Diffusion MRI and Cognitive Dysfunction in Post-Hospitalised COVID-19 Patients

Candidate Number: 8270T

Supervisor: Dr Marta Correia

Online Notebook Link: https://universityofcambridgecloud-my.sharepoint.com/:o:/r/personal/ld558_cam_ac_uk/Documents/Project%20Notebook%20Uploaded?d=we09088709e0a438b87c6c17f0a82081c&csf=1&web=1&e=P3enuS

Except where specific reference is made to the work of others, this work is original and has not been already submitted either wholly or in part to satisfy any degree requirement at this or any other university.

Abstract

Diffusion MRI (dMRI) leverages water molecule movement to give perspective on the microstructural organisation of brain tissue. Previous studies have established that dMRI biomarkers can be used to assess the severity of generalised abnormalities in central nervous function caused by acute COVID-19 viral infection. This investigation extends those studies by considering how Fractional Anisotropy and Mean Diffusivity are affected longitudinally. Analysis revealed diffusion restoration in white matter (WM) with post-hospitalisation treatment whilst grey matter (GM) exhibited a more heterogenous response with some persistency of microstructural disruption. The right side of the brain was more susceptible to COVID-19 with 9 of 13 significantly affected WM tracts on the right. Cognitive metric performance was then merged with diffusion data. The change in fractional anisotropy in the Middle Cingulate Gyrus, a region clinically known for responsibility in spatial processing, was discovered to be the best predictor of spatial reasoning score ($p = 0.0275$) when a generalised linear model was fit. Thus, dMRI data was seen to be an effective predictor of COVID-19 neuropathological response. Finally, fitting a three component Gaussian Mixture Model (GMM) allowed for voxel-based abnormality scores, providing a more nuanced picture of damage than averaged-out regional FA/MD. Indeed, abnormality score inclusion in the GLM resulted in a 45% reduction in AIC. Such abnormality scores are the best predictors of COVID-19 induced cognitive dysfunction and enable earlier diagnosis.

1. Background Science

1.1 Diffusion MRI

1.1.1 Diffusion

Diffusion imaging monitors water molecule movement to evaluate biological microenvironments. Free water molecules diffuse isotropically under Brownian motion in the absence of cellular obstacles. Restricted anisotropic diffusion describes directed movement such as in axons where myelin layers restrict transversal diffusion[1].

1.1.2 MRI Fundamentals

MRI relies on nuclear magnetic resonance. Exposing protons to an external magnetic field results in precession at a certain Larmor frequency, ω_0[2].

$$\omega_0 = \gamma \cdot B_0$$

Equation 1. Larmor frequency, ω_0. γ = gyromagnetic ratio, B_0 = the strength of the external magnetic field.

A radio frequency pulse applied at ω_0 can be absorbed and then re-emitted by the protons on de-excitation.

© 8270T MMXXIII

Properties associated with this de-excitation indicate proton density and location, yielding MRI contrast.

[See Appendix A for further MRI explanation]

1.1.3 Diffusion MRI

MRI is made sensitive to diffusion through linearly varying the magnetic field strength applied to protons in a region of interest (ROI) and pulsing it. This 'field gradient pulse' devised by Stejskal et Al.[3] causes precession of the protons in the water molecules. Along the gradient, precession occurs at different rates resulting in dispersion of phase and signal loss on de-excitation. Another gradient pulse is applied but with the opposite direction to rephase the spins. For protons in water molecules that remained stationary, this rephasing will reveal a strong contrast signal. For water molecules that diffusively moved, their new position prevents perfect rephasing. Hence, the more a water molecule moves, the more dephased it will be and the weaker the recorded signal[4].

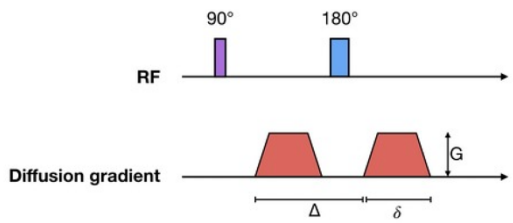

Diagram 1. Diffusion weighted field gradient pulse sequence in MRI. G = gradient pulse magnitude, δ = width of gradient pulse, Δ = times between gradient pulses [5].

1.1.4 Field Gradient Pulse Signal

The pulsed magnetic field gradient for dMRI is characterised by the b-factor.

$$b = \gamma^2 G^2 \tau^2 (\Delta - \tau/3)$$

Equation 2. b-factor. G = gradient pulse magnitude, τ = gradient application time, Δ = times between gradient pulses and γ = the gyromagnetic ratio.

Two diffusion imaging sequences with different b-factors can be used to quantify the degree of molecular mobility through the apparent diffusion coefficient (ADC)[6].

$$ADC = \frac{\log(s_0 - s_1)}{b_1 - b_0}$$

Equation 3. Apparent Diffusion Coefficient. b = b-factor of respective imaging sequence. s = diffusion weighted signal intensity.

ADC calculations along a minimum of six noncollinear directions allows for determination of a diffusion tensor D. The diffusion tensor is a 3x3 symmetric, positive-definite matrix with three orthogonal eigenvectors and three positive eigenvalues. D can be understood as defining an ellipsoid that describes both the diffusivity and isotropy of water molecule diffusion along each ellipsoid axis[7].

$$\boldsymbol{D} = \begin{matrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{matrix}$$

Equation 4. Representation of the diffusion tensor as a 3x3 symmetric matrix.

1.1.5 Diffusion Tensor Scalars

The mean diffusivity (MD) defines the overall magnitude of diffusion.

$$MD = \frac{1}{3}\big(Tr(D)\big)$$

*Equation 5. Mean Diffusivity as the trace of the diffusion tensor, **D**.*

Fractional anisotropy (FA) is a normalised measure ranging from 0 to 1 that defines the fraction of diffusion that is anisotropic. From its equation, FA can be interpreted as the difference of the tensor's ellipsoid shape from that of a perfect sphere[8]. The perfect sphere representing complete isotropic motion with an FA=0.

$$\mathrm{FA} = \sqrt{\frac{3}{2}\sum_{i=1}^{3}\left(\frac{\lambda_i - \frac{\mathrm{Tr}(\boldsymbol{D})}{3}}{\frac{\mathrm{Tr}(\boldsymbol{D})}{3}}\right)^2}$$

*Equation 6. Fractional Anisotropy. λ_i is the ith eigenvalue of the diffusion tensor, **D**.*

1.1.6 Biological Utility of Diffusion Tensor Scalars

Diffusion tensor scalars are useful as subtle neuropathology can be detected beyond conventional MRI resolution. Low FA suggests microstructural disruption to the tissues in WM tracts and abnormal variation in MD suggests oedema[9].

1.1.7 Brain Parcellation

The brain has great spatial heterogeneity[10]. Brain parcellation defines distinct partitions of white/GM. Parcellation can involve traditional literature-based atlases or contemporary mathematically driven region partition.

For GM parcellation, MALPEM[11] uses Expectation-Maximisation of MRI relaxation times. This approach accounts for individual variations in brain structure and hence copes better with brain lesions/neuropathology present in COVID-19 patients.

For WM parcellation, TractSeg[12] uses a deep learning approach for segmentation that remains accurate irrespective of lesions.

Parcellation allows for the characterisation of response of specific brain regions to COVID-19.

1.2 COVID-19 and Brain Pathology

1.2.1 SARS-Cov-2 and the brain

COVID-19 is a contagious disease caused by the SARS-CoV-2 virus.

Brain injury is likely to be a feature of severe viral infection from COVID-19 and is caused by inflammatory response on breaching the brain-blood-barrier. Biomarkers for an inflammatory response in the brain are present both acutely and longitudinally for patients, suggesting ongoing active brain injury post hospitalisation[13].

1.2.2 Cognitive Metrics

COVID-19 neuropathology is seen to impact patient higher-order cognition. Spatial and verbal reasoning are disproportionately impaired when measuring patients with psychometric performance tests[14]. These metrics are a useful indicator of symptomatic response but must be controlled against age and demographic related cognitive decline.

1.3 Project Aims

Newcombe et Al.[9] investigated six patients with severe COVID-19-related respiratory failure with imaging happening on average 19 days post admission. They found pervasive abnormalities on quantitative DTI compared to controls.

Hampshire et Al.[14] assessed the symptomatic cognitive deficits induced by COVID-19 through cognitive metric performance across 81,337 patients. Statistically significant deficits were found across all patients with greater severity for the 192 hospitalised.

This project aims at extending these studies through:

1. Replicating Newcombe et Al.[9] in a longitudinal context to assess post-hospitalisation response of affected brain regions. Persistency of microstructural brain disruption or response to treatment would further affirm the responsibility of COVID-19 for neural distress.

2. Merge the relevant metrics of Hampshire et Al.[14] with dMRI data through multi-modal analysis to better understand how cognitive abnormalities can predict day-to-day patient function. Lower metric scores being associated with greater microstructural damage is expected.

3. Establish a burden of illness[15] using voxel analysis of dMRI data to define abnormality scores.

The last objective is motivated by Addenbrookes clinician Prof. David Menon who cited Coles et Al.'s[15] work on defining a burden as enabling earlier diagnoses. COVID-19 neuropathology treatment would improve by replicating this.

Two competing forces are present in voxel-based abnormality. Looking across multiple voxels can smooth out data to where no abnormalities are seen, but investigating too few voxels is susceptible to noise[16]. Powerful statistical techniques like machine learning are hence needed for this analysis.

2. Methodology

2.1 Replication of Newcombe et Al.[9] (acute study) in Longitudinal Context

2.1.1 Harmonisation

Diffusion data was provided pre-processed for patients and controls for both singleshell and multishell scan regimes. These regimes describe the number of b-values used in imaging. There are a total of 15 patients and 39 controls across both regimes [See Appendix B for a Table 1 with patient profiles]. The mixture of regimes and the limited patient numbers stem from difficulties in data acquisition due to covid restrictions. Limitations with statistical power stemming from sample size were mitigated through data harmonisation across regime datasets.

Harmonisation was conducted through standardisation followed by outlier removal. The resulting z-scores are re-scaled and centred allowing merging of regime datasets.

$$Z = \frac{X - \mu}{\sigma}$$

Equation 7. z-score. X = data point, μ = mean of the dataset, σ = standard deviation of the dataset.

The data in the spreadsheet was not immediately usable; data format inconsistencies needed to be resolved and scans required partitioning into acute and longitudinal. Use of python 'pandas'[17] dataframes and scientific computing libraries enabled creation of custom high-level functions that served to clean up the data [Appendix C: Custom Functions]. For example, a function was needed to convert MALPEM parcellation to a proprietary GM grouping, that focused on cortical regions, used in the Newcombe study[9].

2.1.2 Determining Significance

Extending on Newcombe et Al.[9] longitudinally requires comparison of acute and longitudinal scan data to gauge for statistically significant change in diffusion parameters across that time. Significance would indicate that COVID-19 induced dysfunction and subsequent treatment is what motivates changes in neural microstructure measured by dMRI.

Comparisons were done by a Wilcoxon signed-rank test as scan data lacked normality of distribution (as confirmed by a Shaprio-Wilk test). Acute and longitudinal observation pairs are dependent, making a Wilcoxon test appropriate.

2.1.3 Error Analysis in Acute Study Replication

Errors from motion artifacts, Gibbs ringing, and potential scanner drift were eliminated in pre-processing[9].

MALPEM parcellation identifies 138 regions. Significance tests across all regions increases the risk of Type 1 errors in incorrect rejection of the null hypothesis. Hence, correction for multiple comparisons is necessary. Bonferroni adjustment is frequently used[18] but is too stringent in this case of brain parcellation with a high number of partitions. Conversely, Hochberg adjustment[18] has a less conservative character and is more appropriate. It involves incremental adjustment of the significance threshold for each comparison being formulated. Once a comparison is conducted with a p value under the calculated threshold, all subsequent comparisons are considered significant.

$$\alpha'_i = \frac{\alpha}{m - i + 1}$$

Equation 8. Hochberg Adjustment. m = total number of comparisons, i = current comparison considered, α'_i = adjusted significance level for the ith comparison, *α = baseline significance level (0.05).*

Calculation of a standard error in individual FA/MD per ROI per patient is not possible as only a single scan is taken. Nevertheless, outlier removal in data harmonisation means that providing a mean and standard error across patients is a suitable means of expressing results.

2.2 Cognitive Metrics

2.2.1 Verbal and Spatial Reasoning

Clinically relevant cognitive metrics are identified in Hampshire et Al.[14] as being verbal and spatial reasoning.

Using the custom functions, new 'pandas' dataframes were created, tying diffusion data to cognitive scores. To focus the investigation, only the top five most affected regions were considered against the metrics. A generalised linear model (GLM) was fit to assess the contributions of these affected regions to the score. Nuisance regressors like age were also incorporated in the model. The GLM was fit in R [Appendix D: GLM Code].

2.2.2 Generalised Linear Models (GLMs)

A GLM models the relationship between a response and associated explanatory variables. GLMs are ideal as they can simultaneously handle differing data types, from continuous diffusion values to the binary or count regressors in patient profile[19].

The GLM generalises linear regression by allowing the linear model to be related to the response variable through a link function. A GLM consists of three components. The random component refers to the probability of the response variable (Y). The systematic component consists of a linear function of the predictor variables. Predictor variables that affect the response variable but that aren't the focus of the investigation such as age or gender in this context are known as nuisance regressors.

$$\eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Equation 9. GLM Systematic Component. η = linear predictor, β_i = model coefficients, X_i = predictor variables.

Finally, a link function that links the expectation of the response variable to the systematic component.

$$E(Y) = g(\mu) = \eta$$

Equation 10. GLM link Function. E() = expectation, g() = link function, Y = random component, μ = expected value of the response variable, η = linear predictor.

The parameters of a GLM are calculated through numerical techniques of maximum likelihood.

2.2.3 Generalised Linear Models vs Linear Models

Use of a GLM over a linear model is motivated by several reasons in this case. The assumption of normality required for a regular linear model is violated in the data. Cognitive test scores are likely to have a skewed distribution as do age and other nuisance regressors due to limitations in sample size. The assumption of homoscedasticity is violated as it is well established that the variance in cognitive metric scores changes with age [reference]. Further, nuisance regressors in a GLM need not be artificially converted into continuous variables. Overall, the presence of a link function in GLMs allow for effective handling of non-linear relationships and the different data types needed for this investigation[20].

2.2.4 Response Times and Offsets

Performance in the metric comes in two forms: total accuracy and millisecond response time. The latter was concluded as a better response variable to consider. Response time has greater variation than accuracy which improves the statistical power in the GLM fit. Response time also has ideal distribution characteristics that lend itself to a GLM. Times are positively skewed and continuous, meaning it can be well modelled with a Gamma distribution and a logarithmic link function.

$$\mu = \exp(X\beta)$$

Equation 11. Expected value of the response variable given a logarithmic link function. X = predictor variables, β = model coefficients.

FA/MD values can be negative after standardisation into z-scores. Thus, for a Gamma distribution, applying some constant offset is necessary to make z-scores always positive. The logarithm link function is useful in the case of covid neuropathology where we expect microstructural disruption and changes in diffusion to have a multiplicative rather than additive effect to the cognitive metric[9].

2.2.5 Errors in GLMs

Exclusion of confounding variables like patient's education level is likely to introduce errors into the model.

Performance of the GLM was analysed through considering the diagnostic plots given by calling

'plot(model)' in R. The Akaike Information Criterion (AIC) was also used to assess fit of model.

In visualisation of the GLM, the 'geom_smooth' function from the 'ggplot2'[21] package was used and a 95% confidence interval plotted. This interval was used to characterise uncertainty in model fit.

2.3 Abnormality Scores

2.3.1 Gaussian Mixture Models (GMM)

A Gaussian Mixture Model (GMM) is a probabilistic model that leverages unsupervised learning to solve a clustering problem[22]. Clustering involves identifying points in a dataset that share common characteristics. A GMM solves this through assuming that data points can be generated from a mixture of a finite number of Gaussian distributions with unknown parameters. These are then found using an Expectation-Maximisation algorithm.

A GMM can be represented as a weighted sum of M component Gaussian densities.

$$p(x) = \sum_{i=1}^{M} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

Equation 12. Defining a probability density function assigning a data point weighting for each Gaussian component it belongs to. p = probability, π = weighting, $\mathcal{N}$ = Gaussian distribution and M = number of Gaussian components.

An abnormality score is defined by calculating the probability of a new data point fitting one of these Gaussian components. If a voxel has an intensity unlikely to be in the Gaussian components expected in healthy subjects, it is indicative of an abnormality[23]. A probability map can then give a spatial representation abnormality for visualisation of burdens of illness in brain scan image space.

In context of the project, when looking at the distribution of voxel intensities for the diffusion imaged data, the histogram of these values looks like a Gaussian. Hence, a GMM is a suitable method for defining abnormality thresholds per voxel.

2.3.2 Implementation of a GMM

Access to raw voxel data for abnormality calculations required use of CSD3[24] as to preserve patient anonymity.

Scripts for the GMM and abnormality probability maps were written in python [Appendix E: GMM Code]. Image data was stored in 'NifTI' files and loaded into python using the 'nibabel'[25] library. After passing an image, a histogram of voxel intensity data is formed from which three Gaussian components are fit. Curve fitting the components of the GMM was done using 'scipy.curve_fit'[26] which minimises the least squares error between data and model to find the best parameters. Initial parameter guesses were based on the histogram of voxel data and were tinkered with to ensure 'curve_fit' reached a global minimum. Once the parameters of the three Gaussian distributions were found, probability maps were created for the three components. These maps are standardised and saved as separate 'NifTi' image files.

2.3.3 Abnormality Score and Probability Maps

For an abnormality score, we calculate the difference in probability maps between a patient and the average of all the controls. To average probability maps across controls, it is necessary to first transfer the maps from subject space to template space. Subject space encompasses the individual geometry of a subject's brain. However, two diffusion images in subject space for two different subjects would be incompatible given the differing geometries of their brain. Standardisation by first converting the probability images into template space is done through use of the inbuilt 'fsl'[27] module commands 'tbss_1_preproc' and 'tbss_2_reg'.

The most abnormal voxels were identified using the 'peak_local_max' function from the 'skimage.feature'[28] module.

2.4 Programming Techniques

2.4.1 Refactorability and Efficiency

Code was written with refactorability in mind, allowing scripts to be reused on many different datasets. 'Pickle' serialisation was used for time efficiency purposes facilitating small modifications to code without

having to re-run expensive raw data reading tasks each time. For the GMM, multithreading was used to parallelise time intensive computations dealing with large scan image files. Use of the python 'Pool()' function in the multiprocessing library yielded a speedup on the magnitude of ~10^2 as measured by the 'time' package. 'Numpy'[29] and vectorisation in calculations also assisted in efficiency.

2.4.2 Accuracy and Testing

Exceptions, try/else structures, and logging were all extensively used to guarantee accuracy of outputs through stopping execution in the event of calculation inconsistencies. Testing scripts were also used with additional redundancy in the form of pen-and-paper calculations to ensure code function.

3. Results and Discussion

3.1 Identifying Significantly Changing Regions

Tests for significance of FA/MD changes in GM/WM regions between acute and longitudinal patients were successfully conducted. Hochberg adjustment found an adjusted significance threshold of 0.0038 when accounting for multiple comparisons.

All comparisons are compiled thoroughly in tables in the appendix [Appendix F: Significance Tables].

Fewer significant comparisons in MA/FD change were observed with the longitudinal dataset than that of the Newcombe study[9]. There is a 61% reduction (from 33 to 13) in observed significantly affected WM tracts when considering MD. This is likely down to a selection bias in patients involved in the study. Newcombe featured patients with severe clinical presentations of covid, requiring intubation and with one death. In contrast, the longitudinal patient dataset inherently has individuals well enough to recover and return for imaging post hospitalisation.

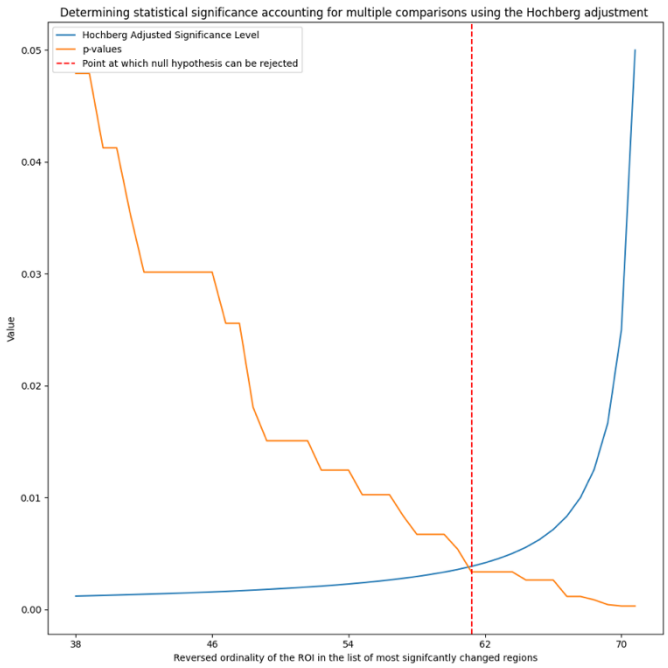

Figure 1. Hochberg adjustment in comparison of the MD of WM regions. The horizontal axis gives the reversed ordinality of the significance of these regions. There are 72 regions in the TractSeg parcellation and hence the 72nd region is that with the smallest p-value in MD change. At the 59th region considered, the p-value of the significance test for this region goes below that of the adjusted significance level from the Hochberg correction. Hence, regions 59 to 72 in an ordered list of TractSeg WM regions by largest to smallest p-value pass the significance threshold and are considered significant.

3.2 White Matter Change

Considering FA, the WM tract of the Corpus-Callosum-Posterior-Midbody changed significantly from an average value of 0.416±0.001 acutely to 0.419±0.002 in the follow-up scan. This suggests that indeed, COVID-19 caused microstructural disruption to neuronal axons which is now being repaired with time towards a restoration of proper directed axonal diffusion (the longitudinal value is within 0.6σ of the control value of 0.422±0.003). The significant nature of this change, despite only a 0.9% difference across the time frames, affirms the necessity of use of diffusion tensor imaging as opposed to conventional MRI.

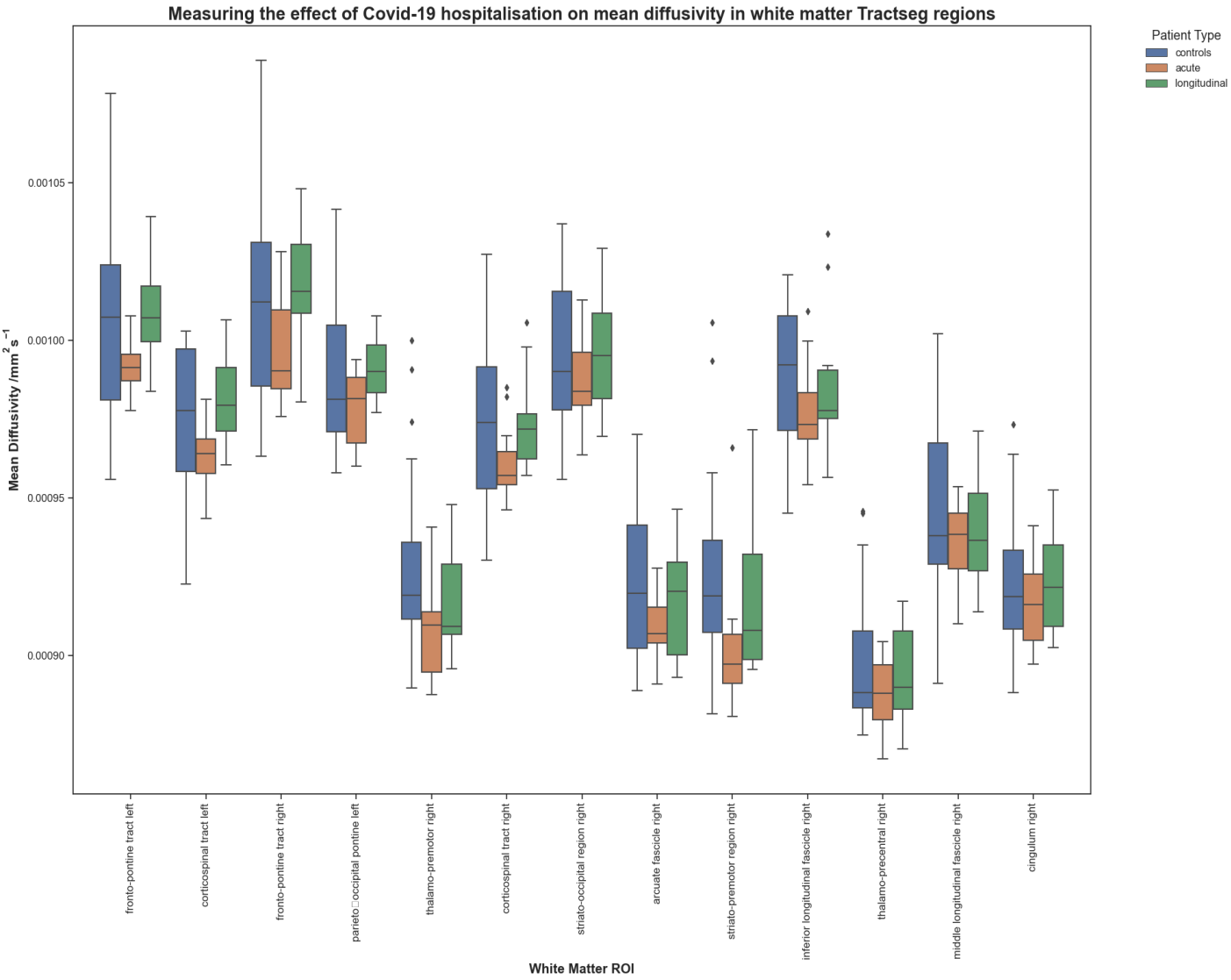

Figure 2. Mean Diffusivity in the most significantly affected white matter regions.

.

Figure 2 shows this trend in WM tract recovery with time. After sustaining an average MD change in these significant regions of 0.84±0.27% acutely, MD recovers towards baseline control levels longitudinally.

Figure 3 corroborates these findings with the change in average MD relative to the controls going from (-4.67±0.16) x10^{-6}mm^2/s to (0.109±0.17) x10^{-6}mm^2/s. The initial decrease in MD can be attributed to cytotoxic oedema[30]. Swelling of neurons restricts the extracellular space in these WM tracts, limiting the total diffusion of water. This happens acutely when the body's immune response is highly active. Restoration to the control value is hence seen due to a reduction in inflammation, and remyelination/ repair of axonal damage. The fact that the median patient in the acute timeframe charts to a similar position in the longitudinal case suggests that this pattern of improvement is not an effect of averaging across patients but is indeed an accurate representation of individual patient neuropathology.

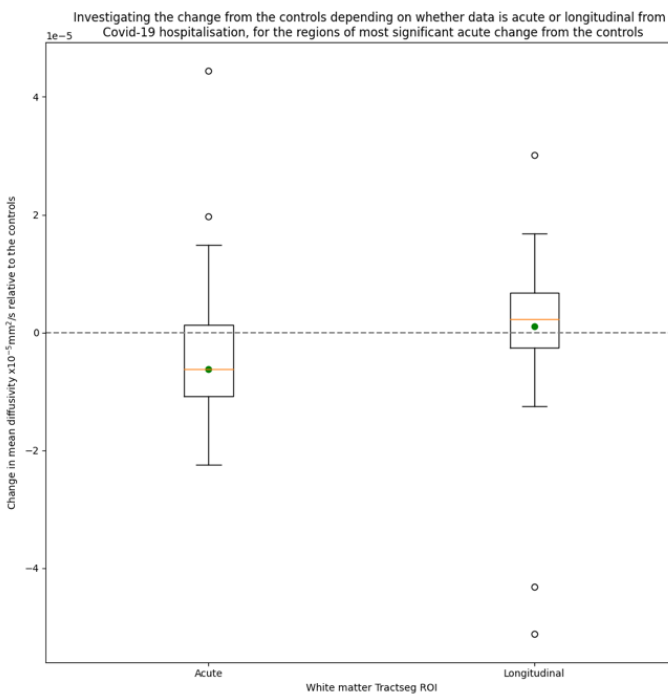

Figure 3. Change in MD values against the mean from the normative control dataset. The orange line represents the median. The green dot is the value for the patient with the median entry in the acute change and is plotted to track the individual behaviour of change across timeframes.

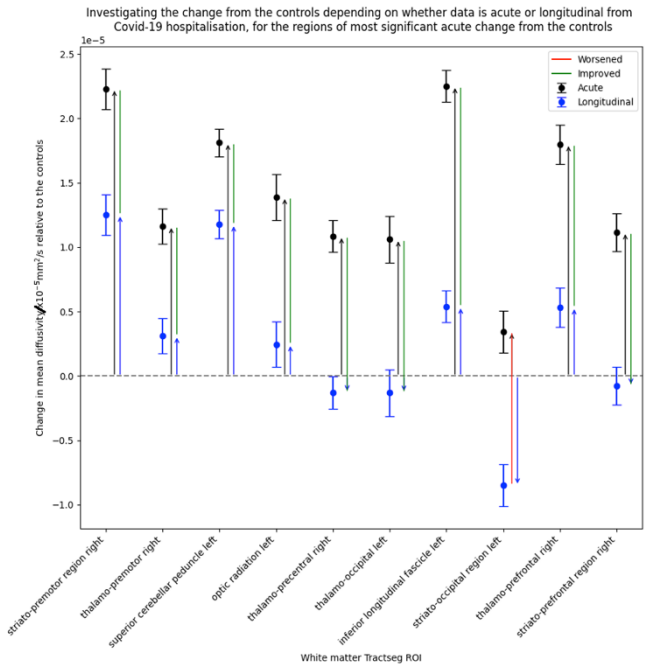

Figure 4. Restoration of WM Mean Diffusivity. The arrows show the difference in MD value from the baseline of the mean of the control dataset. A shorter blue arrow shows that longitudinally, the MD value in these regions is closer to the control value. 12 out of 13 regions showed improvement.

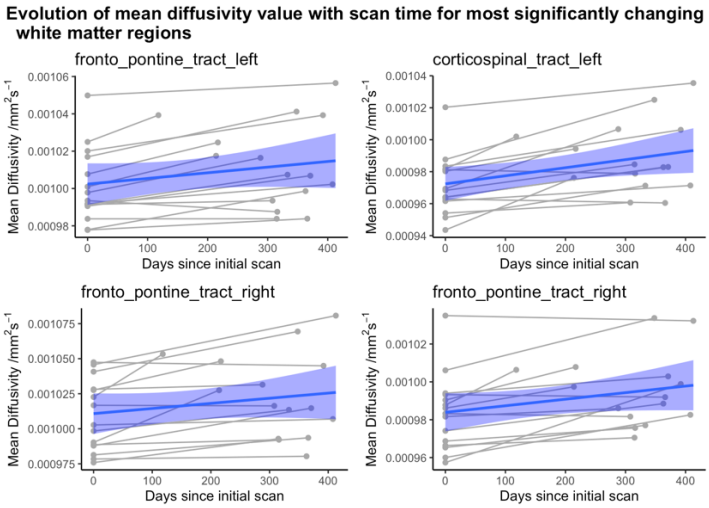

Figure 5. Time Evolution of White Matter Mean Diffusivity. The plots show the change in MD in days since the initial scan for the top 4 most significant regions. Increases in all 4 regions are witnessed.

Figures 4 and 5 further show this clear picture of WM restoration with time, implying the legitimacy of initial COVID-19 induced damage.

3.3 GM Change

In contrast, the long-term response of GM to COVID-19 viral infection shows a more complicated picture. Newcombe et Al.[9] shows an increase in MD in frontal, temporal, parietal and occipital cortices consistent with vasogenic oedema. This is mimicked in the behaviour of change in the acute dataset relative to the controls as seen in Figure 6. However, the longitudinal scans show a varied response to this increase. In contrast to the restoration seen in WM, GM regions show heterogenous activity with time. The varied and more severe response in GM is likely due to significant regional variability in response to damage[31]. Some regions may have sustained irreversible damage and hence show persistent changes to MD like the right frontal lobe.

Curiously, the right occipital lobe, relative to the controls, undergoes an average 1.2±0.6% MD increase acutely but then a 4.3±0.6% decrease longitudinally. This overshooting in correction can be explained by neuroplasticity and compensatory changes. Healing of the region has likely led to synaptic remodelling and changes in connectivity that has adjusted the baseline MD value[32].

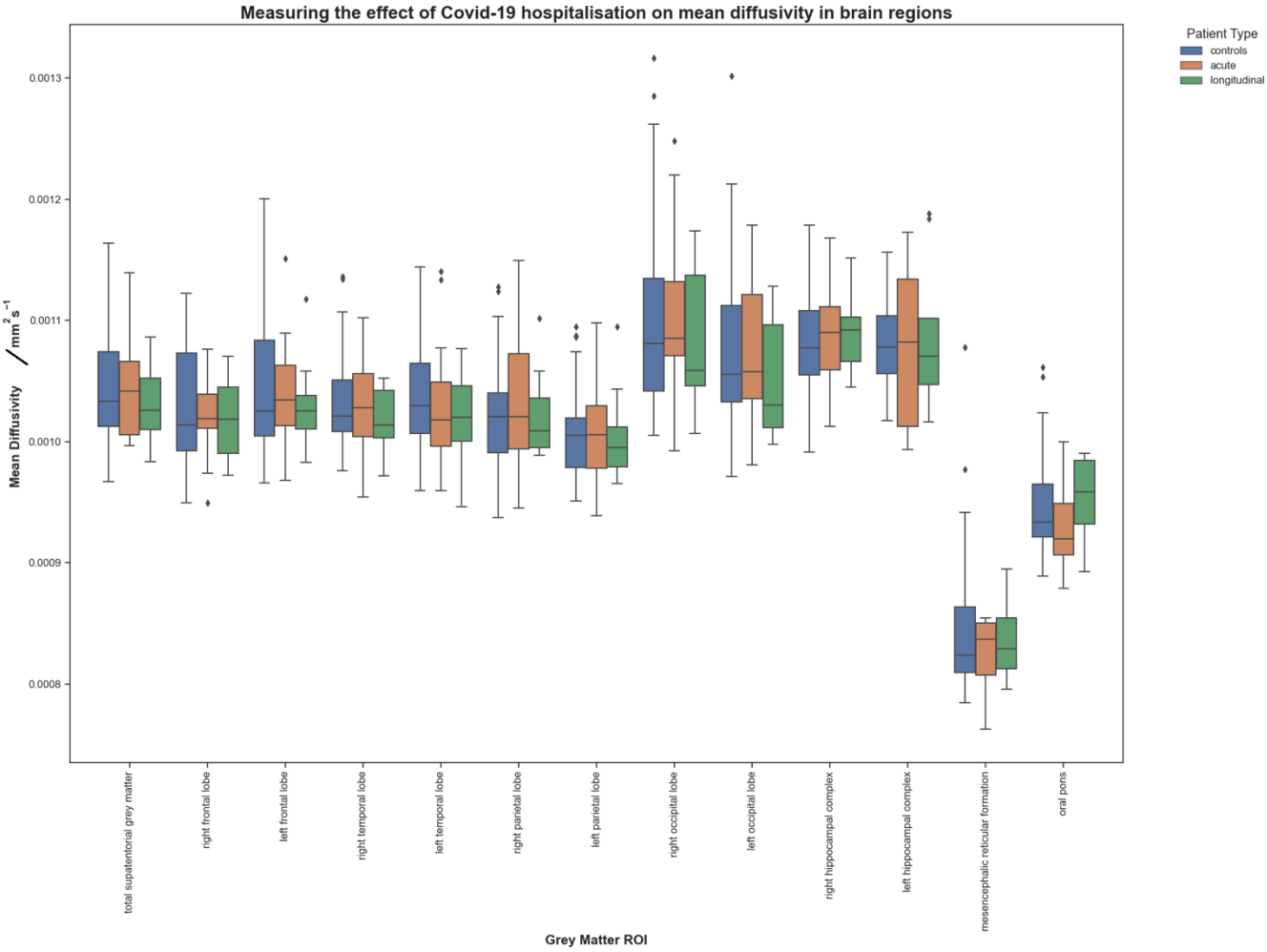

Figure 6. Mean Diffusivity in the most significantly affected grey matter regions.

The difference in observed response between WM and GM is likely due to distinct structural characteristics. GM mainly consists of neuronal cell bodies and lacks myelinated axons meaning that it is denser and more susceptible to damage from inflammation. GM is also more involved in higher order functions and has a more complicated compensatory response than the comparatively simple remyelination of WM following viral infection[33].

3.4 Lateral Response

A strange lateralisation phenomenon was observed in statistically significant MD changing WM regions. Figure 7 shows an apparent bias towards regions in the right side of the brain being more significantly affected by COVID-19 viral infection. To gauge whether this lateralisation was an anomaly or an authentic observation of the dataset, the two regions with right and left pairs, that being the Fronto-Pontine and Corticospinal tracts, were investigated further.

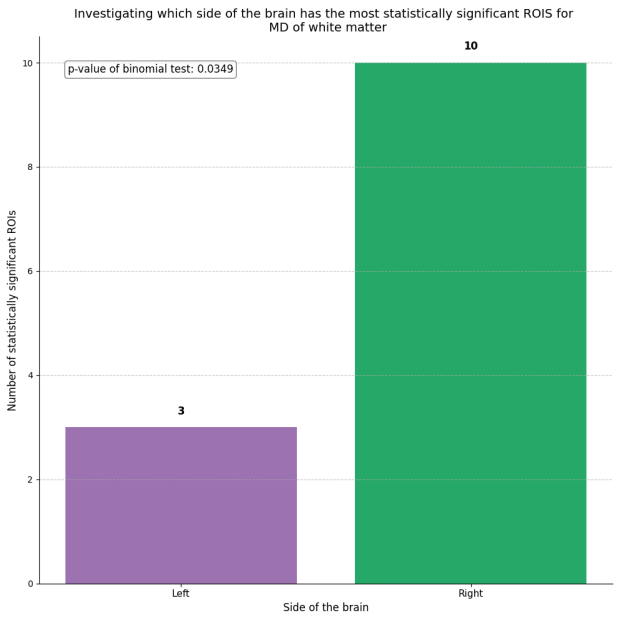

Figure 7. Lateralisation in WM mean diffusivity response. 10 out of the 13 most affected WM regions were on the right side of the brain. p-value for such an imbalance in 0.0349, significant given a 0.05 threshold.

Looking closer at the Fronto-Pontine tract, figure 8 shows a significant difference of (11±2) $x10^{-6}mm^2/s$ with a p-value of 0.003 between left and right MD values for the longitudinal scan dataset. However, figure 9 shows that when looking at the change in MD from acute to longitudinal scans, the difference in value between left and right is only (3.4±0.4) $x10^{-6}mm^2/s$ and not significant. This suggests that there is indeed a lateral response at the point of viral infection and not because of the recovery process. The right side of the brain appears more susceptible to damage from COVID-19 neuropathology. Discussing this with clinicians, this lateralisation phenomenon is indeed observed in other studies and requires further study to reach an explanation[9].

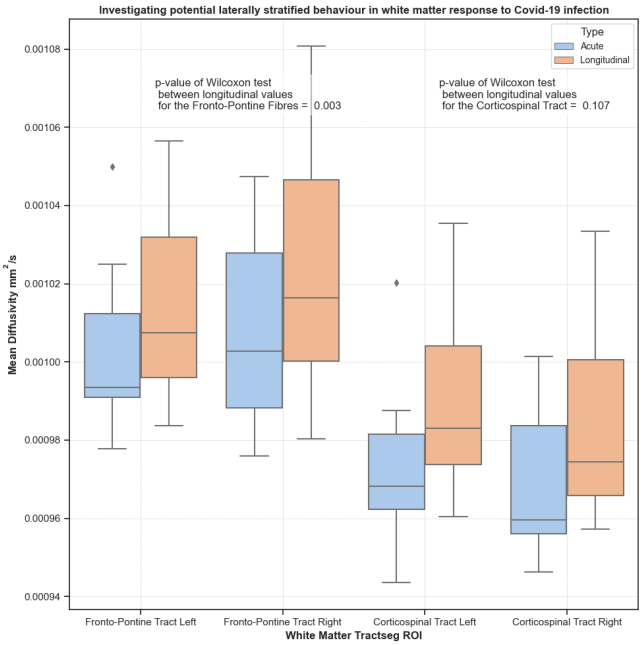

Figure 8. Lateralisation of response. Mean Diffusivity for acute and longitudinal datasets across the most affected regions with bilateral pairs. Significant difference is observed in the longitudinal MD values.

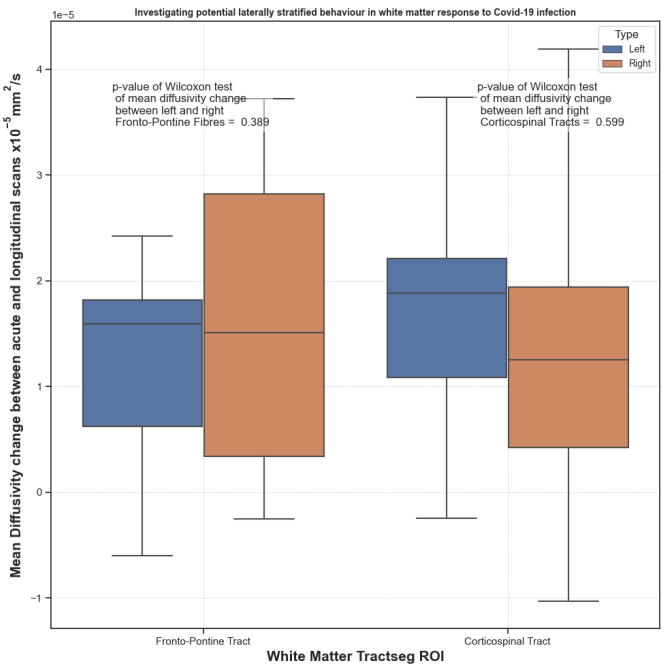

Figure 9. Lateralisation of response. Mean Diffusivity change from acute to longitudinal scans. No significance in this change between left and right suggests lateral response happens during the viral infection.

3.5 Verbal Reasoning as GLM Response Variable

A GLM was fit for verbal reasoning score against the FA value of significantly changed brain regions. Initial bivariate analysis between all predictors, including nuisance regressors was done. Figure 10 shows the result of this analysis. Nuisance regressors with no clear relationship like age and the date difference between scans were excluded from the model based on this analysis. Exclusion of these regressors produced a model with the lowest AIC value of 66 suggesting the best model and selection of predictors given the data.

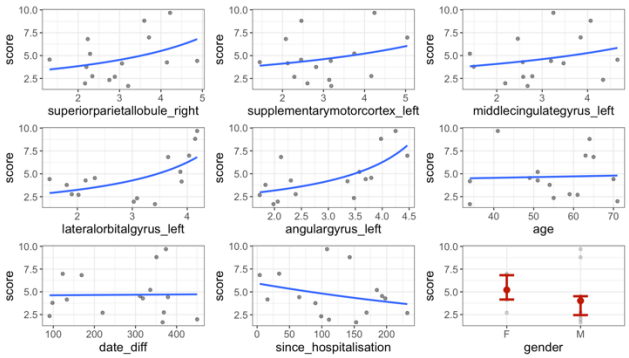

Figure 10. Bivariate analysis between verbal score and predictor variables. Flat lines indicate no significant explanatory ability and thus exclusion of that nuisance regressor is warranted for a better model fit.

Figure 11 shows the relationship between the most significant predictor of the Left-Angular-Gyrus and the verbal reasoning score. The Left-Angular-Gyrus had the largest coefficient of all regressors at (0.5±0.2) with a p-value of 0.0261. A non-linear positive regression is observed, with increase in FA corresponding to an increase in performance in verbal reasoning score. The 95% confidence interval is tight towards the regression line suggesting that FA value change in the Left-Angular-Gyrus can yield a precise estimate in the mean response of verbal score.

A good model fit is implied by the diagnostic plots in Figure 12. Despite the assumption of normally distributed residuals not being a necessity for a Gamma GLM, this happening suggests that there is no deviation from the model assumptions.

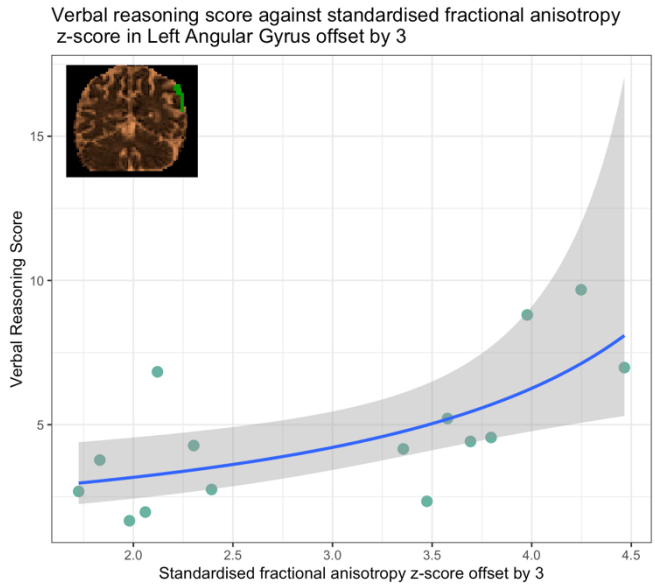

Figure 11. Regression plot between verbal reasoning score and FA z-score in the Left-Angular-Gyrus. Grey shading is the 95% confidence interval. Top left shows a brain scan image with the green contrast being location of the Left-Angular-Gyrus. Note: image is laterally flipped.

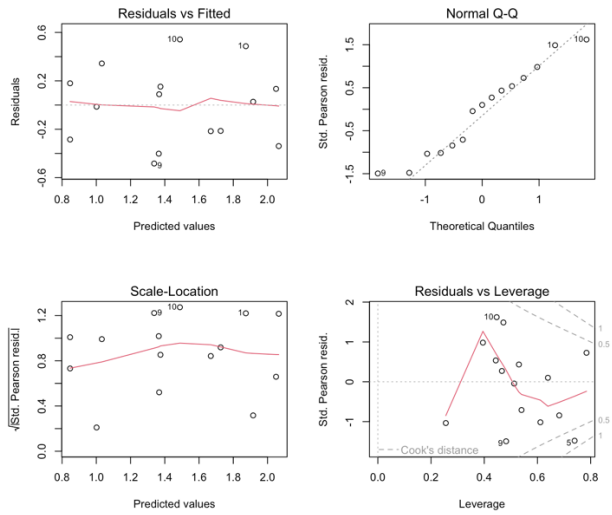

Figure 12: Verbal GLM diagnostic plots. Residuals roughly align along the diagonal in the normal 'Q-Q plot' and show no obvious pattern in the 'Residuals vs Fitted'. Datapoint 5 has greater leverage but doesn't surpass a Cook's distance of 1.

The Angular-Gyrus is a region clinically demonstrated to be responsible for speech processing and formation. The fact that higher FA and hence less damage sustained to the Angular-Gyrus predicts better performance in a verbal task is thus consistent with literature given its importance in enabling communication[34].

3.5 Spatial Reasoning as GLM Response Variable

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   0.918620   0.863358   1.064   0.3226
superiorparietallobule_right  0.174302   0.122222   1.426   0.1969
supplementarymotorcortex_left 0.173216   0.137261   1.262   0.2474
middlecingulategyrus_left     0.439770   0.158546   2.774   0.0275 *
lateralorbitalgyrus_left     -0.156827   0.146689  -1.069   0.3205
angulargyrus_left            -0.290708   0.207185  -1.403   0.2033
age                          -0.024426   0.009903  -2.467   0.0431 *
since_hospitalisation         0.003217   0.001710   1.882   0.1019
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.1613676)

    Null deviance: 4.7271  on 14  degrees of freedom
Residual deviance: 1.1619  on  7  degrees of freedom
AIC: 40.598

Number of Fisher Scoring iterations: 7
```

Figure 13. Output of the GLM fit for spatial reasoning score.

Now assessing spatial reasoning score, the Cingulate Gyrus was the best predictor of score response with a p-value of 0.0275 and a coefficient 1.5x greater in magnitude than the next most influential region. Figure 14 shows that better cellular microstructure in the Cingulate-Gyrus, as seen through FA, predicts higher spatial reasoning scores. Again, this is consistent with literature[35] whereby the Cingulate-Gyrus is attributed responsibility in higher order brain function that leads to movement.

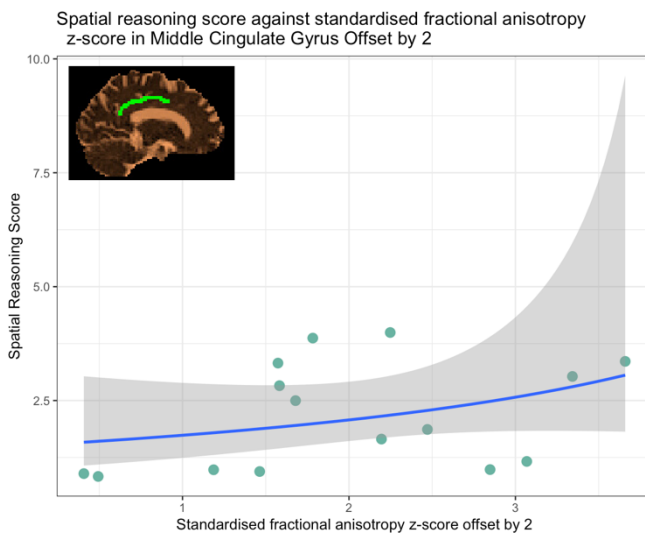

Figure 14. Regression plot between spatial reasoning score and FA z-score in the Cingulate-Gyrus. Top left shows a brain scan image with the green contrast being location of the Cingulate-Gyrus.

In both instance of spatial and verbal scores, the best predictors for response in score are FA improvements in the regions that are responsible for controlling the mechanism in which the cognitive metric is assessed. This is clinically useful as in the absence of an expensive MRI scan, cognitive tests can be used as a proxy for microstructural damage induced by a viral infection. Alternatively, monitoring reduction in the FA of these regions can indicate the severity of psychological response to COVID-19 for an intubated/unconscious patient.

Figure 13 also shows age as an effective predictor of spatial score despite being a nuisance regressor. This is due to the nature of the spatial cognitive test being response time to rapidly appearing targets[14]. Older patients are likely to possess mobility issues and have slower reflexes meaning they are more disadvantaged in test performance than by means of neuropathology alone.

The null deviance is low at 4.7271 and the residual deviance 75% lower implying a good model that outperforms a null model with no predictors. The diagnostic plots in Figure 15 suggest a good fit also. Less normality is seen in the residuals suggesting heteroscedasticity, but this is well handled by the combination of a Gamma GLM and a logarithm link function as reinforced by the low AIC of 40.

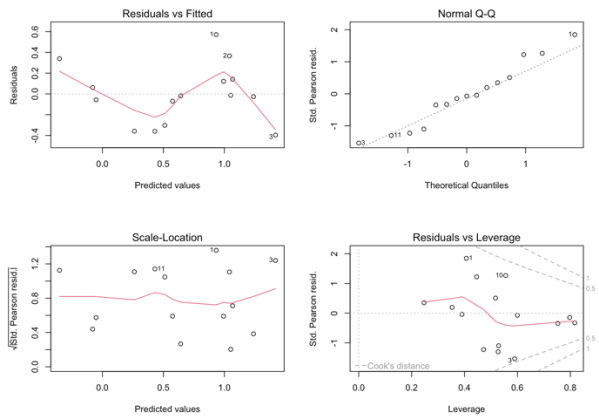

Figure 15. Spatial GLM diagnostic plots.

Overall, it is seen that the impact on cognitive metric performance that COVID-19 has on patients is indeed reflected in the diffusion imaging of affected brain regions. These regions form good predictors of the variance in the score, and this lends itself to clinical application in assessing symptomatic response of such virally induced neuropathology from imaging alone.

3.6 GMM Fit Problems

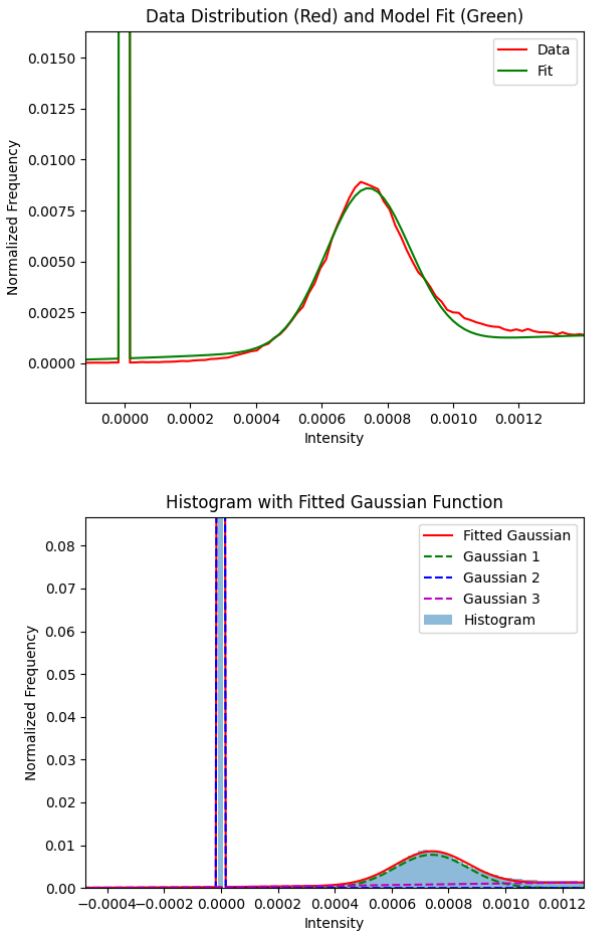

Figure 16. GMM Fit. top) Data distribution vs fit. The red shows distribution of voxel intensities and the green the Gaussian fit to model this. Note: data line is jagged due to bin digitisation error. bottom) Fitted Gaussian Components. The collapse of a diffusion brain image into a histogram of intensity values per voxel from which, 3 Gaussian components were fit.

Fitting the GMM faced issues. Figures 16 show a mismatch in the fit of the model and data for one patient.

In processing of the image, a finite number of intensity bins must be used, this leads to some digitisation error in formation of the image histogram meaning that a perfect Gaussian can't be exactly fit. Further, it is evident that more than 3 gaussian components are needed to capture image complexity for some patients.

A mismatch also implies that 'curve_fit' identified a local optimum as opposed to a global one. The peak in frequency at 0 is due to the background of the image (voxels in the scan with no brain and thus no diffusion). This was ignored with a mask when fitting probability maps.

Two patients had scan images that failed to have a GMM fit: the 'curve_fit' function failed to identify any optima within a computationally reasonable number of iterations. This is likely due to problems with initialisation. The same initial parameter calculation was done irrespective of whether the image was for a control or for a patient with severe neuropathology. An individual initialisation process for each scan image would lead to a better fit.

3.7 Probability Maps

In Figure 17, the third Gaussian component abnormality map shows substantial yellow in the central part of the brain near the brainstem. Scores here exceed 0.8 suggesting that MD values in these voxels are considerably different to those expected in the control dataset. Indeed, the brainstem is known to be a region especially sensitive to COVID-19 induced neural dysfunction[36] and this heightened abnormality is corroborated in the analyses conducted at the start of the project where 9 of the 13 significantly affected WM tracts in the MD case were in the brainstem. The yellow horn-looking structures are the Corticospinal tracts which along with the Fronto-Pontine tract were the only pair of right and left tracts to show significant MD change from acute to longitudinal scans. Their bright yellow appearance here reinforces that even on an individual voxel basis, these regions show damage.

The voxel with the greatest abnormality was that located at coordinates (137,147,50), a voxel associated the Inferior-Longitudinal-Fascicle. This region was one with a significant change in MD previously calculated with a p-value of 0.0034 and a change of $(8.5\pm0.9)\times10^{-5}$ mm^2/s [Appendix F: Significance Tables]. The voxel had an abnormality score of 0.92. Again, this substantiates the effectiveness of the GMM and use of Gaussian maps for abnormality prediction. There is good agreement between the affected regions identified through analysis of FA/MD changes as well as abnormal regions identified through probability maps.

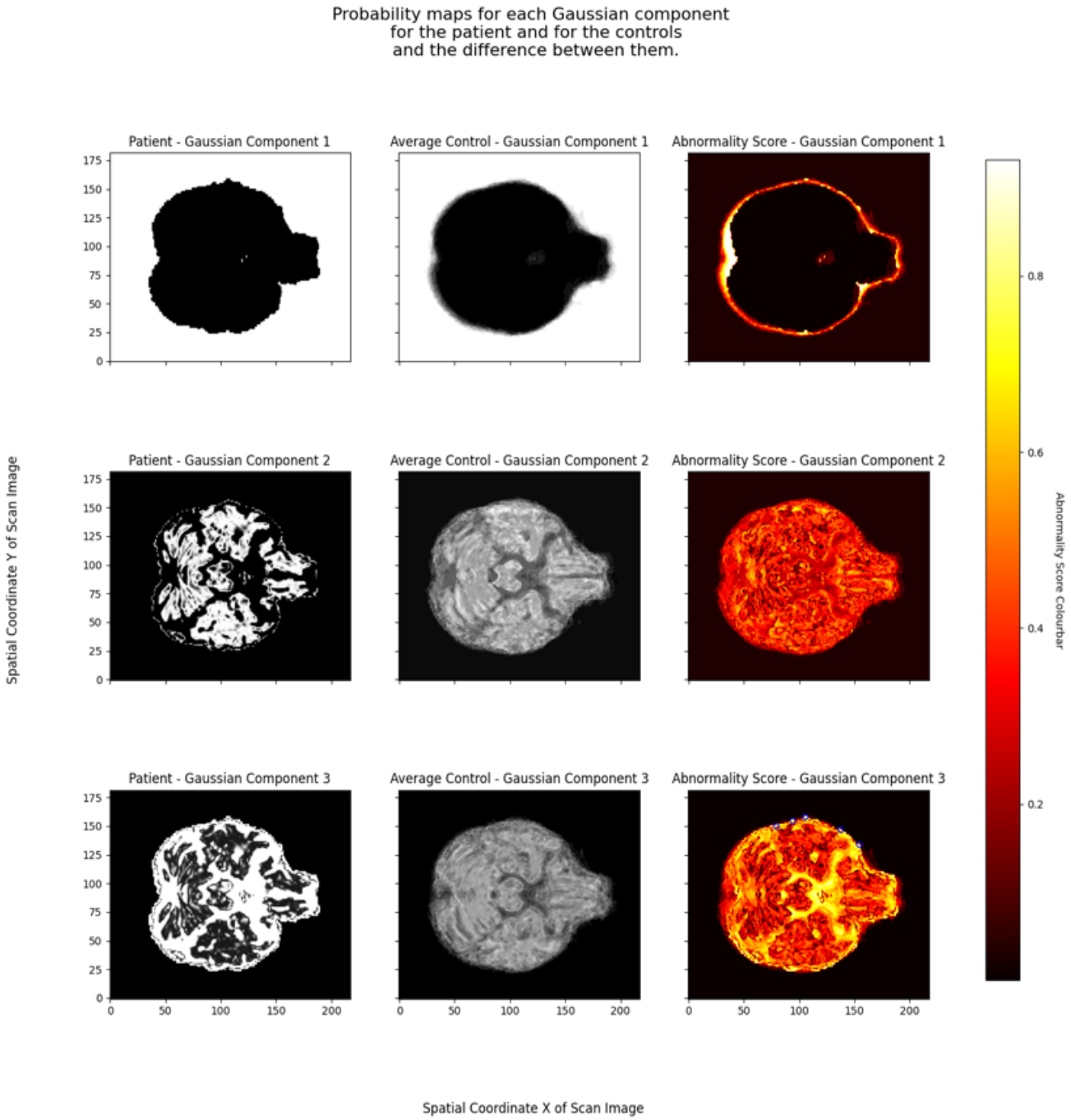

Figure 17. Abnormality Score Probability Maps. Taken for a COVID-19 patient at the z=50 slice of the brain. Description: The first column features maps representing the spatial distribution of the three Gaussian components in the patient's brain image. The second column is the average of the probability maps of the control group. The third column shows the absolute difference in the two probability maps. The brighter the colour, the more abnormal the voxel as it suggests an intensity value with a low probability density of being found in the control model.

The Inferior-Longitudinal-Fascicle being abnormal is congruent with the most affected metrics of COVID-19 examined by Hampshire et Al.[14] The tract connects the occipital and temporal lobe and abnormalities have been associated with deficits in visual processing and language comprehension in dementia patients[37].

3.7 Abnormality as a Predictor Variable

Total abnormality score was then used as a regressor in the previous GLM for spatial score. Figure 18 shows the expected relationship of a declining spatial score with greater total abnormality of the brain. Indeed, abnormality score had a coefficient 3.13x greater than the previous best predictor of FA change in the Cingulate Gyrus.

	Coefficient	Std. Error	p-value
Cingulate-Gyrus	4.8	0.9	0.0189
Abnormality	-15	3	0.0058

Table 1. Comparing the coefficients and p-values from the spatial score GLM with abnormality score now included as a regressor.

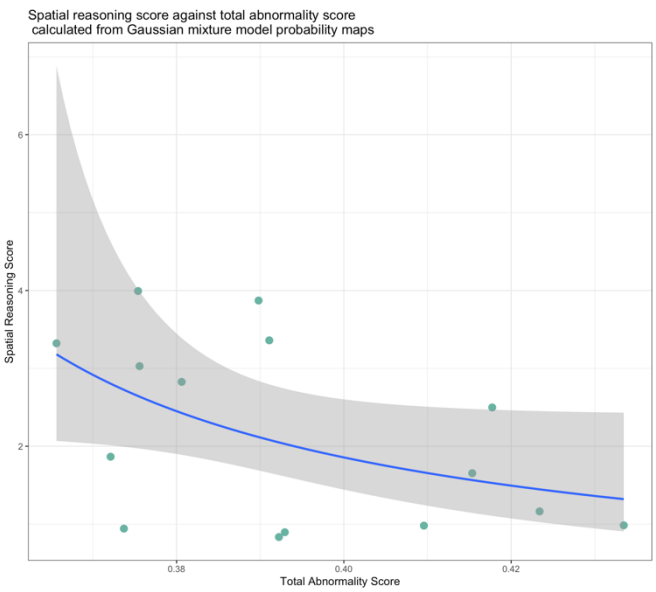

Figure 18. Spatial GLM with abnormality score now included as a regressor. Regression between spatial reasoning score and total abnormality score shows expected negative relationship.

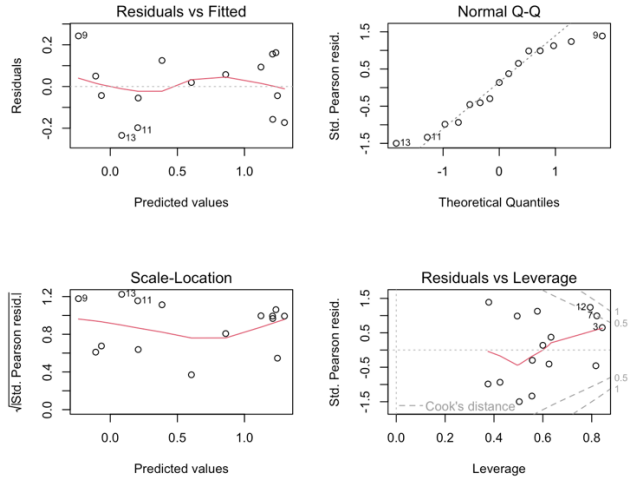

Figure 19. Diagnostic plots for spatial GLM with abnormality score.

Inclusion of abnormality score as a regressor lowered the AIC by 45% and the residual deviance by 74%. Inclusion of abnormality thus considerably improves the ability for the model to predict spatial score. The diagnostic plots also show greater normality in the behaviour between residuals and predicted values, reinforcing a better model.

Spatial Score GLM	AIC	Null Deviance	Residual Deviance
Before	40.598	4.7271	1.1619
Abnormality Score Included	22.364	4.72707	0.30442

Table 2. Comparing the AIC and deviances from the spatial GLM with abnormality score now included as a regressor.

Abnormality score being a superior predictor of cognitive metrics motivates their use to define a burden of illness. Abnormality in individual voxels allows for finer change in brain structure to be seen that may not be apparent after the smoothing effects of an FA/MD mean across the brain region. Hence, abnormality scores in voxels can be used to assess severity of illness post COVID-19 infection.

4. Conclusions

4.1 Overview

An investigation was conducted into COVID-19 neuropathology and dMRI. Longitudinally, WM regions underwent a restoration in MD/FA value but remained deviant from controls. GM showed a heterogenous response of persistent damage owing to structural susceptibility to inflammation. A right-side bias in brain damage was seen. Diffusion in affected regions were then effectively used as regressors in a GLM for prediction of cognitive metric scores. Abnormality scores were then found using voxel-based GMM probability maps. Greatest abnormality was seen in the brainstem, showing good agreement with literature-based expectation. Abnormality served to be a better predictor in the spatial score GLM, with a 45% reduction in AIC following inclusion. This validates the practicality of defining a burden of illness, using voxel-based abnormality as an early indicator of COVID-19 cognitive dysfunction.

4.2 Improvements

Predictive validation could've been leveraged with a larger dataset by enabling the creation of a training set and a test set. GMM and GLM performance could then be evaluated on how the models performed on unseen data. This would warrant greater statistical power in selecting between models and identifying the best brain region predictors for metric scores.

Use of more powerful GPU-clusters on the CSD3 would enable a GMM fit with more Gaussian components, individual starting guess parameter initialisation and more timesteps for execution of the Expectation-Maximisation algorithm. This would ensure an accurate fit with correct location of the global optimum, avoiding a reduction in efficacy in abnormality calculation due to discarded data.

Further investigation of the lateralisation phenomenon is needed. Forming a GLM with relevant data like the handedness of the patient or the severity of their COVID-19 case could shed light on the reasoning behind a right-side bias to affected region.

Finally, an error associated with voxel FA/MD intensity is achievable given repeat scans. Error analysis of abnormality scores could hence be conducted leading to a more rigorous burden of illness.

References

[1] Gibby. Basic principles of magnetic resonance imaging. Neurosurgery clinics of North America. 2005 Jan;16(1):1- 64.

[2] Pooley. AAPM/RSNA physics tutorial for residents: fundamental physics of MR imaging. Radiographics. 2005 Jul-Aug;25(4):1087-99.

[3] Stejskal EO, Tanner JE (1 January 1965). "Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-Dependent Field Gradient". The Journal of Chemical Physics. 42 (1): 288–292. Bibcode:1965JChPh..42..288S. doi:10.1063/1.1695690.

[4] Luypaert, Boujraf. Diffusion and perfusion MRI: basic physics. European journal of radiology. 2001 Apr;38(1):19-27.

[5] Bashir U, Baba Y. Diffusion-weighted imaging, 2012. URL https://radiopaedia.org/ articles/16718.

[6] Hagmann, Jonasson. Understanding diffusion MR imaging techniques: from scalar diffusion-weighted imaging to diffusion tensor imaging and beyond. Radiographics. 2006 Oct;26 Suppl 1:S205-23

[7] Soares JM, Marques P, Alves V, Sousa N. A Hitchhiker's Guide to Diffusion Tensor Imaging. Front. Neurosci., 7(31), 2013.

[8] O'Donnell, L.J. and Westin, C.-F. (2011) 'An introduction to diffusion tensor image analysis', Neurosurgery Clinics of North America, 22(2), pp. 185–196. doi:10.1016/j.nec.2010.12.004.

[9] Newcombe, V.F. et al. (2020) 'Neuroanatomical substrates of generalized brain dysfunction in covid-19', Intensive Care Medicine, 47(1), pp. 116–118. doi:10.1007/s00134-020-06241-w.

[10] Eickhoff, S.B., Yeo, B.T.T. & Genon, S. Imaging-based parcellations of the human brain. Nat Rev Neurosci 19, 672–686 (2018). https://doi.org/10.1038/s41583-018-0071-7

[11] Ledigchr (no date) LEDIGCHR/Malpem: Malpem whole-brain segmentation framework, GitHub. Available at: https://github.com/ledigchr/MALPEM (Accessed: 14 May 2023).

[12] Wasserthal, J., Neher, P. and Maier-Hein, K.H. (2018) 'Tractseg - fast and accurate white matter tract segmentation', NeuroImage, 183, pp. 239–253. doi:10.1016/j.neuroimage.2018.07.070.

[13] Needham. Brain injury in COVID-19 is associated with dysregulated innate and adaptive immune response. Cambridge NIHR Clinical Research Facility.

[14] Hampshire, A. Multivariate profile and acute-phase correlates of cognitive deficits in a COVID-19 hospitalised cohort. eClinicalMedicine, Volume 47, May 2022, 101417

[15] Coles, J.P. et al. (2004) 'Defining ischemic burden after traumatic brain injury using O15 pet imaging of Cerebral Physiology', Journal of Cerebral Blood Flow & Metabolism, 24(2), pp. 191–201. doi:10.1097/01.wcb.0000100045.07481.de.

[16] Kuperman V (2000). Magnetic Resonance Imaging: Physical Principles and Applications. Academic Press

[17] The pandas development team. (2023). pandas-dev/pandas: Pandas (v2.0.1). Zenodo. https://doi.org/10.5281/zenodo.7857418
[18] Coles, J.P. et al. (2004) 'Defining ischemic burden after traumatic brain injury using O15 pet imaging of Cerebral Physiology', Journal of Cerebral Blood Flow & Metabolism, 24(2), pp. 191–201. doi:10.1097/01.wcb.0000100045.07481.de.
[19] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics.
[20] Nelder, J. A., & Wedderburn, R. W. (1972). Generalized Linear Models. Journal of the Royal Statistical Society: Series A (General), 135(3), 370–384.
[21] Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.
[22] Fraley, C. and Raftery, A.E., 2002. Model-Based Clustering, Discriminant Analysis, and Density Estimation. Journal of the American Statistical Association, 97(458), pp.611–631. DOI: 10.1198/016214502760047131.
[23] Bishop, C.M., 2006. Pattern Recognition and Machine Learning. New York: Springer. DOI: 10.5555/1162264.
[24] Research Computing Services HPC documentation¶ (no date) Research Computing Services HPC Documentation - CSD3 1.0 documentation. Available at: https://docs.hpc.cam.ac.uk/hpc/# (Accessed: 14 May 2023).
[25] Brett, M., Hanke, M., Côté, M., Markiewicz, C. J., Côté, M. A., & Whitcher, B. (2021). NIPY/NiBabel: Release 3.2.1. Zenodo. https://doi.org/10.5281/zenodo.2652411
[26] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, 17(3), 261-272. https://doi.org/10.1038/s41592-019-0686-2
[27] FMRIB Analysis Group, University of Oxford. (n.d.). FMRIB Software Library (FSL). Retrieved from https://fsl.fmrib.ox.ac.uk/
[28] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... & Yu, T. (2014). scikit-image: image processing in Python. PeerJ, 2, e453. https://doi.org/10.7717/peerj.453
[29] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362.
[30] Huisman, T. A. (2003). Diffusion-weighted imaging: basic concepts and application in cerebral stroke and head trauma. European Radiology, 13(10), 2283-2297.
[31] Sykova, E., & Nicholson, C. (2008). Diffusion in brain extracellular space. Physiological Reviews, 88(4), 1277-1340. https://doi.org/10.1152/physrev.00027.2007
[32] Klintsova, A.Y. and Greenough, W.T., 1999. Synaptic plasticity in cortical systems. Current opinion in neurobiology, 9(2), pp.203-208.
[33] Franklin, R.J. and Ffrench-Constant, C., 2008. Remyelination in the CNS: from biology to therapy. Nature Reviews Neuroscience, 9(11), pp.839-855.
[34] John, Hall (2010). Guyton and Hall Textbook of Medical Physiology. Saunders. p. 699. ISBN 978-1416045748.
[35] Hayden, B. Y.; Platt, M. L. (2010). "Neurons in Anterior Cingulate Cortex Multiplex Information about Reward and Action". Journal of Neuroscience. 30 (9): 3339–3346. doi:10.1523/JNEUROSCI.4874-09.2010. PMC 2847481. PMID 20203193.
[36] Yong, S.J., 2021. Persistent brainstem dysfunction in long-COVID: a hypothesis. ACS chemical neuroscience, 12(4), pp.573-580.
[37] Shin, J. et al. (2019) 'Inferior longitudinal fasciculus' role in visual processing and language comprehension: A combined Meg-DTI Study', Frontiers in Neuroscience, 13. doi:10.3389/fnins.2019.00875.

Acknowledgements

I would like to especially thank my supervisor Dr Marta Correia for all her guidance and wisdom in assisting me with the undertaking of this project. I could not have done without her explanations and demonstrations.

Appendices

The appendices contain supplementary information to the report. Code and scripts included here are the highest-level and most important ones to achieve the analysis conducted, further scripts are included in the .zip folder accompanying the write-up submission. Further plots and tables are also included in the .zip folder.

Appendix A: Further Explanation of MRI

Appendix A gives some further insight into precession characteristics of a proton after it is truck by a radiofrequency pulse as part of MRI.

A.1 Radiation and Relaxation Times

Precession of the protons exhibits a peculiar quality that is exploited for imaging: magnetic resonance. Exposure of the individual protons to radio frequency radiation at the Larmor frequency causes the proton to absorb this radiation and fall out of alignment with the external magnetic field by jumping to a higher energy state. Eventually these protons will relax and return to lower energy states by re-releasing radiation. This re-released radiation can be detected as an induced signal in a coil to yield useful information.

Having applied a RF pulse, the time it takes a proton to return to its original spin state after re- emitting radiation is called the spin-lattice relaxation time and is characterised by the time constant T1[1]. This time encompasses the proton falling from the excited state back to its ground state of alignment with the external magnetic field. In other words, excitation results in a change in the longitudinal magnetisation direction and eventually after re-emission of radiation, this decays back into alignment with the external magnetic field.

$$M_z = M_0\left(1 - e^{-\frac{t}{T_1}}\right)$$

Equation 1A. T1 Relaxation Time. M_Z = the longitudinal magnetisation, M_0 = initial longitudinal magnetisation which should be aligned with B_0, t = time after radiation emission.

Similarly, transverse magnetisation decay also occurs because application of the RF signal to the protons puts them all into phase but re-emission of the radiation results in a loss of phase coherence and a signal decay with time constant T2.

$$M_{xy} = M_{xy0}e^{-\frac{t}{T_2}}$$

Equation 2A. T2 Relaxation Time, M_{xy} = the transverse magnetisation and M_{xy0} = the maximum transverse magnetisation in phase.

T1 and T2 are tissue properties which when measured indirectly relate to proton density and hence structure of a hydrogen containing object being imaged. The reason hydrogen is investigated instead of other spin-half particles that show precession in an external magnetic field is that hydrogen is naturally abundant in biological organisms and is particularly contained in water and fat. T1 and T2 can be used to indicate whether organisms have more or less water or fat content which provides enough contrast to image the body.

There are two key parameters of an MRI imaging sequence. TR (repetition time) determines how longitudinal magnetisation recovers between each radio frequency pulse. The longer the time required to recover to a net longitudinal magnetisation vector parallel to the external magnetic field, the lower the hydrogen density. The other technique is TE (echo time) which measures the time between the radio frequency pulse and the peak of the induced

signal from the re-emission of the radiation. Protons in a fluid or not constrained by structures like axons remain in phase longer and hence there will be a shorter time to peak of the induced signal[2].

A.2 Slicing and Imaging

For imaging of a large hydrogen containing entity like a body, we must have spatial localisation of measurements. This is achieved through use of magnetic gradients. If we just used one magnetic field magnitude throughout the body, every hydrogen nucleus in every section of the body would have the same Larmor frequency and hence all regions of the body would contribute to T1 and T2. However, with a magnetic gradient, there will be different Larmor frequencies across the body and hence it is possible to image slice across the body by using different Larmor frequencies of RF radiation depending on which section of the body is to be considered[3]. Protons not in the slice considered will not get excited since their Larmor frequency will not match that of the RF pulse used. A magnetic gradient can be achieved through placement of additional electrical coils in the MRI machine.

Thus, using the fact that protons precess in an external magnetic field at a certain Larmor frequency. And that they can absorb a RF pulse at this Larmor frequency and re emit it to undergo de excitation. Properties associated with the de- excitation of the protons can give invaluable insights into proton density and location of protons in certain biological microstructures, both of which constitute the cornerstone of magnetic resonance imaging.

Appendix A References

[1] Haacke EM, Brown RF, Thompson M, Venkatesan R (1999). Magnetic resonance imaging: Physical principles and sequence design. New York: J. Wiley & Sons
[2] Kuperman V (2000). Magnetic Resonance Imaging: Physical Principles and Applications. Academic Press
[3] Pooley. AAPM/RSNA physics tutorial for residents: fundamental physics of MR imaging. Radiographics. 2005 Jul-Aug;25(4):1087-99.

Appendix B: Table 1

Table 1 shows the characteristics and profile of the patients involved in the study. There were a total of 14 patients considered.

Table 1. Subject Characteristics

Patient	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Age (years)	41	64	34	59	49	51	51	61
Sex	M	M	M	M	M	F	M	F
Scanning regime	Multishell	Multishell	Multishell	Multishell	Multishell	Multishell	Multishell	Multishell
Admission date	12/05/2020	04/05/2020	30/03/2020	24/03/2020	03/04/2020	31/03/2020	22/03/2020	31/03/2020
Admission to first imaging (days)	108	143	153	169	193	185	198	232
Days between acute and longitudinal scan	374	351	366	369	311	336	318	220

Patient	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Age (years)	63	70	54	55	71	34	65
Sex	F	M	M	M	M	F	F
Scanning regime	Singleshell	Singleshell	Singleshell	Singleshell	Singleshell	Singleshell	Singleshell
Admission date	09/05/2020	08/04/2020	04/04/2020	05/04/2020	20/03/2020	08/07/2020	26/05/2020
Admission to first imaging (days)	34	65	90	99	111	16	4
Days between acute and longitudinal scan	123	379	98	91	450	133	169

Appendix C: Custom Functions Code

Appendix C gives the script containing high-level data handling custom functions that were used to remove data inconsistencies and put it into the correct format for analysis. All other scripts that handled data or conducted analysis called from functions present in this script.

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```python
# sheet function.py
import os
import numpy as np
from matplotlib import pyplot as plt
from segmentation_dicts.MALPEMDict import MALPEMDict
from segmentation_dicts.TractSegDict import TractSegDict
import pandas as pd
import seaborn as sns
from scipy import stats

"""This script comprises custom functions specifically designed for efficient data handling of dMRI and cognitive
metric data. The functions serve the following purposes:

Division of data into acute and longitudinal timeframes.
Identification and categorization of data into different scans per patient.
Separation of data into fractional anisotropy or mean diffusivity.
Removal of outliers from the data.
Calculation of statistical significance in the change between two dataframes.
Application of parcellation to the data and assignment of ROI names.
Plotting data in the form of boxplots.

These functions streamline the data handling process and facilitate effective analysis of dMRI and cognitive
metric data."""


def multiple_scans(df, number: int = 2):
    """
    Function that takes a dataframe and outputs a new dataframe with entries that have subjects that have scanned
    multiple times.
    :param df: Dataframe that takes structural mri scan data with a format corresponding to the Excel data sheet
    :param number: Minimum number of multiple scans to be included
    :return: A new dataframe with entries corresponding to multiple scans by the same subject, for different subjects.
    """
    # Creating a boolean array with markers corresponding to the position of entries from subjects who have multiple
    # scans
    multiple = (df.groupby('subject_id')['subject_id'].transform('size') >= number)
    df_multiple = df[multiple]
    # Restructuring and removing inconsequential data
    df_multiple = df_multiple.drop(['index', 'scheme', 'group'], axis=1)
    df_multiple = df_multiple.sort_values(by=['subject_id']).reset_index()
    return df_multiple


def initial_scans(df):
    """
    Function that returns a dataframe with only the first ever scans completed by a patient.
    :param df: Dataframe contating patient scan data with the scan_date column.
    :return: New dataframe with just the first ever scans performed.
    """
    # Convert scan date column to datetime
    df['scan_date'] = pd.to_datetime(df['scan_date'], format='%Y%m%d')

    # Create two new dataframes to hold earliest and latest scans
    initials = pd.DataFrame()

    # Group scans by subject ID
    groups = df.groupby('subject_id')

    # Loop over groups
    for subject_id, group in groups:
```

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```
        # Find the earliest and latest scan dates for this patient
        earliest_scan_date = group['scan_date'].min()
        initials = pd.concat([initials, group[group['scan_date'] == earliest_scan_date]])

    initials = initials.reset_index()
    initials = initials.drop(['index', 'level_0'], axis=1)

    return initials


def extreme_dataframes(df):
    """
    Function that creates a new dataframe that contains only the latest and earliest scans from each patient and not
    any intermediary scans.
    :param df: Takes a dataframe that contains only patient data with patients that have multiple scans.
    :return: A tuple of dataframes. One is a dataframe of patient scans containing their earliest scan and the other
    is a dataframe that contains their latest scan.
    """
    # Group by individual patient

    subject_grouped = df.groupby('subject_id')
    # An array of the index positions of the extreme time entries in the original data frame
    extreme_indices = []
    # Find these index positions
    for name, group in subject_grouped:
        before_index = group['scan_date'].astype(int).idxmin()
        after_index = group['scan_date'].astype(int).idxmax()
        extreme_indices.append((before_index, after_index))

        # Construct the new dataframes with the before and after scans
        before_rows = df.loc[[i[0] for i in extreme_indices]].reset_index()
        after_rows = df.loc[[i[1] for i in extreme_indices]].reset_index()

        # Remove artefacts introduced in the dataframe
        before_rows = before_rows.drop(['index', 'level_0'], axis=1)
        after_rows = after_rows.drop(['index', 'level_0'], axis=1)

    return before_rows, after_rows


def group_metric_split(df, metric):
    """
    Function that separates the existing dataframe into two dataframes with controls and patients
    respectively and isolates only entries corresponding to the specific provided metric.
    :param df: Dataframe containing the diffusion tensor imaging data in a form that aligns with
    the data spreadsheet.
    :param metric: The desired metric of diffusion tensor imaging to be considered
    :return: A tuple of a controls data frame and a patients dataframe along the provided metric.
    """
    possible_metrics = ['MD', 'FA', 'MK', 'FW']
    if metric not in possible_metrics:
        raise Exception("Invalid metric requested")
    df_controls = df.loc[(df['group'] == 'controls') & (df['metric'] == metric)].reset_index()
    df_patients = df.loc[(df['group'] == 'patients') & (df['metric'] == metric)].reset_index()
    return df_controls, df_patients

```

Page 2 of 7

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```python

def limited_timed_dataframes(df: pd.DataFrame, limit: int):
    """
    A function that takes in a dataframe containing scan data for patients with multiple scans. The function
    returns two new dataframes containing the scan data corresponding to the earliest conducted scan and the
    latest conducted scan for each patient, provided that that latest scan happened less than the provided time
    limit away from the earliest one. If no such latest scan be identified that adheres to the time limit,
    that patient's data is excluded.
    :param df: Dataframe with scan data containing 'scan_date' and 'subject_id'
    data.
    :param limit: The time limit for which longitudinal scans must have happened after the initial earliest
    acute scan.
    :return: Two new dataframes. One with the acute scan data for patients and the other for the
    longitudinal scan data. Provided the longitduinal data is within the limit.
    """
    # Convert scan date column to datetime
    df['scan_date'] = pd.to_datetime(df['scan_date'], format='%Y%m%d')

    # Create two new dataframes to hold earliest and latest scans
    acute_scans = pd.DataFrame()
    long_scans = pd.DataFrame()

    # Group scans by subject ID
    groups = df.groupby('subject_id')

    # Loop over groups
    for subject_id, group in groups:
        # Find the earliest and latest scan dates for this patient
        earliest_scan_date = group['scan_date'].min()
        latest_scan_date = group['scan_date'].max()

        # Check if the latest scan happened more than a year after the earliest scan
        if (latest_scan_date - earliest_scan_date).days > limit:
            # Find the intermediate scans that happened within one year of the earliest scan
            intermediate_scans = group[
                (group['scan_date'] > earliest_scan_date) & (group['scan_date'] < latest_scan_date) & (
                        (group['scan_date'] - earliest_scan_date).dt.days <= limit)]

            # Find the intermediate scan that happened the furthest in time from the earliest scan
            # If there are no intermediate scans and the latest scan exceeds the time limit between scans,
            # do not add any entry to either acute_scans or long_scans dataframes
            if len(intermediate_scans) > 0:
                furthest_scan = intermediate_scans.sort_values(by='scan_date', ascending=False).iloc[0]

                # Add the furthest intermediate scan to the latest scans dataframe
                long_scans = pd.concat([long_scans, pd.DataFrame(furthest_scan).transpose()])

                # Add the earliest scan to the earliest scans dataframe
                acute_scans = pd.concat([acute_scans, group[group['scan_date'] == earliest_scan_date]])

        else:
            # Add the latest scan to the latest scans dataframe
            long_scans = pd.concat([long_scans, group[group['scan_date'] == latest_scan_date]])
```

Page 3 of 7

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```
            # Add the earliest scan to the earliest scans dataframe
            acute_scans = pd.concat([acute_scans, group[group['scan_date'] == earliest_scan_date]])

    # Remove artefacts introduced in the dataframe
    long_scans = long_scans.reset_index()
    acute_scans = acute_scans.reset_index()

    acute_scans = acute_scans.drop(['index', 'level_0'], axis=1)
    long_scans = long_scans.drop(['index', 'level_0'], axis=1)

    return acute_scans, long_scans


def outlier_removal(df):
    """
    Function that removes any data points beyond 3 standard deviations of the mean.
    This is done through calculation of a z-score.
    :param df: Dataframe containing data.
    :return: New dataframe with outliers removed
    """
    df = df[(np.abs(stats.zscore(df)) < 6).all(axis=1)]
    return df


def isolate_regions(df: pd.DataFrame, segmentation: str, keepid: bool = False, keepdatediff=False):
    """
    Function that returns a dataframe with columns containing only scan data. No additional patient identifier or
    other similar supplementary data is contained.
    :param keepdatediff: True/False keep the date_diff column in the dataframe
    :param keepid: True/False keep the subject_id in the dataframe
    :param df: Dataframe containing scan data and other data.
    :param segmentation: The segmentation framework that the data is in. Can be either 'malpem', 'tractseg' or 'jhu'.
    :return: A dataframe with only scan data and the column names corresponding to brain region.
    """
    # Change input to lowercase for recgonisability
    segmentation = segmentation.lower()
    # Depending on which segmentation framework is specified, create an object containing the dictionary of that
    # framework. The dictionary contains a key: value relationship relating index to brain region.

    # If we want to preserve the subject_id column, keep a temporary copy of the original dataframe
    if keepid:
        temp_df = df

    # Determine which segmentation framework is to be used
    if segmentation == 'malpem':
        region_dict = MALPEMDict.malpem_dict
    elif segmentation == 'tractseg':
        region_dict = TractSegDict.tractseg_dict

    # If an invalid segmentation framework is passed as an input, raise an exception.
    else:
        raise Exception('Invalid brain segmentation protocol given.')

    # Isolate only columns in the provided dataframe that correspond with scan data brain regions
    df = df[list(region_dict.keys())]
    # Rename the columns with the names of the brain regions.
```

Page 4 of 7

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```
    df = df.rename(columns=region_dict)
    df.columns = map(str.lower, df.columns)

    # Reinsert subject_id column through copying it from the temporary dataframe copy, if necessary
    if keepid:
        df['subject_id'] = temp_df['subject_id']

    if keepdatediff:
        df['date_diff'] = temp_df['date_diff']

    return df


def boxplotter(dfs: list, rois: list, title: str = '',
               x_label: str = '', y_label: str = '', save_name: str = '',
               types: list = ['controls', 'acute', 'longitudinal']):
    """
    A function that takes a dataframe with columns of scan data and produces a boxplot. The function takes in the
    dataframes associated with different patient types and time nature of the scan. The default is [acute,
    longitudinal, controls] for what these dataframes correspond to. It also takes in a list of relevant regions that
    are to be plotted. It cab also take various inputs to vary the labels on the plot and the savename.
    :param dfs: List of dataframes corresponding to patient or control scan data in the acute or longitudinal timeframe.
    :param rois: List of regions of interests in the brain to be plotted.
    :param types: List detailing what the dataframes correspond to.
    :param title: Title of the plot.
    :param x_label: X label of the plot.
    :param y_label: Y label of the plot.
    :param save_name: Save name for the plot.
    :return: Shows the boxplot and saves it under the save_name accordingly.
    """
    # Checking that a dataframe has been provided for each patient type.
    if len(types) != len(dfs):
        raise Exception('Need same number of patient types as dataframes of patient types.')

    # Make rois input all lowercase
    rois = list(map(lambda x: x.lower(), rois))

    # Remove outliers from the dataframe
    # for i, type_df in enumerate(type_dfs):
    #     type_dfs[i] = outlier_removal(type_df)

    # Create an empty list to store the plot dataframes
    plot_dfs = []

    # We need to flip the table such that we have rows corresponding to each subject and column entries detailing the
    # type of scan and the region of interest, this is necessary for plotting using the seaborn library
    # Loop through each type of data and its corresponding dataframe
    for type, df in zip(types, dfs):
        # Create an empty list to store the dataframes for the same patient type and different regions
        same_type_dfs = []
        # Loop through each region of interest in the dictionary
        for region in rois:
            # Extract the data for the current region from the dataframe and convert it to a list
            temp_df = df[region].tolist()
            # Convert the list to a pandas dataframe with one column named 'value'
            temp_df = pd.DataFrame(temp_df, columns=['value'])
```

Page 5 of 7

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```
            # Add a column named 'region' to the dataframe and set all rows to the current region
            temp_df.insert(0, 'region', region)
            # Add the new dataframe to the list of dataframes for the same patient type
            same_type_dfs.append(temp_df)
        # Concatenate all dataframes for the same patient type and different regions into a single dataframe
        same_type_df = pd.concat(same_type_dfs)
        # Add a column named 'type' to the dataframe and set all rows to the current patient type
        same_type_df.insert(0, 'type', type)
        # Add the new dataframe to the list of plot dataframes
        plot_dfs.append(same_type_df)

    # Concatenate all plot dataframes into a single dataframe and reset the index
    plot_df = pd.concat(plot_dfs).reset_index()

    print(plot_df)

    sns.set(style='ticks', palette='pastel')

    # Increase the size of the figure
    fig, ax = plt.subplots(figsize=(15, 10))

    # Create the plot
    sns.boxplot(x='region', y='value', data=plot_df, hue='type', ax=ax)
    max_y = plot_df['value'].max()
    min_y = plot_df['value'].min()

    ax.set_ylim([min_y * 0.85, max_y * 1.1])

    # Add a title
    ax.set_title(title, fontsize=20, fontweight='bold')

    # Set the x-axis label
    ax.set_xlabel(x_label, fontsize=15, fontweight='bold')

    # Set the y-axis label
    ax.set_ylabel(y_label, fontsize=15, fontweight='bold')

    # Rotate the x-axis labels by 45 degrees
    labels = ax.get_xticklabels()
    ax.set_xticklabels(labels, rotation=45)

    # Add padding to the x-axis labels
    ax.margins(x=0.1)

    # Set the font size for the tick labels
    ax.tick_params(labelsize=12)

    # Add a grid to the plot
    sns.despine(offset=10, trim=True)
    ax.grid(True)

    plt.tight_layout()

    filename = os.path.basename(__file__)[:-3]
    plt.savefig('../plots/' + save_name + '.png')

    plt.show()


def most_significant_regions(df1: pd.DataFrame, df2: pd.DataFrame, related: bool, number_rois: int):
    """
    Determines the p-values for rois conducting the necessary test given if the data
```

File - /Users/leondailani/PycharmProjects/projectpartiii/sheet_functions.py

```
 is related or not.
    :param df1: Dataframe 1
    :param df2: Dataframe 2
    :param related: Boolean to inform whether the dataframes are related or not.
    :param number_rois: Number of rois that are needed.
    :return: Returns a dictionary with keys of roi names and p-values corresponding to the change in the value between
    the two dataframes
    """
    # If the two samples are related, need to perform a Signed-Rank metrics_and_atlases.
    if related:
        # Doing a basic check to see if the groups are indeed related. If related, they should have the same number of
        # entries.
        if len(df1) != len(df2):
            # Raise an exception if this is not the case
            raise Exception('These two dataframes are not of related groups as they contain different number of '
                            'entries.'
                            'The group size is not the same.')
        # Perform the Signed-Rank metrics_and_atlases. p is the p-value.
        stat, p = stats.wilcoxon(df1, df2)

    else:
        # If two samples are independent, want to perform a Wilcoxon Rank Sum Test instead
        stat, p = stats.ranksums(df1, df2)

    # Replace nan values with 1
    p = np.nan_to_num(p, nan=1)

    # Want to isolate and return the top most statistically significant regions for changes
    # First sort the regions in ascending order (smallest p-value first) and get the indexes of those
    # 15 lowest p-value regions
    most_significant_ps = np.sort(p)[:number_rois]
    most_significant_indices = np.argsort(p)[:number_rois]

    # Now want to recover what each region these indices correspond to are.
    # First need to figure out which brain segmentation framework is being used based on length of dictionaries.
    # / the number of columns in the dataframes.

    # Need to add one because malpem segmentation framework dictionaries start from a key value of 1
    # but argsort starts from an index of 0. Tractseg starts from 0, so can avoid doing this.

    number_columns = df1.shape[1]
    if number_columns == len(MALPEMDict.malpem_dict):
        get_dict_value = np.vectorize(lambda key: MALPEMDict.malpem_dict[key])
        most_significant_indices = most_significant_indices + 1

    elif number_columns == len(TractSegDict.tractseg_dict):
        get_dict_value = np.vectorize(lambda key: TractSegDict.tractseg_dict[key])

    most_significant_rois = get_dict_value(most_significant_indices)

    # Create a dictionary linking most significantly changed ROIs as keys with the corresponding value being
    # the p-value
    roi_p_dict = dict(zip(np.char.lower(most_significant_rois), most_significant_ps))

    return roi_p_dict
```

Page 7 of 7

Appendix D: GLM Code

Appendix D shows the top-level R code that conducts initial analysis on the regressors, fits the GLM and plots the bivariate relationship between the most significant predictor region and the response variable. The code looks at the verbal reasoning score. Other similar R scripts were used to fit different models for MD and other cognitive metric tasks.

File - /Users/leondailani/PycharmProjects/projectpartiii/appendix.R

```r
"""
This script looks at the relationship between several predictor variables and a verbal reasoning test score by fitting a GLM.
The predictors include the change in fractional anisotropy for brain regions and patient profile characteristics.
First I check for the distribution of the brain region values and see that they're not normal meaning a GLM is appropriate.
I then look at the bivariate analysis of all the predictors and the response to see which regressors don't have any relationship and can be excluded.
Then I fit the GLM and plot the relationship between the most significant predictor and the response.
"""

# Install required packages
install.packages("devtools")
install.packages('cowplot')
install.packages('readr')

# Load necessary libraries
require(devtools)
require(flexplot)
library('cowplot')
library(readr)

# Read data from a CSV file
d <- read_csv("Documents/Part III Project/d.csv")

# Visualizing the distribution of brain region values using the 'flexplot' function.
# The distribution of these variables is expected to be non-normal, which suggests that a Generalized Linear Model (GLM) might be appropriate.
a = flexplot(superiorparietallobule_right~1,data=d) # Superior Parietal Lobule (Right)
b = flexplot(supplementarymotorcortex_left~1,data=d) # Supplementary Motor Cortex (Left)
b1 = flexplot(middlecingulategyrus_left~1,data=d) # Middle Cingulate Gyrus (Left)
c = flexplot(lateralorbitalgyrus_left~1,data=d) # Lateral Orbital Gyrus (Left)
c1 = flexplot(angulargyrus_left~1,data=d) # Angular Gyrus (Left)
c2 = flexplot(score~1,data=d) # Score
# Plot these distributions in a grid
plot_grid(a,b,b1,c,c1,c2,nrow=2)

# Bivariate analysis of each predictor with the response variable
# This is done to identify which predictors have a relationship with the response variable.
# Different methods are used based on the nature of the predictor variable.
# 'Gamma' is used for continuous predictors and 'poisson' or 'logit' is used for discrete predictors.
a1 = flexplot(score~superiorparietallobule_right,data=d,method='Gamma',se=F,jitter=c(0,.2))
b1 = flexplot(score~supplementarymotorcortex_left,data=d,method='Gamma',se=F,jitter=c(0,.2))
c1 = flexplot(score~middlecingulategyrus_left,data=d,method='Gamma',se=F,jitter=c(0,.2))
d1 = flexplot(score~lateralorbitalgyrus_left,data=d,method='Gamma',se=F,jitter=c(0,.2))
e1 = flexplot(score~angulargyrus_left,data=d,method='Gamma',se=F,jitter=c(0,.2))
a2 = flexplot(score~age,data=d,method='poisson',se=F,jitter=c(0,.2))
b2 = flexplot(score~date_diff,data=d,method='poisson',se=F,jitter=c(0,.2))
c2 = flexplot(score~since_hospitalisation,data=d,method='poisson',se=F,jitter=c(0,.2))
d2 = flexplot(score~gender,data=d,method='logit',se=F,jitter=c(0,.2))
# Plot these relationships in a grid
plot_grid(a1,b1,c1,d1,e1,a2,b2,c2,d2,nrow=3)

# Fit the GLM model using relevant predictor variables
# The 'Gamma' family is chosen because the response variable is not normally distributed.
```

File - /Users/leondailani/PycharmProjects/projectpartiii/appendix.R

```
model <- glm(score ~ supplementarymotorcortex_left + middlecingulategyrus_left + lateralorbitalgyrus_left + angulargyrus_left  + gender, data = d, family=Gamma(link="log"))
# Show the summary statistics of the model
summary(model)
# Plot diagnostic plots of the model to visualize how well the model fits the data.
# This includes a Residuals vs Fitted plot, a Normal Q-Q plot, a Scale-Location plot (sqrt|standardized residuals| vs fitted values), and a Residuals vs Leverage plot.
par(mfrow=c(2,2))
plot(model)

# Load additional libraries for plotting
library(ggplot2)
library(effects)

# Plotting the relationship between the most significant predictor (angulargyrus_left) and the response variable (score)
# The fitted GLM is added to the plot to show the relationship between the predictor and the response.
main_plot <- ggplot(d, aes(x = angulargyrus_left, y = score)) +
  geom_point(size = 3, color = "#69b3a2") +
  theme_bw() +
  geom_smooth(method = "glm", method.args = list(family = "Gamma"))

# Print the plot with a title and labels for the x and y axes.
print(main_plot + ggtitle("Verbal reasoning score against standardised fractional anisotropy \n z-score in Left Angular Gyrus offset by 3")+labs(y="Verbal Reasoning Score", x="Standardised fractional anisotropy z-score offset by 3"))
```

Page 2 of 2

Appendix E: GMM code

Appendix E contains two scripts. One implements the Gaussian Mixture Model and generates the abnormality maps 'abnormality_mulptiple.py' and the other creates the plot (Figure 17) to visualise the abnormality scores 'abnormality_maps.py'.

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_multiple.py

```python
# abnormality_multpiple.py
"""This script implements a pipeline for processing MRI scan data to identify abnormalities by fitting a Gaussian
mixture model. It starts by loading MRI scan data from NIfTI files. Then, it computes a histogram of the data and
fits a Gaussian mixture model with three components to this histogram. The optimized parameters of the Gaussian
mixture model are used to create probability maps for each Gaussian component. These probability maps are saved as
separate NIfTI files for further analysis. The script also includes functionality for visualizing the histogram of
the MRI data and the fitted Gaussian mixture model, as well as the individual Gaussian components. This script
supports parallel processing of multiple files for increased efficiency. The results of the Gaussian mixture model
fitting are logged and can be used for further statistical analysis and comparison between different groups of scans
(e.g., patient vs. control)."""

import logging
import multiprocessing
import nibabel as nib
import numpy as np
import os
import pickle as pkl
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit


# Define the Gaussian mixture model function with three Gaussian components
def gaussian_mixture(x, A1, A2, A3, X1, X2, X3, s1, s2, s3):
    """
    Computes the value of a Gaussian mixture model with three components at given points.

    :param x: Array of points at which to evaluate the Gaussian mixture model.
    :param A1: Amplitude of the first Gaussian component.
    :param A2: Amplitude of the second Gaussian component.
    :param A3: Amplitude of the third Gaussian component.
    :param X1: Mean of the first Gaussian component.
    :param X2: Mean of the second Gaussian component.
    :param X3: Mean of the third Gaussian component.
    :param s1: Standard deviation of the first Gaussian component.
    :param s2: Standard deviation of the second Gaussian component.
    :param s3: Standard deviation of the third Gaussian component.
    :return: Array of Gaussian mixture model values at the points in `x`.
    """
    term1 = A1 * np.exp(-(x - X1) ** 2 / (2 * s1 ** 2))
    term2 = A2 * np.exp(-(x - X2) ** 2 / (2 * s2 ** 2))
    term3 = A3 * np.exp(-(x - X3) ** 2 / (2 * s3 ** 2))
    return term1 + term2 + term3


# Define a single Gaussian function
def gaussian(x, A1, X, s):
    """
    Computes the value of a Gaussian function at given points.

    :param x: Array of points at which to evaluate the Gaussian function.
    :param A1: Amplitude of the Gaussian.
    :param X: Mean of the Gaussian.
    :param s: Standard deviation of the Gaussian.
    :return: Array of Gaussian function values at the points in `x`.
    """
    return A1 * np.exp(-(x - X) ** 2 / (2 * s ** 2))


```

Page 1 of 5

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_multiple.py

```python
# Main function to fit a Gaussian mixture model to the data
def fit_gaussian_mixture(filename):
    """
    Fits a Gaussian mixture model to MRI scan data.

    :param filename: Name of the file containing MRI scan data.
    :return: Tuple containing the optimized parameters, the data points, and the corresponding histogram.
    """
    # Define expected minimum and maximum values
    min_dif = 0
    max_dif = np.inf
    # Load data from the file
    data = load_files(filename)
    # Perform the Gaussian mixture fitting and obtain the optimized parameters
    params, xdata, pbin = do_figure(data, min_dif, max_dif)
    # Save the probability maps to separate NIfTI files
    make_maps(filename, params, data)

    return params, xdata, pbin


# Load the image data from the input NIfTI file
def load_files(filename):
    """
    Loads MRI scan data from a NIfTI file.

    :param filename: Name of the NIfTI file.
    :return: Array containing the MRI scan data.
    """
    image = nib.load(filename)
    data = image.get_fdata()

    return data


# Fit the Gaussian mixture model to the histogram of the data
def do_figure(p, min_dif, max_dif):
    """
    Fits a Gaussian mixture model to the histogram of MRI scan data and plots the results.

    :param p: Array containing the MRI scan data.
    :param min_dif: Minimum expected value.
    :param max_dif: Maximum expected value.
    :return: Tuple containing the optimized parameters, the data points, and the normalized histogram.
    """
    # Calculate the histogram of the data
    pbin, edges = np.histogram(p.ravel(), bins=500)
    xdata = (edges[:-1] + edges[1:]) / 2

    # Normalize the histogram
    num_voxels = np.sum(pbin)
    pbin = pbin / num_voxels

    # Set initial parameter estimates
    X0 = np.zeros(9)

    bounds = (np.zeros(9), np.ones(9))

    X0[0] = np.max(pbin)
    X0[1] = np.max(pbin)
    X0[2] = np.max(pbin)
    X0[3] = np.sum(pbin * xdata) / np.sum(pbin)
    X0[4] = 1.1 * X0[3]
    X0[5] = 1.2 * X0[3]
```

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_multiple.py

```
    Xsq = np.sum(pbin * xdata * xdata) / np.sum(pbin)
    X0[6] = np.sqrt(Xsq - (X0[3] * X0[3]))
    X0[7] = X0[6]
    X0[8] = X0[6]

    # Perform the curve fitting using the Gaussian mixture model function
    print('Working')
    params, _ = curve_fit(gaussian_mixture, xdata, pbin, p0=X0, bounds=bounds, max_nfev=100000)

    # Plot the original histogram data in red
    plt.plot(xdata, pbin, 'r')

    # Plot the fitted Gaussian mixture model in green
    ydata = gaussian_mixture(xdata, *params)
    plt.plot(xdata, ydata, 'g')

    # Set the plot labels and title
    plt.xlabel('Intensity')
    plt.ylabel('Normalized Frequency')
    plt.title('Data Distribution (Red) and Model Fit (Green)')
    plt.legend(['Data', 'Fit'])

    return params, xdata, pbin


# Create probability maps and save them as separate NIfTI files
def make_maps(input_file, params, pin):
    """
    Creates probability maps for each Gaussian component and saves them as separate NIfTI files.

    :param input_file: Name of the file containing the original MRI scan data.
    :param params: Optimized parameters of the Gaussian mixture model.
    :param pin: Array containing the MRI scan data.
    """
    input_img = nib.load(input_file)
    base_path, name, _ = fileparts(input_file)

    # Initialize probability map array
    prob = np.zeros((*input_img.shape, 3))

    # Calculate probability maps for each Gaussian component
    for i in range(3):
        prob[..., i] = params[i] * np.exp(-((pin - params[i + 3]) / params[i + 6]) ** 2)

        # Normalize the probability maps
    sump = np.finfo(float).eps + np.sum(prob, axis=-1)

    for i in range(3):
        prob[..., i] = (prob[..., i] / sump)

    # Save the probability maps as separate NIfTI files
    for i in range(3):
        out_nii = nib.Nifti1Image(prob[..., i], input_img.affine, input_img.header)
        nib.save(out_nii, os.path.join(base_path, f"{name}_component_{i + 1}.nii.gz"))

    # Extract the file path, name, and extension from the input filename


def fileparts(filename):
    """
    Extracts the file path, name, and extension from a filename.

    :param filename: Name of the file.
```

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_multiple.py

```
    :return: Tuple containing the file path, name, and extension.
    """
    base_path, full_name = os.path.split(filename)
    name, ext = os.path.splitext(full_name)

    return base_path, name, ext


def combination_plotter(params, xdata, pbin):
    """
    Plots the histogram of MRI scan data and the fitted Gaussian mixture model, as well as the individual Gaussian components.

    :param params: Optimized parameters of the Gaussian mixture model.
    :param xdata: Array of data points.
    :param pbin: Corresponding histogram.
    """
    # Plot the histogram
    plt.bar(xdata, pbin, width=np.diff(xdata)[0], alpha=0.5, label='Histogram')

    # Plot the fitted Gaussian function
    ydata_fitted = gaussian_mixture(xdata, *params)
    plt.plot(xdata, ydata_fitted, 'r-', label='Fitted Gaussian')

    # Plot individual Gaussian peaks
    gaussian1 = gaussian(xdata, A1=params[0], X=params[3], s=params[6])
    gaussian2 = gaussian(xdata, A1=params[1], X=params[4], s=params[7])
    gaussian3 = gaussian(xdata, A1=params[2], X=params[5], s=params[8])

    plt.plot(xdata, gaussian1, 'g--', label='Gaussian 1')
    plt.plot(xdata, gaussian2, 'b--', label='Gaussian 2')
    plt.plot(xdata, gaussian3, 'm--', label='Gaussian 3')

    # Add legend and labels
    plt.legend()
    plt.xlabel('X-axis Label')
    plt.ylabel('Y-axis Label')
    plt.title('Histogram with Fitted Gaussian Function')

    # Show the plot
    plt.show()


def process_group(file_list):
    """
    Fits a Gaussian mixture model to the MRI scan data in each file of a list in parallel, handling any fitting errors.

    :param file_list: List of file names containing MRI scan data.
    :return: List of results for each file
    """
    # Use parallelisation to speed up execution time
    with multiprocessing.Pool() as pool:
        group_results = []
        for file in file_list:
            try:
                result = pool.apply_async(fit_gaussian_mixture, [file]).get()
                group_results.append(result[0])
            except RuntimeError:
                # If the fitting fails, log the failed file and move on
                logging.exception(f"Failed to fit {file}. Skipping...")
    return group_results


# Main script
if __name__ == "__main__":
    # Define the patient and control file pathnames
```

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_multiple.py

```
    patient_files = ['30292', '30293', '30294', '30295', '30297', '30302']
    control_files = ['10000', '10058', '10179', '10459', '14136', '16242', '18770', '21091', '23108', '24972', '25002',
                     '25044', '27336', '27949', '28069', '28973']

    patient_files = list(map(lambda filename: 'patients_FA_scans/' + filename + 'FA.nii.gz', patient_files))
    control_files = list(map(lambda filename: 'controls_FA_scans/' + filename + 'FA.nii.gz', control_files))

    patient_results = process_group(patient_files)
    control_results = process_group(control_files)

    # Store the Gaussian Components in a pickle file

    with open('results.pkl', 'wb') as f:
        pkl.dump(
            [control_results, patient_results], f
        )
    f.close()

```

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_prob_maps.py

```python
# abnormality_prob_maps
"""This Python script processes and visualizes MRI brain scans to detect and highlight abnormal regions in a
patient's brain compared to a control group. It calculates an abnormality score based on differences in Gaussian
Mixture Model (GMM) components between the patient and control group scans, and generates a set of plots for each GMM
component showing the patient's map, the average control map, and the abnormality score map."""

import numpy as np
import nibabel as nib
import matplotlib.pyplot as plt
from skimage.feature import peak_local_max
import matplotlib.colors as colors

# Define function to display abnormality maps
def display_maps(patient_file, control_files, slice_index, min_distance=5, num_peaks=5):
    """
        Function to display probability maps for each Gaussian component for the patient and for the controls
        and the difference between them.

        :param patient_file: str, file path to the patient's .nii.gz files
        :param control_files: list of str, file paths to the control group's .nii.gz files
        :param slice_index: int, index of the slice to be displayed from the 3D volume
        :param min_distance: int, minimum number of pixels separating peaks in the abnormality score map, defaults to 5
        :param num_peaks: int, number of peaks to identify in the abnormality score map, defaults to 5

        :return: None, this function saves figures to the local filesystem and writes information to .txt files

        The function loads Gaussian component maps for the patient and the control group. It then calculates a
        mean control map for each component and computes the absolute difference between the patient's maps and the mean control maps.

        This difference is considered as an abnormality score which is then written into a .txt file. The function also generates a
        set of plots for each Gaussian component showing the patient's map, the average control map, and the abnormality score map.

        The function also identifies the local maxima in the abnormality score map, which are marked on the plot and saved into a .txt file.
        """

    # Lists to store patient and control maps
    patient_maps = []
    control_maps = []

    # List to store coordinates of maximum values
    maxima_coordinates = []

    # Loop over the three components of the .nii files
    for i in range(1, 4):
        # Load patient map data
        patient_map_file = patient_file + f".nii_component_{i}.nii.gz"
        patient_map = nib.load(patient_map_file).get_fdata()
        patient_maps.append(patient_map)

        # Initialize an empty control map
        control_map_sum = np.zeros_like(patient_map)

```

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_prob_maps.py

```
        # Load each control map data and add to the sum
        for control_file in control_files:
            control_map_file = control_file + f".nii_component_{i}.nii.gz"
            control_map = nib.load(control_map_file).get_fdata()
            control_map_sum += control_map

        # Calculate the average control map and append to control_maps
        control_map_avg = control_map_sum / len(control_files)
        control_maps.append(control_map_avg)

    # Calculate the average control map for all components
    avg_control_maps = np.mean(control_maps, axis=0)

    # Calculate the difference between patient maps and average control maps
    diff_maps = np.abs(patient_maps - avg_control_maps)

    # Calculate a total abnormality score for the patient
    total_abnormality_score = np.mean(diff_maps)

    # Write the total abnormality score to a text file
    with open('FAabnormalities/' + patient_file[-7:] + '.txt', mode='w') as file:
            file.write(str(total_abnormality_score))

    # Create subplots to visualize the maps
    fig, axs = plt.subplots(3, 3, figsize=(15, 15))
    fig.suptitle('Probability maps for each Gaussian component \n for the patient and for the controls\n and the difference between them.', fontsize=16)

    # Add overall labels for all subplots
    fig.text(0.5, 0.04, 'Spatial Coordinate X of Scan Image', ha='center', va='center', fontsize=12)
    fig.text(0.04, 0.5, 'Spatial Coordinate Y of Scan Image', ha='center', va='center', rotation='vertical', fontsize=12)

    # Normalize the colormap so it's consistent across all components
    diff_map_norm = colors.Normalize(vmin=np.min(diff_maps), vmax=np.max(diff_maps))

    # Loop over the three components to plot patient maps, control maps, and abnormality score maps
    for i in range(3):
        axs[i, 0].imshow(patient_maps[i][:, :, slice_index], cmap='gray', origin='lower')
        axs[i, 0].set_title(f'Patient - Gaussian Component {i + 1}')
        axs[i, 1].imshow(control_maps[i][:, :, slice_index], cmap='gray', origin='lower')
        axs[i, 1].set_title(f'Average Control - Gaussian Component {i + 1}')
        diff_map = np.abs(patient_maps[i][:, :, slice_index] - control_maps[i][:, :, slice_index])
        im = axs[i, 2].imshow(diff_map, cmap='hot', norm=diff_map_norm, origin='lower')
        axs[i, 2].set_title(f'Abnormality Score - Gaussian Component {i + 1}')

    for ax in axs.flat:
        ax.label_outer()

    # Find the local maxima in the abnormality score map
    coordinates = peak_local_max(diff_map, min_distance=min_distance, num_peaks=num_peaks)
    maxima_coordinates.append(coordinates)

    # Mark the local maxima on the abnormality score map
    for coord in coordinates:
        axs[i, 2].plot(coord[1], coord[0], 'bo', markersize=5, markeredgewidth=1, markerfacecolor='None')

    # Add a colorbar showing the scale for the colormap
    fig.subplots_adjust(right=0.85)
```

Page 2 of 3

File - /Users/leondailani/PycharmProjects/projectpartiii/abnormality_prob_maps.py

```
    cbar_ax = fig.add_axes([0.9, 0.15, 0.03, 0.7])
    cbar = fig.colorbar(im, cax=cbar_ax, orientation='vertical')
    cbar.set_label('Abnormality Score Colourbar', rotation=270, labelpad=20)

    # Save the plot
    plt.savefig('FAplots/'+str(slice_index) + '_' + patient_file[-7:])

    # Save the outputs of the most abnormal regions
    with open('FAoutputs/' +str(slice_index) + '_' + patient_file[-7:] + '.txt', mode='w') as file:
        for i, coords in enumerate(maxima_coordinates):
            file.write(f"Component {i+1} local maxima coordinates:")
            for coord in coords:
                file.write(f"  - x: {coord[1]}, y: {coord[0]}")

# Example usage
if __name__ == '__main__':
    # Define the patient and control file pathnames
    patient_files = ['30292', '30293', '30295', '30297', '30302']
    control_files = ['10000', '10058', '10179', '10459', '14136', '16242', '18770', '21091', '23108', '24972', '25002',
                     '25044', '27336', '28069', '28973']

    patient_files = list(map(lambda filename: 'patients_FA_scans/' + filename +'FA', patient_files))
    control_files = list(map(lambda filename: 'controls_FA_scans/' + filename +'FA', control_files))

    # Define a slice to be used for analysis
    slice_index = 50

    # Run the function to produce the abnormality score map plots
    for patient_file in patient_files:
        display_maps(patient_file, control_files, slice_index)

```

Page 3 of 3

Appendix F: Significance Tables

Appendix F contains tables that provide comparisons of fractional anisotropy and mean diffusivity between acute and longitudinal timeframes for COVID-19 affected time patients. Asterisks indicated statistical comparisons that survive Hochberg correction for multiple comparisons, which are also highlighted in bold text.

Table 2. Mean diffusivity and fractional anisotropy for grey matter regions. All data reported as median (IQR). P-value from Wilcoxon test of acute and longitudinal scan values.

	Mean Diffusivity ($x10^{-3}$ mm^2 /s)					**Fractional Anisotropy**				
	Controls	**Acute**	**Longitudinal**	**P-value**	**Adjusted Significance Threshold**	**Controls**	**Acute**	**Longitudinal**	**P-value**	**Adjusted Significance Threshold**
Supratentorial	1.04 (0.967-1.72)	1.05 (0.997-1.16)	1.04 (0.983-1.14)	0.617	0.0083	0.133 (0.118-0.152)	0.135 (0.119-0.145)	0.136 (0.125-0.161)	0.268	0.005
Right Frontal Lobe	1.02 (0.95-1.44)	1.02 (0.95-1.2)	1.03 (0.972-1.1)	0.424	0.005	0.147 (0.111-0.173)	0.146 (0.138-0.156)	0.15 (0.142-0.168)	0.352	0.01
Left Frontal Lobe	1.05 (0.966-1.76)	1.05 (0.968-1.19)	1.03 (0.983-1.17)	0.617	0.01	0.147 (0.119-0.163)	0.144 (0.134-0.157)	0.148 (0.14-0.168)	0.33	0.0083
Right Temporal Lobe	1.04 (0.976-1.37)	1.03 (0.955-1.18)	1.02 (0.972-1.13)	0.726	0.025	0.149 (0.138-0.166)	0.149 (0.137-0.163)	0.154 (0.141-0.169)	0.289	0.0062
Left Temporal Lobe	1.04 (0.96-1.39)	1.02 (0.959-1.14)	1.02 (0.946-1.15)	0.726	0.05	0.15 (0.142-0.178)	0.153 (0.129-0.166)	0.153 (0.14-0.188)	0.373	0.017
Right Parietal Lobe	1.03 (0.937-1.99)	1.03 (0.945-1.15)	1.02 (0.989-1.13)	0.634	0.012	0.13 (0.0968-0.141)	0.132 (0.113-0.144)	0.134 (0.12-0.156)	0.165	0.0042
Left Parietal Lobe	1.01 (0.936-1.96)	1.01 (0.939-1.1)	1.0 (0.965-1.09)	0.571	0.0062	0.128 (0.103-0.141)	0.13 (0.118-0.141)	0.133 (0.127-0.156)	0.124	0.0038
Right Occipital Lobe	1.1 (1.01-2.0)	1.09 (0.993-1.25)	1.09 (1.01-1.28)	0.644	0.017	0.108 (0.0729-0.133)	0.111 (0.0906-0.127)	0.111 (0.0926-0.147)	0.233	0.0045
Left Occipital Lobe	1.07 (0.971-1.9)	1.08 (0.981-1.18)	1.06 (0.998-1.21)	0.593	0.0071	0.11 (0.07-0.141)	0.115 (0.0957-0.128)	0.115 (0.0993-0.147)	0.274	0.0056
Right Hippocampal Complex	1.09 (0.991-1.22)	1.09 (1.01-1.17)	1.1 (1.04-1.26)	0.135	0.0038	0.14 (0.119-0.199)	0.139 (0.125-0.16)	0.146 (0.132-0.166)	0.326	0.0071
Left Hippocampal Complex	1.08 (1.02-1.2)	1.07 (0.994-1.17)	1.08 (1.02-1.19)	0.296	0.0045	0.137 (0.12-0.319)	0.136 (0.121-0.196)	0.138 (0.119-0.172)	0.463	0.025
Mesencephalic Reticular Formation	0.841 (0.784-1.23)	0.837 (0.763-0.909)	0.828 (0.795-0.931)	0.443	0.0056	0.183 (0.152-0.283)	0.19 (0.172-0.233)	0.196 (0.168-0.236)	0.358	0.012
Oral Pons	0.938 (0.889-1.13)	0.931 (0.879-1.15)	0.959 (0.893-1.15)	0.142	0.0042	0.234 (0.201-0.33)	0.248 (0.177-0.325)	0.253 (0.171-0.281)	0.525	0.05

Table 3. Mean diffusivity (MD; quantifies overall diffusion of water in tissue compartments) and fractional anisotropy (FA; a measure of directionality of diffusion) for the projection fibre tracts, commissural fibre tracts and thalamic radiations. All data reported as median (IQR). P-value from Wilcoxon test of acute and longitudinal scan values.

	Mean Diffusivity ($x10^{-3}$ mm^2 /s)					**Fractional Anisotropy**				
	Controls	**Acute**	**Longitudinal**	**P-value**	**Adjusted Significance Threshold**	**Controls**	**Acute**	**Longitudinal**	**P-value**	**Adjusted Significance Threshold**
Projection Fibres										
Corticospinal Tract Left	0.98 (0.92-1.1)	0.97 (0.94-1.0)	0.98 (0.96-1.0)	**0.00031***	**0.0038***	0.45 (0.41-0.49)	0.45 (0.43-0.48)	0.45 (0.42-0.48)	0.89	0.0045
Corticospinal Tract Right	0.98 (0.92-1.1)	0.96 (0.95-1.0)	0.97 (0.96-1.0)	**0.0012***	**0.0038***	0.45 (0.41-0.49)	0.44 (0.43-0.48)	0.44 (0.42-0.49)	0.85	0.0029
Commissural Fibres										
CC Genu	0.98 (0.92-1.1)	0.96 (0.93-1.1)	0.97 (0.94-1.1)	0.03	0.0014	0.36 (0.32-0.4)	0.36 (0.33-0.39)	0.36 (0.32-0.4)	0.25	0.00085
CC Posterior Midbody	0.97 (0.89-1.2)	0.96 (0.87-1.1)	0.96 (0.91-1.0)	0.03	0.0014	0.42 (0.38-0.46)	0.41 (0.38-0.47)	0.41 (0.4-0.47)	**0.041***	**0.05***
Splenium Of Corpus Callosum	1.1 (0.98-1.2)	1.1 (1.0-1.1)	1.1 (1.0-1.1)	0.17	0.005	0.39 (0.34-0.43)	0.37 (0.33-0.41)	0.38 (0.34-0.42)	0.45	0.001
Commissure Anterior	1.1 (1.0-1.2)	1.1 (1.0-1.2)	1.1 (1.1-1.2)	0.03	0.0013	0.36 (0.31-0.39)	0.35 (0.32-0.37)	0.34 (0.3-0.37)	**0.03***	**0.05***
Fornix Left	1.9 (1.4-2.4)	1.9 (1.5-2.3)	1.8 (1.4-2.1)	0.25	0.0083	0.43 (0.31-0.52)	0.42 (0.33-0.48)	0.41 (0.34-0.48)	0.64	0.0015
Fornix Right	1.9 (1.5-2.3)	2.0 (1.6-2.3)	1.9 (1.5-2.1)	0.25	0.0071	0.43 (0.33-0.48)	0.41 (0.36-0.5)	0.44 (0.32-0.47)	0.89	0.0038
Thalamic Radiations										
Anterior Thalamic Radiation Left	0.96 (0.9-1.1)	0.96 (0.92-1.0)	0.96 (0.93-1.0)	0.45	0.05	0.35 (0.31-0.37)	0.34 (0.31-0.37)	0.35 (0.32-0.38)	0.17	0.00076
Anterior Thalamic Radiation Right	0.97 (0.89-1.1)	0.95 (0.92-1.0)	0.96 (0.94-1.0)	0.055	0.0017	0.34 (0.31-0.38)	0.34 (0.31-0.37)	0.34 (0.31-0.38)	0.68	0.0017
Superior Thalamic Radiation Left	0.9 (0.84-1.1)	0.9 (0.86-0.93)	0.91 (0.88-0.94)	0.083	0.0025	0.43 (0.39-0.48)	0.43 (0.4-0.46)	0.43 (0.4-0.47)	0.93	0.0071
Superior Thalamic Radiation Right	0.9 (0.84-1.1)	0.89 (0.86-0.96)	0.9 (0.88-0.96)	0.0054	0.00085	0.42 (0.39-0.47)	0.43 (0.39-0.45)	0.43 (0.38-0.46)	0.33	0.00091
Thalamo-Prefrontal Left	0.94 (0.87-1.1)	0.93 (0.89-1.0)	0.94 (0.92-1.0)	0.33	0.025	0.36 (0.33-0.4)	0.36 (0.34-0.39)	0.36 (0.34-0.4)	0.52	0.0012
Thalamo-Prefrontal Right	0.94 (0.88-1.1)	0.93 (0.89-1.0)	0.94 (0.92-1.0)	0.026	0.0012	0.36 (0.32-0.4)	0.36 (0.33-0.39)	0.36 (0.33-0.39)	0.98	0.017
Thalamo-Premotor Left	0.93 (0.85-1.1)	0.92 (0.86-1.0)	0.93 (0.9-1.0)	0.14	0.0038	0.39 (0.33-0.42)	0.39 (0.36-0.43)	0.39 (0.36-0.43)	0.98	0.012
Thalamo-Premotor Right	0.92 (0.87-1.1)	0.91 (0.88-0.99)	0.92 (0.9-0.99)	**0.0012***	**0.0038***	0.39 (0.35-0.43)	0.38 (0.36-0.44)	0.38 (0.36-0.43)	0.14	0.00072
Thalamo-Precentral Left	0.9 (0.84-1.1)	0.91 (0.84-0.98)	0.91 (0.88-0.96)	0.14	0.0036	0.41 (0.38-0.45)	0.41 (0.39-0.44)	0.41 (0.39-0.44)	0.6	0.0013
Thalamo-Precentral Right	0.9 (0.85-1.0)	0.89 (0.84-0.96)	0.9 (0.87-0.96)	**0.0034***	**0.0038***	0.41 (0.37-0.44)	0.4 (0.38-0.44)	0.4 (0.37-0.44)	0.23	0.00083
Thalamo-Postcentral Left	0.91 (0.86-1.0)	0.92 (0.86-0.95)	0.93 (0.9-0.96)	0.14	0.0033	0.4 (0.37-0.46)	0.4 (0.37-0.44)	0.4 (0.37-0.44)	1	0.05
Thalamo-Postcentral Right	0.91 (0.85-0.99)	0.91 (0.86-0.97)	0.92 (0.9-0.97)	0.048	0.0016	0.41 (0.38-0.45)	0.4 (0.36-0.43)	0.41 (0.36-0.43)	0.68	0.0015
Thalamo-Parietal Left	0.96 (0.89-1.1)	0.96 (0.88-1.0)	0.96 (0.93-1.0)	0.17	0.0045	0.38 (0.34-0.41)	0.38 (0.34-0.4)	0.38 (0.35-0.41)	0.93	0.0056
Thalamo-Parietal Right	0.96 (0.9-1.0)	0.96 (0.89-1.0)	0.96 (0.93-1.0)	0.03	0.0012	0.39 (0.35-0.41)	0.38 (0.35-0.39)	0.38 (0.35-0.4)	0.6	0.0014
Thalamo-Occipital Left	1.0 (0.96-1.2)	1.0 (0.96-1.1)	1.0 (0.98-1.1)	0.015	0.0011	0.37 (0.33-0.4)	0.37 (0.33-0.39)	0.36 (0.33-0.41)	0.72	0.0018

Table 4. Mean diffusivity and fractional anisotropy for the striatal fibre tracts and brainstem tracts. All data reported as median (IQR). P-value from Wilcoxon test of acute and longitudinal scan values.

	Mean Diffusivity ($x10^{-3}$ mm^2 /s)					Fractional Anisotropy				
	Controls	Acute	Longitudinal	P-value	Adjusted Significance Threshold	Controls	Acute	Longitudinal	P-value	Adjusted Significance Threshold
Striatal Fibres										
Striato-Fronto-Orbital Region Left	0.98 (0.91-1.1)	0.99 (0.95-1.1)	1.0 (0.96-1.0)	0.25	0.0062	0.35 (0.3-0.39)	0.35 (0.31-0.37)	0.35 (0.31-0.37)	0.15	0.00075
Striato-Fronto-Orbital Region Right	0.99 (0.92-1.1)	0.99 (0.94-1.1)	1.0 (0.96-1.1)	0.041	0.0015	0.35 (0.3-0.39)	0.35 (0.3-0.37)	0.35 (0.3-0.38)	0.72	0.002
Striato-Prefrontal Region Left	0.94 (0.87-1.1)	0.93 (0.9-1.0)	0.94 (0.91-1.0)	0.3	0.01	0.35 (0.31-0.38)	0.35 (0.32-0.38)	0.35 (0.32-0.38)	0.49	0.0011
Striato-Prefrontal Region Right	0.93 (0.88-1.1)	0.92 (0.89-1.0)	0.93 (0.91-1.0)	0.018	0.0011	0.35 (0.31-0.39)	0.35 (0.32-0.38)	0.35 (0.32-0.38)	0.8	0.0026
Striato-Premotor Region Left	0.93 (0.84-1.1)	0.92 (0.86-1.0)	0.93 (0.89-1.0)	0.035	0.0014	0.38 (0.31-0.41)	0.37 (0.33-0.42)	0.38 (0.34-0.42)	0.98	0.01
Striato-Premotor Region Right	0.92 (0.85-1.1)	0.9 (0.87-1.0)	0.91 (0.9-0.99)	**0.0026***	**0.0038***	0.38 (0.34-0.42)	0.38 (0.34-0.43)	0.37 (0.34-0.43)	0.25	0.00088
Striato-Precentral Region Left	0.91 (0.84-1.0)	0.92 (0.85-1.0)	0.92 (0.88-0.97)	0.14	0.0029	0.4 (0.36-0.43)	0.39 (0.36-0.42)	0.4 (0.37-0.43)	0.23	0.00082
Striato-Precentral Region Right	0.9 (0.83-1.0)	0.89 (0.85-0.97)	0.9 (0.88-0.96)	0.01	0.00096	0.4 (0.36-0.43)	0.4 (0.38-0.43)	0.4 (0.37-0.43)	0.45	0.0011
Striato-Postcentral Region Left	0.92 (0.85-1.0)	0.93 (0.86-0.98)	0.93 (0.9-0.96)	0.3	0.012	0.39 (0.35-0.43)	0.39 (0.36-0.42)	0.39 (0.36-0.42)	0.42	0.00096
Striato-Postcentral Region Right	0.91 (0.84-1.0)	0.91 (0.85-0.98)	0.91 (0.89-0.96)	0.083	0.0023	0.4 (0.37-0.43)	0.4 (0.37-0.42)	0.4 (0.37-0.43)	0.64	0.0014
Striato-Parietal Region Left	0.95 (0.88-1.0)	0.96 (0.88-1.0)	0.96 (0.92-1.0)	0.073	0.0021	0.37 (0.34-0.4)	0.37 (0.34-0.4)	0.37 (0.34-0.41)	0.39	0.00094
Striato-Parietal Region Right	0.95 (0.88-1.0)	0.95 (0.88-1.0)	0.95 (0.91-1.0)	0.026	0.0012	0.39 (0.36-0.42)	0.38 (0.35-0.4)	0.38 (0.35-0.41)	0.45	0.0011
Striato-Occipital Region Left	1.0 (0.95-1.1)	1.0 (0.95-1.1)	1.0 (0.97-1.1)	0.015	0.0011	0.38 (0.34-0.41)	0.38 (0.33-0.4)	0.38 (0.33-0.41)	0.85	0.0036
Striato-Occipital Region Right	1.0 (0.94-1.1)	0.99 (0.96-1.1)	1.0 (0.97-1.1)	**0.0026***	**0.0038***	0.37 (0.33-0.4)	0.37 (0.33-0.39)	0.37 (0.33-0.4)	0.89	0.005
Brainstem										
Parieto-Occipital Pontine Left	0.98 (0.94-1.1)	0.98 (0.96-1.0)	0.99 (0.97-1.0)	**0.00085***	**0.0038***	0.41 (0.38-0.44)	0.41 (0.38-0.43)	0.41 (0.38-0.43)	0.8	0.0028
Parieto-Occipital Pontine Right	0.99 (0.93-1.1)	0.99 (0.96-1.0)	1.0 (0.97-1.1)	0.0067	0.00088	0.42 (0.39-0.45)	0.41 (0.39-0.44)	0.41 (0.38-0.44)	1	0.025
Fronto-Pontine Tract Left	1.0 (0.95-1.1)	0.99 (0.98-1.0)	1.0 (0.98-1.1)	**0.00031***	**0.0038***	0.4 (0.37-0.44)	0.41 (0.37-0.44)	0.41 (0.38-0.44)	0.49	0.0011
Fronto-Pontine Tract Right	1.0 (0.95-1.1)	1.0 (0.98-1.0)	1.0 (0.98-1.1)	**0.00043***	**0.0038***	0.41 (0.37-0.45)	0.4 (0.38-0.44)	0.4 (0.38-0.44)	0.89	0.0042
Superior Cerebellar Peduncle Left	1.0 (0.94-1.1)	1.0 (0.96-1.0)	1.0 (0.96-1.1)	0.012	0.001	0.38 (0.35-0.41)	0.39 (0.37-0.4)	0.38 (0.36-0.4)	0.76	0.0024
Superior Cerebellar Peduncle Right	1.0 (0.94-1.1)	1.0 (0.97-1.1)	1.0 (0.96-1.1)	0.14	0.0042	0.38 (0.35-0.4)	0.38 (0.35-0.4)	0.38 (0.34-0.39)	0.98	0.0083
Inferior Cerebellar Peduncle Left	0.94 (0.87-1.0)	0.94 (0.91-0.97)	0.95 (0.91-0.99)	0.048	0.0016	0.37 (0.31-0.4)	0.37 (0.33-0.38)	0.36 (0.34-0.4)	0.76	0.0023
Inferior Cerebellar Peduncle Right	0.93 (0.86-1.0)	0.94 (0.9-1.0)	0.94 (0.9-1.0)	0.23	0.0056	0.35 (0.32-0.38)	0.35 (0.31-0.37)	0.35 (0.33-0.37)	0.45	0.001
Middle Cerebellar Peduncle	1.1 (0.96-1.2)	1.1 (1.0-1.1)	1.1 (1.0-1.2)	0.064	0.0018	0.37 (0.32-0.4)	0.36 (0.35-0.39)	0.37 (0.35-0.39)	0.095	0.00071

Table 5. Mean diffusivity and fractional anisotropy for the association fibre tracts. All data reported as median (IQR). P-value from Wilcoxon test of acute and longitudinal scan values.

	Mean Diffusivity ($x10^{-3}$ mm^2 /s)					Fractional Anisotropy				
	Controls	Acute	Longitudinal	P-value	Adjusted Significance Threshold	Controls	Acute	Longitudinal	P-value	Adjusted Significance Threshold
Association Fibres										
Superior Longitudinal Fascicle I Left	0.89 (0.82-0.95)	0.89 (0.81-0.98)	0.89 (0.87-0.96)	0.14	0.0031	0.4 (0.37-0.43)	0.39 (0.36-0.42)	0.4 (0.37-0.42)	0.56	0.0012
Superior Longitudinal Fascicle I Right	0.9 (0.83-0.96)	0.89 (0.82-0.99)	0.9 (0.87-0.98)	0.015	0.0011	0.4 (0.36-0.44)	0.4 (0.36-0.42)	0.4 (0.37-0.42)	0.8	0.0025
Superior Longitudinal Fascicle Ii Left	0.9 (0.84-0.98)	0.91 (0.84-1.0)	0.91 (0.89-0.99)	0.083	0.0022	0.38 (0.34-0.41)	0.37 (0.33-0.4)	0.37 (0.33-0.41)	0.21	0.00079
Superior Longitudinal Fascicle Ii Right	0.91 (0.83-0.97)	0.9 (0.86-0.99)	0.91 (0.87-0.99)	0.0067	0.00086	0.37 (0.33-0.41)	0.38 (0.33-0.4)	0.38 (0.33-0.4)	0.72	0.0019
Superior Longitudinal Fascicle Iii Left	0.91 (0.85-1.0)	0.91 (0.85-0.98)	0.92 (0.89-1.0)	0.015	0.001	0.38 (0.33-0.42)	0.37 (0.32-0.41)	0.37 (0.32-0.41)	0.93	0.0062
Superior Longitudinal Fascicle Iii Right	0.92 (0.85-1.0)	0.91 (0.87-1.0)	0.92 (0.89-0.98)	0.012	0.001	0.36 (0.32-0.41)	0.37 (0.32-0.38)	0.37 (0.32-0.39)	0.85	0.0033
Uncinate Fascicle Left	1.0 (0.92-1.1)	0.99 (0.96-1.1)	1.0 (0.97-1.1)	0.073	0.002	0.35 (0.32-0.38)	0.34 (0.31-0.37)	0.35 (0.31-0.37)	0.21	0.00078
Uncinate Fascicle Right	1.0 (0.94-1.1)	0.98 (0.95-1.0)	1.0 (0.96-1.1)	0.03	0.0012	0.34 (0.29-0.37)	0.33 (0.29-0.35)	0.33 (0.3-0.36)	0.76	0.0022
Inferior Longitudinal Fascicle Left	1.0 (0.94-1.0)	0.99 (0.95-1.0)	1.0 (0.96-1.1)	0.03	0.0013	0.36 (0.33-0.4)	0.36 (0.32-0.38)	0.36 (0.32-0.4)	0.28	0.00089
Inferior Longitudinal Fascicle Right	0.99 (0.93-1.1)	0.97 (0.95-1.0)	0.98 (0.96-1.1)	**0.0034***	**0.0038***	0.36 (0.33-0.4)	0.36 (0.31-0.38)	0.36 (0.31-0.39)	0.85	0.0031
Arcuate Fascicle Left	0.93 (0.87-1.0)	0.92 (0.88-0.99)	0.92 (0.91-0.99)	0.055	0.0017	0.36 (0.32-0.39)	0.35 (0.31-0.38)	0.35 (0.31-0.39)	0.68	0.0016
Arcuate Fascicle Right	0.92 (0.87-1.0)	0.91 (0.89-0.99)	0.92 (0.89-0.99)	**0.0026***	**0.0038***	0.36 (0.32-0.4)	0.36 (0.32-0.38)	0.36 (0.32-0.39)	0.6	0.0013
Cingulum Left	0.93 (0.87-0.99)	0.93 (0.87-1.0)	0.93 (0.91-0.99)	0.064	0.0019	0.36 (0.32-0.39)	0.36 (0.33-0.38)	0.36 (0.33-0.38)	0.14	0.00074
Cingulum Right	0.92 (0.85-0.98)	0.92 (0.87-1.0)	0.93 (0.9-1.0)	**0.0034***	**0.0038***	0.37 (0.33-0.4)	0.36 (0.33-0.39)	0.36 (0.34-0.4)	0.36	0.00093
Middle Longitudinal Fascicle Left	0.94 (0.89-1.0)	0.93 (0.89-1.0)	0.95 (0.92-1.0)	0.0084	0.00091	0.36 (0.32-0.39)	0.36 (0.3-0.38)	0.35 (0.31-0.39)	0.42	0.001
Middle Longitudinal Fascicle Right	0.94 (0.88-1.0)	0.94 (0.9-1.0)	0.95 (0.91-1.0)	**0.0034***	**0.0038***	0.36 (0.33-0.4)	0.36 (0.31-0.38)	0.36 (0.32-0.38)	0.42	0.00098
Inferior Occipito-Frontal Fascicle Left	1.0 (0.93-1.1)	0.99 (0.95-1.1)	1.0 (0.96-1.1)	0.12	0.0028	0.34 (0.31-0.36)	0.34 (0.29-0.36)	0.34 (0.3-0.37)	0.25	0.00086
Inferior Occipito-Frontal Fascicle Right	1.0 (0.93-1.1)	0.99 (0.97-1.1)	0.99 (0.97-1.1)	0.012	0.00098	0.33 (0.29-0.35)	0.33 (0.29-0.35)	0.33 (0.29-0.36)	0.72	0.0019
Optic Radiation Left	1.0 (0.95-1.2)	1.0 (0.95-1.1)	1.0 (0.97-1.1)	0.01	0.00094	0.38 (0.33-0.41)	0.37 (0.33-0.4)	0.37 (0.34-0.41)	0.72	0.0021
Optic Radiation Right	1.0 (0.94-1.1)	0.99 (0.96-1.1)	1.0 (0.97-1.1)	0.0067	0.00089	0.38 (0.35-0.41)	0.38 (0.34-0.41)	0.38 (0.34-0.42)	0.52	0.0012