



Decentralized Analysis of Brain Imaging Data: Voxel-Based Morphometry and Dynamic Functional Network Connectivity

Harshvardhan Gazula^{1*}, Bradley T. Baker^{1,2*}, Eswar Damaraju^{1,3}, Sergey M. Plis¹, Sandeep R. Panta¹, Rogers F. Silva¹ and Vince D. Calhoun^{1,3}

¹ The Mind Research Network, Albuquerque, NM, United States, ² Department of Computer Science, The University of New Mexico, Albuquerque, NM, United States, ³ Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM, United States

OPEN ACCESS

Edited by:

Sook-Lei Liew,
University of Southern California,
United States

Reviewed by:

Gennady V. Roshchupkin,
Erasmus Medical Center, Erasmus
University Rotterdam, Netherlands
Amir Omidvarnia,
Florey Institute of Neuroscience and
Mental Health, Australia

*Correspondence:

Harshvardhan Gazula
hgazula@mrn.org
Bradley T. Baker
bbaker@mrn.org

Received: 20 March 2018

Accepted: 06 August 2018

Published: 27 August 2018

Citation:

Gazula H, Baker BT, Damaraju E,
Plis SM, Panta SR, Silva RF and
Calhoun VD (2018) Decentralized
Analysis of Brain Imaging Data:
Voxel-Based Morphometry and
Dynamic Functional Network
Connectivity.
Front. Neuroinform. 12:55.
doi: 10.3389/fninf.2018.00055

In the field of neuroimaging, there is a growing interest in developing collaborative frameworks that enable researchers to address challenging questions about the human brain by leveraging data across multiple sites all over the world. Additionally, efforts are also being directed at developing algorithms that enable collaborative analysis and feature learning from multiple sites without requiring the often large data to be centrally located. In this paper, we propose two new decentralized algorithms: (1) A decentralized regression algorithm for performing a voxel-based morphometry analysis on structural magnetic resonance imaging (MRI) data and, (2) A decentralized dynamic functional network connectivity algorithm which includes decentralized group ICA and sliding-window analysis of functional MRI data. We compare results against those obtained from their pooled (or centralized) counterparts on the same data i.e., as if they are at one site. Results produced by the decentralized algorithms are similar to the pooled-case and showcase the potential of performing multi-voxel and multivariate analyses of data located at multiple sites. Such approaches enable many more collaborative and comparative analysis in the context of large-scale neuroimaging studies.

Keywords: decentralized algorithms, COINSTAC, VBM, dFNC, multi-shot

1. INTRODUCTION

In the current times, innovation and discovery are often underpinned by the size of data at one's disposal and this has led to a paradigm shift in scientific research increasing the emphasis on collaborative data-sharing (Cragin et al., 2010; Tenopir et al., 2011). This growing significance of data-sharing is more evident in the field of neuroscience where, in the past few years, there has been a proliferation of efforts (Poldrack et al., 2013) toward enabling researchers to leverage data across multiple sites. In part, this is due to the fact that collecting neuroimaging data is expensive as well as time consuming (Landis et al., 2016) and aggregating or sharing data across various sites provides researchers with an opportunity to uncover important findings that are beyond the scope of the original study (Poldrack et al., 2013). In addition to making predictions more certain by increasing the sample size (Button et al., 2013), sharing data ensures reliability and validity of the results, and safeguards against data fabrication and falsification (Tenopir et al., 2011; Ming et al., 2017).

As mentioned previously, data-specific collaborative efforts include either aggregating the data via a centralized data sharing repository or sharing data via agreement based collaborations, or data usage agreement (DUA) in other words (Thompson et al., 2014, 2017). However, each methodology has its own set of barriers. For example, policy or proprietary restrictions or data re-identification concerns (Sweeney, 2002; Shringarpure and Bustamante, 2015) might hinder data sharing whereas DUAs might take months to complete and even if one comes through, there is no guarantee of the utility of the data until the planned analysis is performed (Baker et al., 2015; Ming et al., 2017). Other significant challenges include the storage and computational resources needed which could prove costly as the volume of the data shared goes up.

Frameworks such as ENIGMA (Thompson et al., 2014, 2017) to some extent bypass the need for DUAs by performing a centrally coordinated analysis at each local site. This enables potentially large data at each local site to stay put allowing a greater level of control as well as privacy. Another framework called ViPAR (Carter et al., 2015) tries to go one step further by, relying on open-source technologies, completely isolating the data at the local site but only pooling them via transfer to perform automated statistical analyses. This repeated pooling of data becomes cumbersome as the number of sites or the size of the data at each site goes up and ENIGMA (Thompson et al., 2014, 2017; Hibar et al., 2015; van Erp et al., 2016) addresses this issue by pooling local statistical results for further analysis, also known as, meta-analysis (Adams et al., 2016). However, the heterogeneity among the local analyses caused by adopting various data collection mechanisms or preprocessing methods can lead to inaccurate meta-analysis findings.

Plis et al. (2016), proposed a web-based framework titled Collaborative Informatics and Neuroimaging Suite Toolkit for Anonymous Computation (COINSTAC) to address the aforementioned issues. COINSTAC provides a platform to analyze data stored locally across multiple organizations without the need for pooling the data at any point during the analysis. It is intended to be an ultimate one-stop shop by which researchers can build any statistical or machine learning model collaboratively in a decentralized fashion. This framework implements a message passing infrastructure that will allow large scale analysis of decentralized data with results on par with those that would have been obtained if the data were in one place. Since, there is no pooling of data it also preserves the privacy of individual datasets.

Some of the decentralized computations discussed in the literature so far include decentralized regression (Plis et al., 2016), joint independent component analysis (Baker et al., 2015), decentralized independent vector analysis (Wojtalewicz et al., 2017), decentralized neural networks (Lewis et al., 2017), decentralized stochastic neighbor embedding (Saha et al., 2017) and many more. To our knowledge, most of these algorithms have been tested on synthetic data. In this work we present two new decentralized algorithms that are widely used in a centralized manner in the imaging community and demonstrate their utility on real world brain imaging data.

Regression, is widely used in neuroimaging studies as it enables one to regress certain covariates, for example- age, diagnosis, gender or treatment response, to study their effects on the structure and function of various brain regions. Some examples of regression related studies in this field include (Fennema-Notestine et al., 2007) where regression was used as a validity test in examining the aggregation of structural imaging across different datasets. In addition, the very successful ENIGMA studies are mostly using regression analyses for a small number of variables. Roshchupkin et al. (2016) presented a framework titled HASE (high-dimensional association analyses) that is capable of analyzing high-dimensional data at full resolution, yielding exact association statistics. While singleshot and multishot regression have been presented previously (Plis et al., 2016), their treatment was cursory in nature without any actual consideration of the appropriate gradient descent scheme or the validity of the methods on real datasets both of which have been presented in this work.

In this paper, in addition to improving the single-shot and multi-shot regression we also present a new variant of decentralized regression- “decentralized regression with normal equation” and extend this work to operate on voxels in an MRI image, in order to implement a voxel-based morphometry (VBM) study in a decentralized framework (Ashburner and Friston, 2000). We implement and evaluate the proposed decentralized VBM approach on the publicly available MIND Clinical Imaging Consortium (MCIC) dataset (available via the COINS data exchange at <https://coins.mrn.org> and contrast the results obtained with those from pooled/centralized regression to validate the proof-of-concept.

Another widely utilized method in neuroimaging analysis is dynamic functional network connectivity (dFNC) (Sakoglu et al., 2010; Allen et al., 2014). dFNC is an analysis pipeline for functional magnetic resonance imaging (fMRI) data, which allows for the identification and analysis of networks of co-activating brain states. In contrast to static approaches (Smith et al., 2009), which take the mean connectivity over time-points, dFNC uses clustering of time varying connectivity estimates computed from sliding-windows taken over subject time-courses, thus becoming desirable in experiments where network connectivity is highly dynamic in the time dimension, for example in experiments which utilize resting-state fMRI (Deco et al., 2013; Damaraju et al., 2014).

Importantly, dFNC is focused on time-courses of networks extracted from a group independent component analysis (ICA), which is a widely used approach for estimating functional brain networks (Calhoun and Adali, 2012) and as such to implement dFNC we needed to also implement a decentralized group ICA approach.

For collaborative neuroimaging applications, a decentralized version of dFNC is desirable for many of the same reasons as regression, and currently, no such decentralized version exists. Unlike regression, however, the dFNC pipeline consists of multiple, distinct stages, all of which require decentralization. In this paper, we present an initial version of decentralized dFNC by providing decentralized approaches to both the group spatial independent component analysis (ICA) and K-Means

clustering steps in the pipeline, which, along with additional preprocessing steps including sliding window correlation, can be implemented together to perform decentralized dFNC. Our resulting methods, dglCA, and ddFNC via dK-Means, provide dynamic connectivity results consistent with established pooled approaches in the literature, thus representing an important step toward more exhaustive analysis of the decentralized approaches to the dFNC pipeline. Our contributions in this paper can thus be summarized as follows.

1. Development of decentralized regression with normal equation, improvement of single-shot and multi-shot regression and their validation on structural MRI data
2. Presentation of a decentralized dynamic functional network connectivity analysis pipeline and its evaluation on functional MRI data

2. METHODS

2.1. Decentralized VBM (i.e., Voxelwise Decentralized Regression)

Statistical analysis plays a key role in the field of neuroimaging studies. Researchers would often want to characterize the effect of various factors such as age, gender, disease condition, etc., on the composition of brain tissue at various regions of the brain. Voxel-based morphometry (VBM) (Ashburner and Friston, 2000) is one such approach that facilitates a comprehensive comparison, via generalized linear modeling, of voxel-wise gray matter concentration between different groups, for example. To enable such statistical assessment on data present at various sites, it is important to develop decentralized tools. In this section, we first provide a brief overview of decentralized regression algorithms (the building blocks of decentralized VBM which is essentially voxel-wise regression) along with some notation.

The goal of decentralized regression is to fit a linear equation (given by Equation 1) relating the covariates at S different sites to the corresponding responses. Assume each site j has data set $\mathcal{D}_j = \{(\mathbf{x}_i, y_i) : i \in \{1, 2, \dots, s_j\}\}$ where $\mathbf{x}_{i,j} \in \mathbb{R}^d$ is a d -dimensional vector of real-values features, and $y_j \in \mathbb{R}$ is a response. We consider fitting the model in Equation 2 where $\mathbf{w}$ is given as $[\mathbf{w}; b]$ and $\mathbf{x}$ as $[\mathbf{x}; 1]$

$$y \approx \mathbf{w}^\top \mathbf{x} + b \quad (1)$$

$$\mathbf{y} \approx \mathbf{w}^\top \mathbf{x} \quad (2)$$

The vector of regression parameters/weights $\mathbf{w}$ is found by minimizing the sum of the squared error given in Equation (3)

$$F(\mathbf{w}) = \sum_{j=1}^S \sum_{i=1}^{s_j} (y_i - \mathbf{w}^\top \mathbf{x}_{i,j})^2 \quad (3)$$

The regression objective function is a linearly separable function, that can be written as sum of a local objective function calculated at each local site as follows:

$$F(\mathbf{w}) = \sum_{j=1}^S F_j(\mathbf{w}) \quad (4)$$

where

$$F_j(\mathbf{w}) = \sum_{i=1}^{s_j} (y_i - \mathbf{w}^\top \mathbf{x}_{i,j}) \quad (5)$$

A central aggregator (AGG) is assumed whose role is to compute the global minimizer $\hat{\mathbf{w}}$ of $F(\mathbf{w})$.

2.1.1. Single-Shot Regression

In one approach to solve the decentralized regression problem, termed the *single-shot* regression (Plis et al., 2016), each site j finds the minimizer $\hat{\mathbf{w}}_j$ of the local objective function $F_j(\mathbf{w})$. This is the same as solving the regression problem at each local site. Once the regression model at each site is fit, the weights are sent to the central aggregator (AGG) where they are aggregated (weighted average) to find the global minimizer or can be used separately to perform a meta-analysis similar to those performed in ENIGMA (using a manual spreadsheet-based approach however) (Turner et al., 2013; van Erp et al., 2016). The pseudocode to perform single-shot decentralized regression (Plis et al., 2016), with a slight modification, is presented here again for completeness.

Algorithm 1 Single-shot Regression

Require: Data $\mathcal{D}_j$ at site j for sites $j = 1, 2, \dots, S$, where $|\mathcal{D}_j| = s_j \forall j$

- 1: **for** $j = 1$ to S **do**
- 2: $\hat{\mathbf{w}}_j = \text{argmin}_{\mathbf{w}} F_j(\mathbf{w})$.
- 3: Node j sends $\hat{\mathbf{w}}_j$ to AGG.
- 4: **end for**
- 5: AGG computes $\hat{\mathbf{w}} = \frac{1}{\sum_{j=1}^S s_j} \sum_{j=1}^S s_j \hat{\mathbf{w}}_j$ and return $\hat{\mathbf{w}}$

2.1.2. Decentralized Regression With Normal Equation

One limitation of single-shot regression is that the “site” level covariates cannot be included at each local site as this leads to collinearity issues. This issue can be offset by utilizing a decentralized version of the analytical solution to the linear regression problem. For a standard regression problem of the form given in Equation (2), the analytical solution is given as

$$\hat{\mathbf{w}} = (\mathbf{x}^\top \mathbf{x})^{-1} \mathbf{x}^\top \mathbf{y} \quad (6)$$

Assuming that the augmented data matrix $\mathbf{x}$ is made up of data from different local sites, i.e.,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_S \end{bmatrix} \quad (7)$$

it's easy to see that $\hat{\mathbf{w}}$ can be written as

$$\hat{\mathbf{w}} = \left(\begin{bmatrix} \mathbf{x}_1^\top & \cdots & \mathbf{x}_S^\top \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_S \end{bmatrix} \right)^{-1} \times \begin{bmatrix} \mathbf{x}_1^\top & \cdots & \mathbf{x}_S^\top \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_S \end{bmatrix} \quad (8)$$

$$\hat{\mathbf{w}} = \left(\sum_{j=1}^S \mathbf{x}_j^\top \mathbf{x}_j \right)^{-1} \times \left(\sum_{j=1}^S \mathbf{x}_j^\top \mathbf{y}_j \right) \quad (9)$$

The above variant of the analytical solution to a regression model shows that even if the data resides in different locations, fitting a global model in the presence of site covariates delivers results that are exactly similar to the pooled case.

Algorithm 2 Decentralized Regression with Normal Equation

Require: Data $\mathcal{D}_j$ at site j for sites $j = 1, 2, \dots, S$, where $|\mathcal{D}_j| = s_j \forall j$

- 1: **for** $j = 1$ to S **do**
- 2: Compute $\text{Cov}(X_j) = \mathbf{x}_j^\top \mathbf{x}_j$
- 3: Compute $\mathbf{x}_j^\top \mathbf{y}_j$
- 4: Node j sends $\text{Cov}(X_j)$ and $\mathbf{x}_j^\top \mathbf{y}_j$ to AGG.
- 5: **end for**
- 6: AGG computes

$$\hat{\mathbf{w}} \leftarrow \left(\sum_{j=1}^S \text{Cov}(X_j) \right)^{-1} \left(\sum_{j=1}^S \mathbf{x}_j^\top \mathbf{y}_j \right) \text{ and return } \hat{\mathbf{w}}$$

2.1.3. Multi-Shot Regression

Decentralized regression with a normal equation is a nice mathematical formulation which produces results that are exactly the same as those from the pooled regression. However, one of the biggest drawback of the analytical form of regression is it becomes computationally expensive to evaluate the inverse of $\mathbf{x}^\top \mathbf{x}$ as the number of features in the dataset ($\mathcal{D}$) increases. While in a neuroimaging setting there might not be as many covariates to make it computationally expensive, it is indeed a challenge while working with datasets where the cardinality of the feature set is usually large (especially in machine learning). One can overcome this drawback by implementing an optimization method in a way that entails the local sites and AGG having to communicate iteratively. This is a type of distributed gradient descent and such a regression is termed “*multi-shot*” regression (Plis et al., 2016).

For a regression model of the form given in Equation 5, the gradient update equation (given a learning rate η) is given as

$$\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t - \eta \cdot \nabla F_j(\hat{\mathbf{w}}) \quad (10)$$

Algorithm 3 Multi-shot Regression

Require: Data $\mathcal{D}_j$ at site j for sites $j = 1, 2, \dots, S$, where $|\mathcal{D}_j| = s_j \forall j$

Require: Step size η (Suggested default: 0.001)

Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates (Suggested defaults: 0.9 and 0.999, respectively)

Require: Small constant δ used for numerical stabilization (Suggested default: 10^{-8})

Require: $\hat{\mathbf{w}}_{t-1} \leftarrow \mathbf{0}$ (Initial parameter vector), $m_0 \leftarrow \mathbf{0}$ (Initialize 1st moment vector), $v_0 \leftarrow \mathbf{0}$ (Initialize 2nd moment vector), $t \leftarrow 0$ (Initialize timestep), tolerance **Tol** $\triangleright$ at AGG

- 1: **while** **True** **do**
 - 2: **for** $j = 1$ to S **do**
 - 3: AGG sends $\hat{\mathbf{w}}_{t-1}$ to node j
 - 4: Node j computes $\nabla F_j(\hat{\mathbf{w}}_{t-1})$
 - 5: Node j sends $\nabla F_j(\hat{\mathbf{w}}_{t-1})$ to AGG.
 - 6: **end for**
 - 7: AGG computes $\nabla F_c \leftarrow \sum_{j=1}^S \nabla F_j(\hat{\mathbf{w}}_{t-1})$ $\triangleright$ aggregate gradient
 - 8: $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla F_c$ $\triangleright$ update biased first moment estimate
 - 9: $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \nabla F_c^2$ $\triangleright$ update biased second moment estimate
 - 10: $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ $\triangleright$ Compute bias-corrected first moment estimate
 - 11: $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ $\triangleright$ Compute bias-corrected second moment estimate
 - 12: AGG computes $\hat{\mathbf{w}}_t \leftarrow \hat{\mathbf{w}}_{t-1} - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \delta)$ $\triangleright$ Update parameters
 - 13: **if** $\|\hat{\mathbf{w}}_t - \hat{\mathbf{w}}_{t-1}\|_2 \leq \text{Tol}$ **then**
 - 14: **break**
 - 15: **end if**
 - 16: $\hat{\mathbf{w}}_{t-1} \leftarrow \hat{\mathbf{w}}_t$
 - 17: **end while**
 - 18: **return** $\hat{\mathbf{w}}_t$ as $\hat{\mathbf{w}}$ $\triangleright$ Resulting parameters
-

where

$$\nabla F_j(\hat{\mathbf{w}}) = \sum_{i=1}^{s_j} (y_i - \hat{\mathbf{w}}^\top \mathbf{x}_{i,j}) \mathbf{x}_{i,j} \quad (11)$$

In multi-shot regression, at every time step the AGG sends the value of $\hat{\mathbf{w}}_{t-1}$ to each of the local sites which then compute their local gradients $\nabla F_j(\hat{\mathbf{w}}_t)$ and send them back to the AGG where it sums up all the local gradients in order to update the parameter vector $\hat{\mathbf{w}}_t$. The need to sum up all the local gradients is explained as follows:

$$\begin{aligned} \text{From Equation (4), } F(\hat{\mathbf{w}}) &= \sum_{j=1}^S F_j(\hat{\mathbf{w}}) \\ \therefore \nabla F(\hat{\mathbf{w}}) &= \sum_{j=1}^S \nabla F_j(\hat{\mathbf{w}}) \end{aligned} \quad (12)$$

To illustrate this using an example, suppose there are 3 sites ($S = 3$) with s_1, s_2 and s_3 number of samples, respectively, at each site. The global objective function $F(\hat{\mathbf{w}})$ can be easily written

as the sum of objective functions from each site (this because the objective function is linear) as follows:

$$\begin{aligned}
 F(\hat{\mathbf{w}}) &= \sum_{j=1}^{s_1+s_2+s_3} (y_j - \hat{\mathbf{w}}^\top x_j)^2 \\
 &= \sum_{j=1}^{s_1} (y_j - \hat{\mathbf{w}}^\top x_j)^2 + \sum_{j=1}^{s_2} (y_j - \hat{\mathbf{w}}^\top x_j)^2 \\
 &\quad + \sum_{j=1}^{s_3} (y_j - \hat{\mathbf{w}}^\top x_j)^2 \\
 &= \sum_{j=1}^{s_1} F_1(\hat{\mathbf{w}}) + \sum_{j=1}^{s_2} F_2(\hat{\mathbf{w}}) + \sum_{j=1}^{s_3} F_3(\hat{\mathbf{w}}) \\
 \therefore \nabla F(\hat{\mathbf{w}}) &= \sum_{j=1}^{s_1} \nabla F_1(\hat{\mathbf{w}}) + \sum_{j=1}^{s_2} \nabla F_2(\hat{\mathbf{w}}) + \sum_{j=1}^{s_3} \nabla F_3(\hat{\mathbf{w}}) \quad (13)
 \end{aligned}$$

From Equation (13), it should be easy to see that the aggregated gradient is just a sum of the gradients from each site. On the other hand, if the mean sum of squared errors is preferred i.e., $F(\hat{\mathbf{w}}) = \frac{1}{m} \sum_{j=1}^m (y_j - \hat{\mathbf{w}}^\top x_j)^2$, which mathematically has the same minimizer as $\sum_{j=1}^m (y_j - \hat{\mathbf{w}}^\top x_j)^2$ since $F(\hat{\mathbf{w}})$ is convex, it can be shown that the aggregated gradient is a weighted average of the gradients from the local sites:

$$\begin{aligned}
 F(\hat{\mathbf{w}}) &= \frac{1}{s_1 + s_2 + s_3} \sum_{j=1}^{s_1+s_2+s_3} F_j(\hat{\mathbf{w}}) \\
 &= \frac{1}{s_1 + s_2 + s_3} \left(\frac{s_1}{s_1} \sum_{j=1}^{s_1} F_j(\hat{\mathbf{w}}) + \frac{s_2}{s_2} \sum_{j=1}^{s_2} F_j(\hat{\mathbf{w}}) \right. \\
 &\quad \left. + \frac{s_3}{s_3} \sum_{j=1}^{s_3} F_j(\hat{\mathbf{w}}) \right) \\
 &= \frac{1}{s_1 + s_2 + s_3} (s_1 F_1(\hat{\mathbf{w}}) + s_2 F_2(\hat{\mathbf{w}}) + s_3 F_3(\hat{\mathbf{w}})) \\
 \therefore \nabla F(\hat{\mathbf{w}}) &= \frac{1}{s_1 + s_2 + s_3} (s_1 \nabla F_1(\hat{\mathbf{w}}) + s_2 \nabla F_2(\hat{\mathbf{w}}) + s_3 \nabla F_3(\hat{\mathbf{w}})) \quad (14)
 \end{aligned}$$

Algorithm 3 shows the steps involved in multi-shot regression. In order to update the parameters (*here*, $\hat{\mathbf{w}}$), any off-the-shelf optimization scheme, for example, gradient descent, adagrad (Duchi et al., 2011), adadelata (Zeiler, 2012), momentum gradient descent (Rumelhart et al., 1986), nesterov accelerated gradient descent (Nesterov et al., 1983), Adam (Kingma and Ba, 2014) could have been used. The choice of scheme adopted could depend on the data being analyzed. Moreover, additional considerations have to be given to the stopping criterion tolerance, the number of iterations, the choice of learning rate and any other additional hyper-parameters depending on the scheme utilized. In some cases, the choice of optimization scheme can result in an analysis which could take minutes, days or years to arrive. In our tests, we found out that the Adam optimization

scheme performs extremely well on the real dataset and hence has been adopted to perform the multi-shot regression.

2.1.4. Other Statistics

In addition to generating the weights of the covariates (regression parameters), one would also be interested in determining the overall model performance given by goodness-of-fit or the coefficient of determination (R^2) as well as the statistical significance of each weight parameter (t -value or p -value).

As demonstrated in Algorithm 4 (Ming et al., 2017), determining R^2 entails calculating the sum-square-of-errors (SSE) as well as total sum of squares (SST) which are evaluated at each local site and then aggregated at the global site to evaluate R^2 given by $1 - \text{SSE}/\text{SST}$. An intermediary step before the calculation of SST is the calculation of the global $\bar{\mathbf{y}}$ which is determined by taking a weighted average of the local $\bar{\mathbf{y}}_j$ weighted on the size of data at each local site.

Algorithm 4 Decentralized R^2 calculation

Require: Data $\mathcal{D}_j$ at site j for sites $j = 1, 2, \dots, S$, where $|\mathcal{D}_j| = s_j \forall j$

- 1: AGG sends $\hat{\mathbf{w}}$ to each local site.
- 2: **for** $j = 1$ to S **do**
- 3: Node j computes $\bar{\mathbf{y}}_j = \frac{1}{s_j} \sum_{i=1}^{s_j} \mathbf{y}_i$
- 4: Node j sends $\bar{\mathbf{y}}_j$ and s_j to AGG.
- 5: **end for**
- 6: AGG computes $\bar{\mathbf{y}} = \frac{\sum_{j=1}^S (s_j \bar{\mathbf{y}}_j)}{\sum_{j=1}^S s_j}$ ▷ global mean
- 7: AGG sends $\bar{\mathbf{y}}$ to local sites
- 8: **for** $j = 1$ to S **do**
- 9: $\text{SST}_j = \sum_{i=1}^{s_j} (y_i - \bar{\mathbf{y}})^2$
- 10: $\hat{\mathbf{y}}_j = \hat{\mathbf{w}} \cdot \mathbf{x}_j$
- 11: $\text{SSE}_j = \sum_{i=1}^{s_j} (y_i - \hat{\mathbf{y}}_j)^2$
- 12: Node j sends SST_j and SSE_j to AGG
- 13: **end for**
- 14: AGG computes $\text{SST} \leftarrow \sum_{j=1}^S \text{SST}_j$,
 $\text{SSE} \leftarrow \sum_{j=1}^S \text{SSE}_j$,
 $R^2 \leftarrow 1 - \frac{\text{SSE}}{\text{SST}}$
- 15: **return** R^2

Algorithm 5 (Ming et al., 2017) details the steps involved in calculating the t -values (and therefore p -values) of each regression parameter. Assuming the weight vector has been calculated using either the single-shot or multi-shot regression, the global weight vector ($\hat{\mathbf{w}}$) is sent to each of the local sites where the local covariance matrix as well as the sum-square-of-errors is calculated and sent back along with the data size to the aggregator (AGG) which then utilizes that information to calculate the t -values for each parameter (or coefficient). Once, the t -values have been calculated, the corresponding two-tailed p -values can be deduced using any publicly available *distributions* library.

Algorithm 5 Decentralized t -value calculation

Require: Data $\mathcal{D}_j$ at site j for sites $j = 1, 2, \dots, S$, where $|\mathcal{D}_j| = s_j \forall j$

- 1: AGG sends $\hat{\mathbf{w}}$ to each local site.
- 2: **for** $j = 1$ to S **do**
- 3: $\hat{\mathbf{y}}_j = \hat{\mathbf{w}} \cdot \mathbf{x}_j$
- 4: $SSE_j = \sum_{i=1}^{s_j} (y_i - \hat{y}_j)^2$
- 5: $Cov(\mathbf{x}_j) = \mathbf{x}_j^T \mathbf{x}_j$
- 6: Node j sends SSE_j , $Cov(\mathbf{x}_j)$ and s_j to AGG.
- 7: **end for**
- 8: AGG computes $Cov(\mathbf{x}) \leftarrow \sum_{j=1}^S Cov(\mathbf{x}_j)$,
 $MSE \leftarrow \frac{1}{\sum_{j=1}^S s_j} \sum_{j=1}^S SSE_j$,
 $SE(W) \leftarrow \sqrt{diag(MSE \cdot Cov(\mathbf{x})^{-1})}$,
 $t \leftarrow \hat{\mathbf{w}}/SE(W)$
- 9: **return** t

2.1.5. Bandwidth and Complexity

For singleshoot regression, each site communicates a local weight vector $\hat{\mathbf{w}}_j$ of size $(d+1)$ to the aggregator in addition to the cardinality of the dataset at each site $|\mathcal{D}_j| = s_j$, a scalar. Once all the information is aggregated, a weighted average of the local $\hat{\mathbf{w}}_j$ s with the weights being s_j performed to get the global weight vector $\hat{\mathbf{w}}$. Assuming $s_j > d$ and that the normal equation is used to get the local weight vectors $\hat{\mathbf{w}}_j$ s, the computational complexity is $\mathcal{O}(d^2 s_j)$ whereas the computational complexity of calculating the weighted average at the AGG is $\mathcal{O}(d)$.

In the case of decentralized regression with normal equation, the first step (at each site) includes the calculation of $\mathbf{x}^T \mathbf{x}$ (at $\mathcal{O}(d^2 s_j)$) and $\mathbf{x}^T \mathbf{y}$ (at $\mathcal{O}(d s_j)$) with an overall complexity of $\mathcal{O}(d^2 s_j)$. A total information of $\sum_{j=1}^S \{s_j \times [(d+1)^2 + (d+1)]\}$ is communicated to the AGG where they are aggregated (as shown in Algorithm 2) to obtain the global weight vector $\hat{\mathbf{w}}$ at $\mathcal{O}(d^3)$.

Contrary to where the computation starts in the case of singleshoot or DRNE, the computation/communication starts from the AGG in multishot regression. The AGG initializes the $\hat{\mathbf{w}}$ and communicates the $(d+1)$ -sized vector to each of the S sites. At every iteration, each site j then calculates the gradient vector ($\mathcal{O}(d)$) and sends it back to the AGG which again means the communication $S \times (d+1)$ accounting for S sites. At the AGG, steps 7 through 12 (refer to Algorithm 3) are performed at an order of $\mathcal{O}(d)$ which are again sent back to each of the local sites, implying a communication of $S \times d$, for the next iteration of the gradient descent.

The above treatment of communication bandwidth and complexity is subject to certain considerations *viz.*, the number of covariates, the number of samples at each site, the optimization scheme used in the calculation of $\mathbf{x}^T \mathbf{x}$, the stopping criterion, etc.

2.2. Decentralized dFNC

In this section, we briefly present our initial work toward performing dynamic functional network connectivity (dFNC) analysis in a decentralized framework. As mentioned earlier, dFNC is a multi-step pipeline finds common states in subject fMRI time-courses (TCs), and is often done by clustering a

sliding window over subject time-courses, as is done (e.g., Allen et al., 2014; Damaraju et al., 2014). Thus, we present methods for decentralized spatial ICA along with decentralized K-Means clustering. Our presentation here is by no means a rigorous take on dFNC which we save for future work.

2.2.1. Decentralized Group Spatial ICA

Following preprocessing, the first step in the dFNC pipeline includes group ICA (Calhoun et al., 2001). Since we are dealing with fMRI data, suppose that we now have data $\mathbf{X} \in \mathbb{R}^{d \times N}$, where d is the voxel-space of the data (in brain voxels), and N is the total number of time-points across all subjects in the network. In linear spatial ICA, we model each individual subject as a mixture of r many statistically independent spatial maps, $\mathbf{A} \in \mathbb{R}^{d \times r}$, and their time-courses, $\mathbf{S} \in \mathbb{R}^{r \times N_i}$, where N_i is the length of the time-course belonging to subject i . In the decentralized case, we can model the global data set $\mathbf{X}$ as the column-wise concatenation of s sites in the temporal dimension, where each site is modeled as a set of subjects concatenated in the temporal dimension:

$$\mathbf{X} = [\mathbf{A}_1 \mathbf{S}_1 \mathbf{A}_2 \mathbf{S}_2 \dots \mathbf{A}_s \mathbf{S}_s] \in \mathbb{R}^{d \times N}.$$

Our goal is to learn a global unmixing matrix, $\mathbf{W}$, such that $\mathbf{X}\mathbf{W} \approx \hat{\mathbf{A}}$, where $\hat{\mathbf{A}} \in \mathbb{R}^{d \times r}$ is a set of unmixed spatially independent components. To this end, we perform a decentralized group independent component analysis (dglICA). Our method consists first of the two-stage GlobalPCA procedure utilized in Baker et al. (2015). In this procedure, each site first performs subject-specific LocalPCA dimension-reduction and whitening to a common k principal components in the temporal dimension. A decentralized, second stage, then produces a global set of r spatial eigenvectors, $\mathbf{V} \in \mathbb{R}^{r \times d}$. As outlined in Baker et al. (2015), this second stage has sites pass locally-reduced eigenvectors to other sites in a peer-to-peer scheme, where upon receiving a set of eigenvectors, a site then stacks them in the column dimension, and performs a further reduction of the stacked matrix, which is then passed to the next peer in the network. This process iterates until the global eigenvectors reach some aggregator (AGG), or otherwise terminal site in the network.

Algorithm 6 Decentralized group ICA algorithm (dglICA)

Require: s sites with data $\{\mathbf{X}_i \in \mathbb{R}^{d \times N_i} : i = 1, 2, \dots, s\}$, intended final rank r , local site rank $k_2 \geq r$, local subject rank k_1 .

- 1: **for all** sites $i = 1, 2, \dots, s$ **do**
- 2: Perform LocalPCA (Baker et al., 2015) on each site $\rightarrow k_1$ eigen-vectors for each subject.
- 3: Perform LocalPCA (Baker et al., 2015) on concatenated subjects $\rightarrow k_2$ eigenvectors at each site.
- 4: Reduce local data set to $\mathbf{X}_{i,red} \in \mathbb{R}^{d \times k_2}$
- 5: **end for**
- 6: Perform GlobalPCA (Baker et al., 2015) to obtain r global eigenvectors, $\mathbf{V}$, at the aggregator.
- 7: On the aggregator, perform ICA to obtain global unmixing matrix, $\mathbf{W}$.

The aggregator site then performs whitening on these resulting eigenvectors, and runs a local ICA algorithm, such as infomax ICA (Bell and Sejnowski, 1995), to produce the spatial unmixing matrix, $\mathbf{W}$. The global spatial eigenvectors, $\mathbf{V}$, are then unmixed to produce $\hat{\mathbf{A}}$ by computing $\hat{\mathbf{A}} \approx \mathbf{V}\mathbf{W}$, which is shared across the decentralized network. Each site then uses this unmixing matrix to produce individual time-courses for each i -th subject by computing $\mathbf{A}_i \approx \mathbf{X}_i^T \mathbf{S}$. Each site can then perform spatio-temporal regression back reconstruction approach (Calhoun et al., 2001; Erhardt et al., 2011) to produce subject-specific spatial maps.

2.2.2. Decentralized Clustering

In order to perform dFNC in a decentralized paradigm, we first require a notion of decentralized clustering. Following the precedent of previous work in dFNC, we focus first on decentralized K-Means optimization, for which there exist a number of pre-established methods for decentralization. A number of methods utilize some manner of weighted centroid averaging, where each site in the network broadcasts updated centroids to an aggregator node which then computes the merged centroids, and rebroadcasts them to the local sites (Forman and Zhang, 2000; Dhillon and Modha, 2000; Jagannathan and Wright, 2005), though completely peer-to-peer approaches have also been proposed (Datta et al., 2006, 2009), as well as methods robust to asynchronous updates (Di Fatta et al., 2013). Though we have not found any methods which do this, methods which compute K-Means via gradient descent (Bottou, 2010) are also amenable to decentralization (Yuan et al., 2016). For simplicity's sake, we take the approach of centroid-averaging outlined in Dhillon and Modha (2000), and leave rigorous presentation and comparison of the remaining methods as future work.

To perform clustering for distributed dFNC, we first have each site separate its subjects into sliding-window time-courses, where the window length is fixed across the decentralized network. Additionally, initial clustering was performed on a subset of windows from each subject, corresponding to windows of maximal variability in correlation across component pairs. To obtain these exemplars, each site computes variance of dynamic

connectivity across all pairs of components at each window. We then select windows corresponding to local maxima in this variance time-course. This resulted in an average of 8 exemplar windows per subject. We then perform decentralized K-Means on the exemplars to obtain a set of centroids, which are shared across the decentralized network, which we feed into a second stage of K-Means clustering.

For the second stage of decentralized clustering, at each iteration, each site computes updated centroids according to Dhillon and Modha (2000), which corresponds to a local K-Means update. These local centroids are then sent to the aggregator node, which computes the weighted average of these updated centroids, and re-broadcasts the updated global centroids until convergence.

2.2.3. Bandwidth and Complexity

To compute the communication and complexity for ddFNC, we separately analyse the novel component algorithms of dglICA and dK-Means.

For decentralized group ICA, the communication of the algorithm is closely related to the communication of GlobalPCA. In the GlobalPCA algorithm given in Baker et al. (2015), each site communicates a $d \times r$ matrix of eigenvectors to the subsequent site until the aggregator is reached. After the aggregator performs ICA to obtain the global unmixing matrix, $\mathbf{W}$, this matrix is broadcast to all other sites in the network. Thus, for a single, non-aggregator site, the total communication for dglICA is exactly $d \times r + r^2$. At the aggregator, the total communication is exactly $d \times r + r^2 \times s$ if the unmixing matrix is broadcast directly to each node. Of course, this cost could be mitigated by following a peer to peer communication scheme, and having other non-aggregator sites broadcast the unmixing matrix as well.

Next, we can compute the overall complexity of dglICA as the total complexity of local site operations. Consider an individual site, i , with m subjects, where the concatenated matrix is given as $\mathbf{X}_i \in \mathbb{R}^{d \times N_i}$. In general, the complexity of SVD on the $N_i \times N_i$ covariance matrix is $\mathcal{O}(N_i^3)$, though this can be improved upon by using iterative methods, such as the MATLAB *svds* function. Thus, the complexity for the two-stage LocalPCA computation on one site is $\mathcal{O}(2N_i^3)$. The per-site complexity for GlobalPCA is given as the complexity of a SVD computed on a $d \times d$ covariance matrix, which is created by concatenating the k_2 eigenvectors from the previous site; i.e., the per-site complexity for GlobalPCA is $\mathcal{O}(d^3)$. Finally, the complexity of ICA is exactly equal to the number of ICA iterations, $\mathcal{I}$, which depends heavily on the choice of ICA algorithm, and hyper-parameter selection (see Bell and Sejnowski, 1995 for more details on the complexity of Infomax, for example). Thus, the total per-site complexity for dglICA is $\mathcal{O}(N_i^3 + d_i^3)$ for non-aggregator nodes, and $\mathcal{O}(N_i^3 + d_i^3 + \mathcal{I})$ on the aggregator node. The overall runtime of dglICA is thus dependent on the computational resources available at each site, as well as the computational resources and ICA parameters chosen by the aggregator site.

Prior to performing K-Means, each site i computes $N_{ij} - w$ windowed time-courses of length w on each subject j , computing the rank r covariance matrix for those windows. Thus, if there are

Algorithm 7 Decentralized dFNC algorithm (ddFNC)

Require: s sites with data $\{\mathbf{X}_i \in \mathbb{R}^{d \times N_i} : i = 1, 2, \dots, s\}$, win-size t , number of clusters k .

- 1: dglICA $\rightarrow \mathbf{W}$, global unmixing matrix, broadcast to sites.
 - 2: **for all** sites $= i = 1, 2, \dots, s$ **do**
 - 3: Back-reconstruct subject TCs
 - 4: Using sliding window of size t , obtain $r \times r$ covariance matrices.
 - 5: Obtain exemplar covariance matrices (Damaraju et al., 2014).
 - 6: **end for**
 - 7: Run K-Means on exemplar covariance matrices to obtain k initial centroids, C_0 .
 - 8: Run K-Means with initial clusters C_0 to obtain k centroids C , and clustering assignment for each instance, L .
-

m_i subjects at site i , the local complexity is $\mathcal{O}(m_i(N - w)r^3)$ for this operation. No inter-site communication occurs during this process.

For decentralized K-Means, the communication between sites depends on the number of “K-Means Iterations,” $\mathcal{J}$, i.e., the number of iterations required for the centroids to stabilize. $\mathcal{J}$ depends heavily on the initial centroids, the distance metric used, the distribution of the global data set, and other factors which make it difficult to compute exactly for arbitrary data. In each iteration of decentralized K-Means, we communicate k many centroids of size $\mathbb{R}^2$, for an average communication of $r^2 \cdot k \cdot \mathcal{J}$ from the sites to the aggregator. The aggregator, then, performs a total of $r^2 \cdot k \cdot \mathcal{J} \cdot s$ communication (Dhillon and Modha, 2000), which again, could be mitigated by passing centroids to intermediate sites, provided those sites can be trusted with the centroid information.

The time complexity of decentralized K-Means is described in Dhillon and Modha (2000). At each site, the distance and centroid recalculation computations come out to per-site complexity of $\mathcal{O}((3kr^2 + M_i k + M_i r^2 + kr^2) \cdot \mathcal{J})$ (Dhillon and Modha, 2000), where M_i is the number of instances at site i . The total number of computations consists of the sum of these site-wise complexities, and the centroid-averaging step with a complexity of $\mathcal{O}(kr^2)$, for a total of $\mathcal{O}((3kr^2 + Mk + Mr^2 + kr^2) \cdot \mathcal{J})$, where M is the total number of data instances in the decentralized network.

Since dK-Means is computed twice for full ddFNC, once on the exemplars, and once on the global set of subject windows, the complete complexity of the clustering stage of the algorithm is given as the dK-Means complexity for $M = \sum E_i$ added to the dK-Means complexity for $M = \sum m_i$, i.e., $\mathcal{O}((3kr^2 + (\sum E_i + \sum m_i)(k + r^2) + kr^2) \cdot \mathcal{J} + kr^2)$.

The overall site-wise complexity and communication for ddFNC is just the sum of the site-wise communication and complexities for each of the stages described here. In the paradigm described here, the communication and complexity on the aggregator is generally more demanding than that on the individual sites, which makes sense for cases where the aggregator has sufficient and reliable network and hardware resources. In cases where this is not necessarily true, some of the aggregation tasks can be distributed to other sites in the network, thus reducing communication and complexity on the final aggregator. In the dglICA algorithm, performing ICA on the aggregator may become a bottleneck if the aggregator does not have sufficient computational resources to perform a standard run of ICA; however, this problem could be mitigated by performing a hardware check on sites in the consortium, and assigning the role of aggregator dynamically based on availability of computational resources. For more discussion of the particularities of network communication and other issues which may arise in decentralized frameworks like the one used for ddFNC, see Plis et al. (2016).

3. DATA

3.1. Structural MRI for Decentralized VBM

As part of validating the proof-of-concept, we applied decentralized VBM to brain structure data collected on

chronic schizophrenic patients and healthy controls. Specifically, the data comes from the Mind Clinical Imaging Consortium (MCIC) collection- a publicly accessible, on-line data repository containing curated anatomical and functional MRI, in addition to other data, collected from individuals with and without a schizophrenia spectrum disorder (Gollub et al., 2013) and available via the COINS data exchange <https://coins.mrn.org> (Scott et al., 2011).

Although more information about the MCIC can be found in Gollub et al. (2013), here we will report numbers for the final data used in this study as some subjects were excluded during the preprocessing phase. The final cohort for whom data are available includes 146 patients and 160 controls with site distribution as follows: Site B (IA) 40 patients/67 controls; Site D (MGH) 32/23; Site C (UMN) 32/26; Site A (UNM) 42/44, respectively. All subjects provided informed consent to participate in the study that was approved by the human research committees at each of the sites.

Briefly, T1-weighted structural MRI (sMRI) images were acquired with the following scan parameters: $TR = 2,530$ ms for 3 T, $TR = 12$ ms for 1.5 T; $TE = 3.79$ ms for 3 T, $TE = 4.76$ ms for 1.5 T; $FA = 7^\circ$ for 3 T, $FA = 20^\circ$ for 1.5 T; $TI = 1100$ ms for 3 T; $Bandwidth = 181$ for 3 T, $Bandwidth = 110$ for 1.5 T; $voxelsize = 0.625 \times 0.625$ mm; slice thickness 1.5 mm; $FOV = 16 - 18$ cm.

The T1-weighted sMRI data were preprocessed using the Statistical Parametric Mapping software using unified segmentation (Ashburner and Friston, 2005), in which image registration, bias correction and tissue classification were performed using a single integrated algorithm resulting in individual brains segmented into gray matter, white matter and cerebrospinal fluid and nonlinearly warped to the Montreal Neurological Institute (MNI) standard space. The resulting gray matter concentration (GMC) images were re-sliced to $2 \times 2 \times 2$ mm, resulting in $91 \times 109 \times 91$ voxels. Although one can obtain both modulated (Jacobian corrected) and unmodulated gray matter segmentations, in this study, we use unmodulated GMC maps to test our regression models.

To test the decentralized regression on the MCIC data described in the previous paragraph, we regress the age, diagnosis, gender and the site covariates on the voxel intensities ($\sim 600,000$ voxels). All the decentralized computations discussed here have been performed on a single machine.

3.2. Functional MRI for dFNC

To evaluate ddFNC, we utilize imaging data from Damaraju et al. (2014) collected from 163 healthy controls (117 males, 46 females; mean age: 36.9 years) and 151 age- and gender matched patients with schizophrenia (114 males, 37 females; mean age: 37.8 years), for a total of 314 subjects.

The scans were collected during an eyes closed resting fMRI protocol at 7 different sites across United States and pass data quality control (see **Supplementary Material**). Informed and written consent was obtained from each participant prior to scanning in accordance with the Internal Review Boards of corresponding institutions (Keator et al., 2016). A total of 162 brain-volumes of echo planar imaging BOLD fMRI data were collected with a temporal resolution of 2 s on 3-Tesla scanners.

Imaging data for six of the seven sites was collected on a 3T Siemens Tim Trio System and on a 3T General Electric Discovery MR750 scanner at one site. Resting state fMRI scans were acquired using a standard gradient-echo echo planar imaging paradigm: FOV of 220×220 mm (64×64 matrix), TR = 2 s, TE = 30 ms, FA = 770, 162 volumes, 32 sequential ascending axial slices of 4 mm thickness and 1 mm skip. Subjects had their eyes closed during the resting state scan. Data preprocessing for dglCA was performed according to the preprocessing steps in Damaraju et al. (2014).

3.3. ddFNC Experimental Parameters

We verify that ddFNC can generate sensible dFNC clusters by replicating the centroids produced in Damaraju et al. (2014). We run both pooled and decentralized versions of our algorithm, and compare our results directly with the results provided by the authors of Damaraju et al. (2014). We thus closely follow the experimental procedure in Damaraju et al. (2014), with some of the additional post-processing omitted for simplicity. To evaluate

the success of our pipeline, we run a simple experiment where we implement the ddFNC pipeline end-to-end on the data, simulating 314 subjects being evenly shared over 2 decentralized sites.

We set a window-length of 22 time-points (44 s), for a total of 140 windows per subject. For dglCA, we first estimate 120 subject-specific principal components locally, and reduce each subject to 120 points in the temporal dimension. Subjects are then concatenated temporally on each site, and we use the GlobalPCA algorithm in Baker et al. (2015) to estimate 100

TABLE 1 | Correlation between SSE from pooled, single-shot and multi-shot regression.

	Pooled	Single-shot	Multi-shot
Pooled	1.000000	0.992905	1.000000
Single-shot	0.992905	1.000000	0.992905
Multi-shot	1.000000	0.992905	1.000000

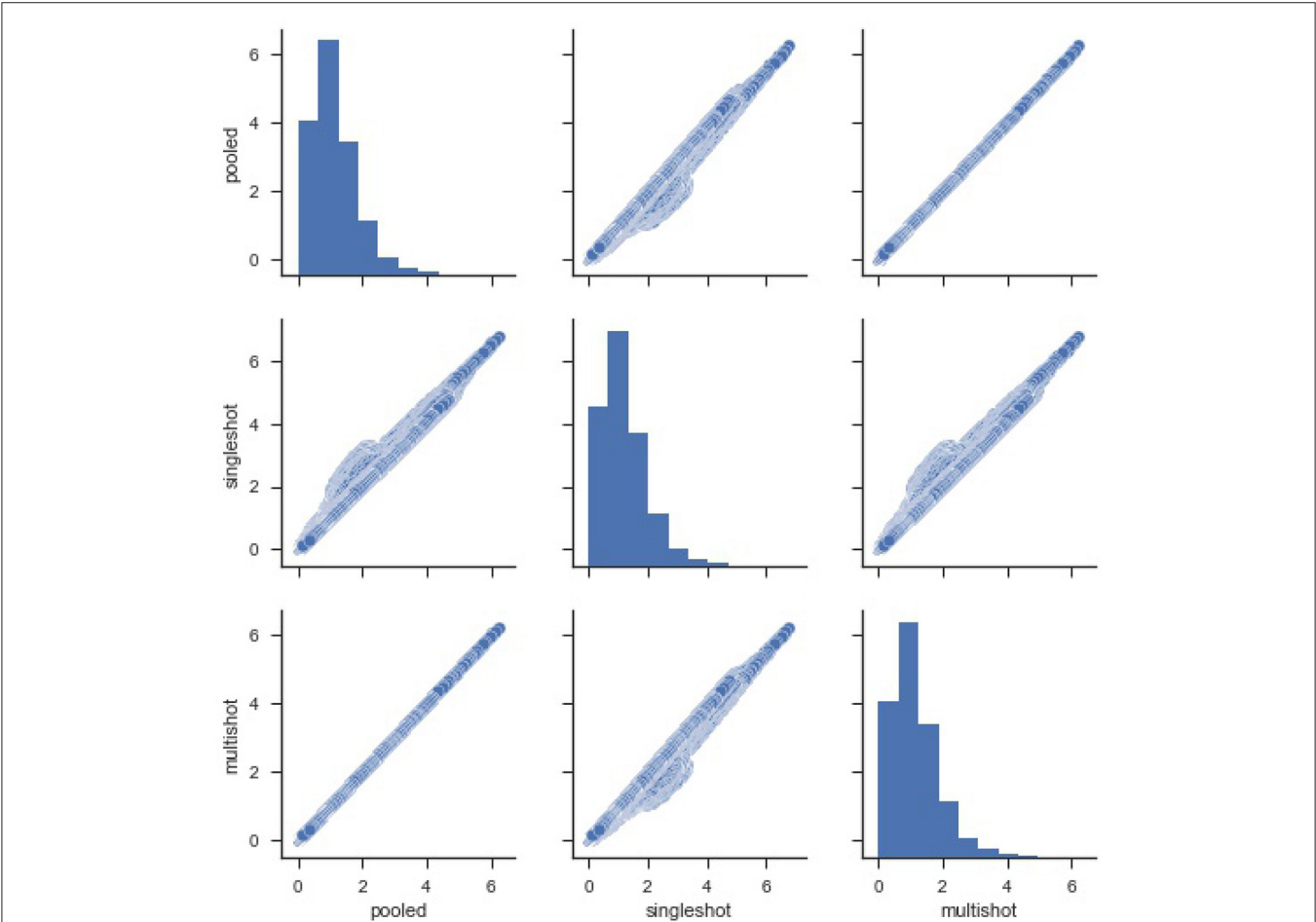


FIGURE 1 | Pairwise plot of Sum Square of Errors (SSE) from pooled, single-shot and multi-shot regression. Although the distribution plot looks similar across the three regressions, the pooled regression vs. multi-shot regression scatter plot demonstrates how identical they are to each other. The scatter plot of pooled regression vs. single-shot regression demonstrates that the SSE values obtained from single-shot regression are on the higher side compared to the values from pooled regression.

spatial components, and perform whitening. We then use local infomax ICA (Bell and Sejnowski, 1995) on the aggregator to estimate the unmixing matrix $\mathbf{W}$, and estimate 100 spatially independent components, $\hat{\mathbf{A}}$. We then broadcast $\hat{\mathbf{A}}$ back to the local sites, and each site computes subject-specific time-courses.

After spatial ICA, we have each site perform a set of additional post-processing steps prior to decentralized dFNC. First, we select 47 components from the initial 100, by computing components which are most highly correlated with the components from Damaraju et al. (2014). We then have each site drop the first 2 points from each subject, regress subject head movement parameters with 6 rigid body estimates, their derivatives and squares (total of 24 parameters). Additionally, any spikes identified are interpolated using 3rd order spline fits to good neighboring data, where spikes are defined as any points exceeding $\text{mean (FD)} + 2.5 \cdot \text{std(FD)}$, where FD is framewise displacement [interpolating 0 to 9 points (mean, sd: 3, 1.76)].

For clustering, we forgo a separate elbow-criterion estimation, and use the optimal number of clusters from Damaraju et al. (2014), setting $k = 5$. For the exemplar stage of clustering, we evaluate 200 runs where we initialize centroids uniformly randomly from local data, and then run dK-Means using the cluster averaging strategy in Dhillon and Modha (2000). For our distance measure, we use scikit-learn (Pedregosa et al., 2011) to compute the correlation distance between covariance matrices following the methods in Damaraju et al. (2014). To keep our implementation simple, unlike Damaraju et al. (2014), we do not utilize graphical LASSO to estimate the covariance matrix, and thus do not optimize for any regularization

parameters. Additionally, we do not perform additional Fisher-Z transformations or perform additional regularization using a previously computed static dFNC result. Future implementations may also utilize a decentralized static functional network connectivity (sFNC) algorithm as preprocessing, as is done for the pooled case in Damaraju et al. (2014). Finally, for the second stage of dK-Means, we initialize using the centroids from the run with the highest silhouette score, computed using the scikit-learn python toolbox (Pedregosa et al., 2011), again running dK-Means to convergence. After computing the centroids, we use the correlation distance and the Hungarian matching algorithm (Kuhn, 1955) to match both plotted spatial components from dglICA and the resulting centroids from dK-Means.

4. RESULTS

4.1. Decentralized VBM Results

For starters, in order to compare the efficacy of each regression (single-shot and multi-shot) against the pooled case, we present a simple pairwise plot of the SSE of the regression performed on every voxel, **Figure 1**. In mathematical terms, the SSE represents lowest objective function value that could be attained from the regression model. It can be seen from **Figure 1** that the SSE from multi-shot and pooled/centralized regression lie perfectly along a diagonal indicating the parameters obtained from them are identical. This can also be verified from **Table 1** showing the correlation between the different SSEs. Please note that results from the decentralized regression with normal equation were not presented as it has been mathematically shown to be equivalent to that of a pooled regression.

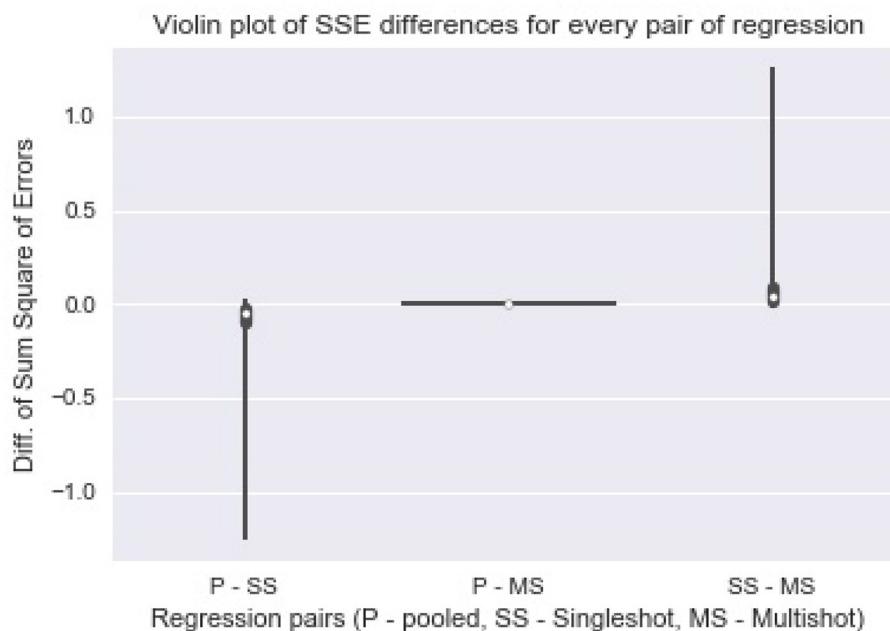


FIGURE 2 | Violin plot of Sum Square of Error differences between every pair of regression. The plot of differences in SSE from pooled regression and multi-shot regression (P-MS) centered around 0 demonstrates how identical the results from the two regressions are. On the other hand, the SSE values from single-shot regression are higher compared to those from the pooled regression.

It can be seen that the correlation between SSE from the centralized regression and multi-shot is 1. On the other hand, it can also be noticed that the SSE correlations between single-shot and pooled or single-shot and multi-shot are slightly lower than perfect correlation. The single-shot approach can be considered to be similar to a meta-analysis, whereas the multi-shot approach is basically a mega-analysis (i.e., equivalent to the pooled analysis).

Figure 2 shows a violin (distribution) plot of the difference in SSE from every pair of regression. Evidently, the differences in SSE between pooled and multi-shot regression are centered around 0. To reinforce our notion that the multi-shot is superior to single-shot we take a look at the R^2 values from the different regressions and compare. It can be seen from **Figure 3** that the R^2 values from multi-shot and pooled regression align perfectly along a diagonal (correlation = 1, refer to **Table 2**) or have exactly

the same distribution, whereas those from single-shot are all over the place.

As noted earlier, in addition to evaluating the regression model parameters, researchers will also be interested in understanding the statistical significance of the various parameter estimates. **Figures 4–6** show the statistical significance of each covariate (age, diagnosis and gender), from both

TABLE 2 | Correlation between R^2 from pooled, single-shot and multi-shot regression.

	Pooled	Single-shot	Multi-shot
Pooled	1.000000	0.906662	1.000000
Single-shot	0.906662	1.000000	0.906662
Multi-shot	1.000000	0.906662	1.000000

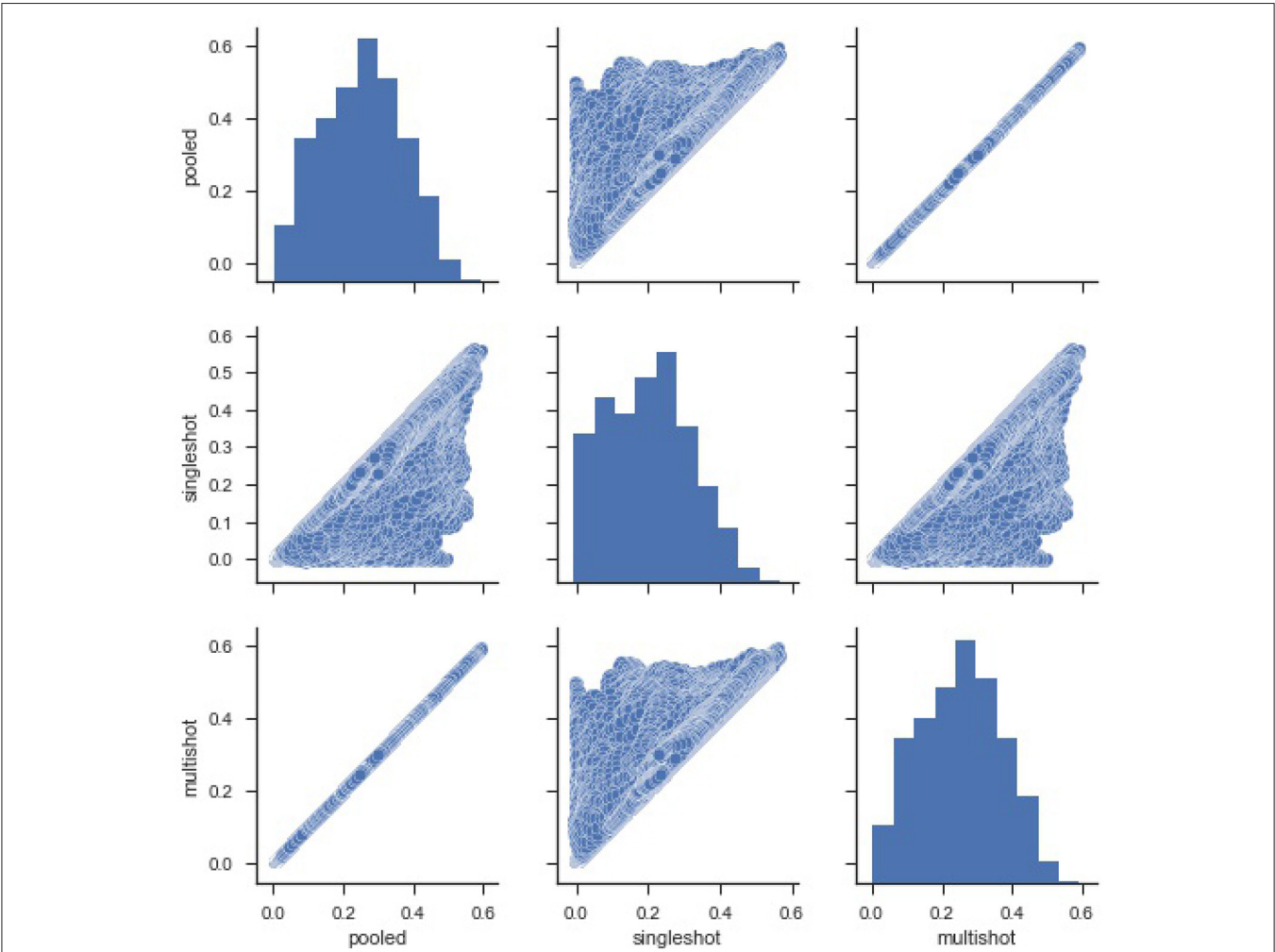


FIGURE 3 | Pairwise scatter plots of Coefficient of Determination R^2 from the three types of regression. It can be seen again that the R^2 values for the regressions from multi-shot regression and pooled regression are exactly equal. The R^2 values from single-shot regression are less than their corresponding values from pooled regression or multi-shot regression because the model being fit in single-shot has fewer covariates (Note, one of the limitations of the single-shot is that the site specific covariates could not be included as it introduces collinearity).

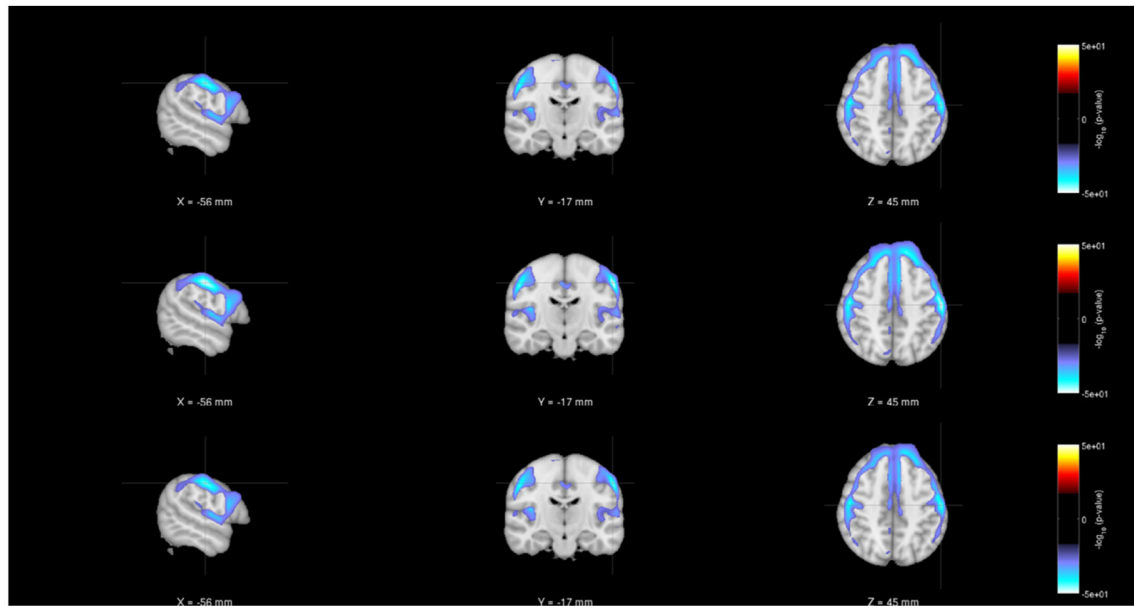


FIGURE 4 | Rendered images of voxel-wise significance values ($-\log_{10}p\text{-value} \times \text{sign}(t)$) for the covariate “Age” from pooled regression (**Top**) and single-shot regression (**Center**), and multi-shot regression (**Bottom**) overlaid on MNI average template. One could see that the regions with expected gray matter decrease as age increases are similar from all kinds of regression. Although the single-shot regression uses fewer covariates, the similarity of the rendered images with those of pooled regression or multi-shot regression indicate the relative weight or orientation of the corresponding β coefficient will be similar to those from pooled/multi-shot regression.

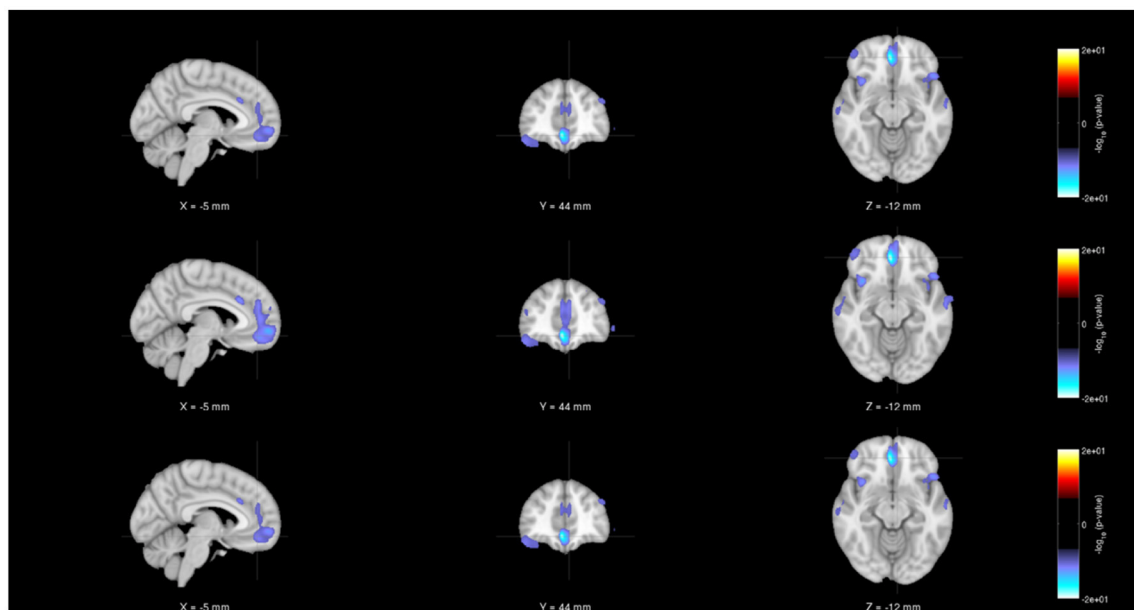


FIGURE 5 | Rendered images of voxel-wise significance values ($-\log_{10}p\text{-value} \times \text{sign}(t)$) for the covariate “Diagnosis” from pooled regression (**Top**) and single-shot regression (**Center**) and multi-shot regression (**Bottom**) overlaid on MNI average template. Regardless of the type of regression performed, the images indicate that in the medial frontal and bilateral temporal lobe/insula there is a significant gray matter density reduction for schizophrenic patients compared to the same regions of the healthy subjects.

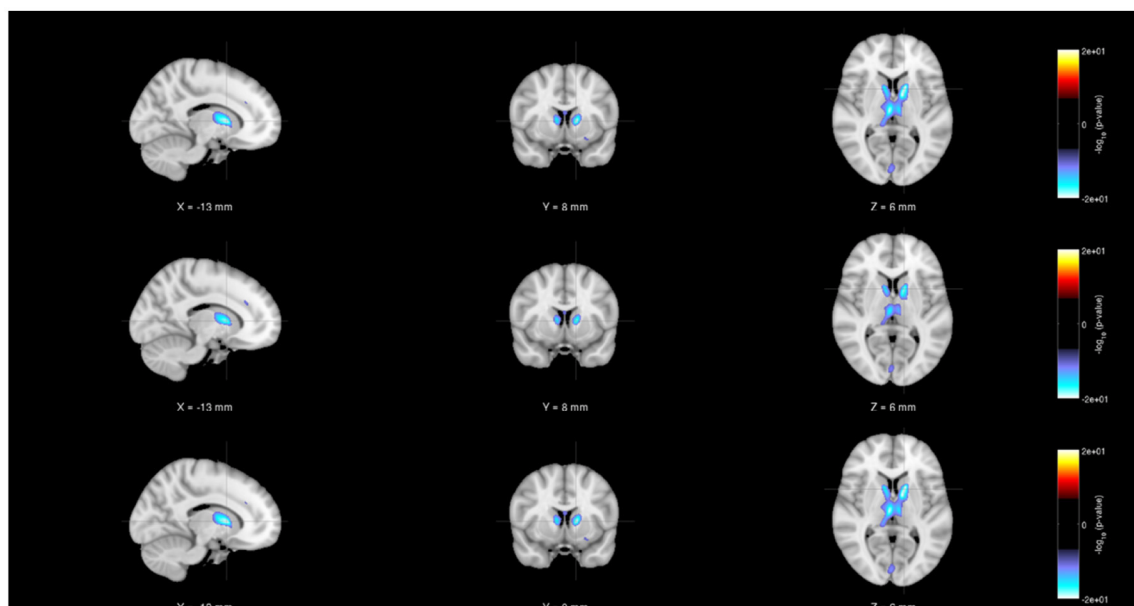


FIGURE 6 | Rendered images of voxel-wise significance values ($-\log_{10}p\text{-value} \times \text{sign}(t)$) for the covariate “Gender” from pooled regression (**Top**) and single-shot regression (**Center**) and multi-shot regression (**Bottom**) overlaid on MNI average template. It can be seen from all the three rendered images that there is a significant amount of gray matter reduction in the sub-cortical regions for males. Since we are using unmodulated gray matter maps, these sex differences could be due to changes in brain volumes.

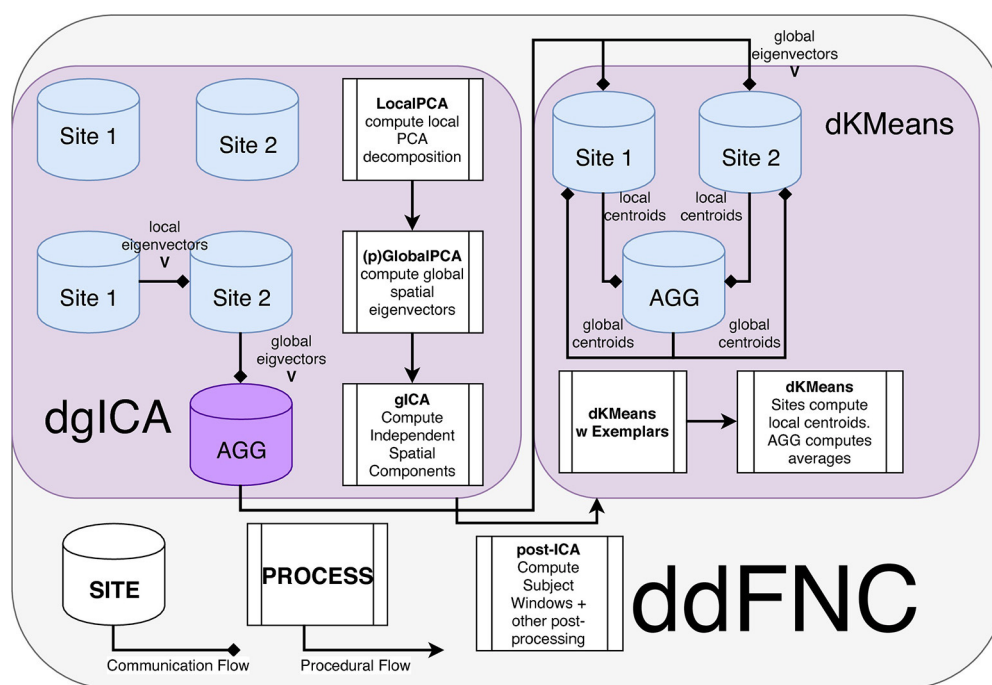


FIGURE 7 | Flowchart of the ddFNC procedure e.g., with 2 sites. To perform dgICA, sites first locally compute subject-specific LocalPCA to reduce the temporal dimension, and then use the GlobalPCA procedure from Baker et al. (2015) to compute global spatial eigenvectors, which are then sent to the aggregator. The aggregator then performs ICA on the global spatial eigenvectors, using InfoMax ICA (Bell and Sejnowski, 1995) for example, and passes the resulting spatial components back to local sites. The dK-Means procedure then iteratively computes global centroids using the procedure outlined in Dhillon and Modha (2000), first computing centroids from subject exemplar dFNC windows, and then using these centroids to initialize clustering over all subject windows.

centralized and decentralized regressions performed against each voxel, plotted on an MNI brain template. **Figure 4** shows the brain images with the $-(\log_{10}p\text{-val} \times \text{sign}(t))$ -values for the weight parameter corresponding to “Age.” It is notable to see that the results from the multi-shot regression have a perfect correlation to those from the pooled version. Moreover, the observations show the expected decrease in gray matter concentration as age increases. **Figures 5, 6** show the rendered images for $-\log_{10}p$ -values for the “Diagnosis” and “Gender” covariate, respectively.

4.2. ddFNC Results

A summary of the complete steps in the decentralized dFNC pipeline is given in **Figure 7**. In **Figure 8**, we plot some examples of the components estimated from decentralized spatial ICA in comparison with the spatial components from Damaraju et al. (2014), after performing Hungarian matching between the estimated spatial maps. We also plot the correlation of the components from our ICA implementation in comparison to the

components from Damaraju et al. (2014). Indeed, the estimated components are highly correlated with the results from Damaraju et al. (2014), for all 100 estimated components, as well for the 47 selected neurological components from Damaraju et al. (2014), indicating that dglICA is able to produce results comparable to the pooled case. We include additional spatial maps for all 47 estimated spatial components in the **Supplementary Material**.

In **Figure 9**, we plot the centroids from Damaraju et al. (2014) (**Figure 9A**), as well as the centroids estimated using decentralized dFNC (**Figure 9B**). Indeed, the centroids found using ddFNC prove similar to the centroids found in Damaraju et al. (2014), with centroids 2 and 3 being the closest matches under correlation distance.

5. DISCUSSION

The results described in the previous section demonstrate the fidelity of decentralized regression and decentralized dynamic function network connectivity in analyzing neuroimaging data.

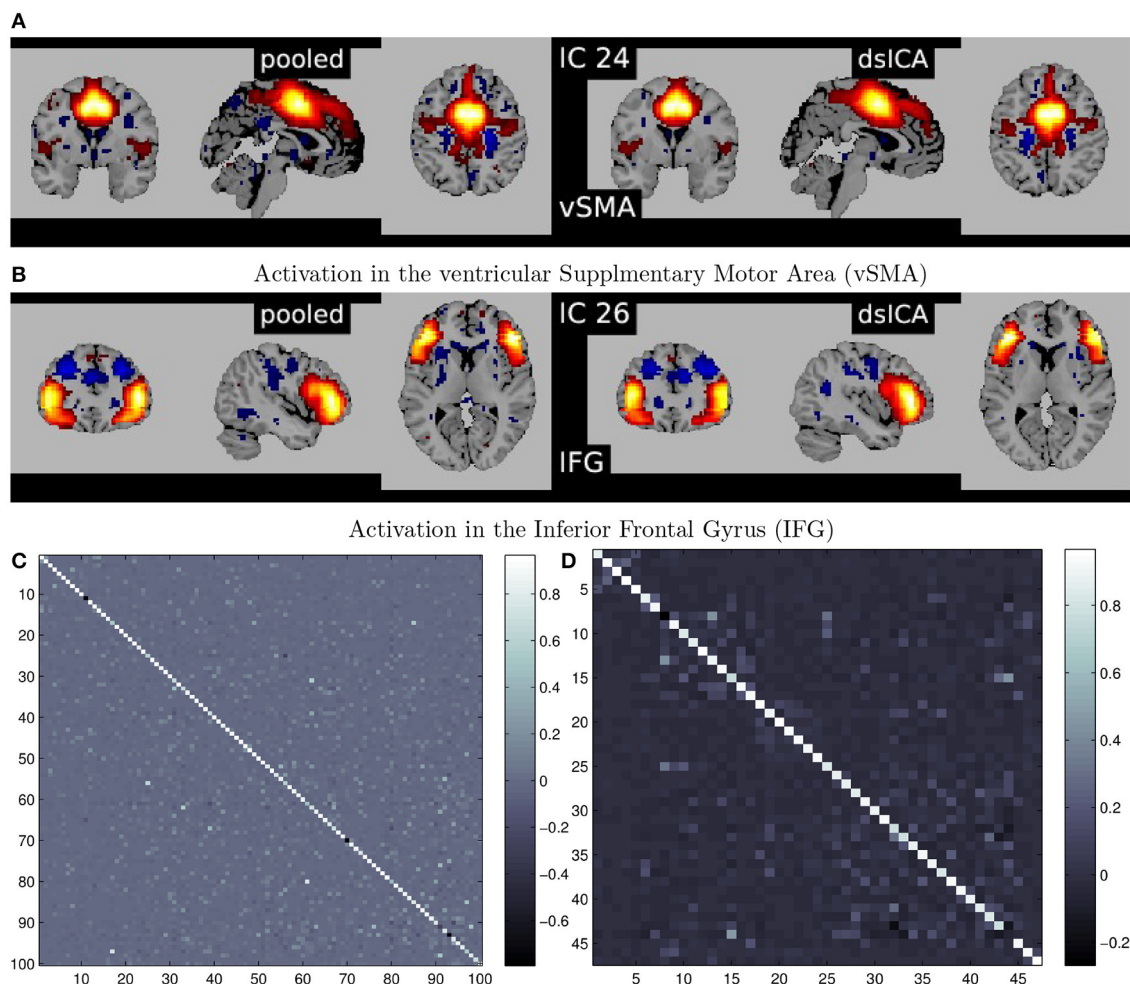
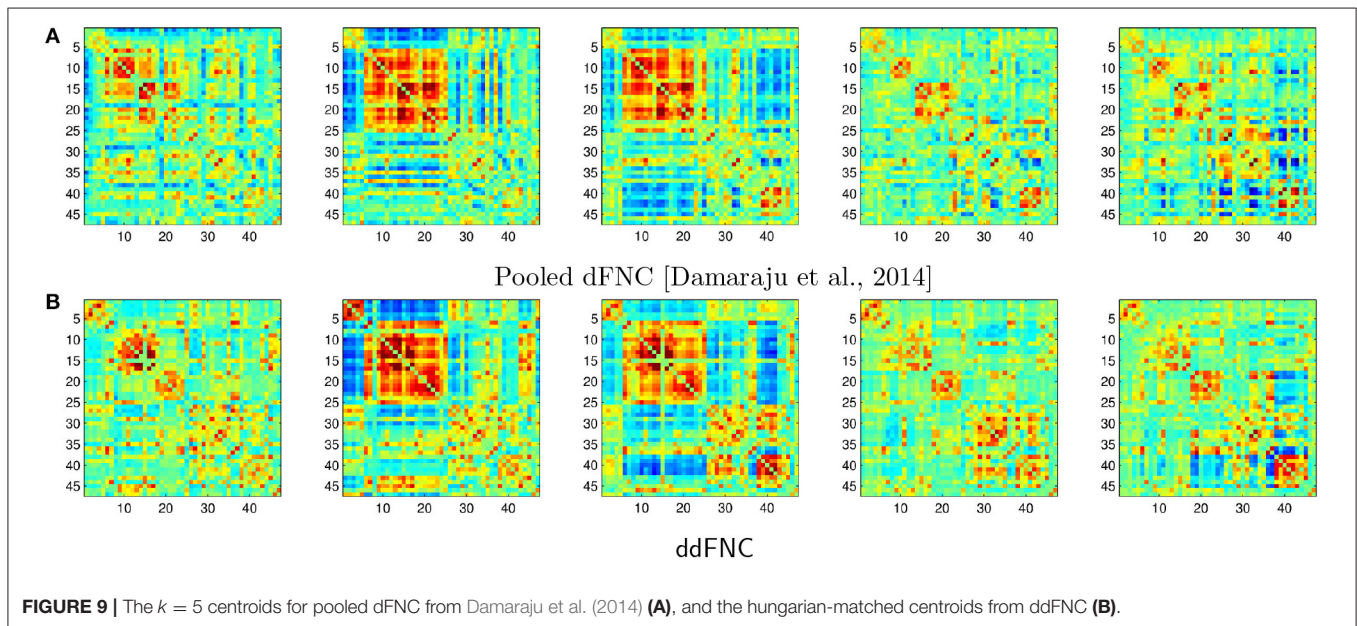


FIGURE 8 | (A,B) Illustrate examples of matched spatial maps from dglICA and pooled ICA. **(C,D)** Show the correlation of the components between pooled spatial ICA and dglICA after hungarian matching. **(C)** Shows correlation between all 100 components, and **(D)** Shows correlation between the 47 neurological components selected in Damaraju et al. (2014).



Although single-shot regression is simple and easy to implement, it limits our ability to incorporate site covariates and thus might not be extremely helpful. The decentralized regression with normal equation and multi-shot regression are superior to single-shot regression because not only do they allow incorporating site related variables but also give exact results as the pooled regression. The linearity and convexity of the regression objective function made this possible and thus are an excellent alternative to perform regression on multi-site datasets.

In terms of the regression objective function, either the sum of squared errors or mean sum of squared errors can be used in practice. However, it's mathematically convenient to use sum of squared errors which subsequently entails (at the AGG) a simple addition of the gradients ($\mathcal{O}(1)$) instead of a weighted average of the gradients ($\mathcal{O}(n)$). Added to that, we also showed how the sample size at the local sites has no bearing on the final results.

On a more practical note, the need for multi-shot regression might not arise often in a neuroimaging setting where the number of covariates is usually small. In such cases, the decentralized regression with normal equation will suffice. However, in decentralized settings where the number of covariates is usually large (machine learning/big data) the multi-shot regression comes to the fore. From a computational time standpoint, and as discussed in the computational complexity section, it should be obvious that the multi-shot regression takes more time to complete than the decentralized regression with normal equation as it involves iteratively passing the gradients between the local nodes and the AGG. It is worth mentioning that although the decentralized regression algorithms demonstrated here pertain to a simple linear regression model, these algorithms can easily be extended to more complex models with polynomial terms or interaction terms as well as to ridge regression, lasso regression, and elastic net regression.

Regarding ddFNC, we plan on performing a more robust analysis, going into the future, as a stand-alone algorithm,

particularly with respect to different variations on the dK-Means optimization and initialization, or with differing versions of ICA on the aggregator (AGG) node, such as fastICA (Koldovský et al., 2006), Entropy Bound Minimization (Li and Adali, 2010), and others. Additionally, the possibility of performing a decentralized static FNC either as a preprocessing step to ddFNC or a separate analysis is attractive. One other avenue worth exploring with ddFNC is the flow of information across the decentralized network. In particular, since the GlobalPCA step in dgICA already makes the procedure partially peer-to-peer, it makes sense to explore adding this functionality to the dK-Means methods to preserve this peer-to-peer structure. Finally, we plan to evaluate privacy-sensitive versions of ddFNC, utilizing differential-privacy or other privacy measures as a way to perform these analyses with some assurance of per-subject privacy in the decentralized network.

Finally, we note that the decentralization of algorithms in a neuroimaging setting emphasizes the importance of analysis on data present at multiple sites, the decentralization discussed herewith is no different from other decentralized algorithms discussed elsewhere in literature. The AGG is not really a master node *per se* but in fact one of the local sites itself. The term AGG was introduced to separate all the other local sites from that site where the results are accumulated.

6. CONCLUSION

In this paper, we presented a simple case study of how voxel-based morphometry and dynamic functional network connectivity analysis can be performed on multi-site data without the need for pooling data at a central site. The study shows that both the decentralized voxel-based morphometry as well as the decentralized dynamic functional network connectivity yield results that are comparable to its pooled counterparts guaranteeing a virtual pooled analysis effect by a chain of computation and communication process. Other advantages

of such a decentralized platform include data privacy and support for large data. In conclusion, the results presented here strongly encourage the use of decentralized algorithms in large neuroimaging studies over systems that are optimized for large-scale centralized data processing.

ETHICS STATEMENT

For the MCIC data, all subjects provided informed consent to participate in the study that was approved by the human research committees at each of the sites (UNM HRRC #03-429; UMin IRB #0404M59124; MGH IRB# 2004P001360; UIowa IRB #1998010017). In addition to the informed consent, all patients successfully completed a questionnaire verifying that they understood the study procedures.

For fBIRN data, all subjects provided informed consent to participate in the study that was approved by the human research committees of each of the participating institutes in the fBIRN data repository.

AUTHOR CONTRIBUTIONS

HG implemented the decentralized regression algorithms on structural MRI data and wrote the regression part of the

paper. BB implemented the decentralized dynamic functional network connectivity pipeline on functional MRI data and wrote that part of the paper. ED contributed immensely to the analysis as well as interpretation of the results from both decentralized regression and decentralized dFNC pipeline. SRP contributed to the brain imaging data preprocessing pipeline. SMP proposed the decentralized data analysis system and led the algorithm development effort. RS helped formulate the decentralized regression with normal equation and development of decentralized spatial ICA. VC led the team and formed the vision.

FUNDING

This work was funded by the National Institutes of Health (grant numbers: P20GM103472/5P20RR021938, R01EB005846, 1R01DA040487) and the National Science Foundation (grant numbers: 1539067 and 1631819).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fninf.2018.00055/full#supplementary-material>

REFERENCES

- Adams, H. H., Adams, H., Launer, L. J., Seshadri, S., Schmidt, R., Bis, J. C., et al. (2016). Partial derivatives meta-analysis: pooled analyses when individual participant data cannot be shared. *bioRxiv* 038893. doi: 10.1101/038893
- Allen, E. A., Damaraju, E., Plis, S. M., Erhardt, E. B., Eichele, T., and Calhoun, V. D. (2014). Tracking whole-brain connectivity dynamics in the resting state. *Cereb. Cortex* 24, 663–676. doi: 10.1093/cercor/bhs352
- Ashburner, J., and Friston, K. J. (2000). Voxel-based morphometry—the methods. *Neuroimage* 11, 805–821. doi: 10.1006/nimg.2000.0582
- Ashburner, J., and Friston, K. J. (2005). Unified segmentation. *NeuroImage* 26, 839–851. doi: 10.1016/j.neuroimage.2005.02.018
- Baker, B. T., Silva, R. F., Calhoun, V. D., Sarwate, A. D., and Plis, S. M. (2015). “Large scale collaboration with autonomy: Decentralized data ICA,” in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, (Boston, MA: IEEE), 1–6.
- Bell, A. J., and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.* 7, 1129–1159. doi: 10.1162/neco.1995.7.6.1129
- Bottou, L. (2010). “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT'2010* (Paris: Springer), 177–186.
- Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., et al. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nat. Rev. Neurosci.* 14:365. doi: 10.1038/nrn3475
- Calhoun, V. D., and Adali, T. (2012). Multisubject independent component analysis of fMRI: a decade of intrinsic networks, default mode, and neurodiagnostic discovery. *IEEE Rev. Biomed. Eng.* 5, 60–73. doi: 10.1109/RBME.2012.2211076
- Calhoun, V. D., Adali, T., Pearson, G. D., and Pekar, J. (2001). A method for making group inferences from functional mri data using independent component analysis. *Hum. Brain Mapp.* 14, 140–151. doi: 10.1002/hbm.1048
- Carter, K. W., Francis, R. W., Carter, K., Francis, R., Bresnahan, M., Gissler, M., et al. (2015). Viper: a software platform for the virtual pooling and analysis of research data. *Int. J. Epidemiol.* 45, 408–416. doi: 10.1093/ije/dyv193
- Cragin, M. H., Palmer, C. L., Carlson, J. R., and Witt, M. (2010). Data sharing, small science and institutional repositories. *Philos. Trans. R. Soc. Lond. A Math. Phys. Eng. Sci.* 368, 4023–4038. doi: 10.1098/rsta.2010.0165
- Damaraju, E., Allen, E. A., Belger, A., Ford, J., McEwen, S., Mathalon, D., et al. (2014). Dynamic functional connectivity analysis reveals transient states of dysconnectivity in schizophrenia. *NeuroImage* 5, 298–308. doi: 10.1016/j.neuroimage.2014.07.003
- Datta, S., Giannella, C., and Kargupta, H. (2006). “K-means clustering over a large, dynamic network,” in *Proceedings of the 2006 SIAM International Conference on Data Mining* (SIAM), 153–164. doi: 10.1137/1.9781611972764.14
- Datta, S., Giannella, C., and Kargupta, H. (2009). Approximate distributed k-means clustering over a peer-to-peer network. *IEEE Trans. Knowl. Data Eng.* 21, 1372–1388. doi: 10.1109/TKDE.2008.222
- Deco, G., Ponce-Alvarez, A., Mantini, D., Romani, G. L., Hagmann, P., and Corbetta, M. (2013). Resting-state functional connectivity emerges from structurally and dynamically shaped slow linear fluctuations. *J. Neurosci.* 33, 11239–11252. doi: 10.1523/JNEUROSCI.1091-13.2013
- Dhillon, I. S., and Modha, D. S. (2000). “A data-clustering algorithm on distributed memory multiprocessors,” in *Large-Scale Parallel Data Mining*, Workshop on Large-Scale Parallel KDD Systems, SIGKDD (Berlin; Heidelberg: Springer), 245–260.
- Di Fatta, G., Blasa, F., Cafiero, S., and Fortino, G. (2013). Fault tolerant decentralised k-means clustering for asynchronous large-scale networks. *J. Paral. Distribut. Comput.* 73, 317–329. doi: 10.1016/j.jpdc.2012.09.009
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 12, 2121–2159.
- Erhardt, E. B., Rachakonda, S., Bedrick, E. J., Allen, E. A., Adali, T., and Calhoun, V. D. (2011). Comparison of multi-subject ica methods for analysis of fMRI data. *Hum. Brain Mapp.* 32, 2075–2095. doi: 10.1002/hbm.21170
- Fennema-Notestine, C., Gamst, A. C., Quinn, B. T., Pacheco, J., Jernigan, T. L., Thal, L., et al. (2007). Feasibility of multi-site clinical structural neuroimaging studies of aging using legacy data. *Neuroinformatics* 5, 235–245. doi: 10.1007/s12021-007-9003-9
- Forman, G., and Zhang, B. (2000). Distributed data clustering can be efficient and exact. *ACM SIGKDD Explor. Newsl.* 2, 34–38. doi: 10.1145/380995.381010

- Gollub, R. L., Shoemaker, J. M., King, M. D., White, T., Ehrlich, S., Sponheim, S. R., et al. (2013). The mcic collection: a shared repository of multi-modal, multi-site brain image data from a clinical investigation of schizophrenia. *Neuroinformatics* 11, 367–388. doi: 10.1007/s12021-013-9184-3
- Hibar, D. P., Stein, J. L., Renteria, M. E., Arias-Vasquez, A., Desrivieres, S., Jahanshad, N., et al. (2015). Common genetic variants influence human subcortical brain structures. *Nature* 520, 224. doi: 10.1038/nature14101
- Jagannathan, G., and Wright, R. N. (2005). “Privacy-preserving distributed k-means clustering over arbitrarily partitioned data,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD’05* (Chicago, IL: ACM), 593–599.
- Keator, D. B., van Erp, T. G., Turner, J. A., Glover, G. H., Mueller, B. A., Liu, T. T., et al. (2016). The function biomedical informatics research network data repository. *Neuroimage* 124, 1074–1079. doi: 10.1016/j.neuroimage.2015.09.003
- Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. arXiv:1412.6980.
- Koldovský, Z., Tichavský, P., and Oja, E. (2006). Efficient variant of algorithm fastica for independent component analysis attaining the cramer-rao lower bound. *IEEE Trans. Neural Netw.* 17, 1265–1277. doi: 10.1109/TNN.2006.875991
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Res. Logist. Q.* 2, 83–97. doi: 10.1002/nav.3800020109
- Landis, D., Courtney, W., Dieringer, C., Kelly, R., King, M., Miller, B., et al. (2016). Coins data exchange: an open platform for compiling, curating, and disseminating neuroimaging data. *NeuroImage* 124, 1084–1088. doi: 10.1016/j.neuroimage.2015.05.049
- Lewis, N., Plis, S., and Calhoun, V. (2017). “Cooperative learning: Decentralized data neural network,” in *2017 International Joint Conference on Neural Networks (IJCNN)* (Anchorage, AK), 324–331.
- Li, X.-L., and Adali, T. (2010). Independent component analysis by entropy bound minimization. *IEEE Trans. Signal Process.* 58, 5151–5164. doi: 10.1109/TSP.2010.2055859
- Ming, J., Verner, E., Sarwate, A., Kelly, R., Reed, C., Kahleck, T., et al. (2017). Coinstac: decentralizing the future of brain imaging analysis. *F1000Res.* 6:1512. doi: 10.12688/f1000research.12353.1
- Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Dokl. AN USSR* 269, 543–547.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Plis, S. M., Sarwate, A. D., Wood, D., Dieringer, C., Landis, D., Reed, C., et al. (2016). Coinstac: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data. *Front. Neurosci.* 10:365. doi: 10.3389/fnins.2016.00365
- Poldrack, R. A., Barch, D. M., Mitchell, J., Wager, T., Wagner, A. D., Devlin, J. T., et al. (2013). Toward open sharing of task-based fmri data: the openfmri project. *Front. Neuroinform.* 7:12. doi: 10.3389/fninf.2013.00012
- Roshchupkin, G. V., Adams, H., Vernooij, M. W., Hofman, A., Van Duijn, C., Ikram, M. A., et al. (2016). Hase: framework for efficient high-dimensional association analyses. *Sci. Rep.* 6:36076. doi: 10.1038/srep36076
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323:533. doi: 10.1038/323533a0
- Saha, D. K., Calhoun, V. D., Panta, S. R., and Plis, S. M. (2017). “See without looking: joint visualization of sensitive multi-site datasets,” in *Proceedings of the Twenty-Sixth International Conference on Artificial Intelligence (IJCAI’2017)* (Melbourne, VIC), 2672–2678.
- Sakoglu, Ü., Pearlson, G. D., Kiehl, K. A., Wang, Y. M., Michael, A. M., and Calhoun, V. D. (2010). A method for evaluating dynamic functional network connectivity and task-modulation: application to schizophrenia. *Magn. Reson. Mater. Phys. Biol. Med.* 23, 351–366. doi: 10.1007/s10334-010-0197-8
- Scott, A., Courtney, W., Wood, D., de la Garza, R., Lane, S., Wang, R., et al. (2011). Coins: an innovative informatics and neuroimaging tool suite built for large heterogeneous datasets. *Front. Neuroinform.* 5:33. doi: 10.3389/fninf.2011.00033
- Shringarpure, S. S., and Bustamante, C. D. (2015). Privacy risks from genomic data-sharing beacons. *Am. J. Hum. Genet.* 97, 631–646. doi: 10.1016/j.ajhg.2015.09.010
- Smith, S. M., Fox, P. T., Miller, K. L., Glahn, D. C., Fox, P. M., Mackay, C. E., et al. (2009). Correspondence of the brain’s functional architecture during activation and rest. *Proc. Natl. Acad. Sci. U.S.A.* 106, 13040–13045. doi: 10.1073/pnas.0905267106
- Sweeney, L. (2002). k-anonymity: a model for protecting privacy. *Int. J. Uncert. Fuzziness Knowl. Based Syst.* 10, 557–570. doi: 10.1142/S0218488502001648
- Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., et al. (2011). Data sharing by scientists: practices and perceptions. *PLoS ONE* 6:e21101. doi: 10.1371/journal.pone.0021101
- Thompson, P. M., Andreassen, O. A., Arias-Vasquez, A., Bearden, C. E., Boedhoe, P. S., Brouwer, R. M., et al. (2017). Enigma and the individual: predicting factors that affect the brain in 35 countries worldwide. *Neuroimage* 145, 389–408. doi: 10.1016/j.neuroimage.2015.11.057
- Thompson, P. M., Stein, J. L., Medland, S. E., Hibar, D. P., Vasquez, A. A., Renteria, M. E., et al. (2014). The enigma consortium: large-scale collaborative analyses of neuroimaging and genetic data. *Brain Imaging Behav.* 8, 153–182. doi: 10.1007/s11682-013-9269-5
- Turner, J. A., Damaraju, E., Van Erp, T. G., Mathalon, D. H., Ford, J. M., Voyvodic, J., et al. (2013). A multi-site resting state fmri study on the amplitude of low frequency fluctuations in schizophrenia. *Front. Neurosci.* 7:137. doi: 10.3389/fnins.2013.00137
- van Erp, T. G., Hibar, D. P., Rasmussen, J. M., Glahn, D. C., Pearlson, G. D., Andreassen, O. A., et al. (2016). Subcortical brain volume abnormalities in 2028 individuals with schizophrenia and 2540 healthy controls via the enigma consortium. *Mol. Psychiatry* 21:547. doi: 10.1038/mp.2015.63
- Wojtalewicz, N. P., Silva, R. F., Calhoun, V. D., Sarwate, A. D., and Plis, S. M. (2017). “Decentralized independent vector analysis,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (New Orleans, LA: IEEE), 826–830.
- Yuan, K., Ling, Q., and Yin, W. (2016). On the convergence of decentralized gradient descent. *SIAM J. Optim.* 26, 1835–1854. doi: 10.1137/130943170
- Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. arXiv[preprint] arXiv:1212.5701.

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

Copyright © 2018 Gazula, Baker, Damaraju, Plis, Panta, Silva and Calhoun. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.