

Tutorials

A vendor-neutral functional MRI acquisition protocol for multi-site studies

Jon-Fredrik Nielsen, Ph.D.^{1,2a}, Maximillian K. Egan, Ph.D.¹, Qingping Chen, Ph.D.³, Mojtaba Shafiekhani, M.S.³, James H. Wang, M.S.^{4,5}, Samuel A. Hurley, Ph.D.⁵, Jeffrey S. Soldate, Ph.D.⁶, Jonathan Lisinski, Ph.D.⁶, Stephen M. LaConte, Ph.D.^{6,7}, Bradley P. Sutton, Ph.D.⁸, R. Allen Waggoner, Ph.D.⁹, Rüdiger Stirnberg, Ph.D.¹⁰, Tony Stöcker, Ph.D.^{10,11}, Jeffrey A. Fessler, Ph.D.^{2,12,13}, Douglas C. Noll, Ph.D.^{1,2}, Maxim Zaitsev, Ph.D.³, Scott J. Peltier, Ph.D.^{1,2}

¹ Functional MRI Laboratory, Department of Radiology, University of Michigan, ² Department of Biomedical Engineering, University of Michigan, ³ Division of Medical Physics, Department of Radiology, University Medical Center Freiburg, ⁴ Department of Radiology, University of Wisconsin–Madison, ⁵ Department of Medical Physics, University of Wisconsin–Madison, ⁶ Fralin Biomedical Research Institute at VTC, ⁷ Department of Biomedical Engineering, Virginia Tech, ⁸ Grainger College of Engineering, Bioengineering Department and Beckman Institute, University of Illinois Urbana-Champaign, ⁹ RIKEN Center for Brain Science, ¹⁰ German Center for Neurodegenerative Diseases, ¹¹ Department of Physics & Astronomy, University of Bonn, ¹² Department of Electrical Engineering and Computer Science, University of Michigan, ¹³ Department of Radiology, University of Michigan

Keywords: fMRI pulse sequences, data harmonization, reproducibility, functional MRI, fMRI

<https://doi.org/10.52294/001c.155279>

Aperture Neuro

Vol. 6, Issue SI 1, 2026

We present an open, vendor-neutral BOLD SMS-EPI protocol tailored for multi-site fMRI studies, intended as a drop-in replacement for conventional vendor-specific acquisition and reconstruction pipelines. Built on Pulseseq—an emerging standard for cross-platform MRI pulse sequence development—our protocol ensures identical SMS-EPI pulse sequences and image reconstruction across scanner vendors. This provides, for the first time, known and consistent experimental conditions across sites and scanner software versions. We begin by reviewing the current capabilities of the Pulseseq framework, including vendor support and safety considerations. We then detail our SMS-EPI implementation and demonstrate its performance using resting-state fMRI pilot data from healthy volunteers, showing reduced site variance compared to corresponding vendor protocols on Siemens and GE scanners. To support adoption, we provide practical resources to help researchers integrate Pulseseq fMRI into their studies, including example text for grant proposals and IRB submissions. These resources are freely available at <https://github.com/HarmonizedMRI/Functional>. Our vision is for Pulseseq fMRI to become the standard for multi-site research, enabling more reproducible science and serving as a reference for the development of novel acquisition and reconstruction methods.

INTRODUCTION

Pooling functional magnetic resonance imaging (fMRI) data across multiple sites can enhance statistical power and support the inclusion of more diverse and representative subject populations. However, these benefits can only be fully realized if acquisition protocols are standardized, transparent, and reproducible, thereby not introducing additional variance to the data. In current practice, most fMRI workflows depend on vendor-specific, partially opaque acquisition and image reconstruction pipelines. However, the scanner software is typically optimized for operator convenience rather than scientific rigor, making the protocols

based on vendor-provided sequences non-portable, difficult to replicate, and poorly suited for cross-site harmonization.

As a result, the stability of the fMRI image formation process cannot be guaranteed across scanners from different MRI vendors—or even across different software versions on the same scanner. For example, Keenan et al.¹ observed inconsistent flip angles (radiofrequency [RF] transmit amplitude) following a scanner software upgrade. Consequently, observed differences in functional measures between sites or over time may reflect systematic differences in protocol implementation rather than genuine biological variation, and make acquisition of new data with identical conditions impossible due to the inability to “roll back” to a previous software version. This lack of reproducibility also

^a Corresponding Author:
Jon-Fredrik Nielsen
jfnielse@umich.edu

creates a barrier to innovation, as implementing new fMRI acquisition strategies within proprietary vendor environments is technically demanding and time-consuming.

This article describes a vendor-neutral, open-source, and fully harmonized fMRI protocol that addresses these challenges. Our acquisition protocol is implemented in Pulseq,² an emerging open standard for describing MRI pulse sequences in a vendor-independent way. This allows for *exact* harmonization of the pulse sequence across vendor platforms down to the individual RF/gradient waveform level, which in turn ensures that important aspects of the acquisition, such as the nominal slice profile and choice of fat suppression method, remain constant across sites and over time. Equally important, Pulseq enables harmonization of the image reconstruction step: since the raw (k-space) data and coil calibration data are acquired in a known way, the image time-series can be reconstructed in exactly the same way regardless of which scanner it came from. This stands in contrast to vendor-specific product protocols, which often employ different—and typically opaque—techniques for parallel imaging (e.g., SENSE,³ GRAPPA,⁴ or deep learning-based methods) and image pre-processing. Pulseq therefore creates an opportunity for the fMRI community to standardize not only the pulse sequence, but also the image reconstruction algorithms and associated scanner and coil calibration scans.

We begin by outlining the fundamentals and current capabilities of the Pulseq framework, including how it enables rapid prototyping and deployment of custom MRI sequences across different vendor platforms, and ensures immunity to scanner software upgrades. This section is primarily intended for MRI pulse sequence programmers and protocol maintainers.

We then describe the implementation of our Pulseq fMRI protocol and the practical steps needed to run the sequence on two major MRI vendor platforms—Siemens and General Electric (GE). We also present traveling subject data that illustrate the feasibility of deploying our protocol in a multi-site research setting. Notably, these results show reduced cross-vendor variability in functional connectivity measures compared to standard vendor-specific (product) protocols.

Finally, we provide recommendations and additional resources for fMRI researchers seeking to incorporate Pulseq into their own experimental protocols and grant applications.

THE PULSEQ PLATFORM

The key technical component of our open-source protocol relates to the initial part of the image formation chain: the signal generation and the data acquisition steps, where we employ the Pulseq platform for vendor-neutral MRI pulse sequence programming. Pulseq is both a conceptual sequence representation, a file specification that encapsulates it, and a set of software tools for creating, analyzing, and executing sequence files. Here we attempt to clarify these concepts and provide MRI physicists or protocol

maintainers with enough information to install and run Pulseq sequences on their scanner.

OVERALL WORKFLOW

The key idea behind Pulseq is to write the MRI pulse sequence specification to a vendor-neutral file (called a “.seq” file) that is to be loaded onto a particular MRI scanner via a vendor-specific interpreter ([Figure 1](#)). This file is plain-text and human-readable and contains the complete specification of every real-time event (RF pulses, gradients, signal acquisition) occurring in the sequence, according to an open specification.⁵ This same Pulseq file can also be used for other purposes, such as performing Bloch simulations, and generating sequence timing diagrams or precise k-space trajectories. The Pulseq standard itself is vendor-neutral and agnostic to the specifics of a particular vendor platform and the implementation details of the specific interpreter.

Once the Pulseq interpreter is installed on a particular scanner, different pulse sequences can be executed by simply selecting the appropriate sequence file from the user interface of the scanner console; the interpreter itself is sequence-agnostic. Importantly, maintaining Pulseq sequences across system software upgrades is simply a matter of recompiling the interpreter, since the sequence files are agnostic to the scanner software version. This greatly simplifies the task of maintaining fMRI protocols across scanner software versions.

A DEEPER DIVE INTO PULSEQ

While this article focuses on a specific simultaneous multi-slice (SMS)-echo planar imaging (EPI) fMRI protocol developed and validated by the authors, researchers across the broader fMRI community may want to modify the sequence to suit their own needs. Whether modifying an existing Pulseq sequence or creating one from scratch, it is important to understand how Pulseq represents the various elements of the sequence, what software tools are available to implement them, and how to ensure compatibility with a particular scanner make and model. This section is intended to support that understanding. However, readers who are primarily interested in using the pre-configured fMRI sequence files provided by the authors may choose to skip it.

SEQUENCE REPRESENTATION

In Pulseq, an MRI pulse sequence is represented as a series of sequential, non-overlapping blocks ([Figure 2](#)). Each block can contain at most one RF pulse, one gradient waveform per axis, and one data acquisition (ADC) window. Apart from this constraint, and vendor-specific hardware limits on peak waveform amplitude, gradient slew rate, and transmit/receive (T/R) switching times, the sequence designer is free to choose where to place the block boundaries. This also means that a gradient event does not need to have zero amplitude at a block boundary, as long as no amplitude discontinuity is introduced. The sequence is

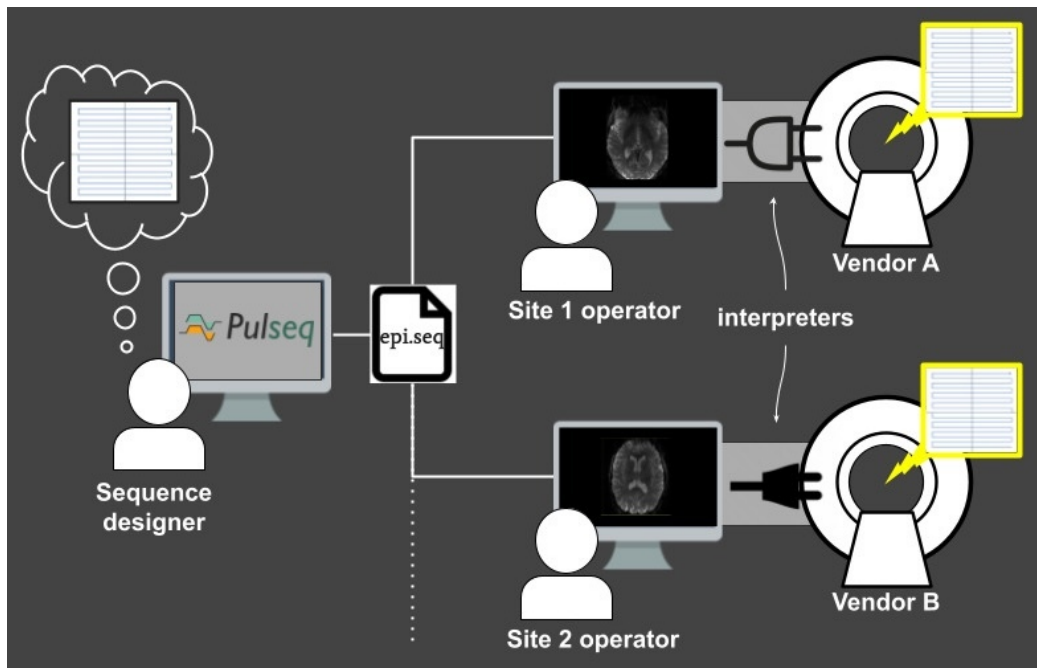


Figure 1. Overview of the Pulseseq platform for implementing and sharing vendor-neutral pulse sequences. The sequence designer creates a .seq file that specifies all low-level details of the sequence that she envisions. That .seq file is shared with scanner operators at different sites. At each site, a vendor-specific and sequence-agnostic interpreter executable loads and executes the sequence on the scanner. This workflow ensures perfect harmonization of the pulse sequence across sites and is easy to maintain across scanner system upgrades since only the interpreter needs to be recompiled. Computer screen image downloaded from <https://commons.wikimedia.org/>.

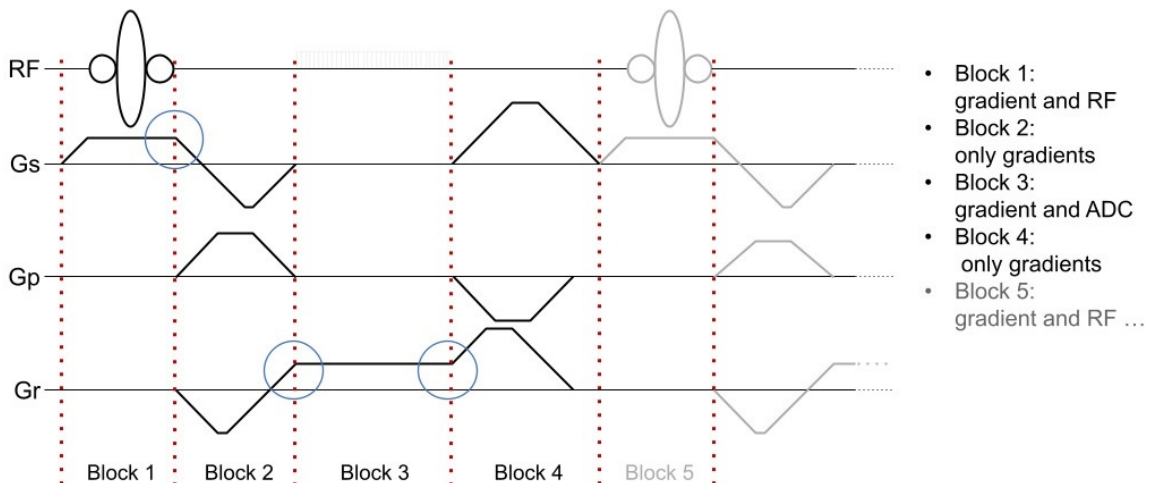


Figure 2. Pulseseq sequence representation. Non-overlapping blocks are executed consecutively and without gaps. Each waveform channel can contain at most one event within a block. Gradient waveforms must be continuous across block boundaries (blue circles). RF: radiofrequency; Gs: gradient along slice-select (“z”) axis; Gp: gradient along phase-encode (y) axis; Gr: gradient along readout (x) axis.

defined explicitly, without formal relationships between blocks or higher-level constructs such as loops or defined repetition time (TR) intervals.

SOFTWARE TOOLS FOR CREATING PULSESEQ FILES

Apart from the ‘official’ MATLAB toolbox, sequence designers can choose from several alternative ways to create a Pulseseq (.seq) file. pyPulseseq⁶ is an open Python implementation of the MATLAB toolbox, and enables sequence creation in environments where it may be difficult to obtain

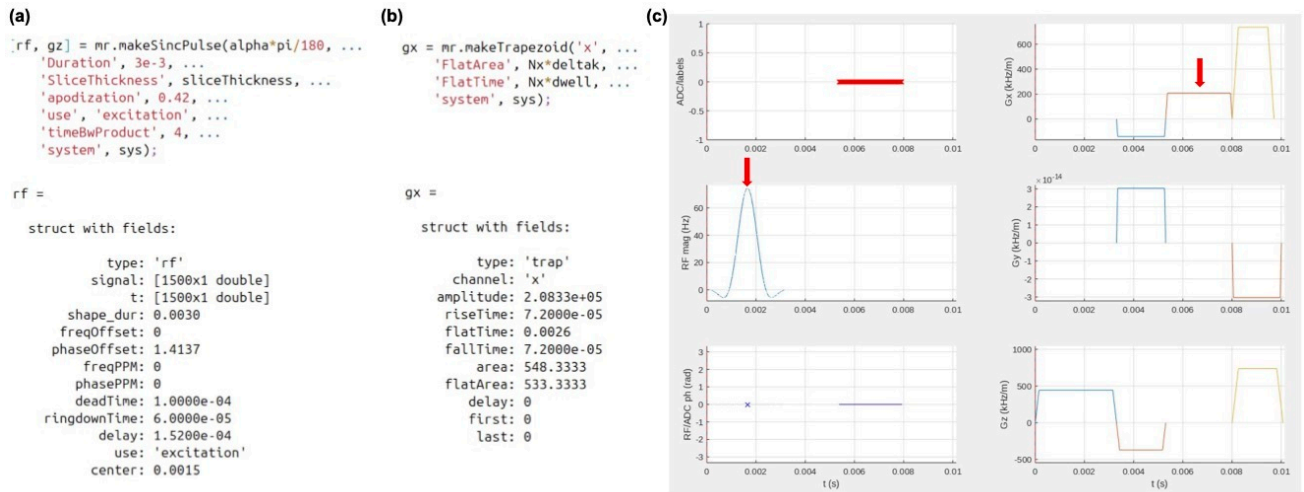


Figure 3. Examples of (a) RF and (b) gradient event creation using the Pulseseq MATLAB toolbox. Once the RF or gradient struct is created, it can be freely modified if desired; it is a simple struct data type. (c) Sequence timing diagram showing the location of these RF and gradient events (red arrows) in a gradient echo sequence.

a MATLAB license. Higher-level design tools also exist,^{7,8} however, the most common way to create Pulseseq sequence files is to work directly with the MATLAB^{2,9} and Python⁶ toolboxes.

DEFINING SEQUENCE EVENTS

The core task for a Pulseseq sequence designer working directly with the MATLAB or Python toolbox is to define the RF, gradient, and ADC events that make up the sequence. In both of these frameworks, each event is described using a small set of parameters encapsulated in a data entity (e.g. an object or a struct). For instance, an RF event must specify its start time relative to the beginning of the block (delay), the complex waveform samples in physical units (signal) and their corresponding time points (t), the frequency offset in Hz (freqOffset) or in parts-per-million (ppmOffset), and the phase offset in radians (phaseOffset). Additionally, system-specific parameters such as deadTime and ringdownTime, which reflect RF coil and amplifier switching behavior, affect the timing of the RF events. The sequence designer can either populate the fields of the structure directly, or use the helper functions provided by the chosen framework to generate common pulse shapes (e.g. sinc RF pulses).^{6,9} The same approach applies to gradient and ADC events that can be created using library functions or by directly populating the required parameters.

ASSEMBLING THE SEQUENCE

Once the individual events have been defined, they need to be assigned to blocks and added to a sequence object. This process typically begins by creating an empty sequence object that is configured for a particular scanner hardware platform, followed by inserting blocks into that sequence object using the `addBlock()` member function. Blocks are added in the order of execution, and the Pulseseq toolbox automatically manages the internal organization of the se-

quence for you. Although the Pulseseq representation does not formally define relationships between blocks, it is common practice to use a loop structure when assembling the sequence in code. Examples of MATLAB and Python code for creating many basic and advanced sequences are freely available on Github; some of these sites are listed in [Table 1](#). Once the sequence object is populated with blocks, it can be plotted for visual inspection and written to a file.

ENSURING COMPATIBILITY WITH SCANNER HARDWARE CONSTRAINTS

It is the responsibility of the sequence designer to ensure that the various RF/gradient/ADC events specified in the .seq file are compatible with the scanner hardware configuration. To aid with this task, Pulseseq defines a list of generic hardware parameters that are set by the sequence designer. These include: RF amplitude, dead time, ringdown time, and raster time; gradient amplitude, slew rate, and raster time; ADC dead time and dwell time; and block duration raster time. The Pulseseq toolbox includes both internal and user-callable checks to ensure that the sequence object adheres to these settings. For a sequence to run on a specific scanner, its design specifications must be equal to or lower than the actual capabilities of that scanner.

AVAILABILITY OF PULSESEQ INTERPRETERS ON VARIOUS MRI VENDOR PLATFORMS

The Pulseseq standard is built on a strict separation between the pulse sequence definition (encapsulated in a Pulseseq object or .seq file) and the system or software that ‘consumes’ it. This has enabled both the research community and scanner vendors to independently develop various third-party Pulseseq tools, including the interpreters that now exist on multiple vendor platforms. [Table 2](#) lists the various interpreters that we are aware of at the time of writing.

Table 1. List of websites containing collections of Pulseseq sequence examples.

URL	Description
https://github.com/pulseseq/pulseseq/tree/master/matlab/demoSeq	A collection of many standard sequences maintained by the Pulseseq development team.
https://github.com/pulseseq/tutorials	A collection of various Pulseseq tutorials covering FID, PRESS, GRE, EPI and some non-Cartesian imaging examples along with the MATLAB source code, raw data examples, and sample reconstruction scripts.
https://github.com/pulseseq/mr-physics-with-pulseseq	Materials for the training program “MR Physics with Pulseseq”. In this course, basic and advanced concepts of MR physics are introduced through theoretical lectures, followed by the corresponding hands-on exercises in pyPulseseq.
https://harmonizedmri.github.io	A list of community-developed Pulseseq protocols. Users can add their own project to this list, by uploading a URL and logo.
https://github.com/HarmonizedMRI	A loose collection of Pulseseq protocols and related tools, developed by the HarmonizedMRI team.

Table 2. List of available Pulseseq interpreters on various human MRI scanner platforms.

Vendor	Developer	Uses an intermediate sequence representation
Siemens	Community researchers ²	No
GE	Community researchers ¹⁰	Yes
Philips	Community researchers ¹¹	No
Philips	Community researchers ¹²	Yes
United Imaging	Vendor	No
Canon Medical	Vendor	No

These interpreters are at various stages of development, and we encourage the reader to contact the respective developer or their vendor representative for the most up-to-date information. However, some general observations can be made. First, several interpreters have been developed by community researchers using only the vendors’ standard pulse sequence application programming interfaces (APIs)—IDEA for Siemens, EPIC for GE, and PARADISE for Philips. These interpreters are openly available within each vendor’s respective research user community. The fact that they rely solely on widely accessible development tools, combined with their open availability, supports the ongoing development and long-term sustainability of these interpreters. Another important benefit of implementing the interpreters using the vendor APIs is that Pulseseq sequences can be plotted and simulated using the standard vendor tools. [Figure 5\(a\)](#) shows an example of this, where our SMS-EPI sequence is simulated and plotted using GE’s native sequence tools (WTools and Pulse View, respectively).

Second, some interpreters require an intermediate sequence representation before execution can proceed. This is the case for GE scanners, where the internal sequence format depends on recognizing repeating subunits, such as a sequence TR or a magnetization preparation segment. Currently, users can convert sequences into this intermediate form manually using openly available tools,¹³ or take advantage of the recently introduced Pulserver framework¹⁴ that automates the conversion process. In addition to simplifying sequence deployment, Pulserver also enables

more interactive sequence prescription directly on the scanner console.

Third, vendors have generally demonstrated strong support for the Pulseseq platform, as evidenced by the development of vendor-authored interpreters and the substantial technical assistance provided to community developers.^{14, 15} Vendors appear to recognize the value of supporting their research users through Pulseseq, both as a rapid prototyping tool and as a means to facilitate protocol harmonization across sites.

Lastly, as broad vendor platform support continues to emerge—each with its own software and hardware constraints—there will be a growing need for free and open-source software tools that enable researchers to verify the compatibility of a Pulseseq sequence with the available scanner hardware. Developing such tools offers an opportunity for community researchers and developers to actively contribute to the Pulseseq project.

PATIENT SAFETY AND SYSTEM HARDWARE PROTECTION

Deploying an MRI pulse sequence requires careful consideration of both subject safety and hardware protection. Subject safety focuses primarily on limiting RF power deposition (specific absorption rate [SAR]) and peripheral nerve stimulation (PNS), while hardware protection involves ensuring the integrity of the RF and gradient subsystems. In sequences developed using vendor-provided pulse programming toolkits (APIs), these safety checks are typically

handled either automatically by the scanner or manually by the sequence designer through API-accessible safety routines. The Pulseq interpreters also benefit from these built-in system and API-level safety mechanisms. Additionally, Pulseq allows SAR and PNS to be estimated during the sequence design phase—before the sequence is deployed to the scanner. Here we summarize the various safety checks employed by Pulseq, focusing on the two vendor platforms used so far in this project (Siemens and GE).

RF POWER DEPOSITION (SAR)

To our knowledge, all human MRI scanners are equipped with hardware for monitoring RF power deposition. This hardware monitor is always fully engaged, for Pulseq and non-Pulseq sequences alike. In addition, the Siemens and GE interpreters use the API safety routines in the same way as native (product/research) sequences. The Pulseq sequence designer can perform additional checks during the design phase to reduce the chance that the sequence will trip the scanner's built-in and API SAR checks, such as comparing the nominal RF power to a known reference scan; planning ahead in this way is particularly useful for multi-site studies.

PERIPHERAL NERVE STIMULATION (PNS)

The sequence designer can (and should!) model PNS during the design stage using either the SAFE model for Siemens scanners,¹⁶⁻¹⁸ or the IEC nerve impulse response function model for GE scanners.¹⁹ Open MATLAB implementations by our group and others of these PNS models are freely available on Github.^{16,20} In addition, some interpreters (e.g., Siemens) perform online PNS monitoring, while others are planning to do so in a future release (e.g., the community-developed GE interpreter). Developing a vendor-neutral, standardized PNS model that can be employed across a broader range of vendors and scanner models is an open problem.

HARDWARE PROTECTION

Like pulse sequences implemented with the native API, the Pulseq interpreters rely on API-accessible safety routines to ensure hardware safety. For example, the GE API (EPIC) defines the subroutine `minseq` to calculate the minimum allowable TR for a given gradient waveform. RF subsystem checks are performed in a similar way. In this way, the vendor does not need to expose any proprietary details about the underlying safety checking algorithms or hardware performance; only the subroutine output (e.g., minimum allowable TR) is needed by the interpreter. This also means that responsibility for hardware protection rests solely on the interpreter, and not on the sequence designer.

PROPOSED VENDOR-NEUTRAL, OPEN-SOURCE FMRI PROTOCOL

PULSE SEQUENCE IMPLEMENTATION

We implemented a BOLD SMS-EPI sequence with parameters that closely match the ABCD protocol²¹: 2.4 mm isotropic resolution, 90x90 matrix size, 60 slices without gaps, SMS factor 6, 1x6_{z2} CAIPI sampling pattern,²² TR 0.8 s, partial Fourier factor 0.8. We created the various sequence events, and assembled the Pulseq sequence object, using the MATLAB Pulseq toolbox. The MATLAB scripts are freely and openly available on Github,²³ enabling users to modify the acquisition parameters to suit their own needs.

[Figure 5\(a\)](#) shows the timing diagram for one sequence TR, which consists of fat saturation using a minimum-phase Shinnar-Le Roux (SLR) pulse,²⁴ SMS excitation using SLR sub-pulses with phase offsets following the scheme by Wong,²⁵ and ramp-sampled EPI readout. To reduce the risk of PNS during the EPI train, we derated the slew rate to ensure that the predicted PNS, based on the nerve impulse response model¹⁹ stays below 80% of the stimulation limit (normal mode). The sequence is RF-spoiled to suppress steady-state signal contributions from spin- and stimulated-echo pathways.²⁶

IMAGE RECONSTRUCTION

To reconstruct images from the acquired raw (k-space) SMS-EPI data, the user is free to choose from several available image reconstruction pipelines. The HarmonizedMRI development team has implemented both 3D SENSE²⁷ and slice GRAPPA²⁸ SMS-EPI reconstructions in MATLAB that are part of the QA protocol described below. Other community-developed implementations are also available.²⁹⁻³³ [Figure 5\(b\)](#) shows the results of reconstructing Pulseq SMS-EPI data using 3D SENSE²⁷ implemented in BART.³² [Figure 6](#) demonstrates that our Pulseq protocol achieves similar image contrast and overall image quality as the matched product protocols.

In our current workflow, image reconstruction is done with slice GRAPPA due to its fast computation speed and robust image quality, and zero-filling partial Fourier (PF) reconstruction. Other choices are possible, such as multi-coil reconstructions with spatial regularization, and homodyne PF reconstruction. The extent to which these image reconstruction and post-processing choices impact fMRI reproducibility is an open research question.

RESULTS

IMAGE QUALITY METRICS

To quantitatively assess overall image quality, we analyzed time-series data in phantoms and a volunteer obtained with both the proposed Pulseq SMS-EPI protocol and the vendors' native (product) protocols on two vendor platforms (GE and Siemens). For the product protocols, images were reconstructed using the vendor-provided reconstructions

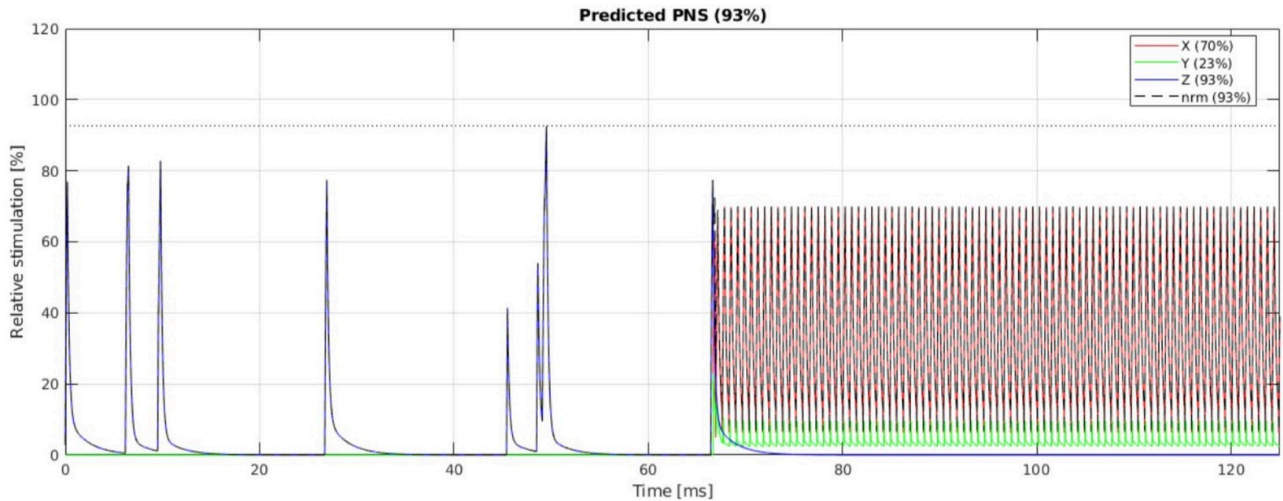


Figure 4. Example of PNS prediction in MATLAB for a Siemens PRISMA scanner using the SAFE¹⁷ model implementation by Szczepankiewicz et al.¹⁶ Such modeling can be done prior to executing the sequence on the scanner, as a step toward ensuring patient safety.

available on the scanner. [Figure 7\(a\)](#) displays the ABCD quality metrics calculated for phantom data acquired on both a GE MR750 and a GE UHP scanner. These values are compared against reference values acquired in the ABCD study. We observe that Pulseseq performs favorably against the vendor ABCD protocol. The higher SNR in the ABCD UHP acquisition is partly ascribed to the full ky coverage, as opposed to the fractional ky coverage on the MR750 and in the Pulseseq acquisition. As the Pulseseq acquisition is the same across scanners, SNR values are more similar across the two scanners as compared to the ABCD protocol.

[Figure 7\(b\)](#) plots the FWHM values for the product multi-band sequences (blue) vs Pulseseq implementation (red) in both GE and Siemens scanners, for resting-state data acquired in a traveling subject, with four repeats at each site. Full width at half maximum (FWHM) is a measure of spatial autocorrelation, with lower values being desirable (indicating higher resolution). It is seen that there is reduced variance across scanners and vendors with the Pulseseq ABCD acquisition compared to the product ABCD acquisition (ABCD: Adolescent Brain and Cognitive Development study²¹).

CROSS-VENDOR REPRODUCIBILITY OF RESTING-STATE FUNCTIONAL CONNECTIVITY

To assess the potential of our Pulseseq fMRI protocol for improving cross-vendor reproducibility of resting-state fMRI metrics, we conducted two small traveling subject studies. In the first study, one traveling subject visited 5 separate scanners, and underwent four runs of a five-minute resting-state scan at each site. [Figure 8](#) summarizes the results of this study. It is seen that the Pulseseq average connectivity matrix is comparable to the Product connectivity matrix, and that the standard deviation has fewer regions of relatively high cross-scanner variation (though this may be due to subject effects). In other words, this data suggests that harmonizing the acquisition (with Pulseseq) as well as the

SMS-EPI reconstruction results in more consistent resting-state functional connectivity (rsFC) metrics across these 5 scanners.

In the second study, 7 volunteers each underwent two separate scan sessions on different days and on different MRI vendor platforms (Siemens and GE). Each session consisted of a T1 anatomical scan and two 5-minute rest scans, one using the Pulseseq protocol and the other the product protocol. In the same session we also acquired 4 repetitions of a block finger-tapping and visual activation task using both Pulseseq and product protocols; analysis of the task data is ongoing and will be presented elsewhere. [Figure 9](#) summarizes the resting-state analysis from this study. We observed that the within-subject, cross-scanner resting-state FC matrices tend to be more similar when using the proposed Pulseseq protocol compared to the product protocols.

FUNCTIONAL CONNECTIVITY METHODS

RsFC matrices were calculated for each scan using the CONN toolbox³⁵ release 22.v2407³⁶ and SPM³⁷ release 12.7771. Functional and anatomical data were preprocessed using CONN's default preprocessing pipeline³⁸ (minus slice time correction) including realignment, outlier detection, direct segmentation and MNI-space normalization, and smoothing with a 6mm kernel. In addition, functional data were denoised using CONN's standard denoising pipeline³⁸ including the regression of potential confounding effects characterized by white matter and cerebrospinal fluid (CSF) timeseries, motion parameters and their first order derivatives, outlier scans, session and scan effects and their first order derivatives, and linear trends within each resting state run, followed by bandpass frequency filtering of the timeseries between 0.008 Hz and 0.09 Hz. The rsFC matrices were calculated using the nodes from the Power functional network atlas.³⁹

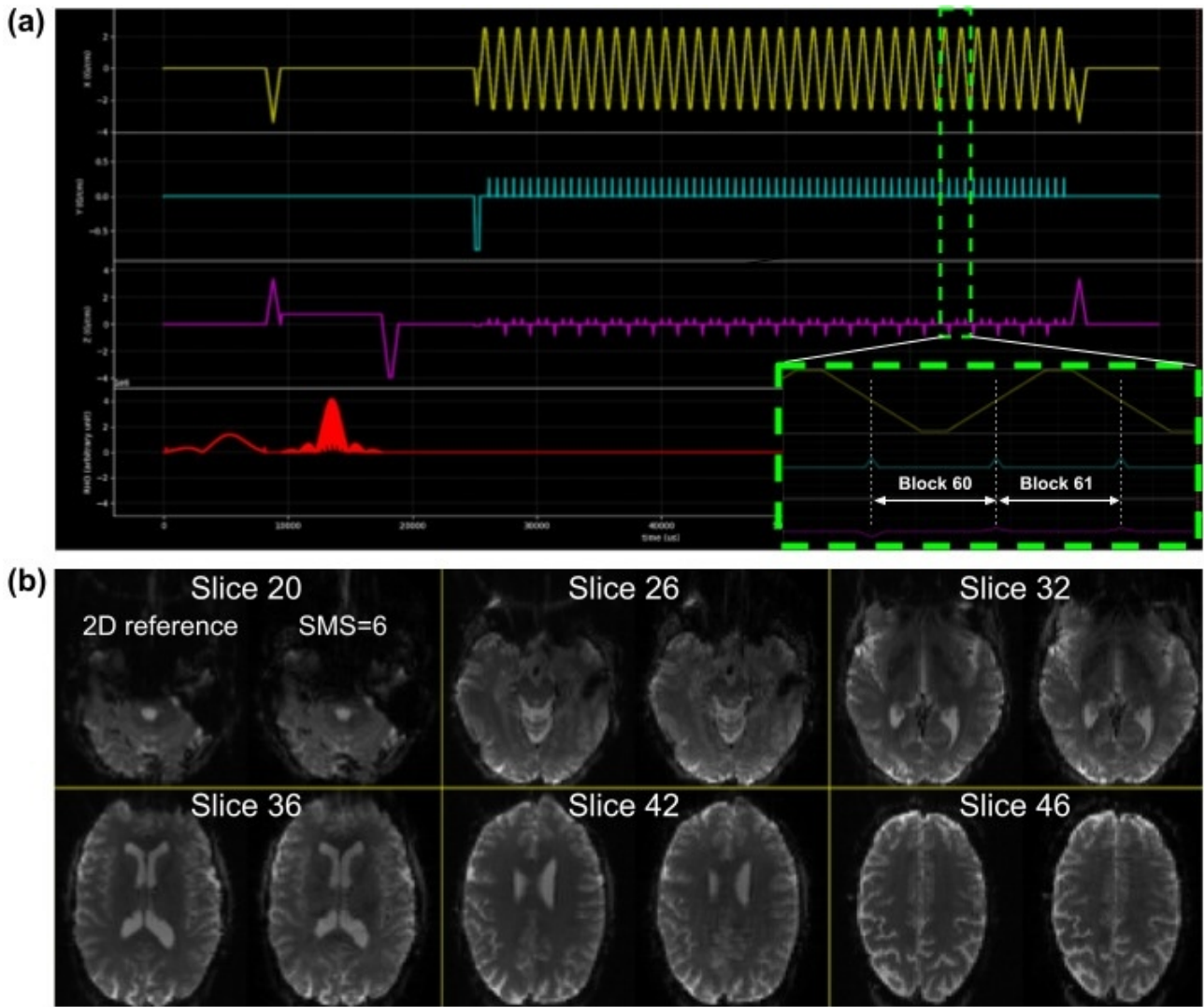


Figure 5. (a) Proposed Pulseseq SMS-EPI sequence simulated in GE’s sequence simulator Pulse Studio. The sequence closely matches the ABCD SMS-EPI protocol (SMS factor 6; 1x6z2 CAIPI sampling pattern; 2.4 mm iso; FOV = [21.6 21.6 14.4] cm). The inset shows gradient waveform details for two consecutive blocks within the EPI train. (b) Volunteer brain images obtained with the sequence in (a). Six slices are shown: each panel shows a non-SMS (2D) reference EPI image on the left, and the SMS=6 image on the right. The latter was reconstructed with SENSE using BART.

EXAMPLE USAGE

To help users get started, we created a complete example showcasing the entire workflow — from data acquisition to generating an image time series. You can explore it here: <https://github.com/HarmonizedMRI/SMS-EPI/tree/main/example>. The demo includes step-by-step instructions, links to all source code, and access to a sample dataset (available upon request via a Google form).

INCORPORATING PULSESEQ FMRI IN YOUR RESEARCH: RECOMMENDATIONS AND ADDITIONAL RESOURCES

Thus far, we have focused on the technical implementation of the Pulseseq fMRI protocol and presented traveling subject data that demonstrate its scientific value. Based on the in-

formation provided in the preceding sections and the associated GitHub repositories, technically proficient users should be able to install the Pulseseq SMS-EPI protocol, customize it as needed, and assess its suitability for their research. In this section, we provide additional guidance and recommendations for users at varying levels of engagement and pulse sequence expertise, from initial installation and testing to the planning and execution of multi-site research studies.

QUALITY ASSURANCE (QA) PROTOCOLS

To ensure correct installation of the proposed Pulseseq fMRI protocol and consistent performance across sites, we recommend that the user executes two QA protocols. The first is described in detail in Chen et al.⁴⁰ and involves scanning a uniform ball phantom (preferably FBIRN/EZfMRI) using a set of separate Pulseseq sequences available at

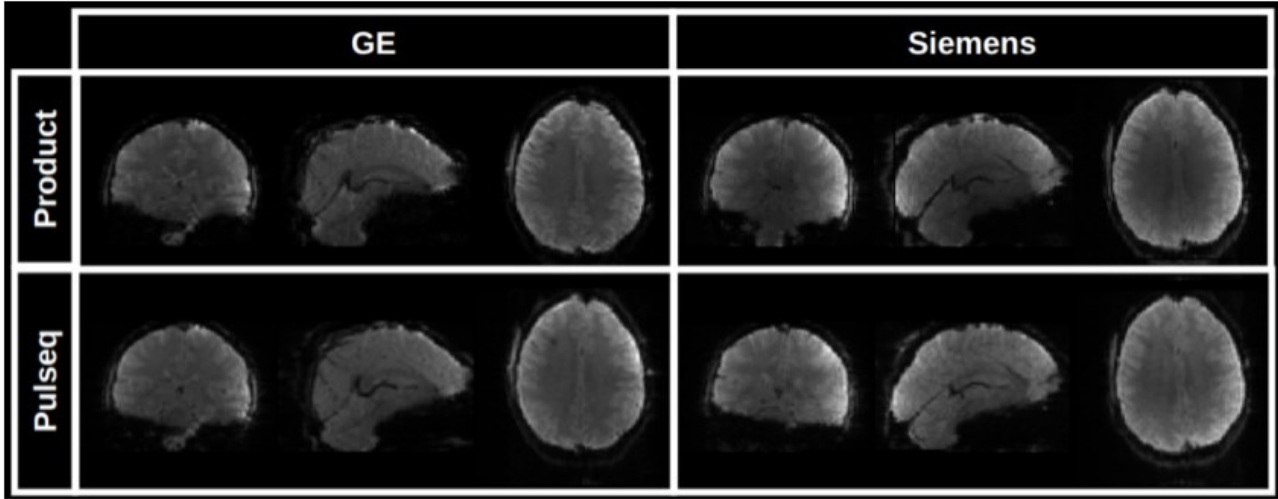


Figure 6. Image comparison between the product (vendor) SMS-EPI protocols (top) and our proposed Pulseseq SMS-EPI protocol (bottom). The same subject was scanned on two different scanners (GE and Siemens). The vendor protocols relied on each vendor's built-in—and closed—image reconstruction algorithm. We obtained the Pulseseq images by reconstructing the data offline in MATLAB, using slice GRAPPA.

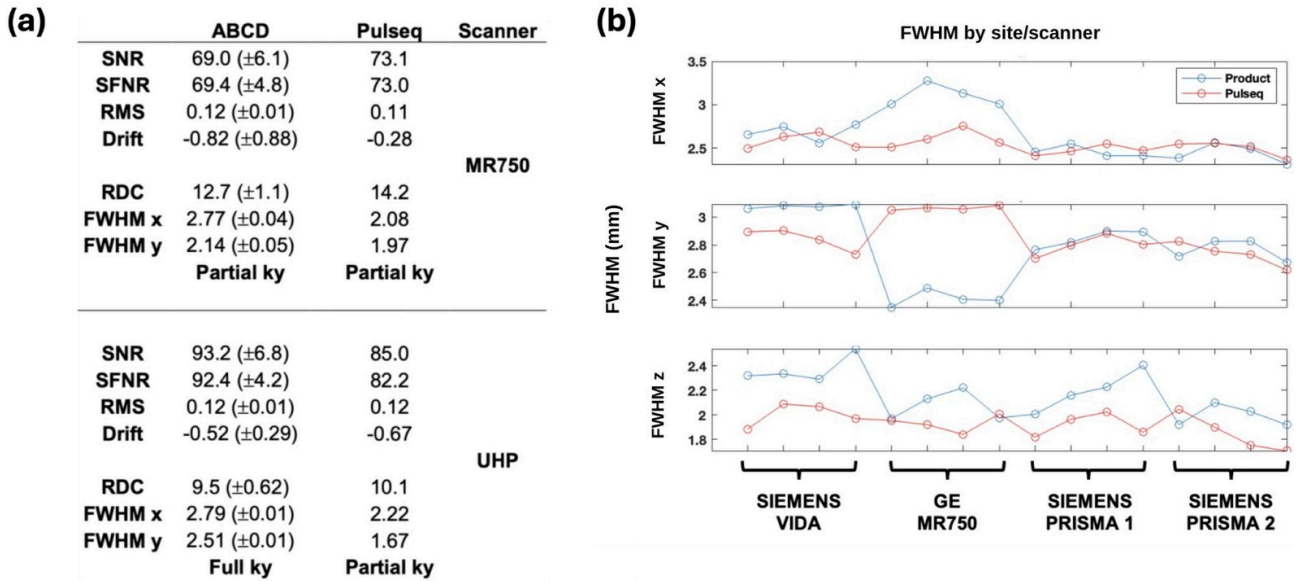


Figure 7. (a) FBIRN phantom metrics for GE MR750 (top) and GE UHP (bottom) MRI scanners, using either ABCD (left) or Pulseseq implementation of SMS-EPI. ABCD phantom metrics were taken as the average of multiple phantom scans (MR750: N=120, UHP: N=189). ABCD implementation of SMS used partial ky acquisition on MR750, full on UHP. (b) FWHM metrics for a traveling subject across four scanners, with four resting-state scans on each.

SNR: image signal-to-noise; SFNR: signal fluctuation to noise ratio; RMS: root mean square residual after polynomial fitting; RDC: radius of decorrelation; FWHM: spatial auto-correlation full width half maximum. ABCD: adolescent brain and cognitive development study.

<https://github.com/HarmonizedMRI/qualityAssurance/tree/main>. The data is then processed using a set of MATLAB scripts that are also available at that Github site, and that produce performance metrics including signal-to-noise ratio (SNR), temporal SNR, and EPI ghost to signal ratio.

We also recommend calculating quality metrics using the ABCD QC protocol (which is based on the fBIRN protocol) on an fBIRN/EZfMRI phantom, as implemented in the ABCD study (<https://github.com/ABCD-STUDY/FIONA-QC-PHANTOM>). Besides the metrics mentioned above, this protocol also calculates RDC (radius of decorrelation), and

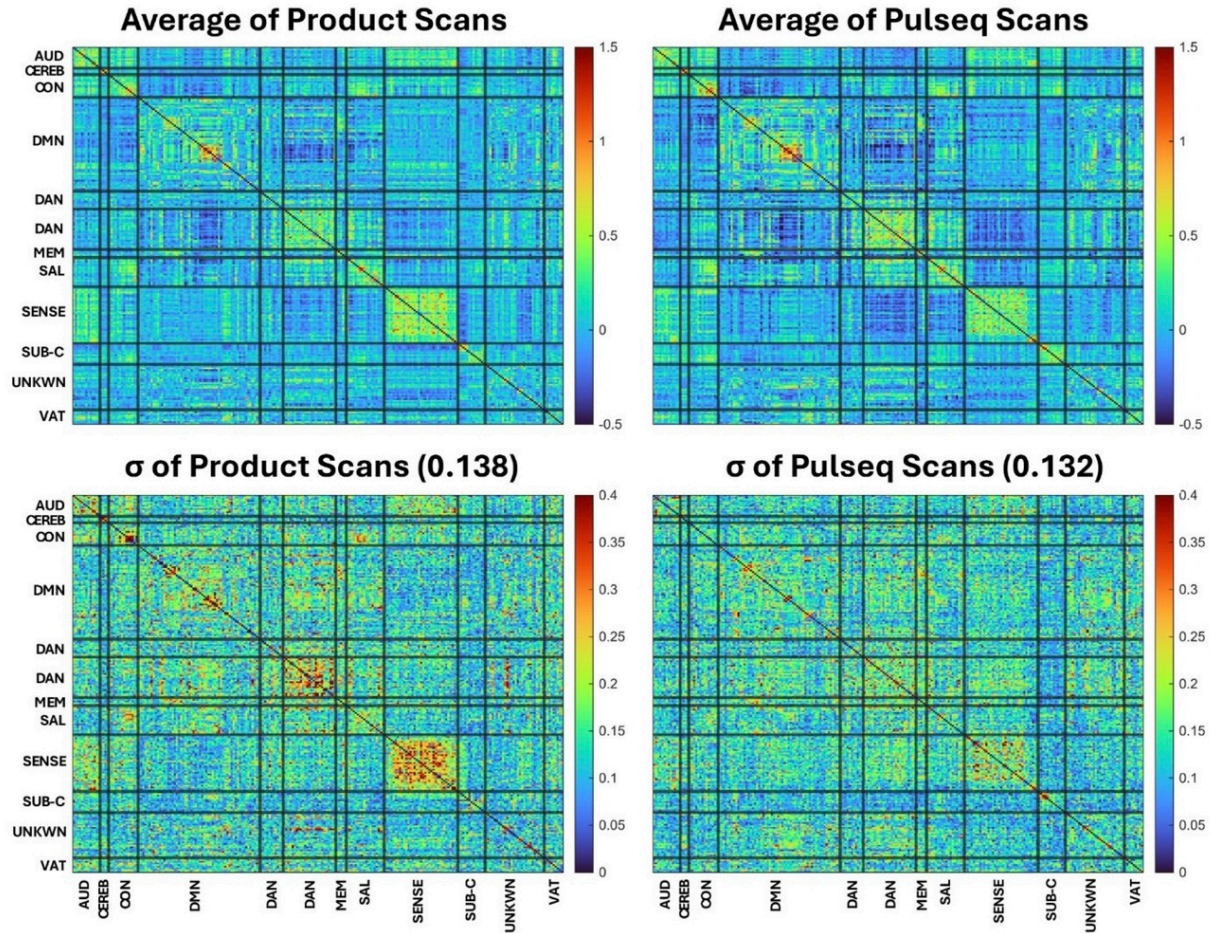


Figure 8. Reproducibility of resting-state functional connectivity (rsFC) matrices in the same subject across 5 different 3T scanners (2 GE; 3 Siemens). A resting-state connectivity matrix is first created from each run by correlating the time-series signals averaged over an atlas-based region of interest (ROI) with the time-series signals from all other ROIs, and calculating the Fisher z transformation³⁴ of the resulting Pearson correlation coefficients. The resulting rsFC matrices (5 Product, 5 Pulseseq) were then averaged, separately for the Product and Pulseseq scans, to produce the two maps in the top row. As expected, these mean rsFC maps are qualitatively similar. The bottom row shows the corresponding standard deviation across the 5 scanners, which is a measure of across-scanner reproducibility; we observe that the Pulseseq protocol produces more consistent rsFC maps (lower standard deviation) for several of the larger networks (e.g., sensory and frontoparietal networks).

FWHM for each spatial axis; both of which can help detect issues in image quality.

For both QA protocols, we further recommend that the user compare their results against the product protocol on their scanner, and against normative values that we provide.

REPRODUCIBILITY BEST PRACTICES

To maximize transparency and reproducibility, we recommend that users follow standard reproducibility best practices, by:

- Using and documenting a specific version of the framework used to create the .seq file.
- Using and documenting a specific version of the interpreter.

- Using and documenting a specific version of the QA protocol sequences.

We also recommend that users consider preregistering their study with a link to a specific version of the sequence (e.g., Zenodo DOI linked to a specific Github commit), for maximum reproducibility. For further guidance on making MRI pipelines more reproducible, we refer the reader to a recent review article by Tamir et al.⁴¹

GETTING TECHNICAL HELP AND CONNECTING WITH OTHER USERS

Our open fMRI protocol is part of a broader ecosystem of open-source software for sequence design and image reconstruction, and vendor-specific Pulseseq interpreters that are typically only available within each vendor's research user

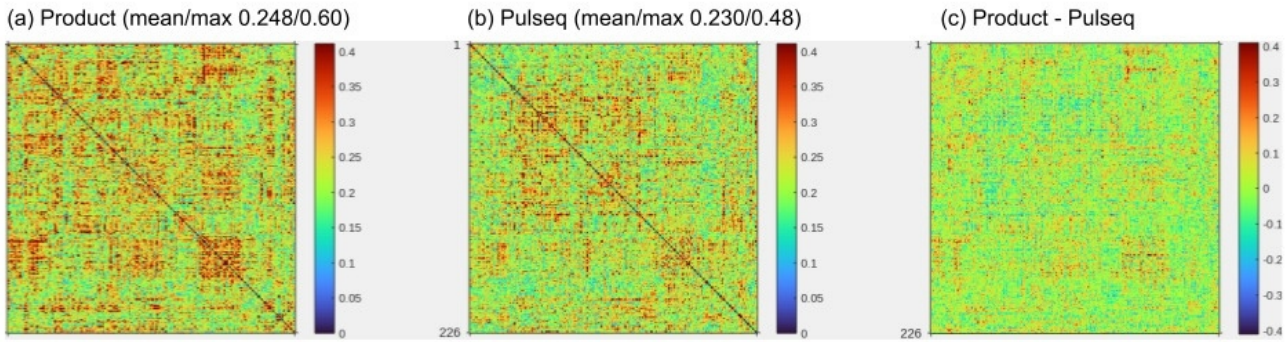


Figure 9. Summary of resting-state SMS-EPI data in 7 volunteers, each scanned on two different 3T scanners (GE MR750 and Siemens Vida). (a-b) Within-subject absolute difference in rsFC maps between the two scanners, averaged over all subjects. The mean (across subjects) absolute difference between the (a) product and (b) Pulseq protocols are shown. The mean and maximum value are indicated. Smaller values (blue and green) are desirable, as it indicates closer agreement between Siemens and GE. The maximum cross-vendor difference is reduced by 20% using Pulseq compared to the product protocols. On average, Pulseq achieves a 7% reduction in cross-vendor rsFC differences. These rsFC difference maps were calculated from images that were smoothed with a 6 mm smoothing kernel as is standard practice, which is likely to mask any differences between the scans. (c) Difference map obtained by subtracting (b) from (a).

community. As such it can be difficult for the resource user to know where to look for guidance on specific components of the proposed fMRI workflow, particularly in a multi-vendor, multi-site setting. Here we provide a few resources to help orient the user. For help getting started with the proposed SMS-EPI fMRI protocol that is the focus of this article, see <https://github.com/HarmonizedMRI/Functional>.

GENERAL HELP WITH PULSEQ

The formal Pulseq specification and the source code for the MATLAB toolbox are available at <https://github.com/pulseq/pulseq>. That site also contains an active discussion forum and a bug report forum; these are named ‘Discussions’ and ‘Issues’, respectively, following the usual Github platform convention. In addition, several recorded Pulseq talks are available online.

PULSEQ USER GROUP MEETINGS

Beginning in September 2025, we will host a regular virtual meeting for the Pulseq user community, to provide technical help in an ‘office hour’ format and give community researchers an opportunity to present and discuss their work in an informal setting. This meeting is open to everyone – a registration link is provided on the project home page (<https://github.com/HarmonizedMRI/Functional>).

DIRECT HELP FROM MEMBERS OF THE HARMONIZEDMRI STUDY TEAM

We are making a fee-based service package available to funded fMRI studies that includes assistance with protocol installation and QA, protocol maintenance across scanner software upgrades, and consultation. We suggest that PIs budget for this service package in their grant proposals; suggested template text to include in the National Insti-

tutes of Health (NIH) Budget Justification section is described below.

WRITING A MULTI-SITE RESEARCH STUDY GRANT PROPOSAL

To assist Principal Investigators (PIs) in integrating the proposed fMRI protocol into future multi-site research studies, we offer example/template text suitable for inclusion in various sections of a grant proposal.⁴² While the template is primarily designed for NIH applications, it can be adapted to meet the requirements of other funding agencies. Users are encouraged to tailor the text as needed to align with their specific project goals and proposal style. The templates include the following:

- Scientific and technical rationale for using Pulseq (Research Plan section)
- QA report from all study sites (Research Plan or Facilities & Resources)
- Budget justification section:
 - Funding for FBIRN phantom (for QA protocol)
 - Tiered support subscription service (optional)
- Human subjects protection section (Pulseq-specific statements regarding safety)
- Letter of support from MRI scanner manufacturer(s) (optional)

While optional, we recommend including letters of support from the vendors. From the vendors’ standpoint, executing a Pulseq sequence is equivalent to running any custom pulse sequence developed by a research user through the vendor’s programming APIs. Consequently, each scanner participating in a proposed multi-site study must have an active research license. Provided this requirement is met, vendors should be able to furnish a letter of support granting permission to run the Pulseq fMRI protocol, con-

firming access to the standard technical support channels available to research users on that platform, and—ideally—endorsing the scientific rationale of both the Pulseseq fMRI protocol and the broader Pulseseq initiative. To facilitate this process, we provide a sample letter of support that incorporates these elements, that users are free to adapt to their own needs.

PREPARING THE IRB PROTOCOL

PULSEQ-SPECIFIC IRB PROTOCOL

As noted above, Pulseseq sequences are implemented using the standard pulse sequence programming environments provided by each vendor (e.g., EPIC for GE, IDEA for Siemens) and are subject to the same safety checks and hardware constraints as any other research sequence. We nevertheless recommend preparing a dedicated IRB protocol specifically for general scanning with Pulseseq sequences. This protocol only needs to be created once and can significantly reduce administrative burden by eliminating the need to repeatedly explain the nature and risks of Pulseseq in each new study. The Pulseseq IRB protocol should be used in conjunction with the study-specific IRB protocol, and participants should provide informed consent for both.

JOINING THE UNIVERSITY OF MICHIGAN MULTI-SITE (SINGLE) IRB PROTOCOL

An alternative for researchers based in the United States is to cede IRB oversight to the University of Michigan. This eliminates the need to prepare a site-specific Pulseseq IRB application, and further encourages scientific collaboration and data sharing between research sites. The University of Michigan multi-site (“single”) IRB protocol covers general Pulseseq scanning including both functional and structural MRI, and is currently in preparation. We share the scientific protocol text publicly,⁴³ which can also be used as a template for preparing a site-specific Pulseseq IRB application. The University of Michigan Pulseseq IRB protocol should be used in conjunction with the study-specific (typically local) IRB protocol, and participants should provide informed consent for both.

DISCUSSION

The proposed Pulseseq fMRI protocol enables identical sequence execution and image reconstruction across multiple scanner platforms, providing highly reproducible conditions for multi-site fMRI research. While this approach minimizes variability, some inter-site differences inevitably remain—for example, variations in receive coil arrays and RF transmit coil configurations. Another source of variability is the B0 shimming protocols employed by different vendors; however, recent efforts by our group⁴⁴ and others⁴⁵ have begun to address this issue. A more subtle factor is the accuracy of gradient trajectories, which can be influenced by differences in gradient hardware performance and vendor-specific eddy current compensation routines. Developing a standardized method for measuring gradient trajec-

tories—and evaluating their impact on the quality of the derived functional maps/measures—remains an area for future research.

As currently implemented, our Pulseseq fMRI protocol only allows limited prescription-time flexibility. While the field-of-view (FOV) can be freely moved and rotated, most other acquisition parameters are fixed including the matrix size and SMS acceleration factor; modifying these parameters requires editing the MATLAB script and creating a new .seq file. In some imaging settings, notably during the course of a study, fixing the FOV and matrix size can be an important beneficial feature to ensure maximum reproducibility. On the other hand, during initial image quality assessment and protocol optimization, it can be beneficial to allow a more interactive sequence prescription, ideally using the vendor’s built-in user interface (UI). Pulserver¹⁴ is a recent effort to provide this functionality, but has so far only been demonstrated on one vendor platform. Another approach is the gammaSTAR framework,⁴⁶ which uses a dedicated user interface and sequence generation backend. Ongoing work is exploring the use of these frameworks to support interactive, vendor-neutral MRI protocol development.

At present, our fMRI protocol includes offline image reconstruction, which is generally sufficient for functional imaging studies. However, clinical research settings can benefit from online (on-the-host) image reconstruction and display similar to the built-in sequences, e.g., as an immediate image quality check and to integrate directly with the standard DICOM image workflow and storage. To our knowledge, most MRI vendors now offer containerized, online image reconstruction environments that allow research users to plug their own custom algorithms into the scanner’s image reconstruction and display pipeline. Together with a more interactive sequence prescription as just described, these containerized environments will enable vendor-neutral (f)MRI protocols that have a similar ‘look and feel’ as built-in protocols.

The vendor interpreters we have tested so far (Siemens, GE) do not yet allow real-time sequence updates, as would be necessary for, e.g., navigator-based prospective motion correction. While existing fMRI protocols in routine use rely on retrospective image co-registration, prospective motion correction may have certain advantages, such as reduced slice interpolation artifacts. To our knowledge, the vendors’ real-time sequence update mechanisms are available to their research users (including the Pulseseq interpreter developers), making it possible to modify the Pulseseq interpreters accordingly. Implementing and testing real-time sequence updates for Pulseseq remains an area for future research.

The current SMS-EPI acquisition and reconstruction methods we offer are based on well-established approaches, but there remains potential for further improvement. We anticipate—and actively encourage—the research community to use this implementation as a foundation or benchmark for developing improved fMRI protocols. Furthermore, as improved methods become available, individual components of our workflow can be replaced with state-

of-the-art alternatives. For example, in our current Pulseq implementation, the block boundaries within the EPI train are centered on the ky/kz blips. This choice is robust but memory-intensive, because each blip is shared between neighboring blocks and must therefore be represented with unique gradient shapes rather than a single scaled trapezoid. On the reconstruction side, advances in spatio-temporal reconstruction are likely to yield more robust functional measures and may require frame-to-frame variation in ky/kz sampling patterns. Other groups have also implemented fMRI and EPI sequences in Pulseq,^{47,48} that may be better suited for such emerging acquisition and reconstruction strategies.

The data presented here do not identify the specific sources of vendor-related differences in functional measures—whether they arise from hardware, pulse-sequence implementation, or reconstruction algorithms. Determining the relative contributions of these factors is an important direction for future research. In particular, if vendor-specific sequence components that lie outside the MRI physicist’s control (e.g., fat-suppression method, slice profile) prove to have only a minor impact on reproducibility, future multi-site (f)MRI studies could focus harmonization efforts on high-level acquisition parameters and the reconstruction stage. Several open-source reconstruction packages are available to support such efforts, including BART,³² MIRT,^{49,50} grappa-tools,²⁹ and the slice GRAPPA MATLAB implementation developed for the present work.⁵¹

In addition to the patient and scanner safety considerations discussed above, the use of Pulseq-based sequences for clinical diagnosis requires assurance that the resulting images are geometrically accurate and free of artifacts beyond those expected from comparable product sequences. Basic system-dependent factors—such as the sign conventions for gradient waveforms or the RF frequency offset—vary across vendors and, if not correctly handled, can lead to left–right image flips, excitation of incorrect slices, or inadequate fat suppression. To address these issues, a standardized Pulseq calibration protocol will be needed to characterize these vendor-specific properties of an MRI system. The Pulseq interpreter would then use this information to ensure that any .seq file is executed exactly as intended.

CONCLUSION

The proposed fMRI protocol harmonizes all software-related aspects of an SMS-EPI protocol, from data acquisition to image reconstruction. The protocol is fully open source, and has so far been tested on two vendor platforms (Siemens and GE) with more expected in the near future. We provide resources to help researchers draft grant and IRB applications that use the proposed Pulseq fMRI protocol, and customer support for sites without a dedicated MRI physicist.

DATA AND CODE AVAILABILITY

Code for Pulseq sequence file generation and image reconstruction: The code used in this manuscript to create the Pulseq sequence files and reconstruct the data to form time-series images, is publicly available. The home page for this project, <https://github.com/HarmonizedMRI/Functional>, contains links to this code as well as a general overview of the project. A complete, tutorial-style, example of using this code is available at <https://github.com/HarmonizedMRI/SMS-EPI/tree/main/example>.

The versions used to collect the data presented here are:

- **Pulseq sequence files:** <https://github.com/HarmonizedMRI/SMS-EPI/releases/tag/v0.2.0> (sequence generation script is ./sequence/main.m)
- **Image reconstruction:** https://github.com/HarmonizedMRI/Utils/releases/tag/Harmonized_fMRI_r0

Pulseq interpreters: Source code for the vendor-specific GE and Siemens Pulseq interpreters used in this manuscript is freely available within the respective vendor research user communities. The GE interpreter source code is available at <https://github.com/GEHC-External/pulseq-ge-interpreter> and research users can request access via their local GE representative. Siemens research users can acquire the Pulseq Interpreter under the conditions of the Customer-to-Customer Partnership (C²P, a.k.a. C2P) program, either electronically via the TeamPlay web service or by contacting the authors at pulseq.mr@uniklinik-freiburg.de.

The interpreter versions used to collect the data presented here are:

- GE interpreter: tv6 v1.9.0 and v1.9.1
- Siemens interpreter: v1.4.3

Example text for NIH grant and IRB applications: The project home page (<https://github.com/HarmonizedMRI/Functional>) contains a link to the example grant and IRB text templates mentioned in this manuscript. This text is provided at no cost and without constraints. Users therefore are free to adapt the text to suit their own needs, and to use it for any purpose.

Example volunteer brain fMRI data set: The data set used in the example at <https://github.com/HarmonizedMRI/SMS-EPI/tree/main/example> is available upon request (via a Google form, available at that site). The data set consists of raw, multi-coil, SMS-EPI k-space data. Our current IRB does not permit publicly sharing raw k-space data since such data cannot be de-faced or otherwise made fully anonymous. The user must agree to not attempt to identify any study participants.

FUNDING SOURCES

Funding for this project is provided by NIH grants U24-NS120056 and R03-EB038095.

.....

CONFLICTS OF INTEREST

The authors have no conflicts of interest to disclose in connection with the content of this manuscript.

Submitted: August 26, 2025 CST. Accepted: December 21, 2025 CST. Published: February 13, 2026 CST.



This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CCBY-4.0). View this license's legal deed at <http://creativecommons.org/licenses/by/4.0> and legal code at <http://creativecommons.org/licenses/by/4.0/legalcode> for more information.

REFERENCES

1. Keenan KE, Gimbutas Z, Dienstfrey A, Stupic KF. Assessing effects of scanner upgrades for clinical studies. *J Magn Reson Imaging*. 2019;50(6):1948-1954. doi:[10.1002/jmri.26785](https://doi.org/10.1002/jmri.26785)
2. Layton KJ, Kroboth S, Jia F, et al. Pulseq: a rapid and hardware-independent pulse sequence prototyping framework. *Magnetic resonance in medicine*. 2017;77(4):1544-1552. doi:[10.1002/mrm.26235](https://doi.org/10.1002/mrm.26235)
3. Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*. 1999;42(5):952-962. doi:[10.1002/\(SICI\)1522-2594\(199911\)42:5](https://doi.org/10.1002/(SICI)1522-2594(199911)42:5)
4. Griswold MA, Jakob PM, Heidemann RM, et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*. 2002;47(6):1202-1210. doi:[10.1002/mrm.10171](https://doi.org/10.1002/mrm.10171)
5. Maxim Zaitsev SKKL. Open file format for MR sequences Version 1.5.1. 2025. <https://pulseq.github.io/specification.pdf>
6. Ravi KS, Geethanath S, Vaughan JT. PyPulseq: a python package for MRI pulse sequence design. *Journal of Open Source Software*. 2019;4(42):1725. doi:[10.21105/joss.01725](https://doi.org/10.21105/joss.01725)
7. Stöcker T, Vahedipour K, Pflugfelder D, Shah NJ. High-performance computing MRI simulations. *Magn Reson Med*. 2010;64(1):186-193. doi:[10.1002/mrm.22406](https://doi.org/10.1002/mrm.22406)
8. Xanthis CG, Aletras AH. coreMRI: A high-performance, publicly available MR simulation platform on the cloud. *PLoS One*. 2019;14(5):e0216594. doi:[10.1371/journal.pone.0216594](https://doi.org/10.1371/journal.pone.0216594). PMID:31100074
9. MATLAB Pulseq Design Toolbox. Medical University Freiburg. Accessed August 12, 2025. <https://github.com/pulseq/pulseq/tree/master/matlab>
10. Nielsen JF, Noll DC. TOPPE: A framework for rapid prototyping of MR pulse sequences. *Magnetic resonance in medicine*. 2018;79(6):3128-3134. doi:[10.1002/mrm.26990](https://doi.org/10.1002/mrm.26990). PMID:29096052
11. Roos THM, Versteeg E, Gosselink M, et al. pTx-Pulseq in hybrid sequences: Accessible and advanced hybrid open-source MRI sequences on Philips scanners. *Magnetic Resonance in Medicine*. Published online 2025. doi:[10.1002/mrm.30601](https://doi.org/10.1002/mrm.30601). PMID:40605822
12. Shaik IA, Liu Q, Robison R, et al. Arbitrary pulse sequence execution with Pulseq on Philips MRI scanners. *Proc Int Soc Magn Reson Med*. 2024;32:3234.
13. PulCeq: A C Representation of the Pulseq Specification. Github. Accessed August 12, 2025. <https://github.com/HarmonizedMRI/PulCeq>
14. Cencini M, Wang K, Huang S, et al. Pulserver: An Open-Source Pulseq-Based Client-Server Framework For Vendor Agnostic, Interactive MR Sequence Design. *Proc Int Soc Magn Reson Med*. 2025;33:1275.
15. Liu X, Cui D, Larson PEZ, et al. Open-source implementation of X-nuclear sequences using the Pulseq framework. *Magnetic resonance in medicine*. Published online 2025. doi:[10.1002/mrm.30509](https://doi.org/10.1002/mrm.30509). PMID:40173321
16. Szczepankiewicz F. Safe_pns_prediction: Matlab Implementation of Peripheral Nerve Stimulation (PNS) Prediction at Siemens MRI Systems Based on the SAFE Model by Herbank and Gebhardt. Github. Accessed August 12, 2025. https://github.com/flip-szczepankiewicz/safe_pns_prediction
17. Hebrank FX, Gebhardt M. SAFE-model—a new method for predicting peripheral nerve stimulations in MRI. *Proc Intl Soc Mag Res Med*. Published online 2000. <https://cds.ismrm.org/ismrm-2000/PDF7/2007.PDF>
18. Szczepankiewicz F, Westin CF, Nilsson M. Gradient waveform design for tensor-valued encoding in diffusion MRI. *J Neurosci Methods*. 2021;348(109007):109007. doi:[10.1016/j.jneumeth.2020.109007](https://doi.org/10.1016/j.jneumeth.2020.109007). PMID:33242529
19. Schulte RF, Noeske R. Peripheral nerve stimulation-optimal gradient waveform design. *Magnetic resonance in medicine*. 2015;74(2):518-522. doi:[10.1002/mrm.25440](https://doi.org/10.1002/mrm.25440)
20. MATLAB implementation of the IEC nerve impulse response PNS model. Accessed December 3, 2025. <https://github.com/HarmonizedMRI/PulCeq/blob/tv7/matlab/+pge2/pns.m>

21. Casey BJ, Cannonier T, Conley MI, et al. The Adolescent Brain Cognitive Development (ABCD) study: Imaging acquisition across 21 sites. *Dev Cogn Neurosci*. 2018;32:43-54. doi:[10.1016/j.dcn.2018.03.001](https://doi.org/10.1016/j.dcn.2018.03.001). PMID:29567376
22. Stirnberg R, Stöcker T. Segmented K-space blipped-controlled aliasing in parallel imaging for high spatiotemporal resolution EPI. *Magn Reson Med*. 2021;85(3):1540-1551. doi:[10.1002/mrm.28486](https://doi.org/10.1002/mrm.28486)
23. Vendor-Agnostic SMS-EPI fMRI Sequence and Image Reconstruction. University of Michigan. 2023. <https://github.com/HarmonizedMRI/SMS-EPI>
24. Pauly J, Le Roux P, Nishimura D, Macovski A. Parameter relations for the Shinnar-Le Roux selective excitation pulse design algorithm (NMR imaging). *IEEE transactions on medical imaging*. 1991;10(1):53-65. doi:[10.1109/42.75611](https://doi.org/10.1109/42.75611)
25. Wong E. Optimized phase schedules for minimizing peak RF power in simultaneous multi-slice RF excitation pulses. In: *Proceedings of the 20th Annual Meeting of ISMRM, Melbourne, Australia*. Vol 20. ; 2012:2209.
26. Zur Y, Wood ML, Neuringer LJ. Spoiling of transverse magnetization in steady-state sequences. *Magn Reson Med*. 1991;21(2):251-263. doi:[10.1002/mrm.1910210210](https://doi.org/10.1002/mrm.1910210210)
27. Zahneisen B, Poser BA, Ernst T, Stenger VA. Three-dimensional Fourier encoding of simultaneously excited slices: generalized acquisition and reconstruction framework: 3D Fourier Encoding for SMS Acquisitions. *Magn Reson Med*. 2014;71(6):2071-2081. doi:[10.1002/mrm.24875](https://doi.org/10.1002/mrm.24875). PMID:23878075
28. Setsompop K, Gagoski BA, Polimeni JR, Witzel T, Wedeen VJ, Wald LL. Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty: Blipped-CAIPI for Simultaneous Multislice EPI. *Magn Reson Med*. 2012;67(5):1210-1224. doi:[10.1002/mrm.23097](https://doi.org/10.1002/mrm.23097). PMID:21858868
29. Chiew M. Grappa-Tools: MATLAB Toolkit for GRAPPA and Slice GRAPPA Reconstruction. Github. Accessed August 13, 2025. <https://github.com/mchiew/grappa-tools>
30. McKibben N. Pygrappa: Python Implementations of GRAPPA-like Algorithms. Github. Accessed August 13, 2025. <https://github.com/mckib2/pygrappa>
31. BART Toolbox. Accessed August 13, 2025. <https://mrrecon.github.io/bart/>
32. Uecker M, Tamir JJ, Ong F, Lustig M. The BART toolbox for computational magnetic resonance imaging. 2007;1182:1195.
33. Gadgetron. Accessed August 13, 2025. <https://gadgetron.github.io/>
34. Wikipedia contributors. Fisher transformation. Wikipedia, The Free Encyclopedia. September 4, 2025. https://en.wikipedia.org/w/index.php?title=Fisher_transformation&oldid=1309552852
35. Whitfield-Gabrieli S, Nieto-Castanon A. Conn: a functional connectivity toolbox for correlated and anticorrelated brain networks. *Brain Connect*. 2012;2(3):125-141. doi:[10.1089/brain.2012.0073](https://doi.org/10.1089/brain.2012.0073)
36. Nieto-Castanon S, Whitfield-Gabrieli A. CONN functional connectivity toolbox: RRID SCR_009550, release 22. Published online 2022. doi:[10.56441/hilbertpress.2246.5840](https://doi.org/10.56441/hilbertpress.2246.5840)
37. Penny WD, Friston KJ, Ashburner JT, Kiebel SJ, Nichols TE. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Elsevier; 2011.
38. Nieto-Castanon A. Handbook of functional connectivity magnetic resonance imaging methods in CONN. Published online 2020. doi:[10.56441/hilbertpress.2207.6598](https://doi.org/10.56441/hilbertpress.2207.6598)
39. Power JD, Cohen AL, Nelson SM, et al. Functional network organization of the human brain. *Neuron*. 2011;72(4):665-678. doi:[10.1016/j.neuron.2011.09.006](https://doi.org/10.1016/j.neuron.2011.09.006). PMID:22099467
40. Chen Q, Wehkamp N, Wan C, et al. Automated, open-source, vendor-independent quality assurance protocol based on the Pulseseq framework. *MAGMA*. 2025;38(3):533-546. doi:[10.1007/s10334-025-01247-1](https://doi.org/10.1007/s10334-025-01247-1). PMID:40274701
41. Tamir JJ, Blumenthal M, Wang J, Oved T, Shimron E, Zaiss M. MRI acquisition and reconstruction cookbook: recipes for reproducibility, served with real-world flavour. *MAGMA*. 2025;38(3):367-385. doi:[10.1007/s10334-025-01236-4](https://doi.org/10.1007/s10334-025-01236-4). PMID:40048131
42. HarmonizedMRI NIH grant text templates. Accessed August 24, 2025. https://drive.google.com/drive/folders/1SaivtmjwFJ_OsU8SBrE1ub1DmhPF6p6W?usp=sharing
43. University of Michigan Pulseseq single IRB: Manual of Operations and Procedures. Accessed August 24, 2025. <https://docs.google.com/document/d/19CjM2TeBdtnhFdxXWSZgqYMGyv21Wj5cetU2D7Mc29w/edit?usp=sharing>

44. Pothoof J, Wehkamp N, Fessler J, Zaitsev M, Noll D, Nielsen JF. Open-Source, Multi-Vendor B0 Shimming Protocol. *Proc Int Soc Magn Reson Med*. 2025;33:3479.
45. D'Astous A, Cereza G, Papp D, et al. Shimming toolbox: An open-source software toolbox for B0 and B1 shimming in MRI. *Magn Reson Med*. 2023;89(4):1401-1417. doi:[10.1002/mrm.29528](https://doi.org/10.1002/mrm.29528). PMID:36441743
46. Konstandin S, Günther M, Hoinkiss DC. gammaSTAR: A framework for the development of dynamic, real-time capable MR sequences. *Magn Reson Med*. 2025;94(4):1485-1499. doi:[10.1002/mrm.30573](https://doi.org/10.1002/mrm.30573). PMID:40391628
47. Ravi KS, Potdar S, Poojar P, et al. Pulseseq-Graphical Programming Interface: Open source visual environment for prototyping pulse sequences and integrated magnetic resonance imaging algorithm development. *Magn Reson Imaging*. 2018;52:9-15. doi:[10.1016/j.mri.2018.03.008](https://doi.org/10.1016/j.mri.2018.03.008)
48. Monreal-Madrigal A, Kurban D, Huber L, Ivanov D, Boulant N, Poser BA. Combining the benefits of 3D acquisitions and spiral readouts for VASO fMRI at UHF. *Imaging Neurosci (Camb)*. 2024;2:1-14. doi:[10.1162/imag_a_00308](https://doi.org/10.1162/imag_a_00308). PMID:40800386
49. Fessler J. Mirt: Michigan Image Reconstruction Toolbox (MIRT) - Matlab Version. Github. Accessed December 2, 2025. <https://github.com/jeffFessler/mirt>
50. Fessler J. MIRT.jl: MIRT: Michigan Image Reconstruction Toolbox (Julia Version). Github. Accessed December 2, 2025. <https://github.com/jeffFessler/MIRT.jl>
51. MATLAB Code for Split Slice GRAPPA Recon of SMS-EPI Data. Github. Accessed December 2, 2025. <https://github.com/HarmonizedMRI/utis/tree/main/%2Bhmriutis/%2Bepi/%2Bslg>