Manual CATO (Connectivity Analysis TOolbox)

Kurz Erklärung: Die Connectivity Analysis Toolbox dient der Rekonstruktion struktureller und funktioneller Gehirnkonnektivität basierend auf Diffusionsgewichteter Bildgebung und Resting-State-fMRT-Daten.

Ziel des Manuals: Wenn ihr Netzwerkanalysen durchführen wollt, seid ihr hier genau richtig! Dieses Manual erläutert zunächst die theoretischen Grundlagen der Netzwerkanalyse und bietet euch anschließend eine Step-by-Step-Anleitung für die Durchführung einer strukturellen Konnektivitätsanalyse mit der CATO-Toolbox. Dafür gibt es zwei Wege: einmal die Anleitung mit Erklärungen und Beispielen (CATO_RUN_EXAMPLE) und einmal ohne. Für die richtige Analyse verwendet den CATO_RUN-Ordner.

Bitte halten Sie den Ordner CATO_RUN(_EXAMPLE) sowie dieses Manual parallel geöffnet, während Sie die Beispiele durchgehen.

Es wird empfohlen, sich vorab die Beispielordner anzusehen, um ein besseres Verständnis für die Funktionsweise der Pipeline zu bekommen. In den Beispielordnern sind bereits Probedaten enthalten, die für eine Beispielausführung verwendet werden können. Dies erleichtert den Einstieg und hilft, die Struktur und den Ablauf der Analyse zu verstehen.

Ihr findet alle benötigten Daten

unter: /net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/02
_code/01_CATO/CATO_MANUAL/



(Stein et al., 2023)

Jannik Lepper

Inhaltsverzeichnis

Glossar	1
Einleitung	3
Kapitel 1. Kurze Theorie Netzwerkanalyse Strukturelle Konnektivität Funktionelle Konnektivität	4 4 5
Effektive Konnektivität	5
Kapitel 2. Theorie CATO Was ist die CATO? Kurz erklärt was macht die CATO?	6 6 6
Anleitung Structural pipeline mit Beispiel (für Erstanwender)	11
Kapitel 3. PrePro-Datenvorbereitung Daten Voraussetzungen Programmvoraussetzungen	11 11 12
Kapitel 4. Ausführen der CATO-Pipeline Ausführung der CATO Pipeline	15 17
Kapitel 5. Abschluss der Datenvorbereitung und Beispielanalyse Analyse der Daten	18 19
Kapitel 6. CATO Second Level Analyse	21
Nur Anleitung ohne Erklärung für Fortgeschrittene	23
Literaturverzeichnis	31
Anhang	32

Glossar

DCM

Dynamic Causal Modeling: Ein Rahmenwerk zur Inferenz und Quantifizierung kausaler Interaktionen zwischen Hirnregionen.

DWI

Diffusionsgewichtete Bildgebung: Ein MRT-Verfahren, das die Diffusion von Wassermolekülen nutzt, um Kontraste in MR-Bildern zu erzeugen.

BOLD

Blood Oxygen Level Dependent: Bezieht sich auf das BOLD-Signal, das in der funktionellen MRT verwendet wird, um Hirnaktivität zu detektieren.

fMRT

Funktionelle Magnetresonanztomographie: Ein MRT-Verfahren zur Messung und Darstellung von Hirnaktivität durch Veränderung des Blutflusses.

DTI

Diffusionstensorbildgebung: Ein MRT-basiertes Neuroimaging-Verfahren, das die eingeschränkte Diffusion von Wasser im Gewebe misst, um neuronale Bahnen darzustellen.

GQI

Generalized Q-Sampling Imaging: Eine fortgeschrittene Diffusions-MRI-Technik zur Schätzung der Faserorientierungsverteilung.

CSD

Constrained Spherical Deconvolution: Eine Methode in der Diffusions-MRI zur Schätzung der Faserorientierungsverteilungen.

AD

Axiale Diffusivität: Ein DTI-Metrik, die die Diffusion entlang der Hauptachse eines Fasertrakts misst.

RD

Radiale Diffusivität: Misst die Diffusion senkrecht zur Hauptachse eines Fasertrakts.

MD

Mittlere Diffusivität: Der durchschnittliche Diffusionswert in alle Richtungen, eine weitere DTI-Metrik.

FACT

Fiber Assignment by Continuous Tracking: Ein deterministischer Traktographie-Algorithmus, der in der DTI verwendet wird.

GFA

Generalisierte Fractional Anisotropie: Ein Maß in der Diffusions-MRI, das den Grad der Anisotropie der Wasserdiffusion beschreibt.

NOS

Number of Streamlines: Anzahl der Strahlen, die in der Traktographie verfolgt werden.

SSD

Sum of Squared Differences: Ein Maß zur Ähnlichkeit zwischen zwei Datensätzen.

FD

Framewise Displacement: Ein Maß für Kopfbewegungen in fMRT-Daten.

DVARS

Derivative of Variance of timecourses: Ein Maß zur Erkennung von Bewegungsartefakten in fMRT-Daten.

QC

Qualitätskontrolle: Verfahren zur Sicherstellung der Datenqualität.

RC

Rich Club: Ein Netzwerkphänomen, bei dem hoch verbundene Knoten dichter miteinander verbunden sind.

L

Pfadlänge: Ein Netzwerkmesswert, der die durchschnittliche kürzeste Entfernung zwischen allen Knotepaaren misst.

GE

Globale Effizienz: Ein Netzwerkmesswert, der misst, wie effizient Informationen im Netzwerk ausgetauscht werden.

С

Clustering-Koeffizient: Ein Netzwerkmesswert, der den Grad der Clusterbildung in einem Netzwerk anzeigt.

Lnorm

Normalisierte Pfadlänge: Pfadlänge normalisiert gegenüber einem zufälligen Netzwerk.

GEnorm

Normalisierte Globale Effizienz: Globale Effizienz normalisiert gegenüber einem zufälligen Netzwerk.

Cnorm

Normalisierter Clustering-Koeffizient: Clustering-Koeffizient normalisiert gegenüber einem zufälligen Netzwerk.

SW

Small-Worldness: Ein Maß für das Gleichgewicht zwischen lokaler Clusterbildung und globaler Integration in einem Netzwerk.

Einleitung

Dieses Manual bietet einen Überblick über die theoretischen Grundlagen von CATO und erklärt die Funktionsweise der Toolbox. Darüber hinaus enthält es eine detaillierte Anleitung zur eigenständigen Analyse, einschließlich einer Beispielpipeline. Das Manual wird zusammen mit den zugehörigen Skripten und Programmen auf dem Server unter: /net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/02_code /01 CATO/CATO MANUAL/bereitgestellt.

(Ausschließlich Erklärung der strukturellen konnektivitäts Pipeline.)

Nach einer kurzen theoretischen Einführung in die Netzwerkanalyse und die Grundlagen für CATO folgt die Anleitung zur Anwendung der Toolbox. Diese umfasst die Voraussetzungen, das Preprocessing (PrePro), die Durchführung der Pipeline, die Qualitätskontrolle (QC) sowie die Analyse der Netzwerke. Für diejenigen, die sich ausschließlich für die Umsetzung interessieren und nicht für die Theorie sowie eine Erklärung der einzelnen Schritte, empfiehlt es sich, direkt zu "Kapitel 3. PrePro-Datenvorbereitung" zu springen und für die Anleitung ohne Erklärungen und Beispielen zu "Nur Anleitung ohne Erklärung für Fortgeschrittene" zu springen. Wenn ihr den fertigen Output sehen wollt, findet ihr fertige Daten für 3 Probanden im

/net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/02_code
/01_CATO/CATO_MANUAL/CATO_EXAMPLE_Processed.

Kapitel 1. Kurze Theorie Netzwerkanalyse

Netzwerkanalysen werden in den Neurowissenschaften zunehmend eingesetzt und erfreuen sich wachsender Beliebtheit. Es stehen vielfältige Methoden zur Verfügung, beispielsweise EEG, MEG, ECOG, fNIRS, fMRT, rsMRT, DWI und weitere, die sich in ihrer temporalen und räumlichen Auflösung unterscheiden. Daher muss zu Beginn einer jeden Untersuchung klar definiert werden, welche Fragestellung adressiert werden soll, und anschließend die entsprechenden Daten erhoben werden. Grundsätzlich unterscheidet man zwischen struktureller, funktioneller und effektiver Konnektivität, wobei jede dieser Kategorien unterschiedliche methodische Ansätze und Anwendungsmöglichkeiten erfordert. So bietet beispielsweise EEG eine hohe zeitliche, aber eine vergleichsweise geringe räumliche Auflösung, während das MRT exzellente räumliche, jedoch eine im Vergleich niedrigere zeitliche Auflösung besitzt. Darüber hinaus steigt nicht nur die Vielfalt der Methoden und damit auch die Zahl der Datentypen, sondern auch die Zahl der verfügbaren Programme (wie SPM, CONN, MATLAB, R, CATO) und Analysemethoden etwa ICA, netzwerkbasierte Statistik (NBS), Dynamic Causal Modeling (DCM) (Bassett & Sporns, 2017). Es ist daher stets zu hinterfragen, welcher Datentyp der Ausgangspunkt der Untersuchung ist und welche spezifischen Analysen damit grundsätzlich möglich sind. Beispielsweise gestaltet es sich schwierig, aus DWI-Daten mithilfe von DCM die effektive Konnektivität abzuleiten oder aus strukturelle Konnektivität zu rekonstruieren, funktionellen Daten die da die zugrundeliegenden physiologischen Informationen grundsätzlich unterschiedlich sind.

Da CATO speziell mit MRT-Daten arbeitet, konzentriert sich die theoretische Grundlage dieses Manuals ausschließlich auf diese Daten Art. An MRT-Scannern stehen hierfür unterschiedliche Sequenzen zur Verfügung, die jeweils spezifische Aspekte der Konnektivitätsanalyse beleuchten. Für die strukturelle Konnektivität werden in der Regel Diffusionsgewichtete Bildgebungsdaten (DWI) herangezogen, welche die Diffusion von Wasser entlang von Faserbahnen messen, ergänzt durch T1-gewichtete, hochauflösende anatomische Aufnahmen. Für die funktionelle und effektive Konnektivität kommen hingegen wesentlich fMRT-Sequenzen zum Einsatz. Beim Resting-State-fMRT liegen die Probanden in Ruhe ohne spezifische Aufgaben, während bei Task-fMRT BOLD (Blood Oxygen Level Dependent) Signaländerungen erfasst werden, die während der Ausführung spezifischer Aufgaben auftreten (Huettel et al., 2009). Nun folgt eine kurze Erklärung der verschiedenen Konnektivitäts Arten.

Strukturelle Konnektivität

Die strukturelle Konnektivität stellt die physischen Verbindung zwischen Gehirnarealen dar. Sie basiert auf bildgebenden Verfahren, die strukturell gewichtete MRT-Daten, wie etwa DWI, verwenden. Mit Hilfe von Traktographieverfahren wird aus diesen Daten analysiert, wie Hirnregionen über die weiße Substanz verbunden sind. Diese Methode ermöglicht es, die anatomischen Verbindungen zwischen Gehirnarealen sichtbar zu machen und bietet eine Grundlage, um Netzwerke auf Basis ihrer strukturellen Integrität zu rekonstruieren (Huettel et al., 2009).

Funktionelle Konnektivität

Die funktionelle Konnektivität beschreibt Zusammenhänge in der Aktivierung verschiedener Gehirnregionen, die entweder während spezifischer Aufgaben (Task-fMRT) oder im Ruhezustand (Resting-State fMRT) gemessen werden. Hierbei werden zeitliche Korrelationsmuster der BOLD-Signale analysiert, um Bereiche zu identifizieren, die synchron aktiviert werden. Die zugrunde liegende Annahme ist, dass Regionen, die in ihren Aktivitätsmustern übereinstimmen, funktionell miteinander vernetzt sind und möglicherweise an derselben kognitiven oder sensorischen Verarbeitung beteiligt sind (Friston, 2011).

Effektive Konnektivität

Die effektive Konnektivität geht einen Schritt weiter, indem sie die kausalen Zusammenhänge zwischen Gehirnregionen untersucht. Im Gegensatz zur funktionellen Konnektivität, die lediglich statistische Abhängigkeiten ermittelt, fokussiert sich die effektive Konnektivität darauf, Richtungen und Einflussnahmen der Interaktionen zwischen Gehirnbereichen zu bestimmen. Hierzu werden Methoden wie DCM oder Granger-Kausalitätsanalysen angewendet, die versuchen, die gerichtete Informationsweitergabe und die zugrundeliegenden neuronalen Mechanismen zu modellieren. Somit liefert die effektive Konnektivität nicht nur Hinweise darauf, welche Regionen gemeinsam aktiviert werden, sondern auch darauf, wie diese Interaktionen in kausaler Weise gesteuert werden (Friston, 2011).

Kapitel 2. Theorie CATO

Das zweite Kapitel beschäftigt sich mit den Fähigkeiten der CATO-Toolbox, den theoretischen Grundlagen und ihrer Funktionsweise. Dieses Kapitel ist in zwei Abschnitte unterteilt: "Was ist die CATO?" und "Was macht die CATO?".

Was ist die CATO?

Die CATO ist eine modulare Software zur Rekonstruktion und Analyse von struktureller und funktioneller Konnektivität, die auf DWI- und Resting-State fMRT-Daten basiert. Sie ermöglicht eine End-to-End-Verarbeitung von Rohdaten bis hin zu vollständigen Connektomkarten und bietet dabei flexible Konfigurationsmöglichkeiten, die auf unterschiedliche Analyseanforderungen abgestimmt werden können. In ihrem Funktionsumfang integriert CATO etablierte Algorithmen, wie beispielsweise Diffusion Tensor Imaging (DTI), Generalized Q-Sampling Imaging (GQI) und Constrained Spherical Deconvolution (CSD) zur Rekonstruktion der weißen Substanz, während für die Analyse der funktionellen Konnektivität Methoden wie Pearson-Korrelationen oder benutzerdefinierte Maße herangezogen werden. Die Besonderheit der Toolbox liegt darin, dass sowohl strukturelle als auch funktionelle Konnektivitätsanalysen auf der Grundlage desselben Atlas-Templates durchgeführt werden können. Dies ermöglicht es, Wechselwirkungen zwischen den beiden Konnektivitätsformen zu untersuchen und so neue Einblicke in die Dynamik neuronaler Netzwerke sowie in die Mechanismen von Hirnerkrankungen zu gewinnen. CATO ist als MATLAB-Toolbox, als eigenständige Anwendung oder in einem Docker-Container verfügbar. In diesem Manual wird ausschließlich die eigenständige Anwendung "executable" erläutert, und die vorliegende Beispielanalyse bezieht sich auf die strukturelle Konnektivität (De Lange et al., 2023).

Kurz erklärt was macht die CATO?

Für eine detaillierte Beschreibung verweise ich auf das entsprechende Paper von De Lange et al., 2023, in dem die Methodik ausführlich dargestellt ist. Das Paper ist in dem Ordner /net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/02_code /01_CATO/CATO_MANUAL/CATO_LIT/ abgelegt.

Structural pipeline:

Im ersten Schritt, dem Structural Preprocessing, werden vorverarbeitete T1-gewichtete anatomische Daten und ein DWI-Datensatz benötigt. Hier wird empfohlen, vor der Analyse mit FreeSurfer eine kortikale Rekonstruktion durchzuführen, da diese nicht nur die rechnerische Effizienz steigert, sondern auch essentielle Dateien für die weiteren Verarbeitungsschritte bereitstellt. Während des Preprocessings werden sowohl die FreeSurfer-Daten als auch die DWI-Daten genutzt, um Zwischenprodukte für die nachfolgenden Schritte zu generieren. Anwender haben die Möglichkeit, eines der vorgegebenen Preprocessing-Skript (wie preprocess_topup_eddy.sh, preprocess_eddy.sh oder preprocess_minimal.sh) zu verwenden oder ein eigenes Skript zu entwickeln, das optimal auf die spezifischen Daten abgestimmt ist. So umfasst beispielsweise das Skript preprocess_topup_eddy.sh mehrere Verarbeitungsschritte: Zunächst werden mithilfe von FSL Topup und FSL Eddy Suszeptibilitäts-bedingte Verzerrungen, Eddy-Strom-Artefakte und Bewegungsartefakte in den DWI-Daten korrigiert. Dabei werden die b-Vektoren entsprechend angepasst und ein DWI-Referenzbild auf Basis der korrigierten b0-Volumina generiert. Anschließend wird mit bbregister die Registrierungs-Matrix zwischen dem DWI-Referenzbild und den T1-Daten berechnet, sodass die Segmentierungsdaten von FreeSurfer auf das DWI-Referenzbild übertragen werden können.

Schritt	Minimal	Eddy	Topup Eddy	Custom
Korrektur von Suszeptibilitäts-Verzerrungen (mit FSL Topup)	nein	nein	ја	?
Korrektur von Eddy-Strom-Verzerrungen und Bewegungsartefakten in den DWI-Daten (mit FSL Eddy)	nein	ја	ja	?
Aktualisierung der b-Vektoren zur Anpassung an die DWI-Korrekturen	nein	ја	ја	?
Berechnung eines DWI-Referenzbildes basierend auf den diffusionsgewichteten (b0) Volumina (mit FSL)	ја	ја	ја	?
Berechnung der Registrierungs-Matrix zwischen dem DWI-Referenzbild und dem T1-Bild (mit FreeSurfer)	ја	ја	ја	?
Übertragung der FreeSurfer-Segmentierung auf das DWI-Referenzbild (mit FreeSurfer)	ја	ја	ја	?

Eine Übersicht, welches Skript welche Verarbeitungsschritte enthält, findet sich in Tabelle 1.

Tabelle 1: Überblick der Vorverarbeitungsschritte

Im zweiten Schritt erfolgt die anatomische Verarbeitung. Hier wird die Hirnoberfläche anhand Referenzatlanten parzelliert. Standardmäßig stehen in der CATO-Toolbox verschiedene Atlanten zur Verfügung, wie der Desikan-Killiany-Atlas, diverse Subparzellierungen des Cammoun-Atlases sowie der Von Economo-Koskinas-Atlas. FreeSurfer wird eingesetzt, um die notwendigen Annotationen zu erzeugen, die es erlauben, die corticalen Parzellierungen mit subkortikalen Regionen zu kombinieren. Zudem werden wichtige statistische Eigenschaften wie die Anzahl der Vertices, die Oberflächenfläche, das Volumen der grauen Substanz und die durchschnittliche Dicke für jede Region berechnet und in einer zusammenfassenden Datei abgelegt, welche in den folgenden Schritten der Analyse Verwendung findet.

Im dritten Schritt, der Reconstruction Diffusion, wird die Organisation der weißen Substanz anhand der DWI-Daten rekonstruiert. Dieser Schritt basiert auf der Analyse von Diffusionsmustern, die durch die Einschränkung der Diffusion von Wassermolekülen entlang neuronaler Fasern entstehen. Es stehen dabei drei verschiedene Methoden zur Auswahl:

- Diffusion Tensor Imaging (DTI): Hier wird ein Tensor-Modell verwendet, um in jedem Voxel eine primäre Diffusionsrichtung zu schätzen. Dabei kommt der informed RESTORE-Algorithmus zum Einsatz, um physiologische Artefakte zu minimieren und Ausreißer zu erkennen bzw. zu entfernen. - Constrained Spherical Deconvolution (CSD): Diese Methode ermöglicht die Rekonstruktion mehrerer Diffusionspeaks pro Voxel durch Dekonvolution des gemessenen Signals. Negative Signalwerte werden eliminiert, um Rauschen zu reduzieren und klar definierte Peaks zu erhalten.

- Generalized Q-sampling Imaging (GQI): Hierbei wird das Diffusionssignal extrapoliert, um die Diffusionsrichtungen mit dem höchsten Signal als Peaks zu identifizieren.

Die gewünschte Methode oder eine Kombination verschiedener Ansätze wird im Konfigurationsparameter "reconstructionMethods" festgelegt. Durch die Kombination von DTI mit CSD oder GQI kann CATO sowohl Regionen mit einer dominanten Diffusionsrichtung als auch komplexere Regionen mit mehreren Diffusionspeaks adäquat modellieren. Zudem bietet dieser Schritt die Möglichkeit, wichtige Diffusionsmaße wie Fractional Anisotropy (FA), Axial Diffusivity (AD), Radial Diffusivity (RD) und Mean Diffusivity (MD) in Form von NIFTI-Dateien zu exportieren, eine Funktion, die durch entsprechende Parameter in der Konfigurationsdatei aktiviert werden kann.

Der vierte Schritt, die Reconstruction Fibers, umfasst die Faserrekonstruktion basierend auf den zuvor ermittelten Diffusionspeaks in jedem Voxel. Hierbei wird eine erweiterte Version des deterministischen Fiber Assignment by Continuous Tracking (FACT)-Algorithmus eingesetzt. Die Rekonstruktion startet an einer oder mehreren Seeds in allen Voxeln, die anhand der Segmentierungscodes in "startRegions" festgelegt wurden, und propagiert Streamlines entlang der Hauptdiffusionsrichtung. Die Richtung der Propagation wird dabei jedes Mal aktualisiert, wenn die Spitze der Faser ein neues Voxel erreicht. Die Faserrekonstruktion endet, wenn bestimmte Abbruchkriterien erfüllt sind, etwa wenn die FA im aktuellen Voxel unter den Schwellenwert "minFA" fällt, das Voxel zu einer in "stopRegions" definierten Region gehört, den Voxel erneut berechnen würde, eine verbotene Region definiert in "forbiddenRegions" betreten wird oder eine abrupte Richtungsänderung (kleiner als der in "maxAngleDeg" definierte Wert auftritt. Die Parameter "maxAngleDeg" und "minFA" sind dabei sowohl für diesen als auch für den folgenden Schritt der Netzwerkrekonstruktion von zentraler Bedeutung und können individuell angepasst werden.

Im fünften Schritt, Reconstruction Fiber Properties, werden die rekonstruierten Fasersegmente analysiert, die verschiedene Gehirnregionen verbinden. Für jeden verwendeten Atlas und jede Rekonstruktionsmethode iteriert dieser Schritt über alle Fasern. Er identifiziert diejenigen Fasersegmente, die zwei oder mehr definierte Hirnregionen (basierend auf der ROIsFile) durchqueren, und speichert das kürzeste Segment zwischen den Regionen. Neben den Start- und Endpunkten jedes Segments sowie dem zugehörigen Regionenpaar werden hier detaillierte Eigenschaften wie die maximale Krümmung (in Radiant), die minimale und durchschnittliche FA, die physische Faserlänge (in Millimetern) sowie Maße AD, RD und MD sowie die Generalized Fractional Anisotropy (GFA) berechnet. Die Diffusionsmaße werden dabei voxelübergreifend gemittelt, wobei die Länge des innerhalb eines Voxels durchlaufenen Pfads als Gewichtungsfaktor dient.

Im sechsten und letzten Schritt, der Network Reconstruction, erfolgt die Erstellung der Konnektivitätsmatrizen für die ausgewählten (sub-)kortikalen Atlanten und Rekonstruktionsmethoden. Die in der Konnektivitätsmatrix enthaltenen Gehirnregionen sowie deren Reihenfolge werden durch die Regionsdatei (ROIsFile) definiert. Zur Generierung der Matrix werden alle Fasersegmente aus der Datei "fiberPropertiesFile" herangezogen, die entsprechende Regionen miteinander verbinden. Dabei können Filterkriterien wie eine Mindestprojektionlänge, ein Mindestwert für FA oder eine maximale Krümmung angewendet werden. Die Verbindungen zwischen den Regionen werden anhand diverser Maße gewichtet, beispielsweise der Anzahl der Streamlines (NOS), der Faserlänge, der mittleren FA sowie der AD, RD, MD und GFA. Darüber hinaus werden weitere Gewichtungen wie die Streamline Volume Density (SVD) definiert als Anzahl der Streamlines geteilt durch das durchschnittliche Volumen der verbundenen Regionen sowie die Streamline Surface Density (SSD) also die Anzahl der Streamlines relativ zur durchschnittlichen Oberflächenfläche der Regionen berechnet. Auch hier erfolgt die Gewichtung der Diffusionsmaße voxelübergreifend unter Berücksichtigung des durchquerten Pfadsegments als Gewichtungsfaktor. Das Ergebnis dieses Schrittes sind die Konnektivitätsmatrizen, die als Basis für weiterführende Analysen neuronaler Netzwerke dienen (De Lange et al., 2023).

Functional pipeline:

Im Folgenden wird die funktionelle Pipeline von CATO beschrieben, die in vier Hauptschritte unterteilt ist wobei der finale Schritt, die Network Reconstruction, nochmals in vier Unterabschnitte gegliedert wird (Regression, Bandpassfilterung, Scrubbing und Korrelationsanalyse).

Zunächst erfolgt ein Preprocessing der funktionellen Daten. In diesem Schritt können Nutzer ihre rs-fMRI-Daten mithilfe der Software ihrer Wahl vorverarbeiten. Die Wahl des Preprocessing-Skript wird über den entsprechenden das Parameter *preprocessingScript* festgelegt. Als Beispiel bietet CATO Skript preprocess default.sh an, das eine Reihe standardisierter Verarbeitungsschritte durchführt. Zunächst wird eine Slice-Timing-Korrektur unter Anwendung des FSL-Tools sliceTimer vorgenommen, sofern dies in der Konfigurationsdatei mittels des Parameters sliceTimingCorrection aktiviert wurde. Die spezifischen Optionen können über *sliceTimerOptions* festgelegt werden. Im Anschluss erfolgt eine Bewegungskorrektur der rs-fMRI-Daten mit dem FSL-Tool MCFLIRT, um Bewegungsartefakte zu minimieren. Danach wird ein Referenzbild erstellt, indem alle bewegungskorrigierten Zeitpunkte gemittelt werden. Dieses Referenzbild wird anschließend mithilfe von FreeSurfer (zur Berechnung der Registrierungs-Matrix) mit dem T1-gewichteten Bild registriert, sodass die T1-Parzellierung, welche wichtige anatomische Informationen enthält, auf das rs-fMRI-Referenzbild übertragen werden kann und somit eine präzise Abstimmung zwischen anatomischen und funktionellen Daten gewährleistet ist.

Der zweite Schritt der funktionellen Pipeline umfasst die anatomische Verarbeitung. Hierbei werden, analog zur strukturellen Pipeline dieselben Methoden genutzt wie die strukturelle Pipeline, um die Hirnoberfläche anhand von Referenzatlanten in Gehirnregionen zu parzellieren. Dabei werden anatomische Statistiken für diese Regionen berechnet. Für eine detaillierte Beschreibung siehe strukturelle Pipeline.

Im dritten Schritt, der Berechnung von Bewegungsmetriken (Compute Motion Metrics), werden für jedes einzelne Bild (Frame) der rs-fMRI-Daten wichtige Indikatoren zur Erfassung von Bewegungsartefakten berechnet. Konkret werden hierfür der Framewise Displacement (FD) und DVARS (D temporal derivative VARiance over voxelS) berechnet. Der FD-Wert ergibt sich als Summe der absoluten translatorischen und rotatorischen Verschiebungen, wobei letztere ursprünglich in Grad angegeben unter Annahme eines kugelförmigen Modells der Großhirnrinde (Radius ca. 50 mm) in Millimeter umgerechnet werden. DVARS quantifiziert die Veränderungen der Signalintensität zwischen aufeinanderfolgenden Frames, indem die Quadratwurzel des durchschnittlichen quadratischen Intensitätsunterschieds aller Gehirnvoxel berechnet wird. Beide Maßzahlen werden aus den Bewegungsparametern (z. B. aus der mit MCFLIRT erzeugten *motionParametersFile*) abgeleitet und in einer entsprechenden Datei gespeichert, um als Grundlage für eine Qualitätssicherung der funktionellen Daten zu dienen.

Als letzter und vierter Schritt die Network Reconstruction berechnet die funktionelle Konnektivität zwischen Gehirnregionen auf Basis von Korrelationen. Dieser Prozess besteht aus mehreren optionalen Vorverarbeitungsschritten und einer abschließenden Korrelationsanalyse und ist in vier aufeinanderfolgende Abschnitte unterteilt:

Zunächst werden im Abschnitt Regression unerwünschte Kovariaten aus den Signalintensitäts-Zeitreihen der rs-fMRI-Daten entfernt. Dies erfolgt durch die Berechnung der Residuen eines linearen Modells, in dem die Kovariaten als Prädiktoren verwendet werden. Standardmäßig umfassen die Regessoren unter anderem lineare Trends der sechs Bewegungsparameter, deren erste Ableitungen, die mittlere Signalintensität von Voxeln in weißer Substanz und Liquor sowie optional der globale Mittelwert des Gehirns berücksichtigt werden.

Im zweiten Abschnitt, der Bandpassfilterung die optional ist, wird ein Butterworth-Bandpassfilter angewandt, um störende Frequenzen zu entfernen. Ziel ist es dabei, sehr niedrige Frequenzen etwa infolge von Scanner-Drift oder langsamen vaskulären Oszillationen sowie hohe Frequenzen wie durch Atmung oder Herzschlag hervorgerufen zu eliminieren. Die entsprechenden unteren und oberen Grenzfrequenzen werden über den Parameter *bandpass_filter.frequencies* festgelegt.

Anschließend folgt der abschnitt Scrubbing der auch optional ist. Hierbei werden Frames, die Bewegungsartefakte Schwellenwerte signifikante aufweisen, anhand definierter Konkret erfolgt dies, wenn der FD-Wert eines Frames ausgeschlossen. den in scrubbing.maxFD festgelegten Wert überschreitet oder wenn die DVARS-Werte eines Frames über einem Schwellenwert liegen, der als Q3 plus einem Multiplikator der Interquartilspanne (IQR) definiert wird. Ferner werden Frames, die eine Mindestzahl an Verstößen definiert durch scrubbing.minViolations aufweisen, komplett ausgeschlossen, wobei auch benachbarte Frames über die Parameter scrubbing.backwardNeighbors und scrubbing.forwardNeighbors mit entfernt werden können, um den Einfluss von Bewegungsartefakten zu minimieren.

Abschnitt der funktionellen Network Im letzten Reconstruction erfolgt die Korrelationsanalyse. Hierbei wird die funktionelle Konnektivität zwischen Gehirnregionen definiert durch die Regionsdatei (ROIsFile) auf Basis der mittleren Signalintensität der jeweiligen Regionen über die Zeit berechnet. Die zugrunde liegende Korrelationsmethode wird über den Parameter reconstructionMethod bestimmt, wobei beispielsweise die Pearson-Korrelation oder partielle Korrelation gewählt werden kann. Die resultierenden Konnektivitätsmatrizen werden in der Datei connectivityMatrixFilegespeichert, während die extrahierten Zeitreihen der einzelnen Regionen in der Datei timeSeriesFile abgelegt werden. Dieser finale Schritt liefert die Basis für weiterführende Analysen funktionaler Konnektivitätsnetzwerke (De Lange et al., 2023).

Anleitung Structural pipeline mit Beispiel (für Erstanwender)

Es folgt eine Schritt-für-Schritt-Anleitung zur Nutzung der CATO-Toolbox auf dem Server, unterteilt in fünf Kapitel: PrePro-Datenvorbereitung, Ausführen der CATO-Pipeline, Qualitätskontrolle und Second-Level-Analyse sowie Visualisieren der Ergebnisse.

Kapitel 3. PrePro-Datenvorbereitung

Im Idealfall liegen die Daten bereits vorverarbeitet mithilfe von FreeSurfer, FSL oder anderen etablierten Tools zur Datenvorverarbeitung vor, da CATO für die kortikale Rekonstruktion vorverarbeitete T1-Daten benötigt. Ein großer Vorteil von CATO ist seine modulare Struktur, die eine flexible Anpassung der Pipeline ermöglicht. Beispielsweise kann die kortikale Rekonstruktion mit FreeSurfer direkt in den Preprocessing-Schritt von CATO integriert werden. Wie bereits erwähnt, wird dies jedoch aufgrund der rechenökonomischen Effizienz nicht empfohlen. Für Nutzer, die FreeSurfer einsetzen möchten, sei darauf hingewiesen, dass FreeSurfer bereits auf dem Server installiert ist. In diesem Zusammenhang empfehle ich die die Skripte des ENIGMA-Protokolls, unter /net/scratch/GROUPS/for2107/for2107 project/02 material and methods/02 code/02 Freesurfer/ verfügbar sind, falls noch keine Berechnung der benötigten Daten stattgefunden hat. Auf die Details dieser Skripte und die Verarbeitung mit FreeSurfer wird hier nicht weiter eingegangen, da davon ausgegangen wird, dass die benötigten Daten bereits vorliegen und einer Qualitätskontrolle (QC) unterzogen wurden. Sollte dies nicht der Fall sein, ist die Durchführung einer Vorverarbeitung mit FreeSurfer, FSL oder alternativen Tools sowie einer QC der vorverarbeiteten Daten der notwendige erste Schritt, bevor mit der CATO-Analyse fortgefahren werden kann.

Daten Voraussetzungen

Ihr benötigt, wie bereits beschrieben, die kortikalen Rekonstruktionen aus vorverarbeiteten T1-Daten sowie die drei DWI-Dateien: *dwi.nii.gz* (Diffusionsbilder(b0)), *dwi.bvals* (B-Werte) und *dwi.bvecs* (B-Vektoren). Diese Dateien können je nach Arbeitsgruppe in unterschiedlichen Strukturen und mit variierenden Benennungen sowie Formaten vorliegen. Idealerweise sollten die Daten im BIDS-Format organisiert sein, da dieses Format als Standard in vielen neuroimaging Analysen gilt.

Um CATO seriell für mehrere Probanden auszuführen, ist eine einheitliche Benennung der Probanden erforderlich. Dies ist wichtig, da das verwendete Skript, das auf die CATO.conf-Datei zugreift, bei abweichenden Namenskonventionen auf Probleme stoßen könnte. Werden hingegen einzelne Probanden analysiert, kann dies problemlos über den bereitgestellten Configuration Assistant erfolgen:

(http://www.dutchconnectomelab.nl/CATO/docs/configuration_assistant.html).

Zusammenfassend benötigt ihr also die FreeSurfer-Daten sowie die entsprechenden DWI-Dateien (*dwi.nii.gz, dwi.bvals, dwi.bvecs*), die jeweils für den entsprechenden Run bzw. die Session vorliegen müssen.

Programmvoraussetzungen

Für die Nutzung der CATO-Toolbox werden folgende Programme benötigt: MATLAB Compiler 6.5 (R2017b), FreeSurfer-7.4.1, FSL-6.0.7.6, CATO-3.2.2, (BrainNet Viewer-1.7).

Im Folgenden wird ein detailliertes Step-by-Step-Beispiel zur Vorbereitung und Ausführung der CATO-Toolbox (CATO_RUN_EXAMPLE) beschrieben. Dabei müsst ihr in den Skripten anstelle von Platzhaltern (XXX) eure konkreten Pfade eintragen. Bitte beachtet, dass die hier verwendeten Ordnernamen lediglich Beispiele sind und in eurem Fall abweichen können. Grundvoraussetzung ist, dass ihr bereits die benötigten DWI-Daten und FreeSurfer-Ergebnisse habt.

Für die richtige Analyse CATO_RUN Ordner verwenden !!!

Organisation der Daten

Zieht den Ordner CATO_MANUAL bitte in euer /net/scratch/XXX-Verzeichnis.

(Bitte darauf achten, dass dies eine Anleitung für ein Beispiel und die richtige Analyse ist. Überprüft entsprechend die Pfade und passt diese ggf. an CATO_RUN oder CATO_RUN_EXAMPLE an.)

Schritt 1: Legt eure DWI-Daten in den Ordner 1_Input_DWI ab.

- Achtet darauf, dass die Daten bereits so vorbereitet sind, dass jeweils der passende Run/Session und die zugehörigen DWI-Dateien (dwi.nii.gz, dwi.bvals, dwi.bvecs) pro Proband enthalten sind.
- Beispielhafte Ordnerstruktur (Namen können variieren):



 Achtung: Es ist sehr hilfreich, wenn ihr pro Proband einen eigenen DWI-Ordner anlegt. Für die Umbenennung können einfache Bash-Skript verwendet werden. Zum Beispiel so eins (es muss natürlich an eure Datenstruktur angepasst werden):

```
# Quellverzeichnis mit den DWI-Dateien
SOURCE DIR="/path/to/source directory"
# Zielverzeichnis, in dem die strukturierte Ordnung erstellt wird
TARGET DIR="/path/to/1 Input DWI"
# Schleife über alle Dateien im Quellverzeichnis
for file in "$SOURCE DIR"/*; do
    # Extrahiere die Probanden-ID aus dem Dateinamen (z. B. "sub 1"
aus "sub 1.bvals")
   base name=$(basename "$file")
   subject id=$(echo "$base name" | cut -d'.' -f1) # Teil vor dem
Punkt
    # Erstelle das Zielverzeichnis für den Probanden und den DWI-
Ordner
   mkdir -p "$TARGET DIR/$subject id/DWI"
    # Verschiebe die Datei in den entsprechenden DWI-Ordner
   mv "$file" "$TARGET DIR/$subject id/DWI/"
done
```

Schritt 2: Legt die FreeSurfer-Ergebnisse in den Ordner 2 Input FreeSurfer ab.

- Wichtig: Die Namen der Probandenordner müssen exakt mit denen aus dem DWI-Input übereinstimmen, da das Skript sonst nicht funktioniert.
- Beispielhafte Ordnerstruktur:

```
2_Input_FreeSurfer/
    — sub 1/
       L____ FreeSurfer 1/
             --- label/
               - mri/
              — scripts/
               - stats/
               - surf/
               - tmp/
               — trash/
    - sub 12/
       L___ FreeSurfer 12/
            (entsprechende Ordnerstruktur)
     - sub 123/
       L___ FreeSurfer 123/
             (entsprechende Ordnerstruktur)
```

• Wichtig: Die interne Ordnerstruktur muss so beibehalten werden, dass nicht einfach alle FreeSurfer Datein direkt in den Probandenordner kopiert werden. Es wird erwartet, dass der Name der Probandenordner einheitlich ist und anschließend der interne FreeSurfer-Ordner (z. B. "FreeSurfer_1") folgt so wie in dem Bsp.

Schritt 3: Anpassen des Merge-Skriptes

- Öffnet das Skript merge_folders.sh (das sich in eurem CATO_RUN-Verzeichnis befinden sollte).
- Ersetzt die Platzhalter (XXX) bei den Quellpfaden durch eure tatsächlichen Pfadangaben.

```
Beispiel:
dwi_dir="/home/users/lepperja/CATO_MANUAL/CATO_RUN/1_Input_DWI"
fs_dir="/home/users/lepperja/CATO_MANUAL/CATO_RUN/2_Input_FreeSurfer"
merged_dir="/home/users/lepperja/CATO_MANUAL/CATO_RUN/3_Merged"
```

Speichert das Skript ab.

Schritt 4: Mach das Merge-Skript ausführbar

• Öffnet ein Terminal im CATO_RUN(EXAMPLE)-Verzeichnis und gebt folgenden Befehl ein:

chmod +x 4_merge_folders.sh

Schritt 5: Ausführen des Merge-Skriptes

• Führt das Skript aus, indem ihr im selben Terminal eingibt:

./4_merge_folders.sh

• Das Skript führt die beiden Datensätze (DWI + FreeSurfer) zusammen und erstellt in 3 Merged die nötige Ordnerstruktur für die weitere Analyse.

Schritt 6: Anpassen des List-Skriptes

- Öffnet das Skript 5_list_merged_folders.sh und passt die entsprechenden Input/Output Pfade in dem Skript an.
- Speichert das Skript ab.
- Das Skript erstellt ein Textdokument, das bei der späteren Umbenennung der Ordner hilft. Dieses Dokument dokumentiert die aktuelle Namenskonvention, sodass eine Zuordnung möglich ist.

Schritt 7: Mach das List-Skript ausführbar

• Öffnet ein Terminal im CATO_RUN-Verzeichnis und führt aus:

chmod +x 5_list_merged_folders.sh

Schritt 8: Ausführen des List-Skriptes

• Gebt im Terminal ein:

 $./5_list_merged_folders.sh$

 Um CATO seriell f
ür mehrere Probanden auszuf
ühren, ist eine einheitliche Benennung der Probanden erforderlich. Dies ist wichtig, da das verwendete Skript, das auf die CATO.conf-Datei zugreift, bei abweichenden Namenskonventionen auf Probleme stößt.

Schritt 9: Anpassen des Rename-Skriptes

ACHTUNG: Das Skript benennt die Daten **gnadenlos** um! Bitte testet vorher, ob die Benennung korrekt ist. Es dürfen keine anderen Ordner in 3_Merged liegen, außer die Probanden-Ordner, da diese sonst ebenfalls umbenannt werden!

- Öffnet das Skript 6_rename_merged_data.sh und passt den Pfad bei BASE_DIR so an, dass er zu eurem 3 Merged-Ordner führt.
- Speichert das Skript ab.

Schritt 10: Das Rename-Skript ausführbar machen

• Öffnet ein Terminal im CATO RUN (EXAMPLE) - Verzeichnis und gebt folgendes ein:

chmod +x 6_rename_merged_data.sh

Schritt 11: Ausführen des Rename-Skriptes

• Startet das Skript im Terminal durch Eingabe von:

./6_rename_merged_data.sh

• Ihr werdet aufgefordert, spezifische Informationen zur Logik eurer Datennamen einzugeben. Zum Beispiel:

```
Bitte gib den Basisnamen für DWI-Dateien ein (z.B. sub_): sub
Bitte gib den Zielnamen für den FreeSurfer-Ordner ein (z.B.
FreeSurfer): Free
```

• Der Präfix kann bei euch variieren. Nachdem das Skript erfolgreich ausgeführt wurde, sind alle zusammengeführten Daten einheitlich benannt und bereit für die CATO-Analyse.

Kapitel 4. Ausführen der CATO-Pipeline

Schritt 12: Anpassen der Konfigurationsdatei

- Öffnen der Datei: Öffnet die Datei CATO.conf mit dem "Text Editor", die sich im Ordner 3_Merged befindet.
- Überprüfen und Anpassen grundlegender Parameter: Stellt sicher, dass die folgenden Parameter korrekt gesetzt sind (in der Regel müssen diese Einträge nicht verändert werden und können unverändert bleiben):

"fslRootDir": "/net/shared/fsl/fsl-6.0.7.6/",

```
"freesurferRootDir": "/usr/local/freesurfer/7.4.1/",
"rawBvalsFile": "DWI/SUBJECT_dwi.bvals",
"rawBvecsFile": "DWI/SUBJECT_dwi.bvecs",
"dwiFile": "DWI/SUBJECT dwi.nii.gz"
```

• Wichtig ! Der Pfad muss an den Speicherort der Toolbox angepasst werden. Beispiel: "preprocessingScript": "/net/scratch/lepperja/CATO_MANUAL/Cato/structural_preprocessing/prep rocess eddy.sh",

• Optionale Parameter:

Falls vorhanden, könnt ihr auch Dateien für Reversed b0-weighted Scans oder Reversed DWI angeben.

Unter den Template-Einstellungen in der CATO.conf Datei könnt ihr die gewünschten Atlanten auswählen, die für eure Analyse verwendet werden sollen.

```
Zum Beispiel: "templates": [
    "aparc",
    "lausanne120",
    "lausanne250" ]
```

• Diffusionsrekonstruktion:

Unter reconstructionMethods wählt ihr die Methode aus, die zur Diffusionsrekonstruktion herangezogen werden soll (z. B. DTI, CSD, GQI oder eine Kombination).

Unter reconstructionSteps legt ihr fest, welche Verarbeitungsschritte durchgeführt werden sollen.

• Preprocessing-Skript :

Optional könnt ihr den Parameter preprocessingScript anpassen. Ihr könnt dabei eine der vorgegebenen Vorlagen (Minimal, Eddy, Topup Eddy) verwenden oder ein eigenes Skript erstellen.

```
Beispiel: "preprocessingScript":
    "XXX/structural preprocessing/preprocess minimal.sh"
```

• Eddy-Konfiguration und Phase-Encoding:

Weitere Parameter betreffen das verwendete eddy-Programm. Hier könnt ihr angeben, ob

beispielsweise 'eddy', 'eddy_openmp' oder 'eddy_cuda' verwendet werden soll wobei 'eddy' der Standard ist und bei keiner Parallelisierung genutzt werden sollte.

Der Parameter "revPhaseEncDim": 1 steht für die Phase-Encoding-Richtung. Bei einer DWI-Erfassung in RAS-Orientierung gibt 1 an, dass die Phase-Encoding in $R \rightarrow L$ (rechts nach links) erfolgt.

• FreeSurfer-Daten:

Der Parameter "forceFreesurferOverwrite": false sollte angepasst werden, falls ihr plant, die FreeSurfer-Daten bei Bedarf zu überschreiben.

• Faser- und Netzwerkrekonstruktion:

Für die Faserrekonstruktion sind Parameter wie NumberOfSeedsPerVoxel, minFA, maxAngleDeg, startRegions, forbidd enRegions und stopRegions relevant.

Ebenso legt ihr in diesem Abschnitt Parameter für die Netzwerkrekonstruktion fest, wie minLengthMM.

• Weiterführende Informationen:

Da die einzelnen Parameter und ihre Auswirkungen bereits in Kapitel 3 grob beschrieben wurden und eine detaillierte Erläuterung den Rahmen dieses Manuals sprengen würde, verweise ich an dieser Stelle auf die vollständige Dokumentation: http://www.dutchconnectomelab.nl/CATO/docs/structural_configuration.html#.

Schritt 13: Speichern der CATO.conf

• Nachdem ihr alle relevanten Parameter angepasst habt, speichert die CATO.conf-Datei.

Ausführung der CATO Pipeline

Schritt 14: Anpassen des Run-Skriptes

• Öffnet das Skript run_pipeline_for_all_subjects.sh und passt den Parameter base dir so an, dass er auf euren Datenordner verweist.

Schritt 15: Mach das Run-Skript ausführbar

• Navigiert im Terminal in das 3_Merged-Verzeichnis und gebt ein:

chmod +x run_pipeline_for_all_subjects.sh

Schritt 16: Kopieren des CATO-Ordners

 Kopiert den Ordner "Cato" (den CATO-Source-Code) in euer net/scratch-Verzeichnis, falls dies noch nicht geschehen ist oder lasst ihn im /net/scratch/XXX/CATO_MANUAL/Cato Ordner.

Schritt 17: Setzen der Umgebungsvariablen

- Navigieren Sie zurück zum Ordner CATO_RUN bzw. CATO_RUN_EXAMPLE.
- Öffnet das Skript set_env.sh und passt den Pfad für setfree an, z. B.:

setfree /home/users/lepperja/CATO_MANUAL/CATO_RUN_EXAMPLE/3_Merged

• Speichert das Skript ab und macht es ausführbar wieder den Terminal im 3_Merged-Verzeichnis öffnen:

chmod +x set_env.sh

• Führt das Skript im Terminal aus, indem ihr eingebt:

source set_env.sh

Schritt 18: Ausführen der Pipeline

• Führt im selben Terminal das Run-Skript aus:

./run pipeline for all subjects.sh

• Für jeden Probanden wird nun ein neuer Ordner DWI_processed erzeugt, der die verarbeiteten Ergebnisse enthält.

Schritt 19: Abschluss

- Die Pipeline ist abgeschlossen, wenn in jedem DWI_processed-Ordner die Datei structural pipeline.finished zu finden ist.
- Anschließend sollte eine QC durchgeführt werden, die unter Schritt 23 beschrieben ist.

Die funktionale Pipeline funktioniert mit kleinen Anpassungen nach demselben Schema.

Kapitel 5. Abschluss der Datenvorbereitung und Beispielanalyse

Nachdem ihr die Daten erfolgreich vorbereitet habt, könnt ihr diese je nach Bedarf analysieren. Im Folgenden wird eine Beispielanalyse beschrieben. Für Interessierte sind die Ausgabe- sowie Analysemöglichkeiten im Anhang in zwei Tabellen dargestellt, wobei *subject1* als Beispiel dient. Natürlich können/müssen die Daten auf Gruppenebenen analysiert werden.

Analyse der Daten

Vor der Analyse sollte eine QC durchgeführt werden. Dafür werden die zusammengefassten Daten der Probanden in Tabellenform benötigt. Die folgenden Schritte beschreiben, wie diese Tabellen erstellt werden können.

Schritt 20: Erstellen von Gruppentabellen:

- Skripte: Diese Skripte die im Ordner: SCRIPTS_ANALYSE liegen, kombinieren die Konnektivitätsdaten aller Probanden und erstellen Gruppendateien pro Gewicht, um eine spätere Analyse zu erleichtern.
- combine_connectivity_data.m: MATLAB-Skript zum Laden und Kombinieren der Konnektivitätsdaten aller Probanden und Erstellen von Gruppendateien pro Gewicht.
- combine_fiber_properties_data.m: MATLAB-Skript zum Laden und Kombinieren der Fiber-Properties-Daten aller Probanden.
- extract_connectivity_matrices.m: MATLAB-Skript zur Extraktion und Speicherung der per-Proband-Konnektivitätsmatrizen in Excel/CSV.

(Mit "Gewicht" ist eine spezifische Eigenschaft oder Metrik gemeint, die den Verbindungen zwischen Hirnregionen zugeordnet ist, wie z. B. "number of streamlines", "length", "fractional anisotropy")

- **Hinweis:** In diesem Manual wird nicht erklärt, wie jedes Skript funktioniert. Die Skript enthalten jedoch Kommentare, die jede Zeile erläutern. Bei Interesse könnt ihr diese Kommentare lesen und die Skript nach Bedarf anpassen.
- Wichtig: Achtet stets darauf, welche Methode ihr zur Rekonstruktion verwendet habt, und passt die Skripte gegebenenfalls entsprechend an.

Schritt 21: Ausführen der Skripte in MATLAB:

- Öffnet MATLAB auf dem Server mit dem Befehl matlab im Terminal.
- Ladet das Skript combine_connectivity_data.m und passt die folgenden Parameter an:

```
% Anzahl der Probanden
numSubjects = 3;
% Pfad zu den Daten
folderPath =
'/home/users/lepperja/CATO_MANUAL/CATO_RUN_EXAMPLE/3_Merged';
```

• Führt das Skript aus, indem ihr auf "Run" klickt (das gilt für alle MATLAB Skripte). Es wird eine kombinierte .mat-Datei pro Gewicht erstellt. Die Gewichte sind in MATLAB unter weightDescriptions zu finden. Mit diesen Daten könnt ihr nun weitere Analysen durchführen, da sie beispielsweise als 1x313 82x82 double strukturiert sind, wobei 82 die ROIs darstellen und n=313 die Anzahl der Probanden repräsentiert. Der MATLAB-Auszug stammt aus der

Datei group_connectivity_csd_aparc_Weight7.mat, wobei Weight7 für 'generalized fractional anisotropy' steht.

ROIs_list 1x313 237880 cell group_connectivity 1x313 235749096 cell group_connectivity_w 1x313 16869448 cell regionDescriptions 1x313 3736594 cell validSubjects 1x313 38600 cell weightDescriptions 1x313 708006 cell

• Alternative: Dasselbe Vorgehen kann bei Bedarf auch mit den Faser-Daten durchgeführt werden, indem ihr das Skript combine_fiber_properties_data.m verwendet und die entsprechenden Parameter anpasst.

Schritt 22: Erstellen von CSV-Tabellen:

- Zur Erleichterung der Berichterstellung könnt ihr CSV-Tabellen erstellen. Verwendet hierfür das MATLAB-Skript extract_connectivity_matrices.m, das die Konnektivitätsmatrizen pro Proband in Excel/CSV-Dateien extrahiert und speichert.
- Anpassung: Auch hier muss der Parameter dataDir entsprechend angepasst werden.

Schritt 23: Qualitätskontrolle (QC):

Alle Skripte zur Qualitätskontrolle befinden sich im Ordner $scripts_Analyse/$. Folgende Schritte sind zu beachten:

Schritt 24: Einfaches QC-Skript anwenden:

- **Skript:**QC_CATO.m
- Beschreibung: Dieses Skript wendet ein einfaches QC-Verfahren, nach Van Den Heuvel et al., 2019 an. Es berücksichtigt vier Metriken: MeanStreamlines, MeanFA, PrevalenceConnections, und ConnectedRegions. Eine IQR-basierte Ausreißererkennung wird durchgeführt.
- Ausgabe: Die Daten werden im neuen Ordner QC_CATO gespeichert.
- Anpassung: Öffnet das Skript und passt den Parameter dataDir an euren Datenpfad an.

Schritt 25: Visualisierung der QC-Ergebnisse:

- **Skript:**visualize_qc_metrics.m
- **Beschreibung:** Dieses Skript visualisiert die Verteilungen der QC-Metriken inkl. Ausreißer. Nach dem Laden aller QC-Ergebnisdateien aus QC_CATO/ werden alle Daten (inkl. Outliers) geplottet.
- Anpassung: Passt den Pfad im Skript entsprechend an.

Schritt 26: Zusammenfassung der Ausreißer:

• **Skript:**QC_sum_Outliers.m

- **Beschreibung:** Dieses Skript fasst alle Ausreißer aus den QC-Ergebnisdateien zusammen und gibt den Grund für die Ausreißer an (welche Metrik(en) die Ausreißerbedingung ausgelöst haben).
- Anpassung: Passt den Pfad im Skript entsprechend an.

Optionale Erweiterung:

• Künstliche Ausreißer hinzufügen: Für Testzwecke können künstliche Ausreißer hinzugefügt werden, indem ihr die Tabellen aus dem Ordner Künstliche_Out in den Ordner QC_CATO überführt. Dies ermöglicht die Überprüfung der Robustheit der QC-Skripte.

Robustheitstests:

• **Hinweis:** Es können auch Robustness-Checks für Netzwerkbasierte Statistik (NBS) Analysen durchgeführt werden, dies wurde jedoch in diesem Beispiel nicht berücksichtigt.

Nachdem die Ausreißer händisch oder mit einem Skript, entfernt wurden, könnt ihr mit der weiteren Analyse fortfahren.

Kapitel 6. CATO Second Level Analyse

Schritt 27: Thresholding der Konnektivitätsdaten:

- **Skript:**Threshold_3.m
- Beschreibung: Setzt im ersten Layer ("number of streamlines") alle Werte < 3 auf 0, um die Sensitivität und Spezifität der Konnektivitätsmatrix zu gewährleisten.
- Anpassung: Passt die Input- und Output-Pfade im Skript an.

Export der Daten zu CSV:

- **Skript:**export_Filtered_To_CSV.m
- **Beschreibung:** Exportiert die gefilterten Daten zu CSV, um das Arbeiten zu erleichtern.
- Anpassung: Passt die Pfade im Skript an.
- Simulation von Kovariaten und Faktoren:
- Es wurde eine fiktive Faktorenanalyse mit fünf Faktoren sowie simulierte Kovariaten (Age, Sex, ScannerSite, TIV, HeadMotion) durchgeführt, um die Daten weiter zu analysieren. Diese simulierten Daten befinden sich in den Dateien Simulated_Covariates.xlsx und Simulated_Factors_313_Participant s.xlsx. die im Ordner Künstliche Out zu finden sind.

Schritt 28: Graphentheoretische Analyse:

- Toolbox: BCT (Brain Connectivity Toolbox)
- Die BCT Toolbox liegt mit im Hauptverzeichnis wo auch CATO zu finden ist: /net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/0 2_code/01_CATO/CATO_MANUAL/BCT

- Einrichtung:
- Fügt die BCT-Toolbox zu MATLAB hinzu:

addpath('euer pfad/BCT');

• Überprüft die Installation mit:

help distance_wei

- **Skript:**compute_Graph_Metrics_All_Subjects.m
- **Beschreibung:** Führt für jede gefundene *_filtered.mat-Datei eine Rich-Club-Analyse und Berechnung graphentheoretischer Maße durch.
- Bestimmt die obersten 15% der Knoten als Rich-Club (RC).
- Kategorisiert die Kanten in RC-Verbindungen, Feeder und Lokalverbindungen.
- Berechnet graphentheoretische Maße wie Pfadlänge (L), globale Effizienz (GE) und Clustering-Koeffizient (C).
- Erzeugt zufällige Netzwerke zur Normalisierung (Lnorm, GEnorm, Cnorm).
- Berechnet den Small-World-Index (SW = Cnorm / Lnorm).
- Speichert alle Ergebnisse in Subject_Graph_Metrics/subjectX/..._allMetrics.mat.
- Anpassung: Passt die Pfade im Skript entsprechend an.

Schritt 29: Weitere Analysen:

Optionale weitere Analysen wie diese:

Das analysis_all_subjects-Skript analysiert Netzwerkmetriken (z. B. Effizienz, Clustering, Small-Worldness) aus den Konnektivitätsdaten aller Probanden. Dabei kombiniert es diese Metriken mit Teilnehmerdaten (z. B. Age, Sex, usw.) aus einer CSV-Datei. Anschließend führt das Skript robuste Regressionsanalysen durch, um den Einfluss mehrerer Faktoren (z. B. psychologische Dimensionen) auf die Netzwerkmetriken zu untersuchen. Schließlich werden die Ergebnisse mittels BH-FDR Korrektur bereinigt, um verlässliche p-Werte trotz Mehrfachvergleichen zu gewährleisten.

- **Skript:**analysis_all_subjects.m
- **Beschreibung:** Führt umfassende Analysen über alle Probanden durch, einschließlich Robuster Regression und BH-FDR-Korrektur der p-Werte.
- Anpassung: Passt die Pfade im Skript entsprechend an.
- Achtung: Mit den bereitgestellten Beispieldaten ist die Analyse nicht möglich, da nur drei Probanden vorhanden sind und dies nicht ausreichend Datenpunkte für statistisch valide Ergebnisse liefert.

Schritt 30: Visualisierung der Ergebnisse:

- BrainNet Viewer
- Im letzten Schritt folgt die Visualisierung mit dem BrainNet Viewer.
- Einrichtung:
- Fügt den BrainNet Viewer zu MATLAB hinzu:

addpath('/net/shared/matlab/BrainNetViewer');

- Startet den BrainNet Viewer mit: BrainNet.
- Manual: Ein Handbuch für den BrainNet Viewer ist im

Ordner /net/scratch/GROUPS/for2107/for2107_project/02_material_and_me thods/02_code/01_CATO/CATO_MANUAL/BNV/ abgelegt und kann bei Bedarf genutzt werden. Die Daten müssen dafür umgewandelt werden siehe Handbuch.



Schritt 31: Rückbenennung der Daten (Optional)

- **Skript:**rerename_subjects.sh
- Beschreibung: Ermöglicht das Zurückbenennen der obersten Ordner.
- Anpassung: Passt den Pfad und die Namen der Textdatei sowie die Namen für die Analyse entsprechend an.

Schritt 32: Abschluss

- **Aufräumen:** Vergesst nicht, alle unnötigen Ordner und Dateien zu löschen oder Programme zu entfernen, die doppelt vorhanden sind.
- Für Interessierte ist weitere Literatur, wie ein Beispiel-Paper, hier zu finden: /net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/0 2_code/01_CATO/CATO_MANUAL/CATO_LIT
- Viel Spaß beim Ausprobieren und Analysieren eurer Daten mit der CATO-Toolbox!

Nur Anleitung ohne Erklärung für Fortgeschrittene

Schritt 1: Organisation der DWI-Daten

- Legt eure DWI-Daten in den Ordner 1_Input_DWI ab.
- Beispielhafte Ordnerstruktur:



• Achtung: Pro Proband einen eigenen DWI-Ordner anlegen.

Schritt 2: Organisation der FreeSurfer-Daten

- Legt die FreeSurfer-Ergebnisse in den Ordner 2_Input_FreeSurfer ab.
- Beispielhafte Ordnerstruktur:



• Wichtig: Interne Ordnerstruktur beibehalten.

Schritt 3: Anpassen des Merge-Skriptes

- Öffnet das Skript merge folders.sh im CATO RUN-Verzeichnis.
- Ersetzt die Platzhalter (XXX) durch eure tatsächlichen Pfadangaben:

dwi_dir="/home/users/lepperja/CATO_MANUAL/CATO_RUN/1_Input_DWI"

fs_dir="/home/users/lepperja/CATO_MANUAL/CATO_RUN/2_Input_FreeSurfer"

merged_dir="/home/users/lepperja/CATO_MANUAL/CATO_RUN/3_Merged"

• Speichert das Skript ab.

Schritt 4: Mach das Merge-Skript ausführbar

• Öffnet ein Terminal im CATO RUN-Verzeichnis.

• Führt aus:

chmod +x 4 merge folders.sh

Schritt 5: Ausführen des Merge-Skriptes

- Gebt im Terminal ein:
- ./4_merge_folders.sh

Schritt 6: Anpassen des List-Skriptes

- Öffnet das Skript 5_list_merged_folders.sh.
- Passt die entsprechenden Pfade (XXX) an.
- Speichert das Skript ab.

Schritt 7: Mach das List-Skript ausführbar

- Öffnet ein Terminal im CATO RUN-Verzeichnis.
- Führt aus:

chmod +x 5_list_merged_folders.sh

Schritt 8: Ausführen des List-Skriptes

- Gebt im Terminal ein:
- ./5_list_merged_folders.sh

Schritt 9: Anpassen des Rename-Skriptes

- Öffnet das Skript 6 rename merged data.sh.
- Passt den Pfad bei BASE_DIR an eure Gegebenheiten an.
- Speichert das Skript ab.

Schritt 10: Mach das Rename-Skript ausführbar

- Öffnet ein Terminal im CATO RUN-Verzeichnis.
- Gebt ein:

chmod +x 6_rename_merged_data.sh

Schritt 11: Ausführen des Rename-Skriptes

- Startet das Skript im Terminal:
- ./6_rename_merged_data.sh
- Gebt die geforderten Präfixe ein:

Bitte gib den Basisnamen für DWI-Dateien ein (z.B. sub_): sub

Bitte gib den Zielnamen für den FreeSurfer-Ordner ein (z.B. FreeSurfer): Free

Schritt 12: Anpassen der Konfigurationsdatei

- Öffnet die Datei CATO.conf im Ordner 3 Merged.
- Überprüft und stellt sicher, dass folgende Parameter korrekt gesetzt sind:

```
"fslRootDir": "/net/shared/fsl/fsl-6.0.7.6/",
```

"freesurferRootDir": "/usr/local/freesurfer/7.4.1/",

"rawBvalsFile": "DWI/SUBJECT dwi.bvals",

"rawBvecsFile": "DWI/SUBJECT dwi.bvecs",

"dwiFile": "DWI/SUBJECT dwi.nii.gz"

- Wichtig ! Der Pfad muss an den Speicherort der Toolbox angepasst werden. Beispiel: "preprocessingScript": "/net/scratch/lepperja/CATO_MANUAL/Cato/structural_preprocessing/prep rocess eddy.sh",
- Optionale Parameter:

Reversed b0-weighted Scans oder Reversed DWI-Dateien angeben. Gewünschte Atlanten unter Template-Einstellungen auswählen.

• Diffusionsrekonstruktion:

Unter reconstructionMethods Methode auswählen (z.B. DTI, CSD, GQI). Unter reconstructionSteps Verarbeitungsschritte festlegen.

• Preprocessing-Skript :

Optional preprocessingScript **anpassen**:

```
"preprocessingScript":
"TOOLBOXDIR/structural_preprocessing/preprocess_minimal.s
h"
```

• Eddy-Konfiguration und Phase-Encoding:

Eddy-Programm angeben (eddy, eddy_openmp, eddy_cuda). Phase-Encoding-Richtung festlegen: "revPhaseEncDim": 1

• FreeSurfer-Daten:

Parameter "forceFreesurferOverwrite": false anpassen.

• Faser- und Netzwerkrekonstruktion:

Parameter

 $wie \ \texttt{NumberOfSeedsPerVoxel, minFA, maxAngleDeg, startRegions, for bidden Regions, stop Regions festlegen.}$

Netzwerkrekonstruktionsparameter wie minLengthMM einstellen.

 Weiterführende Informationen unter: http://www.dutchconnectomelab.nl/CATO/docs/structural_configuration.htm

Schritt 13: Speichern der CATO.conf

• Speichert die angepasste CATO.conf-Datei ab.

Schritt 14: Anpassen des Run-Skriptes

- Öffnet das Skript run pipeline for all subjects.sh.
- Passt den Parameter base dir an euren Datenordner an (XXX).

Schritt 15: Mach das Run-Skript ausführbar

- Navigiert im Terminal in das 3 Merged-Verzeichnis.
- Führt aus:

chmod +x run pipeline for all subjects.sh

Schritt 16: Kopieren des CATO-Ordners

• Kopiert den Ordner "Cato" in euer net/scratch-Verzeichnis

Schritt 17: Setzen der Umgebungsvariablen

- Öffnet das Skript set env.sh.
- Passt den Pfad für setfree an:

setfree XXX/CATO_MANUAL/CATO_RUN_EXAMPLE/3_Merged

- Speichert das Skript ab.
- Macht das Skript ausführbar:

chmod +x set_env.sh

• Führt das Skript aus:

source set_env.sh

Schritt 18: Ausführen der Pipeline

• Führt das Run-Skript im Terminal aus:

./run_pipeline_for_all_subjects.sh

Schritt 19: Abschluss

• Die Pipeline ist abgeschlossen, wenn in jedem DWI_processed-Ordner die Datei structural_pipeline.finished vorhanden ist.

Schritt 20: Erstellen von Gruppentabellen

- Skripte kombinieren die Konnektivitätsdaten aller Probanden und erstellen Gruppendateien pro Gewicht.:
- combine_connectivity_data.m
- combine_fiber_properties_data.m

• extract_connectivity_matrices.m

Schritt 21: Ausführen der Skripte in MATLAB

- Öffnet MATLAB auf dem Server.
- Ladet das Skript combine connectivity data.m
- Passt die folgenden Parameter an:

% Anzahl der Probanden
numSubjects = 3;
% Pfad zu den Daten
folderPath = 'XXX/CATO MANUAL/CATO RUN/3 Merged';

• Führt das Skript aus bei Bedarf auch combine_fiber_properties_data.m.

Schritt 22: Erstellen von CSV-Tabellen

• Verwendet das MATLAB-Skript extract_connectivity_matrices.m

Passt den Parameter dataDir an.

• Führt das Skript aus.

Schritt 23: Einfaches QC-Skript anwenden

- Öffnet das Skript QC CATO.m.
- Passt den Parameter dataDir an euren Datenpfad an.
- Führt das Skript aus.

Schritt 24: Visualisierung der QC-Ergebnisse

- Öffnet das Skript visualize_qc_metrics.m.
- Passt den Pfad im Skript an.
- Führt das Skript aus.

Schritt 25: Zusammenfassung der Ausreißer

- Öffnet das Skript QC_sum_Outliers.m.
- Passt den Pfad im Skript an.
- Führt das Skript aus.

Schritt 26: Optionale Erweiterung Künstliche Ausreißer hinzufügen

• Fügt die Tabellen aus dem Ordner Künstliche Out in den Ordner QC CATO ein.

• Führt das QC-Skript erneut aus.

Schritt 27: Thresholding der Konnektivitätsdaten

- Öffnet das Skript Threshold 3.m.
- Passt die Input- und Output-Pfade an.
- Führt das Skript aus.

Schritt 28: Export der Daten zu CSV

- Öffnet das Skript export_Filtered_To_CSV.m.
- Passt die Pfade an.
- Führt das Skript aus.

(Schritt 29: Simulation von Kovariaten und Faktoren)

 Verwendet die Dateien Simulated_Covariates.xlsx und Simulated_Factors_313_Participant s.xlsx im Ordner Künstliche Out.

Schritt 30: Graphentheoretische Analyse

• Fügt die BCT-Toolbox zu MATLAB hinzu:

addpath('euer_pfad/BCT');

• Überprüft die Installation :

help distance_wei

- Öffnet das Skript compute_Graph_Metrics_All_Subjects.m.
- Passt die Pfade im Skript an.
- Führt das Skript aus.

Schritt 31: Weitere Analysen (Optional)

- Öffnet das Skript analysis_all_subjects.m.
- Passt die Pfade im Skript an.
- Führt das Skript aus.

Schritt 32: Visualisierung der Ergebnisse

• Fügt den BrainNet Viewer zu MATLAB hinzu:

```
addpath('/net/shared/matlab/BrainNetViewer');
```

• Startet den BrainNet Viewer:

BrainNet

• Verwendet das Handbuch im Ordner /net/scratch/GROUPS/for2107/for2107_project/02_material_and_met hods/02_code/01_CATO/CATO_MANUAL/BNV/.

Schritt 33: Rückbenennung der Ordner

Öffnet das Skript rerename_subjects.sh.

- Passt den Pfad und die Namen der Textdatei sowie die Namen für die Analyse an.
- Führt das Skript aus.

Schritt 34: Aufräumen

- Löscht alle unnötigen Ordner und Dateien.
- Entfernt doppelt vorhandene Programme.
- Für Interessierte ist weitere Literatur, wie ein Beispiel-Paper, hier zu finden: /net/scratch/GROUPS/for2107/for2107_project/02_material_and_methods/0 2_code/01_CATO/CATO_MANUAL/CATO_LIT
- Viel Spaß beim Ausprobieren und Analysieren eurer Daten mit der CATO-Toolbox!

Die funktionale Pipeline funktioniert mit kleinen Anpassungen nach demselben Schema.

Literaturverzeichnis

Bassett, D. S., & Sporns, O. (2017). Network neuroscience. *Nature Neuroscience*, *20*(3), 353–364. https://doi.org/10.1038/nn.4502

De Lange, S. C., Helwegen, K., & Van Den Heuvel, M. P. (2023). Structural and functional connectivity reconstruction with CATO - A Connectivity Analysis TOolbox. *NeuroImage*, *273*, 120108. https://doi.org/10.1016/j.neuroimage.2023.120108

- Friston, K. J. (2011). Functional and Effective Connectivity: A Review. *Brain Connectivity*, 1(1), 13–36. https://doi.org/10.1089/brain.2011.0008
- Huettel, S. A., Song, A. W., & McCarthy, G. (2009). *Functional magnetic resonance imaging* (2. ed). Sinauer.
- Stein, F., Gruber, M., Mauritz, M., Brosch, K., Pfarr, J.-K., Ringwald, K. G., Thomas-Odenthal,
 F., Wroblewski, A., Evermann, U., Steinsträter, O., Grumbach, P., Thiel, K., Winter, A.,
 Bonnekoh, L. M., Flinkenflügel, K., Goltermann, J., Meinert, S., Grotegerd, D., Bauer,
 J., ... Kircher, T. (2023). Brain Structural Network Connectivity of Formal Thought
 Disorder Dimensions in Affective and Psychotic Disorders. *Biological Psychiatry*,
 95(7), 629–638. https://doi.org/10.1016/j.biopsych.2023.05.010
- Van Den Heuvel, M. P., Scholtens, L. H., Van Der Burgh, H. K., Agosta, F., Alloza, C., Arango,
 C., Auyeung, B., Baron-Cohen, S., Basaia, S., Benders, M. J. N. L., Beyer, F., Booij, L.,
 Braun, K. P. J., Filho, G. B., Cahn, W., Cannon, D. M., Chaim-Avancini, T. M., Chan, S. S.
 M., Chen, E. Y. H., ... Lange, S. C. D. (2019). 10Kin1day: A Bottom-Up Neuroimaging
 Initiative. *Frontiers in Neurology*, *10*, 425. https://doi.org/10.3389/fneur.2019.00425

Anhang

Dateiname	Beschreibung
acqp.txt	Enthält die Akquisitionsparameter, die zur
	Korrektur von Suszeptibilitätsartefakten mit
	FSL Topup verwendet werden können.
bvals_processed.txt	Verarbeitete b-Werte, die die
	Diffusionsgewichtung für jede Aufnahme
	angeben.
bvecs_processed.txt	Verarbeitete b-Vektoren, die die Richtung
	der Diffusionsgewichtung für jede Aufnahme
	beschreiben.
index.txt	Ordnet jeder Aufnahme die entsprechende
	Zeile in der acqp-Datei zu, um Bewegungs-
	und Verzerrungskorrekturen durchzuführen.
structural_configuration.json	JSON-Datei mit den
	Konfigurationsparametern, die für die
	Verarbeitung des aktuellen Datasets
	verwendet wurden.
structural_pipeline.finished	Signalisiert, dass die Pipeline erfolgreich
	abgeschlossen wurde.
structural_pipeline.log	Protokolldatei mit Details zu den
	Verarbeitungsschritten und möglichen
	Fehlermeldungen.
<pre>subject1_aparc+aseg_b0.nii.gz</pre>	Registrierte Freesurfer-Parzellierung
	(aparc+aseg) im DWI-B0-Referenzraum.
subject1_aseg_b0.nii.gz	Segmentierung der subkortikalen Regionen
	(aseg) im DWI-BO-Referenzraum.
subject1_axial_diffusivity.nii.gz	Axiale Diffusivität, eine Diffusionsmetrik, die
	die Diffusion entlang der Hauptrichtung des
	Wasserflusses beschreibt.
subject1_b0_to_freesurfer.dat	Registrierungsdatei, die die Transformation
	vom DWI-B0-Referenzbild in den Freesurfer-
	Raum beschreibt.
subject1_connectivity_csd_aparc.mat	Konnektivitätsmatrix (CSD-Methode)
	basierend auf der aparc-Parzellierung.
subject1_connectivity_csd_dti_aparc.mat	Konnektivitätsmatrix (CSD-DTI-Methode)
	basierend auf der aparc-Parzellierung.
subject1_connectivity_csd_dti_lausanne1	Konnektivitätsmatrix (CSD-DTI-Methode)
20.mat	basierend auf der Lausanne-120-
	Parzellierung.
subject1_connectivity_csd_dti_lausanne2	Konnektivitätsmatrix (CSD-DTI-Methode)
50.mat	basierend auf der Lausanne-250-
	Parzellierung.

subject1 connectivity csd lausanne120.	Konnektivitätsmatrix (CSD-Methode)
mat	basierend auf der Lausanne-120-
	Parzellierung
subject1 connectivity csd Jausanne250	Konnektivitätsmatrix (CSD-Methode)
mat	hasierend auf der Lausanne-250-
inat	Darzellierung
subject 1 connectivity dti anare mat	Konnoktivitätsmatrix (DTL Mathada)
subjecti_connectivity_uti_aparc.mat	hasierend auf der anarc-Darzellierung
subject1 connectivity dti lausanne120 m	Konnoktivitätsmatriv (DTI Mathada)
subjecti_connectivity_dti_ladsanne120.nn	hasiorond auf dor Lausanno 120
at	Dasielend auf der Lausanne-120-
subject connectivity dti laucenne250 m	Parzemerung. Konnoktivitätsmatriv (DTI Mathada)
subject1_connectivity_dti_lausanne250.m	konnektivitalsmatrix (DTI-iviethoue)
at	Dasierend auf der Lausanne-250-
	Parzellierung.
subject1_connectivity_gqi_aparc.mat	konnektivitätsmätrix (GQI-ivietnode)
	basierend auf der aparc-Parzeillerung.
subject1_connectivity_gqi_dti_aparc.mat	Konnektivitätsmätrix (GQI-DTI-Methode)
	basierend auf der aparc-Parzellierung.
subject1_connectivity_gqi_dti_lausanne1	Konnektivitätsmatrix (GQI-DTI-Methode)
20.mat	basierend auf der Lausanne-120-
	Parzellierung.
subject1_connectivity_gqi_dti_lausanne2	Konnektivitätsmatrix (GQI-DTI-Methode)
50.mat	basierend auf der Lausanne-250-
	Parzellierung.
subject1_connectivity_gqi_lausanne120.m	Konnektivitätsmatrix (GQI-Methode)
<pre>subject1_connectivity_gqi_lausanne120.m at</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120-
<pre>subject1_connectivity_gqi_lausanne120.m at</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode)
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250-
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat	Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-120-Parzellierung.Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-250-Parzellierung.MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD.Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (CQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi_dti.mat</pre>	Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-120-Parzellierung.Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-250-Parzellierung.MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD.Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi_dti.mat	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi_dti.mat	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-Methode) für alle Voxel, gespeichert als MATLAB-Datei.
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi.mat subject1_diffusion_peaks_gqi.mat</pre>	Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-120-Parzellierung.Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-250-Parzellierung.MATLAB-Datei mit Diffusionsmetriken füralle Voxel, z. B. FA, MD, RD und AD.Diffusionspeaks (CSD-DTI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (CSD-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (DTI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-DTI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Referenzbild basierend auf den B0-Volumendes DWI-Datensatzes.Vorverarbeitete DWI-Bilddaten (korrigiert
subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi.mat subject1_diffusion_peaks_gqi.mat	Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-120-Parzellierung.Konnektivitätsmatrix (GQI-Methode)basierend auf der Lausanne-250-Parzellierung.MATLAB-Datei mit Diffusionsmetriken füralle Voxel, z. B. FA, MD, RD und AD.Diffusionspeaks (CSD-DTI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (CSD-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (DTI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-DTI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Diffusionspeaks (GQI-Methode) für alleVoxel, gespeichert als MATLAB-Datei.Referenzbild basierend auf den B0-Volumendes DWI-Datensatzes.Vorverarbeitete DWI-Bilddaten (korrigiertfür Verzerrungen und Bewegungsartefakte).
<pre>subject1_connectivity_gqi_lausanne120.m at subject1_connectivity_gqi_lausanne250.m at subject1_diffusion_measures.mat subject1_diffusion_peaks_csd_dti.mat subject1_diffusion_peaks_csd.mat subject1_diffusion_peaks_dti.mat subject1_diffusion_peaks_gqi_dti.mat subject1_diffusion_peaks_gqi_mat subject1_diffusion_peaks_gqi.mat subject1_diffusion_peaks_gqi.mat subject1_diffusion_peaks_gqi.mat subject1_diffusion_peaks_gqi.mat</pre>	Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-120- Parzellierung. Konnektivitätsmatrix (GQI-Methode) basierend auf der Lausanne-250- Parzellierung. MATLAB-Datei mit Diffusionsmetriken für alle Voxel, z. B. FA, MD, RD und AD. Diffusionspeaks (CSD-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (CSD-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-DTI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Diffusionspeaks (GQI-Methode) für alle Voxel, gespeichert als MATLAB-Datei. Protokoll mit den von FSL eddy verwendeten

subject1_dwi.nii.gz.eddy_movement_rms	Datei mit RMS-Werten der
	Bewegungskorrektur.
<pre>subject1_dwi.nii.gz.eddy_outlier_map</pre>	Karte, die Ausreißer-Voxel im DWI-Datensatz
	identifiziert.
subject1_dwi.nii.gz.eddy_outlier_report	Bericht über Ausreißer-Voxel und deren
	Verteilung.
subject1_fractional_anisotropy.nii.gz	Bild der Fractional Anisotropy (FA), einer
	Diffusionsmetrik, die die
	Richtungsabhängigkeit der Diffusion
	beschreibt.
subject1_mean_diffusivity.nii.gz	Bild der mittleren Diffusivität (MD), die die
	durchschnittliche Diffusion innerhalb eines
	Voxels beschreibt.
subject1_radial_diffusivity.nii.gz	Bild der radialen Diffusivität (RD), die die
	Diffusion senkrecht zur Hauptrichtung
	beschreibt.
subject1_region_properties_aparc.mat	Eigenschaften der Regionen basierend auf
	der aparc-Parzellierung.
subject1_dwi.nii.gz.eddy_outlier_n_stdev	Karte, die die standardisierte Abweichung
_map	von Ausreißern in den DWI-Daten angibt.
subject1_dwi.nii.gz.eddy_outlier_report	Bericht über Ausreißer-Voxel, der eine
	detaillierte Übersicht über die Verteilung
	gibt.
subject1_dwi.nii.gz.eddy_parameters	Parameter, die von FSL Eddy während der
	Korrekturprozesse verwendet wurden.
subject1_dwi.nii.gz.eddy_post_eddy_shell	Parameter der Ausrichtung zwischen den
_alignment_parameters	DWI-Shells nach Eddy.
subject1_dwi.nii.gz.eddy_post_eddy_shell	Phase-Encoding-Translation zwischen
_PE_translation_parameters	verschiedenen Shells nach Eddy-Korrektur.
subject1_dwi.nii.gz.eddy_restricted_move	Root-Mean-Square-Werte (RMS) der
ment_rms	eingeschränkten Bewegungen.
subject1_dwi.nii.gz.eddy_values_of_all_in	Protokoll aller Eingabeparameter, die von
put_parameters	Eddy verwendet wurden.
subject1_fiber_properties_csd_aparc.mat	Eigenschaften der Fasersegmente (CSD-
	Methode) für die aparc-Parzellierung.
<pre>subject1_fiber_properties_csd_dti_aparc.</pre>	Eigenschaften der Fasersegmente (CSD-DTI-
mat	Methode) für die aparc-Parzellierung.
subject1_fiber_properties_csd_dti_lausan	Eigenschaften der Fasersegmente (CSD-DTI-
ne120.mat	Methode) für Lausanne-120.
subject1_fiber_properties_csd_dti_lausan	Eigenschaften der Fasersegmente (CSD-DTI-
ne250.mat	Methode) für Lausanne-250.
subject1_fiber_properties_csd_lausanne1	Eigenschaften der Fasersegmente (CSD-
20.mat	Methode) für Lausanne-120.
subject1_fiber_properties_csd_lausanne2	Eigenschaften der Fasersegmente (CSD-
50.mat	Methode) für Lausanne-250.
subject1_fiber_properties_dti_aparc.mat	Eigenschaften der Fasersegmente (DTI-
	Methode) für die aparc-Parzellierung.

subject1 fiber properties dti lausanne12	Figenschaften der Fasersegmente (DTI-
0.mat	Methode) für Lausanne-120.
subject1 fiber properties dti lausanne25	Eigenschaften der Fasersegmente (DTI-
0.mat	Methode) für Lausanne-250.
<pre>subject1_fiber_properties_gqi_aparc.mat</pre>	Eigenschaften der Fasersegmente (GQI-
	Methode) für die aparc-Parzellierung.
<pre>subject1_fiber_properties_gqi_dti_aparc.</pre>	Eigenschaften der Fasersegmente (GQI-DTI-
mat	Methode) für die aparc-Parzellierung.
<pre>subject1_fiber_properties_gqi_dti_lausan</pre>	Eigenschaften der Fasersegmente (GQI-DTI-
ne120.mat	Methode) für Lausanne-120.
<pre>subject1_fiber_properties_gqi_dti_lausan</pre>	Eigenschaften der Fasersegmente (GQI-DTI-
ne250.mat	Methode) für Lausanne-250.
<pre>subject1_fiber_properties_gqi_lausanne1</pre>	Eigenschaften der Fasersegmente (GQI-
20.mat	Methode) für Lausanne-120.
subject1_fiber_properties_gqi_lausanne2	Eigenschaften der Fasersegmente (GQI-
50.mat	Methode) für Lausanne-250.
subject1_fibers_csd_dti.trk	Rekonstruierte Fasern (CSD-DTI-Methode)
	im TRK-Format.
subject1_fibers_csd.trk	Rekonstruierte Fasern (CSD-Methode) im
	TRK-Format.
subject1_fibers_dti.trk	Rekonstruierte Fasern (DTI-Methode) im
	TRK-Format.
subject1_fibers_gqi_dti.trk	Rekonstruierte Fasern (GQI-DTI-Methode)
subject1_fibers_gqi_dti.trk	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format.
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format.
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt.
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum.
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum.
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchenenzitliche Diffusion henenziett
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt.
subject1_fibers_gqi_dti.trksubject1_fibers_gqi.trksubject1_fractional_anisotropy.nii.gzsubject1_fractional_anisotropy.nii.gzsubject1_lausanne120+aseg_b0.nii.gzsubject1_lausanne250+aseg_b0.nii.gzsubject1_mean_diffusivity.nii.gzsubject1_radial_diffusivity.nii.gz	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung beschreibt.
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz subject1_region_properties_aparc.mat</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung beschreibt. Eigenschaften der Regionen basierend auf der anare Parzolliorung
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz subject1_region_properties_aparc.mat</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung beschreibt. Eigenschaften der Regionen basierend auf der aparc-Parzellierung.
<pre>subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz subject1_region_properties_aparc.mat subject1_region_properties_lausanne120. mat</pre>	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung beschreibt. Eigenschaften der Regionen basierend auf der aparc-Parzellierung. Eigenschaften der Regionen basierend auf
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz subject1_region_properties_aparc.mat subject1_region_properties_lausanne120. mat	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den B0-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den B0-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung beschreibt. Eigenschaften der Regionen basierend auf der aparc-Parzellierung. Eigenschaften der Regionen basierend auf der Lausanne-120-Parzellierung.
subject1_fibers_gqi_dti.trk subject1_fibers_gqi.trk subject1_fractional_anisotropy.nii.gz subject1_lausanne120+aseg_b0.nii.gz subject1_lausanne250+aseg_b0.nii.gz subject1_mean_diffusivity.nii.gz subject1_radial_diffusivity.nii.gz subject1_region_properties_aparc.mat subject1_region_properties_lausanne120.mat subject1_region_properties_lausanne250.mod	Rekonstruierte Fasern (GQI-DTI-Methode) im TRK-Format. Rekonstruierte Fasern (GQI-Methode) im TRK-Format. Bild der Fractional Anisotropy (FA), die die Richtungsabhängigkeit der Diffusion beschreibt. Parzellierungsbild (Lausanne-120) registriert auf den BO-Referenzraum. Parzellierungsbild (Lausanne-250) registriert auf den BO-Referenzraum. Bild der mittleren Diffusivität (MD), die die durchschnittliche Diffusion beschreibt. Bild der radialen Diffusivität (RD), die die Diffusion senkrecht zur Hauptrichtung beschreibt. Eigenschaften der Regionen basierend auf der Lausanne-120-Parzellierung. Eigenschaften der Regionen basierend auf der Lausanne-120-Parzellierung.

	Analyse	Beschreibung	Dazugehörige Dateien
--	---------	--------------	----------------------

Graphentheoretisc	Berechnung von	Konnektivitätsmatrizen:
ne Analysen Netzwerkmaßen	 subject1_connectivity_csd_aparc.mat 	
	wie Grad, Dichte, Clustering- Koeffizient, Zentralitäten, globale Effizienz	- subject1_connectivity_csd_lausanne120.mat -
		subject1_connectivity_csd_lausanne250.mat
	und Small-World-	- subject1_connectivity_dti_aparc.mat
	Eigenschaften des	- subject1_connectivity_dti_lausanne120.mat
	Gehirns.	- subject1_connectivity_dti_lausanne250.mat
		- subject1_connectivity_gqi_aparc.mat
		- subject1_connectivity_gqi_lausanne120.mat
		subject1 connectivity ggi lausanne250.mat
Vergleichende	Untersuchung von	Konnektivitätsmatrizen:
Analysen zwischen	Unterschieden in	- Alle Dateien mit connectivity im Namen
Individuen oder	Netzwerkmaßen	Regionseigenschaften:
Gruppen	zwischen	 subject1_region_properties_aparc.mat
	verschiedenen Subjekten oder Gruppen (z.B. Patienten vs. Kontrollgruppe).	- subject1_region_properties_lausanne120.ma t
		- subject1_region_properties_lausanne250.ma t
Statistische	Untersuchung von	Diffusionsmaße:
Modellierung und	Korrelationen	 subject1_diffusion_measures.mat
Korrelation mit	zwischen	 subject1_fractional_anisotropy.nii.gz
Verhaltensdaten	Netzwerkmaßen	 subject1_mean_diffusivity.nii.gz
	Diffusionsparamete	 subject1_axial_diffusivity.nii.gz
	rn und	 subject1_radial_diffusivity.nii.gz
	Verhaltensdaten,	Konnektivitätsmatrizen:
kognitiven Funktionen oder klinischen Symptomen.	- Alle Dateien mit connectivity im Namen	
Visualisierung von	Erstellung von	Traktografie-Dateien:
Netzwerken und visuellen Traktografie Darstellungen der	 subject1_fibers_csd.trk 	
	- subject1_fibers_csd_dti.trk	
	Gehirnnetzwerke und Fasertrakte zur qualitativen Analyse oder Präsentation.	 subject1_fibers_dti.trk
		 subject1_fibers_gqi.trk
		 subject1_fibers_gqi_dti.trk
		Parzellierungsbilder:
		 subject1_aparc+aseg_b0.nii.gz
		 subject1_lausanne120+aseg_b0.nii.gz

		 subject1_lausanne250+aseg_b0.nii.gz
Community	Identifikation von	Konnektivitätsmatrizen:
Detection und Modularitätsanalys e	Modulen oder Gemeinschaften innerhalb des Gehirnnetzwerks, um funktionelle Einheiten zu erkennen.	- Alle Dateien mit connectivity im Namen
Topologische	Untersuchung von	Konnektivitätsmatrizen:
Analysen	Netzwerk-Motiven, Robustheit des Netzwerks bei Entfernung von Knoten oder Kanten und anderen topologischen Eigenschaften.	- Alle Dateien mit connectivity im Namen
Diffusionsparamete	Analyse von	 subject1_fractional_anisotropy.nii.gz
r-Analyse	Diffusionsmaßen	 subject1_mean_diffusivity.nii.gz
	wie fraktionelle	 subject1_axial_diffusivity.nii.gz
	Anisotropie (FA), mittlere Diffusivität	 subject1_radial_diffusivity.nii.gz
	mittiere Diffusivität (MD), axiale Diffusivität (AD) und radiale Diffusivität (RD).	 subject1_diffusion_measures.mat
Traktografie-	Untersuchung der	Traktografie-Dateien:
Analyse	Eigenschaften von	 subject1_fibers_csd.trk
	Fasertrakten,	 subject1_fibers_csd_dti.trk
	einschließlich	- subject1_fibers_dti.trk
	der Essern zwischen	 subject1_fibers_gqi.trk
	Regionen und	 subject1_fibers_gqi_dti.trk
	räumliche	Fasereigenschaften:
	Verteilung.	 subject1_fiber_properties_csd_aparc.mat
	- subject1_fiber_properties_csd_lausanne120. mat -	
		subject1_fiber_properties_csd_lausanne250. mat
		- subject1_fiber_properties_dti_aparc.mat
		- subject1_fiber_properties_dti_lausanne120. mat

		- subject1_fiber_properties_dti_lausanne250. mat
		 subject1_fiber_properties_gqi_aparc.mat
		-
		<pre>subject1_fiber_properties_gqi_lausanne120.</pre>
		mat
		-
		subject1_fiber_properties_gqi_lausanne250.
		mat
Regionale Analyse	Untersuchung	 subject1_region_properties_aparc.mat
der Gehirnstruktur	spezifischer	-
	Eigenschaften von	subject1_region_properties_lausanne120.ma
	Gehirnregionen, wie	t
	Volumen, Fläche	-
	oder regionale	subject1_region_properties_lausanne250.ma
	Diffusionsmaße.	t
Vergleich	Vergleich der	Konnektivitätsmatrizen und Traktografie-
verschiedener	Auswirkungen	Dateien für alle Methoden:
Traktographie-	verschiedener	- subject1_connectivity_dti <i>mat -</i>
Methoden	ethoden Traktographie- Methoden (DTI, CSD, GQI) auf die resultierenden	<pre>subject1_connectivity_csdmat</pre>
		 subject1_connectivity_gqi_*.mat
		- subject1_fibers_dti.trk
		- subject1_fibers_csd.trk
	Netzwerke und	- subject1 fibers ggi.trk
	Konnektivitätsmaße	