as the effect of distance on that relationship.

For the QC:RSFC analysis of the MEDN+dGSR data, both the intercept (rank 9996/10000, p = 0.0004) and the slope (rank 9744/10000, p = 0.0256) were significant. For the high-low respiration analysis of the MEDN+dGSR data, the intercept was significant (rank 9748/10000, p = 0.0252), but the slope was not (rank 5639/10000, p = 0.4361). These results indicate that dynamic global signal regression does not eliminate the relationship between RPV and ROI-to-ROI correlations, although the impact of dGSR on the effect of distance on the RPV-RSFC relationship is less clear.

**Distance-dependent mean RV-related artifact analyses**  For the QC:RSFC analysis of the MEDN+GODEC data, the intercept was not significant (rank 7745/10000, p = 0.2255), but the slope was (rank 9618/10000, p = 0.0382). For the
high-low respiration analysis of the MEDN+GODEC data, the intercept was not significant (rank 9230/10000, p = 0.0770), but the slope was (rank 9544/10000, p = 0.0456). These results indicate that GODEC eliminates the overall relationship between mean RV and functional connectivity, but that distance still impacts the relationship.

None of the other analyses using mean RV were significant. The p-values are provided in Table 4.7.

**Distance-dependent mean RVT-related artifact analyses** For the QC:RSFC analysis of the OC data, the intercept was not significant (rank 8498/10000, p = 0.1502); the slope was not significant (rank 6855/10000, p = 0.3145). For the high-low respiration analysis of the OC data, the intercept was not significant (rank 8214/10000, p = 0.1786); the slope was not significant (rank 4857/10000, p = 0.5143).

For the QC:RSFC analysis of the MEDN data, the intercept was not significant (rank 9254/10000, p = 0.0746); the slope was not significant (rank 6469/10000, p = 0.3531). For the high-low respiration analysis of the MEDN data, the intercept was significant (rank 9614/10000, p = 0.0386); the slope was not significant (rank 9431/10000, p = 0.0569).

For the QC:RSFC analysis of the MEDN Noise data, the intercept was not significant (rank 8126/10000, p = 0.1874); the slope was not significant (rank 6142/10000, p = 0.3858). For the high-low respiration analysis of the MEDN Noise data, the intercept was not significant (rank 5258/10000, p = 0.4742); the slope was not significant (rank 4470/10000, p = 0.5530).

For the QC:RSFC analysis of the MEDN+Nuis-Reg data, the intercept was not significant (rank 7916/10000, p = 0.2084); the slope was not significant (rank
For the high-low respiration analysis of the MEDN+Nuis-Reg data, the intercept was significant (rank 9562/10000, p = 0.0438); the slope was not significant (rank 9066/10000, p = 0.0934).

For the QC:RSFC analysis of the MEDN+RV-Reg data, the intercept was significant (rank 9698/10000, p = 0.0302); the slope was not significant (rank 3721/10000, p = 0.6279). For the high-low respiration analysis of the MEDN+RV-Reg data, the intercept was not significant (rank 9066/10000, p = 0.0934); the slope was not significant (rank 9018/10000, p = 0.0982).

For the QC:RSFC analysis of the MEDN+RVT-Reg data, the intercept was significant (rank 9709/10000, p = 0.0291); the slope was not significant (rank 2275/10000, p = 0.7725). For the high-low respiration analysis of the MEDN+RVT-Reg data, the intercept was significant (rank 9557/10000, p = 0.0443); the slope was not significant (rank 8045/10000, p = 0.1955).

For the QC:RSFC analysis of the MEDN+aCompCor data, the intercept was not significant (rank 1409/10000, p = 0.8591); the slope was not significant (rank 7861/10000, p = 0.2139). For the high-low respiration analysis of the MEDN+aCompCor data, the intercept was not significant (rank 8950/10000, p = 0.1050); the slope was not significant (rank 7838/10000, p = 0.2162).

For the QC:RSFC analysis of the MEDN+GODEC data, the intercept was significant (rank 9791/10000, p = 0.0209); the slope was significant (rank 9927/10000, p = 0.0073). For the high-low respiration analysis of the MEDN+GODEC data, the intercept was not significant (rank 9153/10000, p = 0.0847); the slope was not significant (rank 8927/10000, p = 0.1073).

For the QC:RSFC analysis of the MEDN+GSR data, the intercept was not significant (rank 6247/10000, p = 0.3753); the slope was not significant (rank 6593/10000, p = 0.3407). For the high-low respiration analysis of the MEDN+GSR data, the
intercept was not significant (rank 9158/10000, p = 0.0842); the slope was not significant (rank 8606/10000, p = 0.1394).

For the QC:RSFC analysis of the MEDN+MIR data, the intercept was not significant (rank 9108/10000, p = 0.0892); the slope was not significant (rank 2958/10000, p = 0.7042). For the high-low respiration analysis of the MEDN+MIR data, the intercept was not significant (rank 7070/10000, p = 0.2930); the slope was not significant (rank 8975/10000, p = 0.1025).

For the QC:RSFC analysis of the MEDN+dGSR data, the intercept was not significant (rank 8983/10000, p = 0.1017); the slope was not significant (rank 8734/10000, p = 0.1266). For the high-low respiration analysis of the MEDN+dGSR data, the intercept was significant (rank 9730/10000, p = 0.0270); the slope was not significant (rank 9144/10000, p = 0.0856).

The results are presented in Table 4.7.
Table 4.7 **Experiment 1, Analysis Group 5.** Results of the respiration-related artifact analyses in analysis group 5. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with $p < 0.05$ are bolded.

<table>
<thead>
<tr>
<th></th>
<th>RPV</th>
<th>mean RV</th>
<th>mean RVT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0127</td>
<td>0.1700</td>
<td>0.1502</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.0254</td>
<td>0.4584</td>
<td>0.3145</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0074</td>
<td>0.4013</td>
<td>0.1786</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.0430</td>
<td>0.8816</td>
<td>0.5143</td>
</tr>
<tr>
<td>MEDN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0011</td>
<td>0.1756</td>
<td>0.0746</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.1785</td>
<td>0.6121</td>
<td>0.3531</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0204</td>
<td>0.1558</td>
<td>0.0386</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.2894</td>
<td>0.1560</td>
<td>0.0569</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0643</td>
<td>0.1608</td>
<td>0.1874</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.1082</td>
<td>0.5099</td>
<td>0.3858</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0216</td>
<td>0.7498</td>
<td>0.4742</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.0397</td>
<td>0.9905</td>
<td>0.5530</td>
</tr>
<tr>
<td>MEDN+Nuis-Reg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0008</td>
<td>0.3722</td>
<td>0.2084</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.6704</td>
<td>0.4042</td>
<td>0.4189</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0197</td>
<td>0.1856</td>
<td>0.0438</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.3172</td>
<td>0.0545</td>
<td>0.0934</td>
</tr>
<tr>
<td>MEDN+RV-Reg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0050</td>
<td>0.0814</td>
<td>0.0302</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.1271</td>
<td>0.7304</td>
<td>0.6279</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0521</td>
<td>0.1275</td>
<td>0.0934</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.1248</td>
<td>0.1713</td>
<td>0.0982</td>
</tr>
<tr>
<td>MEDN+RVT-Reg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0019</td>
<td>0.0987</td>
<td>0.0291</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.4614</td>
<td>0.7026</td>
<td>0.7725</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0436</td>
<td>0.1037</td>
<td>0.0443</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.3172</td>
<td>0.0545</td>
<td>0.0934</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.3610</td>
<td>0.8466</td>
<td>0.8591</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.1567</td>
<td>0.3313</td>
<td>0.2139</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.6098</td>
<td>0.4542</td>
<td>0.1050</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.4496</td>
<td>0.0641</td>
<td>0.2162</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.5387</td>
<td>0.2255</td>
<td>0.2099</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.4845</td>
<td>0.0382</td>
<td>0.0073</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.6906</td>
<td>0.0770</td>
<td>0.0847</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.5263</td>
<td>0.0456</td>
<td>0.1073</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0255</td>
<td>0.6660</td>
<td>0.3753</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.0175</td>
<td>0.6161</td>
<td>0.3407</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.3909</td>
<td>0.1734</td>
<td>0.0842</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.3388</td>
<td>0.2843</td>
<td>0.1394</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0067</td>
<td>0.3431</td>
<td>0.0892</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.1971</td>
<td>0.7318</td>
<td>0.7042</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.1105</td>
<td>0.7655</td>
<td>0.2930</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.3698</td>
<td>0.2956</td>
<td>0.1025</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QC:RSFC int.</td>
<td>0.0004</td>
<td>0.2829</td>
<td>0.1017</td>
</tr>
<tr>
<td>QC:RSFC slope</td>
<td>0.0256</td>
<td>0.4152</td>
<td>0.1266</td>
</tr>
<tr>
<td>high-low resp. int.</td>
<td>0.0252</td>
<td>0.1257</td>
<td>0.0270</td>
</tr>
<tr>
<td>high-low resp. slope</td>
<td>0.4361</td>
<td>0.1626</td>
<td>0.0856</td>
</tr>
</tbody>
</table>
Analysis Group 6: Does TEDANA retain global BOLD signal in BOLD ICA components?

Carpet plots were generated for all of the included DuPre dataset participants. The plots for two participants, sub-06 and sub-19, are presented in Figure 4.8 and Figure 4.9, respectively. These participants were selected because they exhibit strong banding in the OC data. All other participants’ plots are provided in the supplement.

![Image](image.png)

Figure 4.8 **Experiment 1, Analysis Group 6, Analysis 1, Subject 06**

Based on visual inspection of the carpet plots, apparent banding across gray and white matter in the optimally combined data is also present in both accepted and rejected ICA components. In general, banding in the component carpet plots was not as apparent in these analyses as in the original paper, although this could be
due to differences in the multi-echo denoising approaches. It is important to note that, much like ME-ICA, tedana’s ICA step follows a dimensionality reduction step. However, while ME-ICA’s PCA step identified the maximum possible number of components, and then performed a classification step on those PCA components, tedana, by default, uses moving average PCA instead. Moving average PCA (Li et al., 2007) is much more aggressive about dimensionality reduction than ME-ICA’s original method. While we chose to use the least aggressive moving average PCA criterion (the Akaike information criterion), this aggressiveness results in far fewer ICA components than the original method, which improves the likelihood that the ICA will successfully converge, but also makes visually identifying cross-component patterns more difficult.

Figure 4.9 Experiment 1, Analysis Group 6, Analysis 1, Subject 19.
Correlations between the mean cortical signal from optimally combined data and accepted ICA component time series (M[Z] = 0.015, SD[Z] = 0.069) were not significantly higher than zero, t(29) = 1.191, p = 0.122. Correlations between the mean cortical signal from optimally combined data and rejected ICA component time series (M[Z] = -0.020, SD[Z] = 0.052) also were not significantly higher than zero, t(29) = -2.017, p = 0.973.

Additionally, correlations between mean cortical OC signal and accepted ICA components were significantly higher than correlations between mean cortical OC signal and rejected ICA components, t(29) = 2.337, p = 0.013. The distributions of transformed correlation coefficients for accepted and rejected components are shown in Figure 4.10.

Figure 4.10 Experiment 1, Analysis Group 6, Analysis 2. The distribution of Fisher’s z-transformed correlation coefficients, averaged within participant, for accepted and rejected components from the tedana denoising workflow.
These results indicated that BOLD-like components are significantly more correlated with the global signal of the optimally combined data than rejected components, as predicted.

Correlations between the mean cortical signal from multi-echo denoised signal and that of optimally combined data (M[Z] = 0.925, SD[Z] = 0.395) were significantly higher than zero, \( t(29) = 12.617, p < 0.001 \). The distribution of transformed correlation coefficients is presented in Figure 4.11.

**Figure 4.11** Experiment 1, Analysis Group 6, Analysis 3. The distribution of Fisher’s z-transformed correlation coefficients between mean cortical signal of the OC and MEDN data.

### 4.3.3 Discussion

The purpose of Experiment 1 was to replicate and extend the original paper’s respiration analyses. We chose to separate the individual analyses in this replication into analysis groups testing related hypotheses.

In analysis group 1, we sought to validate the novel RPV metric introduced in the original paper. We correlated RPV with mean RV and mean RVT across
participants, to determine if these metrics measure similar constructs. RPV was not significantly correlated with either measure.

We also compared correlations between the envelope used to calculate RPV and both RV and RVT across participants. The comparison was highly significant for both measures, indicating that RV, RVT, and the envelope are all indexing similar features in the respiration time series. As RPV is defined as the standard deviation of this envelope, while mean RV and mean RVT were compared to RPV, it makes sense that RPV would not be correlated with mean RV or mean RVT.

In analysis group 2, we attempted to replicate findings from the original paper that indicated that RPV is significantly, positively correlated with mean framewise displacement, a measure of total movement across the scan. We successfully replicated the original finding that RPV is significantly, positively correlated with mean FD, indicating that overall measures of motion increase as respiratory variance increases.

The results of the comparisons between framewise displacement and both RV and RVT were less clear. The distribution of correlations between RV and FD not significant and positive, while the distribution of correlations between RVT and FD was not. Together, these indicated that participants with more motion also exhibited higher respiratory variance, but not higher respiratory volume-per-time.

In analysis group 3, we sought to replicate and extend analyses comparing respiratory variability and global BOLD signal variability after different denoising strategies. Specifically, RPV was correlated with the standard deviation of mean cortical signal from a range of denoising outputs. All of the original paper’s findings were replicated, in that TE30, MEDN, FIT-R2, MEDN+Nuis-Reg, MEDN+RV-Reg, and MEDN+RVT-Reg correlations were all significant, and the MEDN+GODEC correlation was not. The correlations for both MEDN+aCompCor and MEDN+GSR
were not significant, as predicted, but the correlations for MEDN+MIR and MEDN+dGSR were significant, which did not match predictions.

In analysis group 4, we sought to replicate analyses comparing heart rate variability and global BOLD signal variability before and after multi-echo denoising. All of the original paper’s findings were replicated, in that TE30, MEDN, and FIT-R2 correlations were all nonsignificant. Together, these results further reinforce previous findings that BOLD signal variability is not significantly associated with heart rate variability, and that multi-echo denoising is not necessary for removing heart rate-related noise from BOLD data.

In analyses group 5, we sought to replicate the original paper’s distance-dependence analyses using respiration measures as the target QC metrics, as well as to extend those analyses to additional denoising derivatives. In the original paper, QC:RSFC analyses were performed on OC, MEDN, MEDN Noise, MEDN+GODEC, and MEDN+GSR derivatives, with RPV as the QC metric. The QC:RSFC intercept analyses of the optimally combined, MEDN Noise, and MEDN datasets were significant, while MEDN+GODEC and MEDN+GSR were not. The QC:RSFC slope analyses of the same derivatives were all non-significant.

In our replication, we found that the OC, MEDN, MEDN Noise, MEDN+Nuis-Reg, MEDN+RV-Reg, MEDN+RVT-Reg, MEDN+GSR, MEDN+MIR, and MEDN+dGSR intercept analyses were significant in either the QC:RSFC intercept analysis, the high-low respiration intercept analysis, or both, indicating that runs with greater RPV exhibit elevated RSFC, even after each of these types of denoising.

With the exception of the MEDN+GSR derivative, these findings replicate the original paper’s. These findings also indicate that several proposed denoising steps, including global signal regression, standard nuisance regression, RV-
regression, RVT-regression, minimum image regression, and dynamic global signal regression, all fail to eliminate the relationship between RPV and RSFC.

The slope analyses were significant for OC, MEDN Noise, MEDN+GSR, and MEDN+dGSR, indicating that runs with higher RPV exhibit higher RSFC correlations among nearby ROIs than among distant ones.

In this replication, OC, MEDN, MEDN Noise, and MEDN+GSR were significant, unlike in the original paper.

In analysis group 6, we sought to determine if tedana’s denoising workflow retains global signal associated with motion or respiration.

Visual inspection of carpet plots, with associated motion parameters and physiological recordings, indicated that banding present in the optimally combined dataset was generally apparent in both accepted and rejected ICA components. This banding was often temporally aligned with large movements or deep breaths.

Additionally, analyses of correlations between component time series and optimally combined global signal indicated that, while neither BOLD and non-BOLD component correlations were significantly greater than zero, they were significantly different from one another. Specifically, BOLD component correlations were higher than non-BOLD correlations, indicating that global signal was retained in the BOLD components more than in the non-BOLD components.

This was further supported by the analysis of correlations between OC global signal and MEDN global signal, which were significantly greater than zero. Together, these results indicate that global signal changes related to motion and respiration are not removed by multi-echo denoising.
4.4 Experiment 2: Replication and Extension of Methods for Removing Brain-Wide Signals

4.4.1 Materials & Methods

The second experiment attempted to replicate and extend analyses from the original paper relating to the removal of spatially diffuse noise in a larger sample. In the original paper, a sample of 89 participants (the Cambridge dataset) was used to evaluate the ability of multi-echo denoising and post-processing methods like global signal regression, Go decomposition, and robust principal components analysis to reduce motion-related patterns of functional connectivity. In this replication and extension, the 89-participant sample was combined with the 31-participant DuPre and 649-participant CamCAN samples to increase power. Additional post-processing methods, first evaluated in Experiment 1, were evaluated as well.

The analyses performed in this experiment broadly divide into four types: visual inspection of carpet plots from data before and after denoising, parametric analyses validating mean cortical signal as a good surrogate for global signal, parametric analyses assessing TE-dependence of global signal, and distance-dependent motion-related artifact analyses evaluating the ability of different denoising strategies to ameliorate motion-induced patterns in ROI-to-ROI functional connectivity.

Data

Two of the datasets used for this experiment were obtained from OpenNeuro. The first dataset (OpenNeuro accession number ds000258 v2) (Kundu et al., 2013b), which we will refer to as the Cambridge dataset, was also used in the original paper.
The Cambridge dataset contains 89 participants, four echoes, and no physiological data. Imaging data were acquired using a 3-Tesla Siemens Trio MRI scanner. One run of resting state multi-echo fMRI data was collected with four echoes (30 slices in interleaved ascending order; repetition time, TR=2470ms; echo time, TE=12, 28, 44, and 60ms; flip angle, FA=78°; field of view, FOV=240x240mm; matrix size=64x64; voxel size=3.75x3.75x4.4mm; in-plane acceleration factor=3). Each run was 9:51 minutes in length, during which 239 functional volumes were acquired. Dicoms were converted to NIfTI-1 format prior to processing. This section was (in part) generated automatically using pybids (Yarkoni et al., 2019).

The second dataset (OpenNeuro accession number ds000210 v2) (DuPre et al., 2016), which we will refer to as the DuPre dataset, was not used in the original paper. This dataset is used in the first experiment, and is described in detail above.

The third dataset was obtained from the Cambridge Centre for Ageing and Neuroscience project (available at https://camcan-archive.mrc-cbu.cam.ac.uk/dataaccess/, pending application) (Shafto et al., 2014; Taylor et al., 2017). This dataset will be referred to as the CamCAN dataset. This dataset contains 649 participants, five echoes, and no physiological data. Imaging data were acquired using a 3-Tesla Siemens Trio MRI scanner. T1-weighted structural MRI data were collected (192 slices; repetition time, TR=2.25s; echo time, TE=2.98ms; flip angle, FA=9°; field of view, FOV=256x256mm; matrix size=256x256; voxel size=1x1x1mm). T2-weighted structural MRI data were also collected (192 slices; repetition time, TR=2.8s; echo time, TE=408ms; flip angle, FA=120°; field of view, FOV=256x256mm; matrix size=256x256; voxel size=1x1x1mm). One run of film-viewing multi-echo fMRI data was collected with five echoes (32 slices in interleaved descending order; repetition time, TR=2470ms; echo time, TE=9.4, 21.23, 33.06, 44.89, 56.72ms; flip angle, FA=78°; field of view, FOV=192x192mm; matrix
size=64x64; voxel size=3.00x3.00x4.44mm; in-plane acceleration factor=3). Each run was 7:57 minutes in length, during which 193 functional volumes were acquired. Dicoms were converted to NIfTI-1 format prior to processing. For the sake of the analyses in this experiment (several of which are designed for resting-state fMRI data), this film-viewing dataset was treated as resting-state.

In the original paper, the 12-participant NA dataset was used to evaluate the ability of multi-echo processing methods to remove respiration-related noise from BOLD data, while the 89-participant Cambridge dataset was used to evaluate the methods’ ability to remove spatially diffuse, BOLD-related (and presumably respiration-induced) noise from data without direct reference to respiration. In this replication, the 31-participant DuPre dataset is used to evaluate respiration-related noise (Experiment 1), while the DuPre, Cambridge, and 649-participant CamCAN datasets are combined to evaluate spatially diffuse noise (Experiment 2). Thus, while the Cambridge dataset was used in the original study, in this replication we have used a different preprocessing pipeline and implementations of the original methods, as well as evaluated additional methods for the removal of physiological noise.

**Statistical Power Analyses**

Statistical power analyses for the parametric analyses from the original paper were performed in R with the pwr package. Five analyses (Table 4.8), all one-sample t-tests, were evaluated. Analyses were first divided into three families based on the hypotheses tested, which were used for Bonferroni alpha correction from a base alpha of 0.05. For analyses with statistically significant results in the original paper, one-tailed tests were used for the power analysis, while two-tailed tests were used for non-significant analyses.
Table 4.8  **Results from experiment 2 power analyses.** The six columns represent the name of the analysis, the family (or group of analyses) to which the analysis was assigned, the original analysis’ correlation coefficient, the original analysis’ sample size, the replication analysis’ sample size, and estimated statistical power for the replication analysis, respectively.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Family</th>
<th>Cohen’s D</th>
<th>Original N</th>
<th>Replication N</th>
<th>Statistical Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean cortical signal correlated with signal of all</td>
<td>1</td>
<td>7.58</td>
<td>89</td>
<td>680</td>
<td>1</td>
</tr>
<tr>
<td>gray matter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean cortical signal correlated with signal of whole</td>
<td>1</td>
<td>7.58</td>
<td>89</td>
<td>680</td>
<td>1</td>
</tr>
<tr>
<td>brain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global signal from MEICA-denoised data correlated</td>
<td>2</td>
<td>3.67</td>
<td>89</td>
<td>680</td>
<td>1</td>
</tr>
<tr>
<td>with global signal from optimally combined data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global signal from MEICA-denoised data correlated</td>
<td>2</td>
<td>4.11</td>
<td>89</td>
<td>680</td>
<td>1</td>
</tr>
<tr>
<td>with global signal from FIT R₂* data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance removed by GODEC correlated with variance</td>
<td>3</td>
<td>2.99</td>
<td>89</td>
<td>680</td>
<td>1</td>
</tr>
<tr>
<td>removed by mean signal regression</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results from the power analyses are summarized in Table 4.8. All of the analyses from the original paper using the Cambridge dataset (n=89) were sufficiently powered (1-B >= 0.8) in the combined DuPre+CamCAN dataset (n=680). Power analyses for these analyses were not performed with the Cambridge dataset for the replication, with the rationale that any estimates of power on the same dataset as was used in the original analyses, even for the sake of a replication, would be akin to *post hoc* power analysis and would be invalid.
Data Censoring & Exclusion

Data were subjected to quality control for the structural and functional MRI data. To this end, MRIQC (Esteban et al., 2017) was run on the neuroimaging data. The resulting MRIQC visual reports for both structural and functional scans were inspected for visually-identifiable artifacts, including ringing due to motion, ghosting, and extreme signal leakage across slices. Runs exhibiting any of these artifacts within the visual report were excluded from fMRIPrep preprocessing, multi-echo denoising, and all analyses.

Null distributions were derived within each dataset for two MRIQC image quality metrics: \textit{ghosting-to-signal-ratio} and \textit{foreground-background energy ratio}. Runs that are outliers in directions indicating poor data quality (upper for ghosting-to-signal ratio and lower for foreground-background energy ratio) were excluded from further analyses. Outliers were defined as values more than three standard deviations from the mean value in each dataset. These metrics were selected because they appeared to index scanner noise rather than motion-related noise. Given that the focus of these analyses was on motion-related artifact, motion was not used directly as a quality control metric for data exclusion.

Based on this quality control step, several subjects were excluded from further analyses in experiment 2. From the Cambridge dataset, subjects sub-20494, sub-20859, and sub-20863 were excluded. From the CamCAN dataset, subjects sub-CC221040, sub-CC221336, sub-CC221595, sub-CC221935, sub-CC223286, sub-CC320336, sub-CC321107, sub-CC510043, sub-CC610061, and sub-CC610658 were excluded. From the DuPre dataset, subject sub-18 was excluded. In total, this quality control step reduced the sample size for experiment 2 analyses from 761 to 747.
One CamCAN subject failed during fMRIPrep, and was thus excluded as well, reducing the sample size to 746. This subject was sub-CC221585.

Additionally, eleven subjects from the CamCAN dataset failed when subjected to tedana’s denoising procedure, so they were excluded from further analyses as well. These subjects were sub-CC110187, sub-CC110411, sub-CC221107, sub-CC221648, sub-CC221733, sub-CC310142, sub-CC310397, sub-CC410325, sub-CC420587, sub-CC420589, and sub-CC620026. This reduced the final sample size for experiment 2 to 735.

**Data Processing**

Data processing for this experiment was the same as that from the first experiment, although all processing related to physiological data was not performed, given that the Cambridge and CamCAN datasets do not have physiological trace data.

The basic processing and target datasets for this Aim are outlined in Figure 4.12.
Analyses

Analysis Group 1: Evaluating $T_2^*$ dependence of global signal  In order to determine if $T_2^*$ (FIT-$R_2$) or $S_0$ (FIT-$S_0$) estimates still showed visually identifiable global patterns (i.e., vertical bands), carpet plots with associated line plots of motion parameters were generated for each participant for the following outputs: OC, FIT-$S_0$, and FIT-$R_2$. It was predicted that vertical bands would remain present (and
temporally aligned with those present in OC data) in the FIT-R\textsubscript{2} data. We did not have any predictions regarding the FIT-S\textsubscript{0} data, given that the original paper did not report on these data.

**Analysis Group 2: Comparing measures of global signal**  Mean cortical signal from MEDN data will be correlated with mean signal across all gray matter (cortex, cerebellum, and subcortical nuclei) from the same data, in order to validate the cortical signal as an analog of global signal. The distribution of z-transformed correlation coefficients is predicted to be statistically significantly greater than zero, as assessed with a one-sample t-test, per the findings of the original paper. Power analysis of the original finding indicates that the combined datasets would be sufficiently powered to detect the hypothesized effect.

Mean cortical signal from MEDN data was correlated with mean signal across the whole brain (white matter, gray matter, and ventricles) from the same data, in order to further validate the cortical signal as an analog of global signal. The distribution of z-transformed correlation coefficients was predicted to be statistically significantly greater than zero, as assessed with a one-sample t-test, as in the original paper. Power analysis of the original finding indicated that the combined datasets would be sufficiently powered to detect the hypothesized effect.

**Analysis Group 3: Evaluating changes in global signal after multi-echo denoising**  Mean cortical signal from OC data was correlated with mean cortical signal from MEDN data to assess whether denoised data retains the global signal present in the original OC data. The distribution of z-transformed correlation coefficients was predicted to be statistically significantly greater than zero, as assessed with a one-sample t-test, as in the original paper. Power analysis of the original
finding indicated that the combined datasets would be sufficiently powered to detect the hypothesized effect.

Mean cortical signal from MEDN data was correlated with mean cortical signal from FIT-R₂ data to assess whether global signal present in denoised data is BOLD-related. The distribution of z-transformed correlation coefficients was predicted to be statistically significantly greater than zero, as assessed with a one-sample t-test, as in the original paper. Power analysis of the original finding indicated that the combined datasets would be sufficiently powered to detect the hypothesized effect.

**Analysis Group 4: Multiple Methods to Remove Brain-Wide Signals**

The abilities of GODEC, GSR, dGSR, MIR, and aCompCor to remove respiration-related signal were evaluated in Experiment 1. In Experiment 2, the abilities of these methods to remove widespread motion-related signal were evaluated in a larger dataset.

Specifically, in order to determine if denoised data still showed visually identifiable global patterns (i.e., vertical bands), carpet plots with associated line plots of motion parameters were generated for each participant for the following outputs: MEDN+GODEC, MEDN+GSR, MEDN+dGSR, MEDN+MIR, and MEDN+aCompCor. It was predicted that vertical bands would not remain present in any of the outputs.

Additionally, the variance removed by GODEC was correlated with the variance removed by GSR across participants. It was predicted, based on the original paper, that the correlation coefficient would be positive and statistically significant, indicating that GODEC and GSR remove similar amounts of global fluctuations across participants.
Analysis Group 5: Evaluating the ability of denoising methods to ameliorate focal and global motion-induced changes to functional connectivity

Motion analyses (described below) were performed on denoised data to clarify the impact of motion on ROI-to-ROI resting-state functional connectivity after denoising. In the original paper, the OC, MEDN, MEDN Noise, MEDN+GODEC, and MEDN+GSR derivatives were analyzed this way. To evaluate other methods for eliminating spatially diffuse noise, these analyses were also applied to the MEDN+GODEC Noise, MEDN+GSR Noise, MEDN+dGSR, MEDN+dGSR Noise, MEDN+aCompCor, MEDN+aCompCor Noise, MEDN+MIR, and MEDN+MIR Noise data.

QC:RSFC motion analysis In the QC:RSFC motion analysis, mean framewise displacement is used as the QC metric. Mean time series were extracted for the 264 Power ROIs, for each participant, and were cross correlated, resulting in a 264x264 correlation matrix. The upper triangle of this correlation matrix was retained and z-transformed, resulting in 34,716 z-values. Additionally, the Euclidean distance between each pair of ROIs was calculated and the upper triangle of the resulting 264x264 matrix was retained. For each pair of ROIs, the z-value and mean FD value across participants was correlated to produce the QC:RSFC metric. QC:RSFC was then analyzed against ROI distance by computing a smoothing curve (i.e., a moving average using sets of 1000 points, moving along the distance axis).

10,000 permutations were performed in which the QC metric was permuted across participants, QC:RSFC correlations were performed, and the smoothing curve was calculated. The rank of the real smoothing curve at 35 mm compared to the permuted smoothing curves was interpreted as a p-value indexing general dependence on motion (i.e., combined focal and global effects of motion). The rank of the
difference from 100 mm to 35mm was interpreted as a p-value indexing distance-dependence (i.e., focal effects of motion).

**High-low motion analysis** In this analysis, participants were separated into high-motion and low-motion groups using a median split on mean FD. Mean time series were extracted for the 264 Power ROIs, for each participant, and were cross correlated, resulting in a 264x264 correlation matrix. The upper triangle of this correlation matrix was retained, resulting in 34,716 correlation coefficients. Additionally, the Euclidean distance between each pair of ROIs was calculated and the upper triangle of the resulting 264x264 matrix was retained. For each pair of ROIs, the mean r-value for the low motion group was subtracted from the mean r-value for the high motion group. This resulted in a difference value (in delta-r) and a distance value (in mm) for each pair of ROIs. A smoothing curve was then computed for these difference values over distance in the same manner as the QC:RSFC analysis.

10,000 permutations were performed in which the assignment of high vs. low motion group is randomized across participants, delta-r values were calculated, and the smoothing curve was generated. The rank of the real smoothing curve at 35 mm compared to the permuted smoothing curves was interpreted as a p-value indexing general dependence on motion (i.e., combined focal and global effects of motion). The rank of the difference from 100 mm to 35mm was interpreted as a p-value indexing distance-dependence (i.e., focal effects of motion).

**Scrubbing analysis** In this analysis, time series in which volumes with frame-wise displacement greater than 0.2mm are censored were compared to uncensored time series. Mean time series were extracted for each of the 264 Power ROIs. Correlation matrices were generated for the mean time series, as well as for a scrubbed version of the time series. The correlation matrix from the scrubbed time series was
subtracted from the correlation matrix from the unscrubbed time series to produce a delta-r matrix. Note that the direction of this subtraction is reversed from that of the original paper. This is because the inference from the original paper for scrubbing analyses was reversed compared to that of the QC:RSFC and high-low motion analyses (i.e., lower ranks were more significant). This change in the procedure allowed us to interpret ranks and plots for the scrubbing analysis in the same manner as those of the QC:RSFC and high-low motion analyses. The upper triangle of this difference matrix was retained, resulting in 34,716 delta-r values. Additionally, the Euclidean distance between each pair of ROIs was calculated and the upper triangle of the resulting 264x264 matrix was retained. This resulted in a difference value (in delta-r) and a distance value (in mm) for each pair of ROIs. A smoothing curve was then computed for these difference values over distance in the same manner as the QC:RSFC analysis.

10,000 permutations were performed in which the indices of high-motion volumes to be scrubbed were randomized within each participant, delta-r values were calculated, and the smoothing curve was generated. The rank of the real smoothing curve at 35 mm compared to the permuted smoothing curves was interpreted as a p-value indexing general dependence on motion (i.e., combined focal and global effects of motion). The rank of the difference from 100 mm to 35mm was interpreted as a p-value indexing distance-dependence (i.e., focal effects of motion).

Unlike the QC:RSFC and high-low motion analyses, there are exclusion criteria for the scrubbing analysis. Specifically, any participants with no high-motion volumes or with >50% high-motion volumes were excluded. Time series with no high-motion volumes cannot be subjected to scrubbing, while it was expected that time series with too many high-motion volumes would be of insufficient quality for the analysis.
**Predicted findings for motion analyses** Based on the findings of the original paper, intercepts and slopes were predicted to be statistically significant for the OC data and for the MEDN Noise data. Intercepts (but not slopes) were predicted to be significant for the MEDN data. Finally, neither intercepts nor slopes were predicted to be significant for the MEDN+GODEC and MEDN+GSR data.

We also hypothesized that the extended derivatives included in the analyses would follow similar patterns. Namely, we predicted that the intercepts, but not the slopes, would be statistically significant for the MEDN+GODEC Noise, MEDN+GSR Noise, MEDN+dGSR Noise, MEDN+aCompCor Noise, and MEDN+MIR Noise data. We also predicted that neither the intercepts nor the slopes would be significant for the MEDN+dGSR, MEDN+aCompCor, and MEDN+MIR data.

### 4.4.2 Results

**Analysis Group 1: Evaluating T2\(^*\) dependence of global signal**

As can be seen in Figures 4.13 and 4.14, there is often prominent banding in the OC carpet plot when there are large movements. These bands generally appear in both the FIT-R2 and FIT-S0 carpet plots as well, indicating that both global signal and motion-related fluctuations remain in the T2\(^*\) and S0 time series. This pattern fit with our predictions, that global signal and motion-related fluctuations are at least partially reflected in fluctuations in T2\(^*\). Moreover, these fluctuations also exhibit an S0-based component, given that the same banding is generally apparent in the FIT-S0 carpet plots as well.
Figure 4.13 Experiment 2, Analysis Group 1, Analysis 1, Cambridge Dataset, Subject 20758

Figure 4.14 Experiment 2, Analysis Group 1, Analysis 1, CamCAN Dataset, Subject CC110101
Analysis Group 2: Comparing measures of global signal

Correlations between the mean multi-echo denoised signal extracted from the cortical ribbon and that extracted from all gray matter (M[Z] = 2.565, SD[Z] = 0.617) were significantly higher than zero, t(734) = 112.690, p < 0.001. Correlations between the mean multi-echo denoised signal extracted from the cortical ribbon and that extracted from the whole brain (M[Z] = 2.501, SD[Z] = 0.441) were significantly higher than zero, t(734) = 153.604, p < 0.001.

The highly-significant results of these two analyses replicate the original paper, and indicate that the cortical ribbon mask serves as a reasonable proxy for the global signal.

Figure 4.15 Experiment 2, Analysis Group 2. (A) Correlations between mean MEDN signal from cortical ribbon and gray matter masks. (B) Correlations between mean MEDN signal from cortical ribbon and whole brain masks.
Analysis Group 3: Evaluating changes in global signal after multi-echo denoising

The two analyses in this analysis group were considered a family, so Bonferroni correction was applied, and the alpha value was adjusted from 0.05 to 0.025.

Correlations between the mean cortical ribbon signal from the multi-echo denoised data and the optimally combined data ($M[Z] = 1.234$, $SD[Z] = 0.475$) were significantly higher than zero, $t(734) = 70.423$, $p < 0.001$. The distribution of transformed correlation coefficients is shown in Figure 4.16A.

Correlations between the mean cortical ribbon signal from the multi-echo denoised data and the FIT-R2 data ($M[Z] = 0.442$, $SD[Z] = 0.443$) were significantly higher than zero, $t(734) = 27.019$, $p < 0.001$. The distribution of transformed correlation coefficients is shown in Figure 4.16B.

These findings replicate the original paper’s, indicating that multi-echo denoising retains the global signal from the original OC data, and that global signal is also present in the $T2^*$ (FIT-R2) signal.
Figure 4.16 **Experiment 2, Analysis Group 3.** (A) Correlations between mean cortical ribbon MEDN signal and mean cortical ribbon OC signal. (B) Correlations between mean cortical ribbon MEDN signal and mean cortical ribbon FIT-R2 signal.

**Analysis Group 4: Multiple Methods to Remove Brain-Wide Signals**

In the first analysis, carpet plots were generated for the MEDN+aCompCor, MEDN+dGSR, MEDN+GODEC, MEDN+GSR, and MEDN+MIR derivatives and were visually inspected for banding. Given that these denoising steps were used to attempt to remove spatially-diffuse noise, we predicted that motion-synchronized banding would diminish or disappear completely in the denoised carpet plots. The carpet plots for one subject, sub-CC321174, from the CamCAN dataset, are provided in Figures 4.17 to 4.21.

Qualitatively, it was apparent that banding was most reduced in the GSR and GODEC derivatives. It was dramatically reduced in the aCompCor and dGSR derivatives, though to a lesser extent than GSR and GODEC. Banding was also reduced, though not nearly as much as in the other derivatives, in the MIR derivatives.
Figure 4.17 Experiment 2, Analysis Group 4 aCompCor. Carpet plot.

Figure 4.18 Experiment 2, Analysis Group 4 dGSR. Carpet plot.
Figure 4.19 Experiment 2, Analysis Group 4 GODEC. Carpet plot.

Figure 4.20 Experiment 2, Analysis Group 4 GSR. Carpet plot.
Next, the percent variance removed by GODEC and GSR were correlated across participants. Variance removed by GODEC and GSR were found to be positively and statistically significantly correlated, $r(733) = 0.98$, $p < 0.001$. The scatter plot of variance removed is shown in Figure 4.22.

This finding replicated the original paper’s, indicating that GODEC and GSR remove consistent amounts of global fluctuations across participants. However, the actual amount of variance removed is quite different between the two methods, as evidenced by the axis limits in Figure 4.22.
Figure 4.22 **Experiment 2, Analysis Group 4.** A scatter plot of the variance removed by GODEC and that removed by GSR.

**Analysis Group 5: Evaluating the ability of denoising methods to ameliorate focal and global motion-induced changes to functional connectivity**

In this analysis group, we performed a series of distance-dependence motion-related artifact analyses on the denoising derivatives, across the DuPre, CamCAN, and Cambridge datasets. For the sake of easy comparison, we have reconstructed the DDMRA results from the original paper in Table 4.9.
Table 4.9 **Reconstruction of distance-dependence analyses from the original publication**

<table>
<thead>
<tr>
<th></th>
<th>QC:RSFC</th>
<th>high-low motion</th>
<th>scrubbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td>int.</td>
<td>0.0260</td>
<td>0.0054</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0006</td>
<td>0.0001</td>
</tr>
<tr>
<td>MEDN</td>
<td>int.</td>
<td>0.0007</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0212</td>
<td>0.0910</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>int.</td>
<td>0.0132</td>
<td>0.0382</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0008</td>
<td>0.0037</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>int.</td>
<td>0.8160</td>
<td>0.7555</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.1095</td>
<td>0.0326</td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>int.</td>
<td>0.4859</td>
<td>0.0464</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.1440</td>
<td>0.0155</td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td>int.</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Confirmatory analyses across datasets** Counter to our predictions, neither slopes nor intercepts were significant for OC or MEDN Noise data in the QC:RSFC and high-low motion analyses, indicating that motion does not impact functional connectivity in OC or MEDN Noise data, nor does it exhibit a relationship with distance. Neither intercepts nor slopes were significant for the MEDN data either, except for the high-low motion analysis’s slope. This is the direct opposite of the findings in the original paper, where the only non-significant result for the MEDN data was the high-low motion analysis’s slope.

Conversely, both the slopes and intercepts were significant for the MEDN+aCompCor and MEDN+GODEC data, indicating that aCompCor and
GODEC denoising fail to eliminate the relationship between motion and functional connectivity, and the relationship is influenced by distance between ROIs. The high-low motion intercept and slope were both significant for the MEDN+GSR data. The QC:RSFC slope and both the intercept and slope of the high-low motion analysis were significant for the MEDN+MIR data.

The MEDN+aCompCor Noise, MEDN+GODEC Noise, MEDN+GSR Noise, MEDN+MIR Noise, and MEDN+dGSR Noise data results were largely nonsignificant, indicating that the variance removed by these denoising methods does not exhibit a relationship between motion and functional connectivity.

Many of the distance-dependence analysis results were extremely significant (i.e., rank 0/10000) or nonsignificant (rank 10000/10000), which we found suspicious. As such, we performed post-hoc analyses of each of the datasets separately (Tables 4.11, A.1 and A.2). We additionally included a linear regression-based significance test, in case the original test, which only evaluated two distances (35mm and 100mm), was obscuring patterns in the data. These tests are described in the supplement (Appendix A), and the results are presented in Tables A.3 to A.6.

Given that the original paper performed these analyses on the Cambridge dataset, we have reported those results below. The results of the other analyses are presented in the supplementary materials.
Table 4.10  **Results of Experiment 2, Analysis Group 5**  Results of the distance-dependent motion-related artifact analyses in analysis group 5. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with p < 0.05 are bolded.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>QC:RSFC int.</th>
<th>high-low motion int.</th>
<th>scrubbing int.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN</td>
<td>1.0000</td>
<td>0.9994</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>0.0586</td>
<td>0.0022</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>1.0000</td>
<td>0.9999</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>0.7152</td>
<td>0.9956</td>
<td>0.6479</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>1.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>0.0395</td>
<td>0.5214</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>0.0873</td>
<td>0.2726</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>0.0574</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>0.0005</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>0.9363</td>
<td>0.9989</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td>0.9273</td>
<td>0.3167</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>0.0127</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

**Exploratory analyses within the Cambridge dataset**  As found in the original study, most of the analyses’ slopes and intercepts were significant for the OC data, except for the high-low motion analysis’s intercept.

The slopes for the three analyses were also significant for the MEDN data, as was the intercept for the scrubbing analysis. This differed from the original paper’s findings, in which the MEDN intercepts were all significant, and the QC:RSFC and scrubbing analyses’ slopes were significant.

The slopes for the QC:RSFC and high-low motion analyses were significant for the MEDN Noise data, while the intercepts for the QC:RSFC and scrubbing analyses...
were significant. This was somewhat similar to the original paper’s findings, in which all of the analyses’ slopes and intercepts were significant, except for the scrubbing analysis’s slope.

For the MEDN+aCompCor data and the MEDN+GODEC data, the QC:RSFC and scrubbing analyses’ intercepts were significant, and the scrubbing analysis’s slope was significant, indicating that aCompCor and GODEC retain at least some focal and global effects of motion on functional connectivity.

For the MEDN+GSR data, the slope and intercept of the scrubbing analysis were significant.

For the MEDN+MIR data, all of the results were significant, except for the high-low motion analysis’s slope, strongly indicating that MIR retains both global and focal effects of motion on functional connectivity.

For the MEDN+dGSR data, the scrubbing analysis’s intercept was significant, as were all of the analyses’ slopes, indicating that dGSR retains focal effects of motion on functional connectivity, but that global effects are not strong.

For the MEDN+aCompCor Noise data, the MEDN+GODEC Noise data, the MEDN+GSR Noise data, and the MEDN+MIR Noise data, and the MEDN+dGSR Noise data, the QC:RSFC and high-low motion analyses’ slopes were significant. We originally predicted that all three analyses’ intercepts would be significant for all of these derivatives, and that the slopes would not be. As such, the findings suggest that, counter to expectations, the rejected variance from these denoising methods exhibits a distance-dependent effect of motion on functional connectivity, but does not exhibit an overall heightened effect of motion on functional connectivity across the brain.
Table 4.11 Distance-dependence analyses of the Cambridge dataset  Results of the distance-dependent motion-related artifact analyses in analysis group 5, limited only to the Cambridge dataset. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with p < 0.05 are bolded.

<table>
<thead>
<tr>
<th></th>
<th>QC:RSFC</th>
<th>high-low motion</th>
<th>scrubbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC</td>
<td>int.</td>
<td>0.0145</td>
<td>0.1068</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0002</td>
<td>0.0035</td>
</tr>
<tr>
<td>MEDN</td>
<td>int.</td>
<td>0.3115</td>
<td>0.3206</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0033</td>
<td>0.0076</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>int.</td>
<td>0.0185</td>
<td>0.1051</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0072</td>
<td>0.0279</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>int.</td>
<td>0.0014</td>
<td>0.1286</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.3673</td>
<td>0.8827</td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>int.</td>
<td>0.9165</td>
<td>0.7926</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0044</td>
<td>0.0124</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>int.</td>
<td>0.0003</td>
<td>0.2245</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.5663</td>
<td>0.9945</td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>int.</td>
<td>0.9772</td>
<td>0.8514</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0013</td>
<td>0.0013</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>int.</td>
<td>0.2967</td>
<td>0.9686</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.5255</td>
<td>0.8944</td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>int.</td>
<td>0.9971</td>
<td>0.9943</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0067</td>
<td>0.0191</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>int.</td>
<td>0.0068</td>
<td>0.0241</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0083</td>
<td>0.1418</td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>int.</td>
<td>0.6219</td>
<td>0.8881</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0013</td>
<td>0.0163</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>int.</td>
<td>0.1750</td>
<td>0.2320</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0100</td>
<td>0.0494</td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td>int.</td>
<td>0.4251</td>
<td>0.1807</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0045</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

4.4.3 Discussion

The purpose of Experiment 2 was to replicate and extend analyses relating to denoising methods designed to remove spatially diffuse noise.

In analysis group 1, we attempted to visually determine if motion-related banding in carpet plots is retained in both T2* and S0 estimates.

In analysis group 2, we compared global signal measures to validate mean cortical ribbon signal as a reasonable measure of global signal. As predicted, mean cortical ribbon signal was strongly correlated with both mean whole brain signal and mean
gray matter signal, indicating that mean cortical ribbon signal is a useful measure of global signal.

In analysis group 3, we sought to determine whether global signal is retained after multi-echo denoising and within T2* estimates. As predicted, mean cortical signal from OC data was significantly correlated with mean cortical signal from both MEDN and FIT-R2 data, indicating that multi-echo denoising retains global signal, and that this global signal is T2*-based.

In analysis group 4, we compared the amount of variance removed by GODEC and GSR. While the amount of variance removed by the two methods was strongly correlated across participants.

**Analysis Group 5**

In analysis group 5, we performed a series of distance dependence analyses in order to determine if different denoising approaches successfully ameliorate focal and global effects of motion on functional connectivity. When performed on all three datasets together, the results of these analyses diverged substantially from the original paper and from our hypotheses. Specifically, most of the slopes and intercepts from the analyses of the OC data were not significant, indicating that optimally combined data do not exhibit any significant relationships between motion and functional connectivity. Additionally, the analyses of the MEDN data and MEDN Noise data were also largely nonsignificant, indicating that neither the variance retained, nor the variance discarded, by multi-echo denoising exhibits a significant relationship between motion and functional connectivity.

Conversely, several analyses’ slopes and intercepts were significant for the MEDN+aCompCor, MEDN+GODEC, and MEDN+MIR data, which indicates that additional denoising steps designed to remove spatially diffuse noise still retain
both focal and global relationships between motion and functional connectivity. It is not possible to determine if these denoising methods *induce* these relationships with the analyses as they were performed; therefore, the nonsignificance of the analyses in the MEDN data does not indicate that denoising introduces motion-related noise.

Using the current DDMRA approach, it is not possible to directly compare the distance-dependence of different preprocessing derivatives. While we believe that the analyses could be extended to perform pairwise comparisons, we did not attempt any such extensions in this replication. As such, our ability to infer effects of different denoising approaches is limited.

In the exploratory DDMRA analyses of each of the separate datasets, we discovered that the CamCAN dataset exhibited results that were roughly the reverse of our hypotheses (Table A.1). Given the sample size of the CamCAN dataset relative to the other two, we believe that CamCAN drives the results of the pooled-dataset analyses.

When we performed the analyses on the Cambridge dataset, for a more direct replication (barring changes in preprocessing and denoising implementations), the results were more consistent with the original paper’s findings, though they still differed. Specifically, although the OC results differed in the scrubbing intercept and the high-low motion intercept, most of the slope and intercept analyses were significant in both the original paper and the Cambridge-only replication.

Additionally, although the QC:RSFC and high-low motion intercepts were not significant in the Cambridge-only replication, the high-low motion slope was the only nonsignificant analysis in the original paper. In the original paper, the MEDN Noise analyses were all significant, except for the scrubbing slope. In the replication, the same analyses were significant, except for the high-low motion intercept.
The MEDN+GODEC and MEDN+GSR analyses were also somewhat similar between the replication and the original paper. The MEDN+GODEC QC:RSFC intercept, high-low motion intercept, and QC:RSFC slope analyses were nonsignificant in the original paper, while the high-low motion intercept, QC:RSFC slope, and high-low motion slope were nonsignificant in the replication. In both cases, half of the analyses were significant, but in the original paper the majority of the significant analyses were in the slopes instead of the intercepts. The MEDN+GSR QC:RSFC intercept and slope were nonsignificant in the original paper, while in the replication both of those analyses were nonsignificant, but so were the high-low motion intercept and slope.

Taken together, these results indicate that the motion analyses may produce very different results, even in the same dataset, depending on the preprocessing and/or denoising implementations.

4.5 Limitations and Future Directions

4.5.1 Limitations to the original paper

As reported in Power et al. (2020), RV, RVT, and RPV (reformulated as the measure ENV) are all metrics that sometimes miss deep breaths, indicating that none is a perfect omnibus measure of respiration.

Carpet plots are inherently qualitative artifact detection tools, and also risk conflating global signal changes with artifacts. Therefore, we stress that results from analyses of carpet plots should be interpreted carefully.

165
4.5.2 Limitations to this replication

During preparation of this manuscript, one multi-echo dataset with 28 participants, physiological recordings, resting-state, and task data was made public (Heunis et al., 2020b). While this dataset would be useful for our replication, the dataset was made publicly available too late for us to incorporate it into our analyses. More generally, we hope that the utility of these and related re-analyses of public multi-echo fMRI data sets will encourage investigators to more broadly share their multi-echo acquisitions.

Three-to-four echo data result in very noisy volume-wise T$_2^*/$S$_0$ estimates, which means that all analyses which rely on FIT-T$_2^*$ and FIT-S$_0$ data should be interpreted carefully. Unfortunately, there are no public multi-echo datasets with more than five echoes, nor are there any with more than three echoes and physiological data.

Chest movement alters the B$_0$ field and can cause a phase shift, which can manifest itself as image displacement in the phase encoding direction. It’s important to be clear that this is NOT human head motion. It is image motion due to a systematic distortion. It can still cause problematic image artifacts, but the source of these artifacts is different from actual head motion or respiration-based blood oxygenation changes. This relates to “pseudomotion” described in other works (Power et al., 2019; Gratton et al., 2020). A number of solutions have been proposed for this pseudomotion, mostly involving applying filters to motion parameters (Power et al., 2019; Gratton et al., 2020; Fair et al., 2020). While we considered investigating this form of pseudomotion outside of the scope of this particular study, we plan to follow up with further replications and extensions of related studies. Moreover, pseudomotion should be most apparent in fMRI data with high temporal resolution (e.g., multiband data). Unfortunately, all of the publicly available datasets used in this study had relatively low temporal resolution (2.47-4s), and thus would
not be optimal choices to investigate the interactions between respiration-related pseudomotion and signal decay.

### 4.5.3 Deviations from the original experiments

For the most part, this replication follows the same methods described in the original publication (barring intentional extensions), although the software used to implement each method is generally different. These deviations, combined with the different datasets employed in this study, prevent direct replication. Instead, this study represents a ”conceptual” replication of Power et al. (2018), in which the same algorithms are employed, and hypotheses tested, as the original study, but on different data and with different implementations.

One potentially significant deviation from the original experiments is the use of fMRIPrep for preprocessing, in lieu of a custom preprocessing pipeline. The processing steps in fMRIPrep (motion correction, slice timing correction, distortion correction, coregistration of functional data to anatomical image, and normalization to standard space) are generally the same as the steps described in the original paper (slice timing correction, motion correction, coregistration to anatomical image, and normalization to standard space). However, there are many differences between the pipelines that may lead to different results. Here we attempt to catalogue differences that we believe could be impactful.

First and foremost, while Power et al. did an exemplary job of describing their preprocessing steps, no textual description of a preprocessing pipeline is sufficient for replication. MRI processing libraries, such as AfNI, include too many tunable parameters to report in text; therefore, without access to the processing code, accurately replicating the original pipeline would be near impossible. As such, we chose
to use fMRIPrep, which is a collaboratively-developed fMRI preprocessing workflow designed with robustness across datasets in mind.

Power et al. used AfNI and Freesurfer for all preprocessing steps, while fMRIPrep has a policy of using whichever tool performs best for each step in its workflow. This means that, even when the preprocessing step is equivalent between the two workflows (e.g., motion correction), the actual function used to perform the step may be different. This can lead to small, but consistent, differences.

Power et al. also processed each individual echo separately, and noted that the calculated parameters (e.g., motion parameters) for the different echoes varied slightly. We recommend against processing echoes in a multi-echo dataset individually, and indeed fMRIPrep calculates transformation parameters on one echo and applies them to all echoes in the run, preserving the original mapping across echoes.

Power et al. transformed their data to Talairach space, but we have chosen to use the MNI template for our standard space target. Additionally, we chose to perform multi-echo denoising in native space instead of standard space. While Power et al. performed their analyses in both standard and native space, and reported no meaningful differences in their results, we recommend against applying nonlinear transformations to multi-echo data before applying multi-echo denoising, so we instead chose to extract intermediate, native-space data from fMRIPrep, perform multi-echo denoising, and then apply fMRIPrep’s transforms to standard space afterwards.

Finally, Power et al. used the ME-ICA toolbox, while we used its successor library, tedana. Tedana was developed from ME-ICA’s codebase in 2018, but there have been a number of improvements to tedana since its inception. While we consider a full description of the differences between ME-ICA and tedana outside the scope of this publication, we do wish to note that tedana includes a number of
enhancements and bug-fixes that may impact results, but we do not believe that the bugs that have been identified in ME-ICA are substantial enough to affect the original paper’s findings.

4.6 Code availability

All code required to reproduce the analyses and figures in this paper are available on GitHub at https://github.com/NBCLab/power-replication.

4.7 Author Contributions

Author MCR contributed to the analysis plan and the quality control plan. Author EMD contributed to the analysis plan. Author KLB contributed to the analysis plan and manual tracing of regions of interest. Authors ARL and DAH contributed to the analysis plan and manuscript preparation. Author TS contributed to the analysis plan, the quality control plan, and manuscript preparation.
Acknowledgment

Support for this project was provided by NSF 1631325, NIH R01 DA041353, and NIH U01 DA041156. Part of the data collection and sharing for this project was provided by the Cambridge Centre for Ageing and Neuroscience (CamCAN). CamCAN funding was provided by the UK Biotechnology and Biological Sciences Research Council (grant number BB/H008217/1), together with support from the UK Medical Research Council and University of Cambridge, UK. Data used for power analyses were provided in part by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University. Additional thanks to the FIU Instructional & Research Computing Center (IRCC, http://ircc.fiu.edu) for providing the HPC and computing resources that contributed to the research results reported within this paper.

We would also like to thank Dr. Jessica Bartley and Julio A. Peraza Goicolea for their help with the mathematical notation for the minimum image regression algorithm.
CHAPTER 5

CONCLUSION

As the culmination of my graduate work, this dissertation reflects the skills I have developed and the lessons I have learned over the course of my graduate training.

Chapters 2 and 3 of this dissertation reflect my dedication to research software development, as two very different software papers summarizing work on two open source libraries I have maintained, or helped to maintain, over the past few years. In that time, I have learned a great deal about what it takes to build and maintain an open source software project.

Research software development is relatively easy compared to maintenance. Graduate students are often funded to develop new methods. Those methods may only be made available to the public through publications, or possibly through software packages. However, even in optimal cases where software is provided for the new methods, graduate students often move on from these projects when they graduate, and the principal investigators rarely have the funding to maintain the projects afterward. It is therefore crucial to either fund individuals to maintain existing libraries, as will be done for NiMARE, or to build volunteer-based maintainer groups, as is the case for tedana. It is also useful for methodologists to contribute to existing libraries, when possible, rather than releasing a separate package for a single new tool. This is one reason why we chose to develop NiMARE; it serves to implement a wide range of methods, including some that were developed by graduate students who later left academia, such as generalized correspondence latent Dirichlet allocation.

While chapters 2 and 3 develop from on my work in research software development, chapter 4 is focused on replicating and extending existing work in an open and reproducible manner, reflecting my commitment in open science. This chap-
ter builds on my work as a maintainer of the tedana library (described in Chapter 2) by evaluating multi-echo denoising strategies and characterizing the nature of TE-dependent noise in fMRI data.

We chose to pursue a unique manuscript format for this chapter- a registered report. One of the benefits of a registered report is that it separates the experimental design from the results. This allows peer reviewers to evaluate the quality and completeness of the planned analyses before they are completed. It also aids the researcher by defining all of the processing and analysis steps ahead of time, so that the researcher will not be tempted to pursue alternative strategies for testing the same hypotheses (also known as p-hacking), nor to update their expectations based on analysis results (also known as hypothesizing after results are known). When we chose to use this type of manuscript structure for our replication and extension, we believed that it would make the second stage of the manuscript (i.e., the results and discussion sections) very straightforward. While this is true to some extent, we did discover that executing a complete analysis plan, and interpreting the results, can nevertheless be a complicated and difficult process.

What we discovered is that, under any circumstances, independent replication is difficult. Even when the analysis plan is defined, peer reviewed, and accepted in principle prior to initiating the analyses, it is very easy to become mired in small, nuanced details of the analyses. This is particularly true when the results do not match expectations in neuroimaging research, where there is often little distinction between quality control and the analyses of interest. Without a detailed analysis plan, it is very easy to unintentionally revise one’s methodology based on the results of their analyses, under the guise of fixing mistakes. In my case, the difficulty lay in the implementation of motion analyses (Section 4.4.2). When the analyses produced results that were often the opposite of my expectations, my natural inclination
was to re-run the analyses with different parameters or on different subsets of the data. These post-hoc analyses are reported in both the chapter and its associated supplementary material. Even after devising and executing these new, necessarily exploratory, analyses, the results did not replicate the original study, though the closest approximation to the original study’s analyses (i.e., a replication only on the same dataset as the original study, rather than a pooled sample of multiple datasets) was more similar than other versions of the analyses.

This road block proved to be an important lesson for me. Firstly, it is obvious that direct replication, using the same data and code as the original paper, is impossible without access to said data, code, and even the development environment. What is less obvious is that even conceptual replication, in which the same methods are employed, according to different implementations, on equivalent data, is very difficult to do. A well-written journal article serves as an essential distillation of research, but it is not sufficient on its own to support replication. There will always be undocumented decisions or settings that could influence results. These variables inevitably translate to researcher degrees of freedom, on the part of both the original experimenter and the researcher performing the replication, that can cause results to diverge. Often, these divergences fall within an acceptable range, but they can also cause confusion and doubt for all involved. It is thus my hope that researchers will continue to make progress in sharing their code and data, in order to facilitate direct replications in the future.

Nevertheless, conceptual replication is an important tool in the scientific process. Reproducing findings with different code and/or different data is necessary to confidently conclude that the original findings are generalizable. While this one replication should not be sufficient to conclude that the original paper is or is not replicable, it will hopefully serve as a useful piece of evidence as researchers refine
their understanding of the nature of noise in fMRI. Moreover, it has taught me a
great deal about what reproducible research truly entails, and how difficult it is to
make one’s work replicable.

In conclusion, my work has produced several research objects, including the
chapters in this dissertation. Collectively, my dissertation has thus allowed me to
pursue my primary research interests in research software development and repro-
ducible, open science. This work has allowed me to develop the skills I plan to use
in my post-graduate career, as a research software developer.
BIBLIOGRAPHY


“one” DLPFC in cognitive action control? evidence for heterogeneity from Co-Activation-Based parcellation.


task and resting state dataset for real-time, multi-echo fmri methods development and validation. bioRxiv.


183


Appendix I: BrainMap Discrete Decoding

The BrainMap discrete decoding method compares the distributions of studies with each label within the sample against those in a larger database while accounting for the number of foci from each study. Broadly speaking, this method assumes that the selection criterion is associated with one peak per study, which means that it is likely only appropriate for selection criteria based around foci, such as regions of interest. One common analysis, meta-analytic clustering, involves dividing studies within a database into meta-analytic groupings based on the spatial similarity of their modeled activation maps (i.e., study-wise pseudo-statistical maps produced by convolving coordinates with a kernel). The resulting sets of studies are often functionally decoded in order to build a functional profile associated with each meta-analytic grouping. While these groupings are defined as subsets of the database, they are not selected based on the location of an individual peak, and so weighting based on the number of foci would be inappropriate.

This decoding method produces four outputs for each label. First, the distribution of studies in the sample with the label are compared to the distributions of other labels within the sample. This consistency analysis produces both a measure of statistical significance (i.e., a p-value) and a measure of effect size (i.e., the likelihood of being selected given the presence of the label). Next, the studies in the sample are compared to the studies in the rest of the database. This specificity analysis produces a p-value and an effect size measure of the posterior probability of having the label given selection into the sample. A detailed algorithm description is presented below.
The BrainMap method for discrete functional decoding performs both forward and reverse inference using an annotated coordinate-based database and a target sample of studies within that database. Unlike the Neurosynth approach, the BrainMap approach incorporates information about the number of foci associated with each study in the database.

1. Select studies in the database according to some criterion (e.g., having at least one peak in an ROI).

2. For each label, studies in the database can now be divided into four groups.
   - Label-positive and selected: \( S_{s+t+} \)
   - Label-negative and selected: \( S_{s+t-} \)
   - Label-positive and unselected: \( S_{s-t+} \)
   - Label-negative and unselected: \( S_{s-t-} \)

3. Additionally, the number of foci associated with each of these groups is extracted.
   - Number of foci from studies with label, \( F_{l+} \)
   - Number of foci from studies without label, \( F_{l-} \)
   - Total number of foci in the database, \( F_{db} = F_{l+} + F_{l-} \)

4. Compute the number of times any label is used in the database, \( L_{db} \) (e.g., if every experiment in the database uses two labels, then this number is \( 2S_{db} \), where \( S_{db} \) is the total number of experiments in the database).

5. Compute the probability of being selected, \( P(s^+) \).
   - \( P(s^+) = \frac{S_{s+}}{F_{db}} \), where \( S_{s+} = S_{s+t+} + S_{s+t-} \)
6. For each label, compute the probability of having the label, $P(l^+)$.

   - $P(l^+) = S_{l^+}/L_{db}$, where $S_{l^+} = S_{s+l^+} + S_{s-l^+}$

7. For each label, compute the probability of being selected given presence of the label, $P(s^+|l^+)$.

   - Can be re-interpreted as the probability of activating the ROI given a mental state.
   - $P(s^+|l^+) = S_{s+l^+}/F_{l^+}$

8. Convert $P(s^+|l^+)$ into the forward inference likelihood, $\mathcal{L}$.

   - $\mathcal{L} = P(s^+|l^+)/P(s^+)$

9. Compute the probability of the label given selection, $P(l^+|s^+)$.

   - Can be re-interpreted as probability of a mental state given activation of the ROI.
   - $P(l^+|s^+) = \frac{P(s^+|l^+)P(l^+)}{P(s^+)}$

   - This is the reverse inference posterior probability.

10. Perform a binomial test to determine if the rate at which studies are selected from the set of studies with the label is significantly different from the base probability of studies being selected across the whole database.

   - The number of successes is $K = S_{s+l^+}$, the number of trials is $n = F_{l^+}$, and the hypothesized probability of success is $p = P(s^+)$

   - If $S_{s+l^+} < 5$, override the p-value from this test with 1, essentially ignoring this label in the analysis.
• Convert p-value to unsigned z-value.

11. Perform a two-way chi-square test to determine if presence of the label and selection are independent.

• If $S_{s+t} < 5$, override the p-value from this test with 1, essentially ignoring this label in the analysis.

• Convert p-value to unsigned z-value.
Appendix II: Neurosynth Discrete Decoding

The implementation of the MKDA Chi-squared meta-analysis method used by Neurosynth is quite similar to BrainMap’s method for decoding, if applied to annotations instead of modeled activation values. This method compares the distributions of studies with each label within the sample against those in a larger database, but, unlike the BrainMap method, does not take foci into account. For this reason, the Neurosynth method would likely be more appropriate for selection criteria not based on regions of interest (e.g., for characterizing meta-analytic groupings from a meta-analytic clustering analysis). However, the Neurosynth method requires user-provided information that BrainMap does not. Namely, in order to estimate probabilities for the consistency and specificity analyses with Bayes’ Theorem, the Neurosynth method requires a prior probability of a given label. Typically, a value of 0.5 is used (i.e., the estimated probability that an individual is undergoing a given mental process described by a label, barring any evidence from neuroimaging data, is predicted to be 50%); this is, admittedly, a poor prediction, which means that probabilities estimated based on this prior are not likely to be accurate, though they may still serve as useful estimates of effect size for the analysis.

Like the BrainMap method, this method produces four outputs for each label. For the consistency analysis, this method produces both a p-value and a conditional probability of selection given the presence of the label and the prior probability of having the label. For the specificity analysis, the Neurosynth method produces both a p-value and a posterior probability of presence of the label given selection and the prior probability of having the label. A detailed algorithm description is presented below.
The Neurosynth method for discrete functional decoding performs both forward and reverse inference using an annotated coordinate-based database and a target sample of studies within that database. Unlike the BrainMap approach, the Neurosynth approach uses an *a priori* value as the prior probability of any given experiment including a given label.

1. Select studies in the database according to some criterion (e.g., having at least one peak in an ROI).

2. For each label, studies in the database can now be divided into four groups:
   - Label-positive and selected: $S_{s+t}$
   - Label-negative and selected: $S_{s+t-}$
   - Label-positive and unselected: $S_{s-t}$
   - Label-negative and unselected: $S_{s-t-}$

3. Set a prior probability $p$ of a given mental state occurring in the real world.
   - Neurosynth uses 0.5 as the default.

4. Compute $P(s^+)$:
   - Probability of being selected, $P(s^+) = S_{s+} / (S_{s+} + S_{s-})$, where $S_{s+} = S_{s+t} + S_{s+t-}$ and $S_{s-} = S_{s-t} + S_{s-t-}$

5. For each label, compute $P(l^+)$:
   - $P(l^+) = S_{l+} / (S_{l+} + S_{l-})$, where $S_{l+} = S_{s+l} + S_{s-l}$ and $S_{l-} = S_{s+l-} + S_{s-l-}$

6. Compute $P(s^+|l^+)$:
7. Compute $P(s^+|l^+)$:

- $P(s^+|l^+) = S_{s+l^+}/Sl^+$

8. Compute $P(s^+|l^+, p)$, where $p$ is the prior probability of a label:

- This is the forward inference posterior probability. Probability of selection given label and given prior probability of label, $p$.
- $P(s^+|l^+, p) = pP(s^+|l^+) + (1 - p)P(s^+|l^-)$

9. Compute $P(l^+|s^+, p)$:

- This is the reverse inference posterior probability. Probability of label given selection and given the prior probability of label.
- $P(l^+|s^+, p) = pP(s^+|l^+)/P(s^+|l^+, p)$

10. Perform a one-way chi-square test to determine if the rate at which studies are selected for a given label is significantly different from the average rate at which studies are selected across labels.

- Convert p-value to signed z-value using whether the number of studies selected for the label is greater than or less than the mean number of studies selected across labels to determine the sign.

11. Perform a two-way chi-square test to determine if presence of the label and selection are independent.

- Convert p-value to signed z-value using $P(s^+|l^-)$ to determine sign.
Supplementary Materials for Power Replication

Exploratory extensions to experiment 2, analysis group 5

In addition to the standard motion analyses, in which the "intercept" was evaluated as the smoothing curve at 35mm and the "slope" was evaluated as the difference in smoothing curve values between 100mm and 35mm, we performed a series of motion analyses in which linear regression was used. In these analyses, the raw analysis values (e.g., QC:RSFC values) were treated as a dependent variate, and the ROI-to-ROI distances were treated as the independent variate. The distances were shifted down 35, so that the 35mm distance would be treated as the intercept. Next, a line was fitted to the analysis values, and the intercept and slope were determined for that line. This procedure was repeated for each of the 10,000 permutations, and the intercept and slope was recorded for each. Finally, significance testing was performed by determining the rank of the original analysis values’ slope and intercept against the permutation-based slope and intercept null distributions.

The linear regression-based analyses produced very similar results to the original analyses, indicating that the choice of intercept and slope measure was not driving the results.
Table A.1 **Distance-dependence analyses of the CamCAN dataset** Results of the distance-dependent motion-related artifact analyses in analysis group 5, limited only to the CamCAN dataset. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with p < 0.05 are bolded.

The results of the CamCAN DDMRA analyses differed considerably from those of the other datasets. In this case, the QC:RSFC and high-low motion analyses’ slopes and intercepts were typically significant for the denoised data (e.g., MEDN+aCompCor, MEDN+GODEC, MEDN+GSR, MEDN+MIR), but not for the discarded noise data or for the undenoised data (e.g., OC).

<table>
<thead>
<tr>
<th></th>
<th>QC:RSFC</th>
<th>high-low motion</th>
<th>scrubbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.6880</td>
<td>0.6135</td>
</tr>
<tr>
<td>MEDN</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.4005</td>
<td><strong>0.0374</strong></td>
</tr>
<tr>
<td>MEDN Noise</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.9800</td>
<td>0.9999</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0030</strong></td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>0.5277</td>
<td><strong>0.0026</strong></td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td><strong>0.0026</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.9991</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>0.9539</td>
<td>0.4698</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td><strong>0.0000</strong></td>
<td><strong>0.0000</strong></td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td></td>
<td></td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>int.</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>1.0000</td>
<td>0.9855</td>
</tr>
</tbody>
</table>
Table A.2 **Distance-dependence analyses of the DuPre dataset**  Results of the distance-dependent motion-related artifact analyses in analysis group 5, limited only to the DuPre dataset. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with p < 0.05 are bolded.

<table>
<thead>
<tr>
<th>Analysis Group</th>
<th>QC:RSFC int.</th>
<th>QC:RSFC slope</th>
<th>high-low motion</th>
<th>scrubbing</th>
<th>MEDN int.</th>
<th>MEDN slope</th>
<th>MEDN Noise int.</th>
<th>MEDN Noise slope</th>
<th>MEDN+aCompCor int.</th>
<th>MEDN+aCompCor slope</th>
<th>MEDN+GODEC int.</th>
<th>MEDN+GODEC slope</th>
<th>MEDN+GSR int.</th>
<th>MEDN+GSR slope</th>
<th>MEDN+MIR int.</th>
<th>MEDN+MIR slope</th>
<th>MEDN+MIR Noise int.</th>
<th>MEDN+MIR Noise slope</th>
<th>MEDN+dGSR int.</th>
<th>MEDN+dGSR slope</th>
<th>MEDN+dGSR Noise int.</th>
<th>MEDN+dGSR Noise slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td>0.0000</td>
<td>0.0090</td>
<td>0.0028</td>
<td>0.0000</td>
<td>0.0059</td>
<td>0.0007</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.0000</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0206</td>
<td>0.0009</td>
<td>0.0000</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0279</td>
<td>0.0021</td>
<td>0.0000</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0217</td>
<td>0.0010</td>
<td>0.0000</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0164</td>
<td>0.0123</td>
<td>0.0000</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0072</td>
<td>0.0057</td>
<td>0.0000</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0073</td>
<td>0.0230</td>
<td>0.6317</td>
<td></td>
<td>0.0000</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
<td>0.2112</td>
<td>0.0909</td>
<td>0.0693</td>
<td>0.0029</td>
<td>0.0451</td>
<td>0.2351</td>
</tr>
</tbody>
</table>

197
Table A.3 **Distance-dependence analyses of the all three datasets, with a regression.** Results of the distance-dependent motion-related artifact analyses in analysis group 5, using a linear regression. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with p < 0.05 are bolded.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>QC:RSFC intercept</th>
<th>high-low motion intercept</th>
<th>scrubbing p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC:RSFC high-low motion</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN</td>
<td>1.0000</td>
<td>0.3485</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>1.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>1.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>1.0000</td>
<td>0.9980</td>
<td>0.9986</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>1.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>1.0000</td>
<td>0.9992</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>0.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>1.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>0.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

198
Table A.4 Distance-dependence analyses of the Cambridge dataset, with a regression. Results of the distance-dependent motion-related artifact analyses in analysis group 5, limited to the Cambridge dataset and using a linear regression. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with \( p < 0.05 \) are bolded.

OC scrubbing slope no longer sig MEDN+aCompCor Noise QC:RSFC and high-low motion slopes no longer sig MEDN+GODEC Noise QC:RSFC and high-low motion slopes no longer sig MEDN+MIR Noise high-low motion slope no longer sig (but it’s close!) MEDN+dGSR QC:RSFC and high-low motion slopes no longer sig

<table>
<thead>
<tr>
<th></th>
<th>QC:RSFC</th>
<th>high-low motion</th>
<th>scrubbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td>int.</td>
<td>0.0275</td>
<td>0.1352</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0016</td>
<td>0.0049</td>
</tr>
<tr>
<td>MEDN</td>
<td>int.</td>
<td>0.3744</td>
<td>0.3186</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0149</td>
<td>0.0383</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>int.</td>
<td>0.0294</td>
<td>0.1344</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0221</td>
<td>0.0342</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>int.</td>
<td>0.0016</td>
<td>0.0936</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.2760</td>
<td>0.8911</td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>int.</td>
<td>0.9391</td>
<td>0.8630</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.1878</td>
<td>0.1331</td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>int.</td>
<td>0.0002</td>
<td>0.1228</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.4220</td>
<td>0.9757</td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>int.</td>
<td>0.9880</td>
<td>0.8766</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.1353</td>
<td>0.1225</td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>int.</td>
<td>0.9145</td>
<td>0.9989</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.9839</td>
<td>0.9997</td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>int.</td>
<td>0.9973</td>
<td>0.9911</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0476</td>
<td>0.0174</td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>int.</td>
<td>0.0117</td>
<td>0.0198</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0193</td>
<td>0.1189</td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>int.</td>
<td>0.6985</td>
<td>0.9142</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0148</td>
<td>0.0616</td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>int.</td>
<td>0.2768</td>
<td>0.2775</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0558</td>
<td>0.2070</td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td>int.</td>
<td>0.4183</td>
<td>0.1468</td>
</tr>
<tr>
<td></td>
<td>slope</td>
<td>0.0203</td>
<td>0.0016</td>
</tr>
</tbody>
</table>
Table A.5 **Distance-dependence analyses of the CamCAN dataset, with a regression** Results of the distance-dependent motion-related artifact analyses in
analysis group 5, limited to the CamCAN dataset and using a linear regression.
The rank-based p-value of both the slope and intercept of each analysis is included.
Analyses with $p < 0.05$ are bolded.

MEDN Noise scrubbing slope no longer sig MEDN+MIR Noise scrubbing intercept now sig MEDN+dGSR QC:RSFC slope now sig

<table>
<thead>
<tr>
<th></th>
<th>QC:RSFC</th>
<th>high-low motion</th>
<th>scrubbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td>int. 1.0000 1.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN</td>
<td>int. 1.0000 1.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>int. 1.0000 1.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>int. 0.0000 0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>int. 1.0000 1.0000</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+GODEC</td>
<td>int. 0.0000 0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>int. 1.0000 1.0000</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+GSR</td>
<td>int. 0.0003 0.0062</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>int. 1.0000 1.0000</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+MIR</td>
<td>int. 0.5004 0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>int. 1.0000 1.0000</td>
<td>1.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+dGSR</td>
<td>int. 0.9728 0.0272</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MEDN+dGSR Noise</td>
<td>int. 1.0000 1.0000</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>
**Table A.6** Distance-dependence analyses of the DuPre dataset, with a regression

Results of the distance-dependent motion-related artifact analyses in analysis group 5, limited to the DuPre dataset and using a linear regression. The rank-based p-value of both the slope and intercept of each analysis is included. Analyses with p < 0.05 are bolded.

MEDN+aCompCor Noise QC:RSFC slope no longer sig MEDN+GODEC Noise QC:RSFC slope and intercept no longer sig MEDN+GSR Noise high-low motion and scrubbing slopes no longer sig MEDN+MIR QC:RSFC slope no longer sig MEDN+MIR Noise high-low motion intercept now sig

<table>
<thead>
<tr>
<th></th>
<th>QC:RSFC</th>
<th>high-low motion</th>
<th>scrubbing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC</td>
<td>0.0000</td>
<td>0.0035</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN</td>
<td>0.0016</td>
<td>0.0176</td>
<td>0.0133</td>
</tr>
<tr>
<td>MEDN Noise</td>
<td>0.0071</td>
<td>0.0007</td>
<td>0.0434</td>
</tr>
<tr>
<td>MEDN+aCompCor</td>
<td>0.0454</td>
<td>0.0271</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GODEC Noise</td>
<td>0.0055</td>
<td>0.0464</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR Noise</td>
<td>0.0732</td>
<td>0.2389</td>
<td>0.9998</td>
</tr>
<tr>
<td>MEDN+MIR Noise</td>
<td>0.2498</td>
<td>0.1819</td>
<td>0.0001</td>
</tr>
<tr>
<td>MEDN+MIR int.</td>
<td>0.0846</td>
<td>0.0108</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+aCompCor Noise</td>
<td>0.0270</td>
<td>0.0012</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+GODEC int.</td>
<td>0.1756</td>
<td>0.2098</td>
<td>0.0001</td>
</tr>
<tr>
<td>MEDN+GSR int.</td>
<td>0.8969</td>
<td>0.3939</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise int.</td>
<td>0.0629</td>
<td>0.0069</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+MIR int.</td>
<td>0.8689</td>
<td>0.2654</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR Noise int.</td>
<td>0.0629</td>
<td>0.0069</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise int.</td>
<td>0.5951</td>
<td>0.8744</td>
<td>0.0023</td>
</tr>
<tr>
<td>MEDN+GSR int.</td>
<td>0.0032</td>
<td>0.0030</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise int.</td>
<td>0.0037</td>
<td>0.0032</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR int.</td>
<td>0.0099</td>
<td>0.0111</td>
<td>1.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise int.</td>
<td>0.0290</td>
<td>0.0211</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR int.</td>
<td>0.1472</td>
<td>0.2508</td>
<td>0.4606</td>
</tr>
<tr>
<td>MEDN+GSR Noise int.</td>
<td>0.0671</td>
<td>0.0218</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR int.</td>
<td>0.0598</td>
<td>0.0426</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR int.</td>
<td>0.0861</td>
<td>0.0826</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+GSR Noise int.</td>
<td>0.0325</td>
<td>0.0136</td>
<td>0.0000</td>
</tr>
<tr>
<td>MEDN+MIR Noise int.</td>
<td>0.0996</td>
<td>0.0362</td>
<td>0.6349</td>
</tr>
<tr>
<td>MEDN+MIR Noise int.</td>
<td>0.0274</td>
<td>0.6754</td>
<td>0.3228</td>
</tr>
</tbody>
</table>
VITA

TAYLOR SALO

2009-2013
B.A., Psychology, Concentration: Behavioral and Evolutionary Neuroscience
Cornell University
Ithaca, New York

2015-2018
M.S., Psychology, Cognitive Neuroscience Program
Florida International University
Miami, Florida

2015-2022
Doctoral Candidate
Florida International University
Miami, Florida

SELECTED PUBLICATIONS


8. Flannery JS, ..., Salo T, ..., & Sutherland MT. (2021). HIV infection is linked with reduced error-related default mode network suppression and poorer med-

¹Co-first authors
ication management abilities. Progress in Neuro-Psychopharmacology and Biological Psychiatry.


