

# **rtcog: Real-Time fMRI Monitoring of Brain Activity and Cognitive Processes**

Marly Rubin <sup>1</sup>, Catherine Walsh <sup>1</sup>, Vinai Roopchansingh <sup>2</sup>, Joshua Faskowitz <sup>1</sup>, Daniel A. Handwerker <sup>1</sup>, Peter Bandettini <sup>1,2</sup>, Javier Gonzalez-Castillo <sup>1</sup>

<sup>1</sup> *Section on Functional Imaging Methods, National Institute of Mental Health, Bethesda, United States*

<sup>2</sup> *Functional MRI Core Facility, National Institute of Mental Health, Bethesda, United States*

## **Introduction**

The biological significance of spontaneous neural activity at rest remains not fully understood. One hypothesis suggests that an aspect of this activity reflects internal conscious thought. However, it is difficult to study the link between spontaneous brain activity and self-driven covert cognition using traditional fMRI paradigms due to the lack of objective, temporally-resolved measures of internal experience. Real-time functional MRI (rtfMRI) offers a means to explore this question by allowing researchers to monitor spontaneous activation of brain configurations of interest as they unfold and trigger introspective surveys at key moments of activity to interrogate the contents of ongoing cognition. However, rtfMRI experiments are difficult to implement because they often require custom-built software, leading to study-specific pipelines that are hard to modify and adapt to new experimental questions. We present *rtcog*, an open-source Python package that provides a configurable framework for building closed-loop rtfMRI experiments.

## **Methods**

*rtcog* is built on top of AFNI real-time functionality, which captures data from the scanner on an acquisition-by-acquisition basis, converts it from DICOM to NIfTI, and performs volume registration. *rtcog* receives each acquisition via a socket connection and performs further preprocessing using a configurable, high-performance pipeline. Default preprocessing steps include low-pass filtering, smoothing, and spatial normalization, with the ability to implement additional user-defined preprocessing steps.

After preprocessing, acquisitions are compared with full-brain, user-defined activation templates using spatial pattern-matching algorithms. Detected onsets of targeted neural activation patterns are termed *hits*. The software currently provides two pattern matching techniques: one based on activity *levels* within a user-provided region of interest, and another based on support vector regression models trained to estimate the expression of user-defined activity *patterns*. Users can also implement custom matching algorithms. Following detection of a pattern of interest, *rtcog* utilizes PsychoPy for automatic survey presentation and participant response logging, including speech recording for freeform responses. Users can also define custom stimuli or actions triggered by each hit, making *rtcog* adaptable to diverse research questions. This software additionally utilizes the HoloViz platform for live data visualization, allowing for real-time data quality assurance and monitoring of detected brain states and participant responses. The overall software architecture and data flow are shown in Figure 1. *rtcog*'s source code is available at <https://github.com/nimh-sfim/rtcog> and documentation at <https://rtcog.readthedocs.io>.

To evaluate performance, we measured software-side processing time for the preprocessing pipeline. We also conducted tests on both 3T GE and Siemens systems to characterize the latency between scanner acquisition and receipt by *rtcog*.

## **Results**

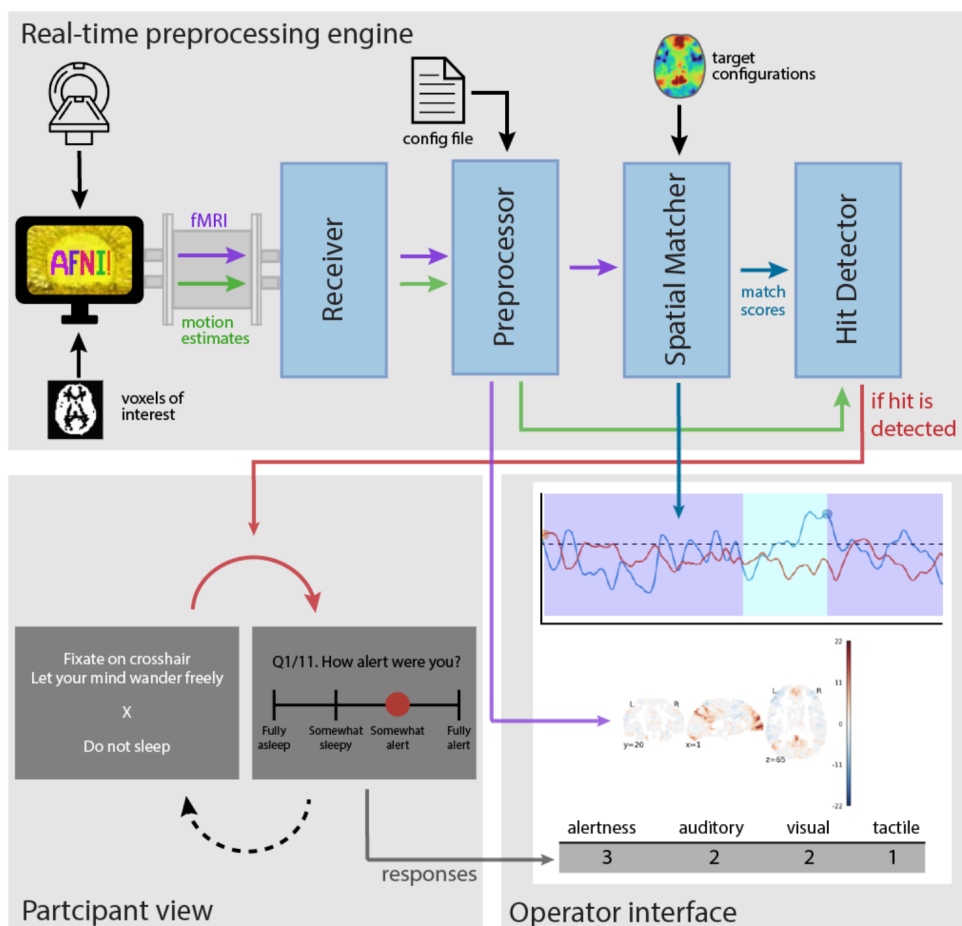
*rtcog* achieves processing speeds of ~0.12 seconds per fMRI volume on an M4 MacBook Pro once the data are received. Initial scanner-side latency tests show that delays depend primarily on slice count, indicating that image reconstruction is the dominant bottleneck. The GE system produced shorter and more stable delays of approximately 2-2.5 seconds, whereas the Siemens system showed longer and more variable delays, requiring reduced slice coverage to approach comparable latency. These delays compound the hemodynamic-driven delay of approximately 5

seconds needed for the BOLD signal to reach its peak. A total delay of 7+ seconds compromises temporal alignment between neural events and survey presentation. Ongoing development seeks to reduce this latency by optimizing acquisition protocols and finding means to capture hemodynamic responses in early phases prior to their peak.

## Conclusions

While the development of rtcog was originally motivated by the study of spontaneous thought, the software is adaptable across real-time neuroimaging paradigms, such as online neuromodulation. It provides a flexible and extensible framework for building custom closed-loop fMRI experiments by making preprocessing, pattern-matching, and experimental stimuli modular and easily configurable. Initial benchmarking suggests that rtcog's preprocessing pipeline is fast relative to scanner-side and hemodynamic delays, highlighting that optimizing acquisition parameters and capturing responses prior to their peak are important future directions for improving temporal precision.

## Figures



**Figure 1.** Overview of rtcog's multiprocessing architecture. AFNI receives scanner data and performs initial conversion and registration. rtcog applies configurable preprocessing, scores incoming acquisitions against user-defined targets, detects hits, then triggers experiment-specific actions. The operator interface streams data in real time as it arrives.