



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Stephan Gerhard

Connectomics - Tools and Applications

ConnectomeViewer and ConnectomeWiki

Master Thesis
April 2009 to January 2010

Supervisor: Prof. Jean-Philippe Thiran
Supervisor: Dr. Patric Hagmann
Mentor: PD Daniel Kiper

Abstract

The field of science called connectomics deals with computer-assisted image acquisition and analysis for the structural mappings of neural connections. In the upcoming decades, such structural maps or connectomes will become available for many species. Advances in Diffusion MRI technology are expected to contribute substantially to this development on the mesoscopic, whole-brain level.

Knowledge about healthy and abnormal brain structure for basic and clinical research can be retrieved by the analysis of these connectomes. A crucial ingredient to make progress in the field of connectomics is the development of advanced methods for visualization and analysis of these forthcoming connectomes.

In the context of this thesis, two tools were developed which contribute to the advancement of connectomics.

Firstly, the software application *ConnectomeViewer* was developed, providing an extensible workspace for connectomics. It serves structural neuroimaging researchers to investigate spatial and topological features of connectomes. Even though the ConnectomeViewer was developed to visualize and analyze mesoscale anatomical datasets, it can be used for functional network analysis, as well as for microscale neural networks. It was programmed in the Python programming language. The *Connectome File Format* was designed. This format enables storage in datatypes relevant to connectomics such as networks, surfaces, volumes, tracks and metadata. Secondly, a web-platform termed *ConnectomeWiki* based on semantic wiki technology was designed and implemented. It serves as a collaborative knowledge base of brain regions and their connectivity across species.

Keywords: Diffusion MRI, neuroanatomy, brain connectivity, connectome, connectomics, Python, network, semantic wiki

Acknowledgements

My family and Patricia for the kind support.

EPFL Jean-Philippe Thiran: Main supervisor, interfacer and enabler, opportunity to host a server.
Patric Hagmann: Main supervisor, motivator; Reto Meuli (CHUV), Leila Cammoun, Alia Lemkaddem, Alessandro Daducci.

INI Daniel Kiper: My mentor at the INI, always having time when I needed; Daniele Oberti for support in the conversion of songbird brain regions and connections to the ConnectomeWiki; Paul Rogister for advice on the beauty of GUIs; Elisha Ruesch for providing neuronal datasets and Stefan Roth for its conversion; Albert Cardona, German Koestinger.

The KEY Institute for Brain-Mind Research Patricia Milz not only for the EEG Attention dataset and valueable feedback; Roberto Pascual-Marqui to make the converter from SLORETA to Connectome File Format possible.

Open Source Community / Others All the unkown mass of open source developers in the Python, Scipy world. Especially I would mention Prabhu Ramachandran and Gael Varoquaux for the nice visualization application Mayavi, and Gael also for the countless bug-fixes and support on the mailing lists. Enthought for their nice application framework Envisage and the Enthought Tool Suite. Mediawiki and Semantic Mediawiki developers. Benjamin Thyreau for helping with 64-bit deployment.

Contents

1 Preface	7
1.1 Introduction	7
1.2 Context	8
1.3 Outline	10
I Tools	13
2 ConnectomeViewer	15
2.1 Introduction	15
2.1.1 State Of The Art	15
2.1.2 Why Python?	17
2.2 Connectome File Format	18
2.2.1 Metadata (XML)	19
2.2.2 Volume data (Nifti)	19
2.2.3 Surface data (Gifti)	20
2.2.4 Networks (GraphML)	22
2.2.5 Track data (TrackVis)	22
2.3 Architecture / Object Model	22
2.4 Features / Plugins	23
2.4.1 Connectome File View	24
2.4.2 Mayavi Visualization Tree and Object	24
2.4.3 3D View	25
2.4.4 Text Editor	25
2.4.5 Interactive Python Shell	25
2.4.6 Analyzer Node View	26
2.4.7 More functionality	26
2.4.8 Getting data into the ConnectomeViewer	27
2.5 Connectome Database	29
2.6 Conclusion	30
3 ConnectomeWiki	33
3.1 Introduction	33
3.2 Basic Ontology	34
3.2.1 Brain Region	34
3.2.2 Brain Region Connection	37
3.2.3 Main Categories	38
3.3 Example Applications	39
3.3.1 Efferent connections of zebra finch robust nucleus	39
3.3.2 Subregions of a brain region STG in Homo sapiens	39
3.3.3 Topographical zebra finch nuclei	40
3.4 Initial Datasets	40
3.5 Conclusion	42

II Applications	45
4 ConnectomeViewer Applications	47
4.1 Introduction	47
4.2 Human Developmental project	47
4.2.1 Analysis Protocol	48
4.2.2 Results	48
4.2.3 Discussion	52
4.2.4 Conclusion	54
4.3 sLORETA EEG Attention Study	55
4.3.1 Visualization Protocol	55
4.3.2 Conclusion	57
4.4 Average Human Connectivity	58
4.4.1 Analysis and Visualization Protocol	58
4.4.2 Conclusion	61
4.5 Cat area 17 cortex	62
4.5.1 Analysis Protocol Aim 1	62
4.5.2 Analysis Protocol Aim 2	64
4.5.3 Conclusion	64
A Homepage and Source Code	69
B Developmental Project Script	71
C Cat area 17 cortical microcircuit	73
D Connectome File XML Schema	75
E ConnectomeViewer Object Model	81

Chapter 1

Preface

1.1 Introduction

Investigations into the anatomical structure of the nervous systems of living organism appears to be a necessary precondition to learn about the functions of these structures, as they act as constraints for informational processes.

The nervous system can be conceived as a complex system. It is embedded into a body, and this body in turn is embedded into an environment, both also conceivable as complex systems. The role of the nervous system is to act as the control system of the organism to survive in this complex world. One might get an idea of the complexity that this control system must have when one reflects on one's own experience and the near-infinite possibilities one is able to sense, perceive, think and act. And this is what we actually encounter when starting with neuroanatomy. A staggering complexity at many spatial levels.

The outcome of the very process of conceptualizing the experienced world into entities and their relations and interactions is expressible in mathematical apt structure, namely networks or graphs. Networks are made up of nodes representing entities and of edges, representing their relation. Traces of this conceptual schema can already be found in philosophical traditions like atomism. The separation of the world into atomar elements and their interactions turned out to be very successful in our culture to describe and understand phenomena.

Before applying this using-networks-to-understand approach to neuroanatomy, some examples of the ubiquity of networks are given. Notions of networks occur in the fields of social and natural sciences, and also extensively in structural sciences such as computer science and mathematics. In the following, specific areas in which networks are employed, are listed with examples.

- biology: neural networks (structural, functional, effective), protein interaction networks (interactome), genetic regulatory networks, protein homology network, biochemical pathways
- social sciences: social networks, migration networks, citation networks, political networks, coauthorship networks
- economics: capital flows, recommender systems, consumer-good networks
- ecology: species interactions, spatial ecology, epidemiology, and evolution in social groups
- logistics: transportation networks (traffic flow), food chains, food webs, energy networks
- semantic networks: topic maps, concept map, mind maps, conceputal graphs
- linguistics: propositional networks
- computer sciences: object models, class hierarchies, entity-relationship models, artifical neuronal networks, distributed systems

- communication networks: world wide web, telecommunication networks
- others: music similarity networks, movie actor collaboration network, factor graphs, graphical models

A collection of network visualizations can be found on the VisualComplexity homepage¹.

In general, two complementary ways of applying the network concept to the study of information processing in the nervous system can be distinguished. These are structural and functional descriptions. Each description employs a different set of methods of investigations with different resolutions in space and time. These are listed in the following. Quasi gold standards that define a ground truth to which results of higher level, lower resolution methods have to be compared are highlighted in bold letters.

structural description A structural description focuses on the physical structures of the nervous tissue. To be recognized as such, these structures need to exhibit a certain stability over time. Neuronal cells and their connections through synapses are thought to be the main physical structures mediating information processing.

Methods: Diffusion MRI, **light microscopy**, **electron microscopy**

functional description A functional description focuses on the time-dependent evolution of signals. The functional role of a structural component (e.g. cortical area) is largely defined by its anatomical connections. The significance of a structural component is defined by its functional role in the system it belongs to. To assess functional connectivity defined as ‘temporal correlations between spatially remote neurophysiological events’ [6] measures are employed which denote the symmetrical statistical association between the system’s structural components: correlational measures and measures of coherence.

Methods: Functional MRI, PET, EEG, MEG, **electrophysiology** (intracellular and extracellular), Voltage-sensitive Dye Imaging

Answers to structural questions such as: „What parts are there?“, „How do we separate them from the rest?“, „How are these parts interconnected?“ lay the foundations of functional questions such as „Who does what and when?“

In the context of this master thesis a software to visualize and analyze network data was developed. As described in the application section, the software developed can be applied to both structural and functional connectivity data. The focus of the thesis presented lays however on structural descriptions, namely neuroanatomical investigations, carried out with Diffusion-Weighted MRI.

1.2 Context

Recent advances in structural neuroimaging on a meso- (Diffusion-Weighted MRI [2]) and on a microscale level [2, 3, 12] has resurrected interest in the research field of connectomics. The term *connectome* is used to describe connectional maps of neural circuitry. Connectomes are thought to be essential to the understanding of the structure and functioning of these circuits. However, the neuroscience community is divided on the feasibility of complete connectomes [4].

On a mesoscale level, such connectomes relate only indirectly to actual neuronal connectivity. Diffusion-weighted MRI, a non-invasive mesoscale technique, acquires gradient maps of the diffusion of water molecules through the tissue of the brain. The data acquired through Diffusion-Weighted MRI is suggested to reflect white matter fiber tract architecture. Water diffusion is constrained by axonal membranes and sheets of myelination, but is eased along the principal directions of white matter bundles. Based on this indirect measures, only coarse estimates of connectomes for the whole brain [6] can be obtained.

Over the last few years, a Connectome Mapping Pipeline² has been established at the Signal Processing Lab 5 at the Ecole Polytechnique Fédérale. It includes all the necessary steps

¹<http://www.visualcomplexity.com/>

²Abbreviated as CMP

to compute whole-brain human mesoscale connectomes from raw Diffusion-Weighted MRI data. Figure 1.1 shows this pipeline's general steps of processing. For a detailed description of the individual steps, see [5]. Total computing time on standard equipment is four days.

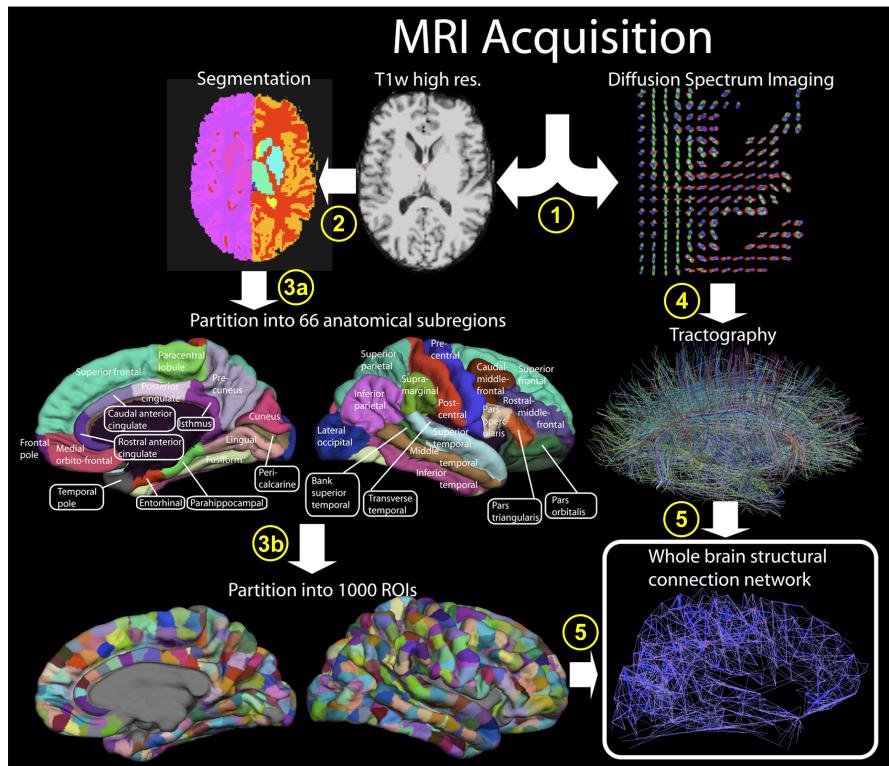


Figure 1.1: Overview of the Connectome Mapping Pipeline developed at EPFL LTS5. Necessary steps to process MRI data into Connectomes.

Using this method, good estimates for mid-range to long-range axonal bundles can be obtained [9]. Reliable quantitative information [10] with good confidence levels [11] can be acquired. The deriving connectomes from Diffusion MRI data are expected to find use in important applications [2]:

- Basic neuroscience: a scaffold for Connectomes at microscale levels, and large-scale computational models
- Clinical neuroscience: measures of white matter change as a clinical marker
- Systems neuroscience: parallel functional and anatomical experiments to derive influence of structure to brain function and behaviour
- Comparative Anatomy: Measuring connectivity in human and other species with the same technique
- Neurosurgery: Knowledge of the locations of key white matter connections is very important

The context of this master thesis is the development of tools and their applications for mesoscale connectomes analysis and visualization. However, it must be expected that a large number of connectomes from a variety of sources will become available in the upcoming years. In anticipation of this development, the tools developed are not dependent on this particular Connectome Mapping Pipeline shown, but can be used in other pipelines as well³. Furthermore, the tools can also be applied to data of structural connectivity on the microscale level (as shown in the application section 4.5).

³The Human Connectome Project, 2010-2015. <http://www.humanconnectomeproject.org/>

1.3 Outline

In the present work, I will not go into details of Diffusion MRI acquisition and processing, brain segmentation and registration, or tractography algorithms [2, 8, 9]. The main focus of this master thesis will lay on the developed tools (Part I) and their applications (Part II) for connectome analysis and visualization:

The **ConnectomeViewer** is a tool for visualization and analysis of structural neuroimaging data on multiple spatial scales and for different modalities. After exploring other related software projects, I first introduce the Connectome File Format containing all relevant data and metadata. Then I present an overview of the ideas behind the software architecture. In the following, a description of implemented features and plugins will be given. I will conclude with a few remarks concerning the suggested beneficial option of the development of a Connectome Database for data sharing [1]. The ConnectomeViewer can be regarded as a natural extension to the Connectome Mapping Pipeline described above. However, by introducing enough abstraction, mainly in the specification of the Connectome File Format (Section 2.2), the ConnectomeViewer is not dependent on this pipeline.

The **ConnectomeWiki** serves as a knowledge resource of neuroanatomy for various species and is loosely coupled to the ConnectomeViewer. Knowledge from well-known neuroanatomy with references to the relevant literature can be retrieved. The main vision is introduced and some relevant architectural features are discussed with some example applications.

In Part II, potential applications of the ConnectomeViewer are shown on unpublished datasets. The first Diffusion MRI dataset contains subjects of ages 2 to 18 years. It is processed according to the Connectome Mapping Pipeline. Another datasets consists of 20 healthy adult subjects that were scanned and also processed with this pipeline, yielding an average human connectome dataset. This dataset was mainly investigated for basic network metrics.

EEG data, when properly analyzed, can yield electrophysiologically meaningful functional connectivity networks, e.g. using measures of lagged coherence. A converter was written to make functional EEG data usable from sLORETA, a widely-used software to analyze and visualize EEG data. The datasets of an attention study (Section 4.3) comprising 23 male subjects' functional connectivity during three conditions was converted. A use case to visualize a massive number of thresholded networks is described.

Cat area 17 cortex data is still heavily investigated at the Institute of Neuroinformatics. Three reconstructed cortical pyramidal cells were converted to the Connectome File Format. A simple network estimation based on a proximity criterium for potential synapses was performed. Additionally, the estimated cortical microcircuitry was modeled and visualized.

Bibliography

- [1] Sporns O, Tononi G, Kötter R. The Human Connectome: A Structural Description of the Human Brain. *PLoS Comput Biol* 1(4): e42, 2005.
- [2] Van Dijk KR, Hedden T, Venkataraman A, Evans KC, Lazar SW, Buckner RL. Intrinsic functional connectivity as a tool for human connectomics: theory, properties, and optimization. *J Neurophysiol*. 103(1): 297-321, 2010.
- [3] Seung HS. Reading the book of memory: sparse sampling versus dense mapping of connectomes. *Neuron* 62(1): 17-29, 2009.
- [4] Lichtman JW, Sanes JR. Ome sweet ome: what can the genome tell us about the connectome? *Curr Opin Neurobiol*, 2008.
- [5] Hagmann P, Cammoun L, Gigandet X, Meuli R, Honey CJ, et al. Mapping the Structural Core of Human Cerebral Cortex. *PLoS Biol* 6(7): e159, 2008.
- [6] Lee L, Harrison LM, Mechelli A. A report of the functional connectivity workshop, Dusseldorf 2002. *Neuroimage* 19:457-65, 2003.
- [7] Johansen-Berg H, Behrens T. Diffusion MRI: From Quantitative Measurement to In-vivo Neuroanatomy. *Academic Press*, 2009.
- [8] Patric Hagmann. From diffusion MRI to brain connectomics. *PhD thesis*, Ecole Polytechnique Fédérale de Lausanne, 2005.
- [9] Xavier Gigandet. Global brain connectivity analysis by diffusion MR tractography: Algorithms, Validation and Applications. *PhD thesis*, Ecole Polytechnique Fédérale de Lausanne, 2009.
- [10] Hagmann P, Gigandet X, Meuli R., Kötter R, Sporns O, Wedeen VJ. Quantitative validation of MR tractography using the CoCoMac database. *Proc. Intl. Soc. Mag. Reson. Med.* 16, Toronto, Canada, 2008.
- [11] Gigandet X, Hagmann P, Kurant M, Cammoun L, Meuli R, Thiran J-P. Estimating the confidence level of white matter connections obtained with MRI tractography. *PLoS ONE*. 3(12):e4006, 2008.
- [12] Lichtman JW, Livet J, Sanes JR. A technicolour approach to the connectome. *Nat Rev Neurosci* 9(6): 417-22, 2008.

Part I

Tools

Chapter 2

ConnectomeViewer

2.1 Introduction

Connectomics dealing with the structure of nervous systems is carried out at many spatial scales. Methodological advances in data acquisition and processing are observable at both the mesoscale with increased performance of Diffusion Weighted MRI [3] as well as at microscopic levels regarding reconstruction of cellular networks [4, 5]. More improvements are likely to occur in the coming decades, since a structural "blueprint" of the nervous systems is a necessary requisite to make real progress in understanding their functional properties.

The ConnectomeViewer application, as a powerful, extensible software tool for visualization and analysis in connectomics, was mainly developed with MRI-derived connectome data in mind. Through a Connectome Mapping Pipeline [6] developed at EPFL Signal Processing Lab 5, Diffusion MRI data is processed yielding large-scale connectivity maps. These connectivity maps are essentially networks, depicting spatial and topological information, and are termed Connectomes [7]. One of the main goals for the ConnectomeViewer was to provide visualization and analysis capabilities for the data output of this pipeline.

However, the usability of the ConnectomeViewer is not limited *by its very design* to a mesoscale level of description, as shown in the application section. In section 4.5 a cortical microcircuitry of the cat primary visual cortex was estimated, including neuronal reconstructions from light microscopy. Moreover, ready usage of functional network data based on sLORETA lagged coherences EEG analysis is shown in section 4.3 to be appropriate for visualization.

In the following sections, a quick overview of state-of-the-art software is given with similar target users. The reasoning behind why Python was chosen as a programming language is given. Furthermore, various relevant data types potentially occurring in connectomics are described and it is shown, how they have been made usable with the Connectome File Format, a novel format acting as data container. The general architecture and already implemented features are then depicted with some use cases. The chapter is closed by some thoughts about a future ConnectomeDatabase, data sharing ideas and future work.

2.1.1 State Of The Art

In the following, well-known, widely-used software tools that have a quite similar target audience as the ConnectomeViewer are described. Differences and similarities are highlighted.

Name	3D Slicer
Description	A multi-platform, free open source software for visualization and image analysis. It is mainly for medical imaging, not limited to neuroimaging research. It has many modules for various processing (segmentation, registration) and visualization task (MRI, DW-MRI). Many publications were enabled using 3D Slicer.
Funding	There are many funding sources and contributors.
Community	There is a big community behind 3D Slicer and many developers.
License	BSD-style open source license
Platform	Windows, Linux and Mac OS X
Source Code	Over one million lines of code, mostly C++.
URL	http://www.slicer.org/

Name	Network Workbench Tool
Description	"A Large-Scale Network Analysis, Modeling and Visualization Toolkit for Biomedical, Social Science and Physics Research. This project will design, evaluate, and operate a unique distributed, shared resources environment for large-scale network analysis, modeling, and visualization, named Network Workbench (NWB)." The NWB has a very broad focus, targeting network researchers from very diverse fields. It is very strong in modeling and analysis, less strong in support for visualization. The GUI provides a task manager for lengthy and computationally intensive calculations. Additionally, analysis tasks and parameters can be invoked from the menus, which makes the tool very user-friendly.
Funding	Involvement of professors (6), 6 developers (current: 6, past: 10), beta testers (16) and an advisory board (10) was made possible with help of research grant money.
Source Code	The Source Code is available. For Documentation, there is a wiki and good introductory tutorials to get started.
Technology	Cyberinfrastructure Shell, JAVA
Platform	Windows, Linux and MacOS X
URL	http://nwb.slis.indiana.edu/

Name	MeVisLab
Description	An development environment for medical image processing and visualization. "A powerful, modular framework for the development of image processing algorithms and visualization and interaction methods, with a special focus on medical imaging. Besides basic image processing and visualization modules, MeVisLab includes advanced medical imaging algorithms for segmentation, registration, and quantitative morphological and functional image analysis."
Funding	Coming from the research labs, MeVis Medical Solutions AG and Fraunhofer MEVIS continue the development and the commercial marketing of MeVisLab.
License	There are Registered and Un-Registered version, for commercial and non-commercial purposes. The licensing differs for each. The Unregistered version can be downloaded for free, but with a restricted set of functionality.
Technology	Qt, Open Inventor, OpenGL, JavaScript and Python for scripting, Insight Toolkit (ITK), Visualization Toolkit (VTK)
Development	For developers, there is a Software Development Kit to write custom modules. A good, comprehensive documentation and reference is available.
Platform:	Windows, Linux, and MacOS X
URL	http://www.mevislab.de/

Name	Node3D
Description	3D graph visualization program written by Issac Trotts in consultation with Shawn Mikula, in the labs of Edward G. Jones.
Features	No sophisticated GUI, but many key bindings for interactivity.
Developers	Last release was 2007, the main developer works now at Google and discontinued development.
Source Code	Available.
Technology	LUA Programming Language (Lua is a powerful, fast, lightweight, embeddable scripting language.)
Platform	Windows, Linux and Mac OS X
URL	http://brainmaps.org/index.php?p=desktop-apps-nodes3d

Name	V3D
Description	“V3D is a handy, fast, and versatile 3D Image Visualization & Analysis System for Bioimages & Surface Objects.” Its target is cellular resolution data. It has volume rendering capabilities.
Developers	Hanchuan Peng Lab (1 head plus 4 people)
License	Custom license (including the agreement not to criticize the code-writing style of the author)
URL	http://penglab.janelia.org/proj/v3d/v3d2.html

How far does the ConnectomeViewer fit into these already existing applications? What are its defining characteristic? As preliminary remark, it must be kept in mind that the development of the ConnectomeViewer was a one-man project. The depicted tools were developed and are maintained with more resources behind, such as more development manyears and financial resources from grants and the industry. Nevertheless, the ConnectomeViewer is competitive in its niche, concerned explicitly with structural neuroimaging research.

The ConnectomeViewer is probably best situated between 3D Slicer and the Network Workbench Tool, but with special emphasis on mesoscale connectomics. Compared to V3D, which mainly targets microscale connectomics, it is more open and extensible. Nodes3D could be seen as a small subset in terms of functionality. The ConnectomeViewer implements no methods for image processing steps such as segmentation or registration, it is assumed that this has already been done. Special-purpose network analysis methods tailored for brain connectivity and group analysis might be advantageous, outperforming the general purpose Network Workbench Tool. Nevertheless ConnectomeViewer networks are easily imported into NWB to carry out analysis there.

2.1.2 Why Python?

The following list depicts advantages of Python [1]. The comments bring them into the context of my project.

excellent for beginners, yet superb for experts There are many resources in the internet to learn. For the experts, Traits [35] allows type definition for objects and many advanced features (validation, delegation, notification, visualization).

rapid development In November 2009, after approximately six month, a fully functional Beta version has been released.

portable, cross-platform Linux, Windows are supported so far for the ConnectomeViewer, Mac OS soon.

easily extensible Using the Envisage Application Framework which provides a plugin architecture, is a perfect tool appropriate for the credo that a good software is never finished and has to provide good interfaces for extensions.

object-oriented Full object-oriented development is supported and was heavily employed in the architecture of the ConnectomeViewer.

you can get the job done What ever you are looking for, may it be statistic packages etc. you will find it.

simple yet elegant The syntax is very easy readable and allows to express the ideas clearly.

stable and mature The first release was 1991. Since then, many years were invested from professional developers all over the world.

powerful standard libs For scientific computing with Python, the `scipy` (scientific python [33]) and `numpy` (matrix operations [34]) libraries are in wide use. `NiPy` [36] is a project for neuroimaging in Python that might be integrated in the future. In general, Python has a growing and motivated scientific community behind.

wealth of 3rd party packages There is an abundant number of libraries existing for any requirements. For example for complex networks research, there is `NetworkX` [16] which is used as the internal network data structure with many ready-to-use analysis algorithms.

Especially in neuroinformatics, apart from the huge number of already existing packages for everyday scientists use, there are more libraries and tools available written in Python every year. As of 2009, this development can be termed a trend. A special issue of *Frontiers in Neuroscience* was dedicated to Python, highlighting recent efforts the development of Python modules in the domain of neuroscience software and neuroinformatics [9]: simulators and simulator interfaces; data collection and analysis; sharing, re-use, storage and databasing of models and data; stimulus generation; parameter search and optimization; visualization; VLSI. hardware interfacing

It is very likely that Python will become the standard programming tool in science.

2.2 Connectome File Format

The ConnectomeViewer application defines its own data format, the Connectome File Format, adhering to open standards [10]. This format integrates widely used and standardized formats in the neuroimaging community. None of the existing formats alone are apt to meet the requirements for a truly integrated endeavor into structural neuroimaging research on multiple levels. As a kind of container format, it was convenient to use a ZIP archive [11] packing and compressing the data. The official file ending is `.cff`. Internally, the structure of a Connectome File¹ is as follows.

```
meta.xml
Gifti/
    testsubject.gii
    testsubject_labels.gii
Network/
    network_res83.graphml
    network_res150.graphml
Nifti/
    ROI_scale33.nii
    ROI_scale60.nii
Tracks/
    fibers_transformed.trk
Pickle/
    network_res83.graphml
    network_res150.graphml
```

First I will discuss the data contained in the `meta.xml` and then explain the different data types in the subsections. The `Pickle` directory will be omitted, since it contains only an automatically generated version of the networks for faster loading.

¹Furthermore, a Connectome File means a file adhering to the Connectome File Format definition.

2.2.1 Metadata (XML)

Using a custom Connectome Schema Definition (Appendix D), XML is apt to store metadata about the Connectome File and link the different data types together. This is done in the meta.xml file. An example excerpt is shown in Listing 2.1.

Listing 2.1: The content of the meta.xml file

```

1 <viewer xmlns="http://www.connectome.ch/2009/Connectome/XMLSchema-instance"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://www.connectome.ch/2009/Connectome/XMLSchema-instance
4   connectome.xsd">
5
6 <viewer-meta version="1.0">
7   <generator>CMT2CFF Converter</generator>
8   <initial-creator>Stephan Gerhard</initial-creator>
9   <creation-date>20-06-2009</creation-date>
10  <modification-date>17-10-2009</modification-date>
11  <name>Single subject connectivity</name>
12  <species>Homo sapiens</species>
13  <legal-notice>Creative Commons Attribution 3.0 Unported</legal-notice>
14  <references>My Publication, 2100, Publication Journal</references>
15  <url>http://www.connectome.ch/viewer</url>
16  <description>This file contains connectivity data for a single subject as done with the ↵
    Connectome Mapping Toolbox developed at EPFL.</description>
17  <nr_of_networks>5</nr_of_networks>
18 </viewer-meta>
19
20 <viewer-network name="Network Lausanne83" src="Network/network_res83.graphml" hierarchical="0" ↵
    hypergraph="0" directed="0">
21   <network-metadata>
22     <data key="mymetadata_key">mymetadata_data</data>
23   </network-metadata>
24   <network-surface name="Individual surfaces" src="Gifti/testsubject.gii" fileformat="gifti" ↵
      addatlas="template_atlas_homo_sapiens_01">
25     <surface-label labelid="Resolution 83" src="Gifti/testsubject_labels.gii" fileformat="gifti" ↵
      "/>
26   </network-surface>
27   <network-volume name="Volume ROIs Resolution 83" src="Nifti/ROI_scale33.nii" fileformat="nifti" ↵
      " segmentation="true">
28     <description>The segmentation of the brain in voxel-format.</description>
29   </network-volume>
30   <network-track name="Trackfile used" src="Tracks/fibers_transformed.trk" fileformat="trk">
31     <description>The track file for visualisation in Trackvis.</description>
32   </network-track>
33 </viewer-network>
34 ...
35 </viewer>
```

The Connectome File metadata is described on lines 7-16, then the first network is defined (lines 22-33). For this network, the paths to the corresponding files for the surface (line 24) including a surface labeling file (line 25), the segmentation volume (line 27) and the track file (line 30) are defined. Additional options are described in the corresponding subsections and in the Appendix D. Flexible annotation possibilities to store any kind of metadata for networks is shown on line 22. What is visible here are the internal IDs (integer values) used to relate a network node to its corresponding surface patch in the surface labeling file, and to the corresponding set of voxels in the segmentation volume file. More details are found in the Appendix D and in the example datasets (Appendix A).

Potentially, any type of metadata could be included conforming to XML standards. For example XCEDE provides an extensive metadata hierarchy for describing and documenting research and clinical studies [12]. Only recently (first meeting January 2010), the INCF program *Minimal Metadata Standards* has been launched, highlighting the importance of relevant metadata especially in neuroimaging for data archiving, storage, sharing and re-use. The goals for the Connectome File Format metadata is to adhere to this standards [13].

2.2.2 Volume data (Nifti)



Nifti is the standard in the neuroimaging community dealing with voxel data. There exists a very good library to use and manipulate the data in Python, PyNifti [14].

Connectomics on a microscale usually deals with image stacks. They are used to segment cells and neurites. This are es-

sentially the same data processing steps as on the mesoscale, whole-brain MRI level. The volume is segmented into gray and white matter voxels. The data structure in both cases is a 3D scalar field. I want to mention this here to point to the potential usage of the Connectome File Format for small data sets on the microscale level. The ConnectomeViewer aims not primarily at providing visualization capabilities for volumetric data, because there exist well-designed tools for this purpose ². However, the Connectome File provides the possibility to store volume data which can be used in scripts. Actually, it is used to interface with TrackVis [17], a tool for visualization of tractography results. The segmented volumes are used to compute the Regions of Interest and select the connecting tracks accordingly.

Thus, raw and segmented volumetric data can be stored in a Connectome File. For the special case of a volumetric dataset containing a segmentation, it is possible to tag this volumes with *segmentation='True'* in the meta.xml (line 27 in section 2.2.1). The voxel scalar values which are the labels should be in correspondence with the ID defined for surface labeling and for nodes to guarantee maximal compatibility. Thanks to Gael Varoquaux, A simple volumetric 3D Viewer was integrated into the ConnectomeViewer to have a quick look at segmentations (See Figure 2.1) ³.

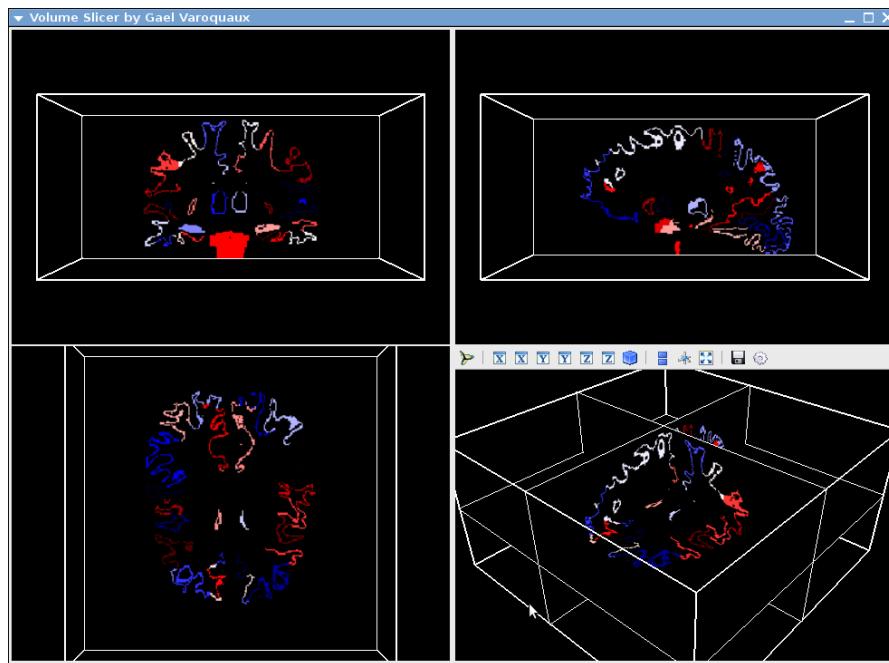
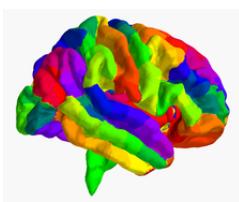


Figure 2.1: This Volume Slicer helper tool was adapted from a Mayavi Example. It allows to have a quick look at the segmentation of volumes.

2.2.3 Surface data (Gifti)



The Connectome Mapping Pipeline uses Freesurfer ⁴ to extract cortical and subcortical surface meshes from MRI data. These surfaces are essentially vertices (points in 3D space) and triangles connecting these. Since there is a jungle of formats to represent surfaces, the neuroimaging community decided to define the Gifti data format [15] which is apt to their needs. ConnectomeViewer supports Gifti files natively to store and render surfaces, and also to label them. With this support, ConnectomeViewer is well prepared for further

²For example, 3D Slicer: <http://www.slicer.org/>

³Thanks goes to Gael Varoquaux to make this possible.

⁴Freesurfer: <http://surfer.nmr.mgh.harvard.edu/>

developments within the neuroimaging communities. A quick overview of Gifiti-classes and their attributes is shown in Figure E.6 in the Appendix.

A system to incorporate template surface atlases into the ConnectomeViewer was developed. In the Connectome Mapping Pipeline, the Freesurfer Average surface is used as a reference template atlas. Individual brain surfaces are registered to this template atlas. Thus it was sensible to incorporate the Freesurfer Average surface directly into the ConnectomeViewer, making the inclusion in individual Connectome Files redundant. This template atlas can be included in the Connectome Files by adding `addatlas="template_atlas_homo_sapiens_01"` to a `<network-surface>` tag (see line 24 in Listing 2.1). The advantage of this option is that is possible to render individual networks with a common reference that allows for better direct visual comparison. Nodes and edges are positioned at the same location for individual subjects.

Future work will be concerned with including other template atlases, and also for non-human species. The SPL-PNL Brain Atlas 2008 [39] and the well known Colin27 brain [40] will be integrated in the official release.

The Connectome Mapping Pipeline yields essentially four types of surface representations for individual surfaces: pial surface, gray-white matter border surface, sphere surface and inflated surfaces. This is also the case for the Freesurfer Average surface. For this average surface, I generated and added also a Flatmap representation of the two cortical hemispheres including the subcortical nuclei. They are included in the `template_atlas_homo_sapiens_01`.

These surface representations are used to position the nodes of the networks in 3D space when no other locations were defined. The node are thereby positioned at the center of gravity of the corresponding ROIs of the surface. Figure 2.2 shows an example for the spherical surface representation of the two hemispheres and subcortical nuclei.

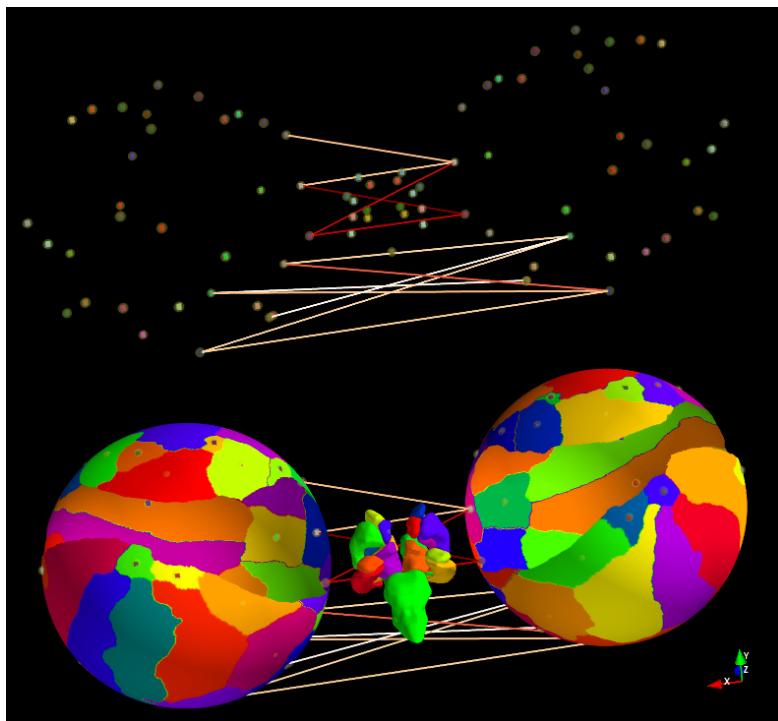
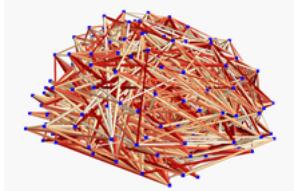


Figure 2.2: The bottom part depicts the ROIs as surfaces of the left and right cortical hemispheres as spheres and subcortical nuclei. The center of gravity of these ROI patches define the position of the nodes for the network layout.

2.2.4 Networks (GraphML)

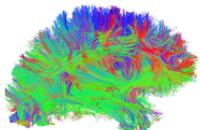


Many standardized file formats exist to express networks or graphs. The GraphML file format [26] was chosen for the following reasons:

- A well-supported data format in various applications which allows easy import and export.
- NetworkX library [16] support for complex network analysis
- Expressiveness for *arbitrary data on nodes and edges*
- Description of hierarchical graphs

Currently, research is concerned with the description of systems and their interactions at various separated levels. It can be expected that in the future, a main concern will be how these various levels of descriptions are interconnected. Such interlevel relations can be described with hierarchical networks [8]. In anticipation of this development, the used GraphML data format structure enables easy integration of hierarchical network descriptions.

2.2.5 Track data (TrackVis)



An important part of the Connectome Mapping Pipeline is tractography, where virtual fiber tracks are reconstructed from the raw Diffusion MRI data. These tracks are readily visualized using the TrackVis application [17]. Through the usage of these tracks to generate the network edges, a certain loss of information occurs (e.g. the detailed spatial structure). Thus, it makes sense

to store track data in the Connectome File for further processing steps. The TrackVis .trk file format is used to store track data. Its specifications are open⁵.

Additionally, if the track data is appropriately linked to the networks and volumetric segmentations within the Connectome File, the user can use an easy interface to invoke TrackVis. Selected network nodes are used to generate automatically the ROIs in TrackVis. These can then be used to filter the tracks. This helps when one carries out network analysis on the networks and finds relevant nodes, that one wants to have a closer look with regard to the detailed virtual fiber tracks. The ConnectomeViewer is able to generate an appropriate Scene file for TrackVis with all the relevant information and invoke TrackVis.

Matlab scripts to manipulate TrackVis data are available from the Brain Connectivity Challenge [27]. There are already parsers for track files available for Python. They are going to be integrated into the final ConnectomeViewer release.

2.3 Architecture / Object Model

Building on top of the Enthought Software stack as seen in Figure 2.3 allows to produce a very flexible software application in a short time. The Envisage application building framework (ETS) [38] provides the bindings to develop software in a modular way based on plugins. Essentially, the ConnectomeViewer application is built using pre-existing plugins (such as Mayavi for 3D Visualization) and custom written plugins, e.g. for the Connectome File handling. Finally, all plugins are integrated into one framework, which is very open for further extensions and modifications.

⁵TrackVis File Format <http://www.trackvis.org/docs/?subsect=fileformat>

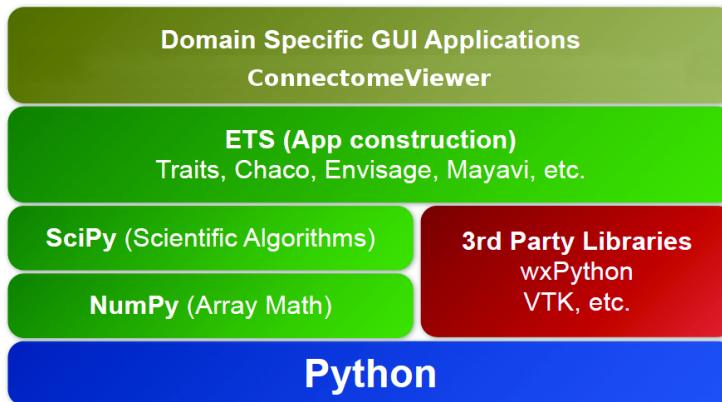


Figure 2.3: Software Application Layers [37]

Only a sketch of the general object classes is given here. A complete overview of the main classes is available in the Appendix E. The complete GPLv3 licensed source code is available from Launchpad [30].

The `CFile` object contains the Connectome File metadata and a list of `Network` objects, which contains lists of `Volume` objects, `SurfaceContainer` objects and `TrackFile` objects. These hierarchy represents all the objects available in a Connectome File. Employing a lazy load strategy for big files with many networks allows to use the memory efficiently. This means that data for a particular networks is only loaded into memory, when it is actually needed. The `CFile` object is exposed for scripting (see Section 2.4.5).

The `Network` objects are in many ways central. They contain the graph with the nodes and edges and its various attributes. Furthermore, a `RenderManager` object and a `DataSourceManager` object is attached to them.

RenderManager Constitutes the manager of the interface between the Connectome File data as exposed in the corresponding classes and the Mayavi visualization plugin. It manages creation of scenes, the generation of the network layouts and surface renderings. It uses the `DataSourceManager`'s `SourceObject` to create Mayavi visualization objects. Furthermore, it implements the node picking functionality for the 3D Views.

DataSourceManager The creation of a data source manager was necessary to structure the data in a good format to create the visualization objects. Visualization objects are the nodes, edges and surfaces as seen in the 3D View in Figure 2.4. Additionally, the interactive data, such as which nodes and edges are selected or which surfaces should be displayed, is managed. To achieve these goals, the data source manager manages a `SourceObject` object containing all this data. All the logic for computing the source object is implemented in the `DataSourceManager`.

The Connectome File View (Section 2.4.1) is actually a plugin, exposing a view on the loaded Connectome File with interaction possibilities. Similarly, the ConnectomeDatabase plugin (Section 2.5) was developed separately and then included into the ConnectomeViewer.

2.4 Features / Plugins

The already implemented and usable functionalities are described in the following subsections with a short description and some application in the end. To understand more elaborate use cases, please refer to the applications part in Chapter 4. The screenshot in Figure 2.4 shows the ConnectomeViewer graphical user interface (GUI) with a loaded Connectome File.

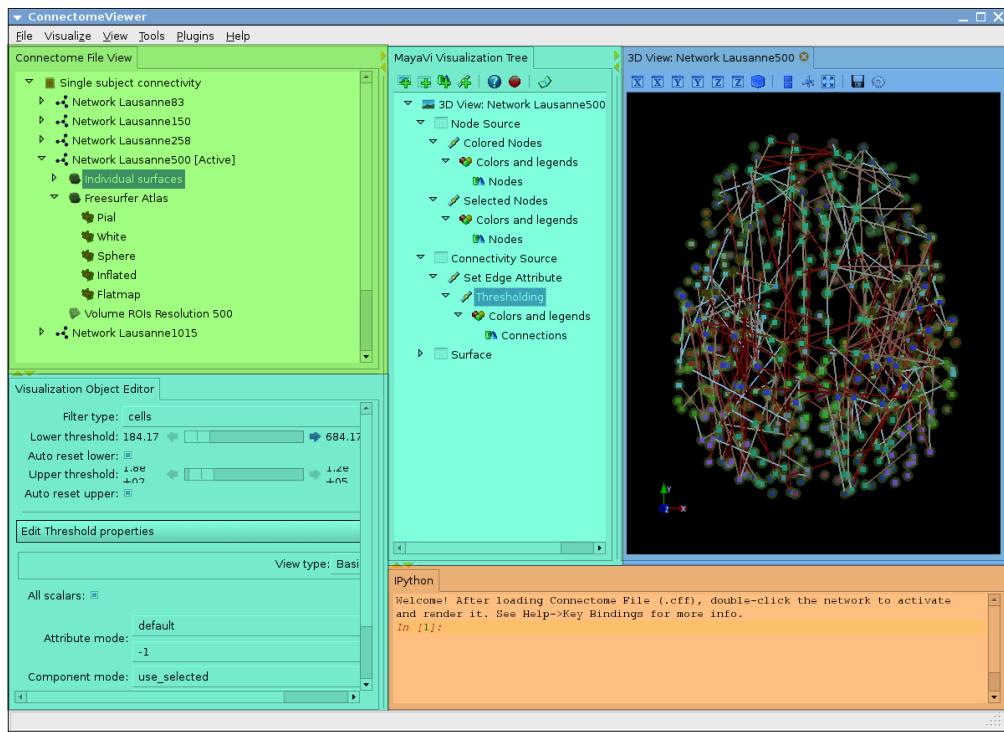


Figure 2.4: Connectome File View (green), Mayavi Visualization Tree and Object (green-blue), 3D View (blue), IPython Shell (orange)

2.4.1 Connectome File View

This view exposes the objects of a loaded Connectome File in a tree structure and allows interactivity. Double-clicking a network node activates the network, loads its data into memory and creates a RenderManager instance which handles the visualization. Right-click on the network opens a context menu, allowing to invoke more functionality. For a network node, this is: Invoking TrackVis as described below; Selecting/Unselecting all network edges in the 3D View; Toggle surface visibility for selected nodes; Deactivate the network.

2.4.2 Mayavi Visualization Tree and Object

Almost any well-known visualization application uses the Visualization Tool Kit (VTK) library [18] which builds on top of OpenGL [19]. Similarly, in the Python world, the tool of choice is Mayavi for 3D Scientific Data Visualization and Plotting [20]. It builds on top of Traits VTK, which extends VTK by a pythonic API. Mayavi modules in turn wrap TVTK objects, which allows to change relevant parameters easily using GUIs (the Mayavi Visualization Object in Figure 2.4).

The Mayavi Visualization Tree depicts essentially a wrapped VTK pipeline with data sources, filters and modules, thereby allowing to benefit from all the capabilities of VTK. Upon loading a Connectome File and activating a network for visualization, appropriate data sources are generated for visualization with the Mayavi plugins. These can then be viewed in the 3D View. The Mayavi Visualization Object View allows to change visualization parameters. An extensive online documentation about the Mayavi capabilities is available⁶

⁶Mayavi Documentation <http://code.enthought.com/projects/mayavi/>

2.4.3 3D View

The ConnectomeViewer GUI in Figure 2.4 displays a rendering of the selected network in the 3D View. The nodes are located at the Center of Gravity for their respective surface patch. Picking functionality allows to select nodes, and get more information about them. Adjacent edges of nodes can be displayed selectively, allowing exploration of dense networks. The Table 2.1 displays Key Bindings that are available in the 3D View.

De-/Select Node and add Edges (neighbor nodes not selected)	1
De-/Select Node and add Edges (neighbor nodes are selected)	2
Toggle the selection of a Node without adding Edges	T or t
Pick point and show coordinates	P or p
Activate Light Manager	L or l
Reset the camera perspective	R or r
Set Focal Point to current Position	f
Save Scene to Image	s or S
Red-Blue Stereo Mode Rendering	3
Rotate object	Arrows
Pan object	Shift - Arrows
Zoom In/Out	+ / -

Table 2.1: Key Bindings available in the 3D View

2.4.4 Text Editor

The Text Editor is used to directly write scripts and execute them. It has Python Syntax Highlighting and line numbers. The generated variables in a script are available in the Interactive Python Shell after execution. Table 2.2 depicts predefined objects that are available for writing scripts and in the IPython shell.

cfile	the currently loaded Connectome File containing all the data, methods to add networks and more
analyze_node	if some analysis were performed from the Analysis Perspective, the results are stored here
mayavi	the currently running instance of Mayavi
application	the currently running Envisage application object
application.active_window	reference to the active window. This can be used e.g. to display messages or change the statusbar

Table 2.2: Default objects available in the IPython shell.

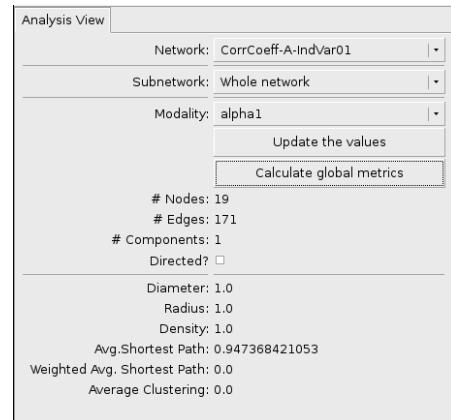
2.4.5 Interactive Python Shell

IPython constitutes an extension to the ordinary Python Shell with an abundant set of functionality [21]. It is completely integrated (as a plugin) into the ConnectomeViewer and allows scripting of the application. With the integrated Text Editor (Section above), scripts can be written and directly executed. All generated variables are available in the IPython Shell for further inspection and manipulation.

This scheme allows for very flexible interaction with the Connectome Data. Examples are shown in the application section, an example script is listed in the Appendix B. Through import of external packages, one can take advantage of their functionality. Such as matplotlib for all kind of plotting activities [22] or NetworkX for complex network analysis [16]. ConnectomeViewer uses the NetworkX library to represent networks.

2.4.6 Analyzer Node View

The Figure on the right shows a view to readily get an overview about global network metrics for a selected network, and its subnetwork. The subnetwork is defined as the subgraph defined by the selected nodes in the 3D View. Furthermore, for weighted measures, the used edge attributes can be specified. The selected network is readily available in the IPython Shell as `analyze_node.current_Graph`. See the applications section for further information about possible ways to analyze networks with the ConnectomeViewer.



2.4.7 More functionality

The loaded Connectome File is exposed in the Ipython Shell as variable `cfile` and contains all data objects that are stored in the Connectome File. For an overview of the object structure and a subset of relevant attributes and functions, see Section 2.3 and Appendix E.

Short Application Example

An averaging operation over all loaded networks from the Connectome File might be a common task. The functional EEG dataset as described in Section 4.3 was used for this short example. The following Listing performs the necessary steps to create an average network for the resting condition with eyes closed in the beta1 frequency band. The script adds the generated average network to the Connectome File View Tree for further inspection. Figure 2.5 shows the resulting average network.

```

1 # imports numerical python library
2 import numpy as np
3 # helper variable number of nodes
4 nr_nodes = len( cfile.networks[0].graph.nodes() )
5 # helper 3d matrix for data storage
6 avg_matrix = np.zeros( (nr_nodes, nr_nodes, len(cfile.networks)) )
7 # loop over all networks to extract the data for averaging
8 for i, netw in enumerate(cfile.networks):
9     # skip the electrodes network
10    if netw.networkname == 'Electrodes':
11        continue
12    # store the connectivity matrix in the helper matrix
13    avg_matrix[:, :, i] = netw.get_matrix(weight_key='R-ROIinLagConn-beta1')
14 # compute the average
15 average = np.average(avg_matrix, 2)
16 # add the network to the connectome file
17 cfile.add_network_from_matrix(name = 'Average Network', matrix = average, like_network = cfile. ←
18     networks[0], directed = 0)
19 # now, we can inspect it as needed
20 # to remove the network again
21 # cfile.remove_network(len(cfile.networks)-1)

```

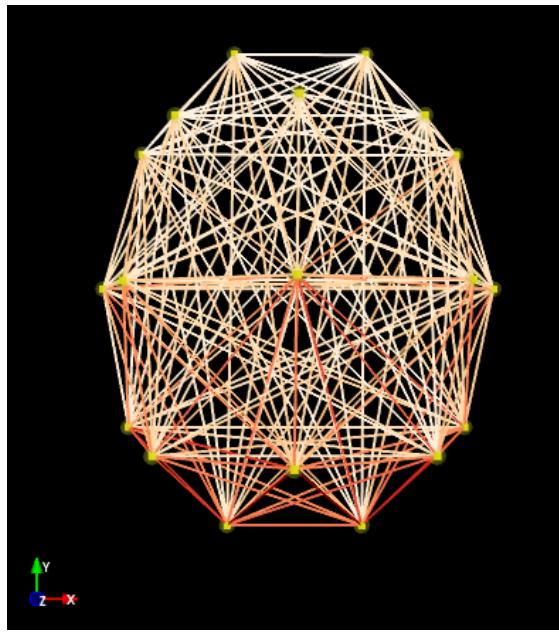


Figure 2.5: Topview of the average EEG attention dataset generated for eye-closed, resting condition in the beta1 frequency range.

Integration with TrackVis

Section 2.2.5 explains how the ConnectomeViewer integrates with TrackVis for tractography visualization. This is helpful, for example, if we want to inspect individual tracts for selected nodes which score high on specific network measures.

In Figure 2.6, two occipital nodes are selected from a single subject Connectome File. This datasets contains track data and a volumetric segmentation of all the nodes. We can invoke TrackVis to generate a Scene file. For the two selected nodes, the corresponding ROIs in TrackVis are automatically generated. Eventually, we are able to inspect the single virtual fibers connecting these two ROIs in TrackVis, as shown in Figure 2.7.

Data Validation

To investigate, whether a particular brain connection was found with other methods, such as neural tract tracing, one has to consult the scientific literature. An extensive knowledge about neuroanatomy would also help. To support the process of finding relevant neuroanatomical literature and looking up known neuroanatomy, the semantic wiki web-platform ConnectomeWiki was developed. The description of the ConnectomeWiki is found in Section 3.

A node of a particular network might contain a link to a ConnectomeWiki page. For instance, all networks that are derived from the Connectome Mapping Pipeline contain such links. In the ConnectomeViewer 3D View, when over a particular node with the mouse pointer, by pressing the key 'i', a node information box appears, as shown in Figure 2.8. This information box allows a one-click access to the corresponding brain region page in the ConnectomeWiki. On this webpage, known afferent and efferent connections from and to this brain region are displayed, allowing to check the reasonableness of the connection, and to start further investigations into the relevant literature.

2.4.8 Getting data into the ConnectomeViewer

As far as converters are concerned to get data into the ConnectomeViewer, they have to comply to the Connectome File Format. Except for Connectome Files to be assembled by hand and scripts, two general converters were written to convert data from the Connectome Mapping

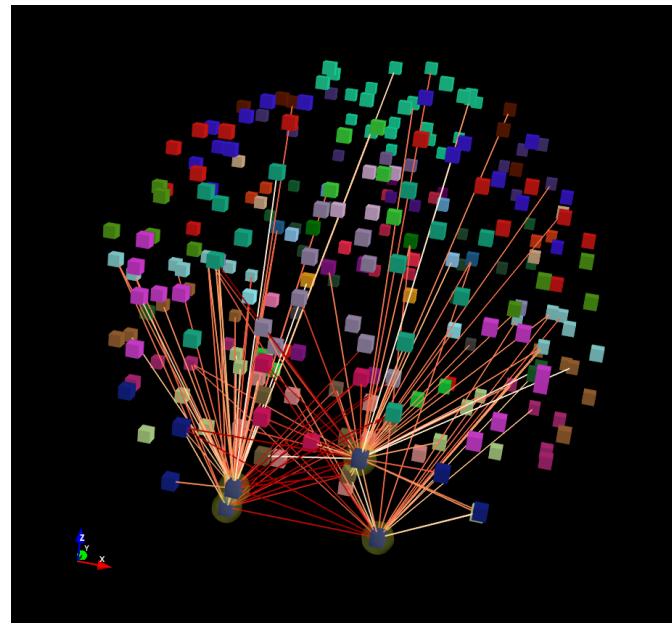


Figure 2.6: Two nodes are selected nodes in the 3D View over the occipital lobe. Adjacent edges are shown. The edges are colorcoded and represent the mean FA (mean fractional anisotropy) in this case.

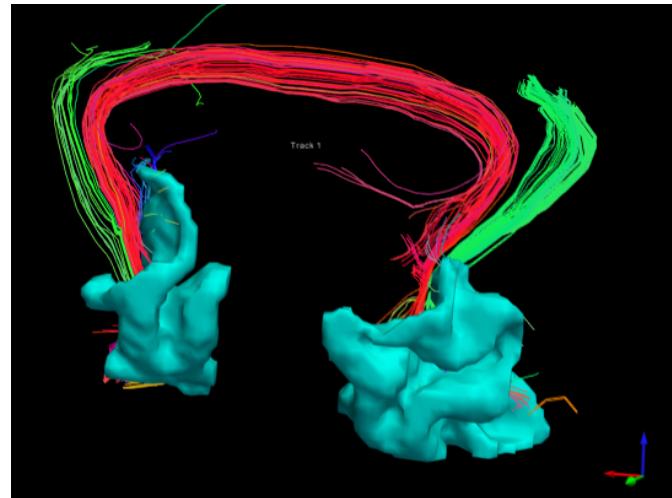


Figure 2.7: Automatically generated occipital ROIs in TrackVis from selected nodes in Figure 2.6. Fibers are filtered by these two ROIs.

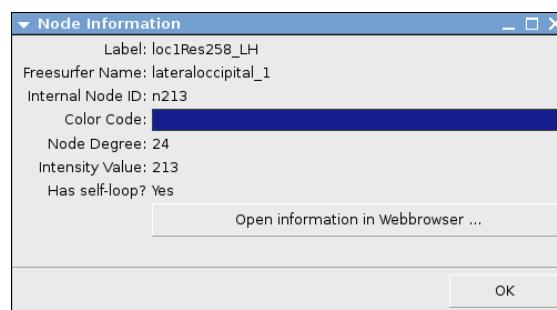


Figure 2.8: The Node Information box appearing after key pressing 'i' in the 3D View.

Pipeline [6] and from the sLORETA [23] application for EEG research into Connectome Files.

These converters will be integrated into the final ConnectomeViewer release.

Furthermore, for data storage, a plugin to access a repository of Connectome Files was developed. Details about this plugin are found in the next section about the ConnectomeDatabase.

2.5 Connectome Database

Data sharing is a key issue affecting many aspects of the scientific endeavour and it is not only facing technological, but also social obstacles. Especially in neuroscience, there are good reasons to open up laboratory data folders to other scientists [24]. Many conferences have been held, and papers and books published concerning this topic [2] but there seems still to be some general reservations among neuroscientist to share their second most valuable resource.

There might be innumerable different ways of how anyone laboratory or consortium manages its scientific data, so it would be a huge effort to integrate them all into one flexible enough application. So data federation strategies [1] are put forward to allow global data sharing, yet not abandoning completely the individual requirements.

Since Connectome data, i.e. brain connectivity information, will be exploitable by many different research fields, it makes sense to adopt some data sharing capabilities right from the start. Given the time frame and resources available, I propose here a rather simple scheme to make Connectome data available.

Separating the data from the actual application that reads and processes them is a well-known scheme and has been very successful since. Similarly, the separation between a Connectome File, and the ConnectomeViewer applications who reads and processes them. So it makes sense to create a repository of Connectome Files that can be accessed online with the ConnectomeViewer. The metadata contained in the Connectome Files should depict all relevant information to know about its content. So, the metadata content was mapped to a relational database table. In Table 2.3, you see the fields of this table.

id	
name	What is the name of this Connectome File?
species	From which species does this Connectome File contain data?
legalnotice	Under what legal conditions is the Connectome File made available?
reference	Journal articles where the dataset was used.
description	Description of the content of the Connectome File, e.g information about data acquisition and processing.
nrofnetworks	The number of networks existing in this Connectome File.
url	A webpage where further information can be found.
generator	Who generated this Connectome File?
cfile_url	Link to Connectome File from where it can be downloaded.
created	When was the Connectome File created?
modified	When was it modified?

Table 2.3: The fields in the SQL Table correspond to the metadata in the Connectome File (see Appendix D)

So far, the Connectome Files are stored as files on a web server and the links to these files are stored in the relational database tables together with the metadata. Please notice that if no further authentication mechanism are implemented, the scheme is not very secure in terms of data protection.

From within the ConnectomeViewer, a ConnectomeDatabase plugin (Figure 2.9) allows to access this database given the correct credentials, and one can select a set of Connectome

files, based on the metadata information, to download them onto the local harddisk.

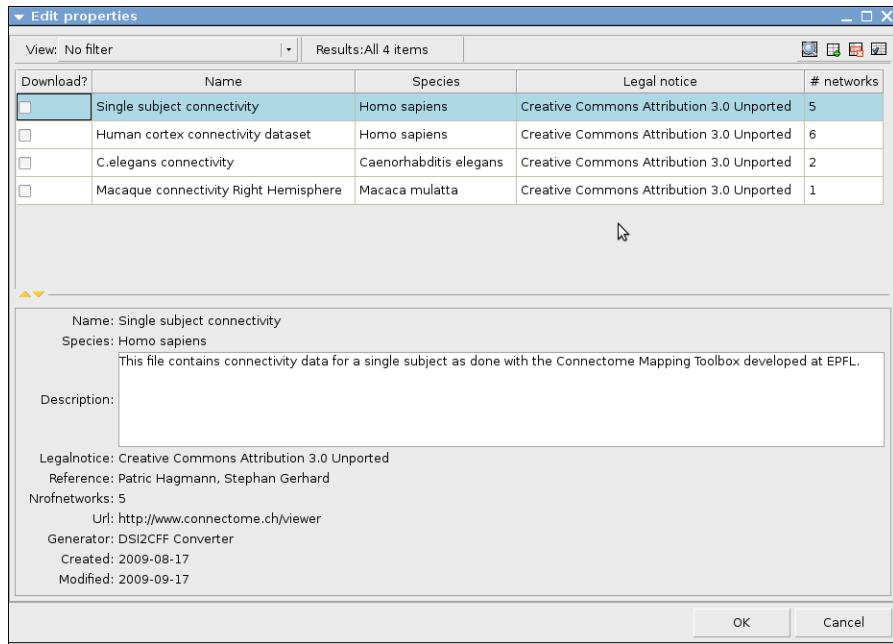


Figure 2.9: The ConnectomeDatabase plugin view from within the ConnectomeViewer provides access to a relational database.

Many more elaborate data repositories can be envisaged, allowing potentially more sophisticated search queries. For instance, spatial databases allow to search for connectivity given a coordinate in a stereotaxic space, such as the Thalamus Connectivity Database [29]. Databasing efforts where large amounts of data are involved are emerging in the fMRI community, such as the fMRI Data Center [25]. To incorporate valuable metadata about subjects, session, study design etc, it has to be agreed on standardized templates to store this information. XCEDE [12] is a promising effort for this purpose.

2.6 Conclusion

This chapter provided a non-technical overview and introduction to the functionality of the ConnectomeViewer software application. It introduced the Connectome File Format for Connectome data storage. This file format is deployed by the ConnectomeViewer. The Graphical User Interface was described and a short application was given. Tutorials and example data are downloadable from the homepage (Appendix A). For further development, the Launchpad web platform is used [30].

Further work concerning the development of the ConnectomeViewer includes:

- Inclusion of more sophisticated Network Measures that are geared toward Brain Connectivity analysis [32].
- Better visualization capabilities for neuroimaging datasets, for instance the implementation of better layouting algorithms, e.g. graph edge bundling techniques [31]
- A parser for TrackVis .trk files to access individual track data and doing pythonic analysis on them, for instance segmentation into labeled bundles.
- Statistics plugin interfacing with many existing descriptive and inferential statistical procedures, e.g. to allow to readily do group-based statistics.
- Inclusion of template atlases of other species than *Homo sapiens*.

Bibliography

- [1] What is Python and Why Python. Retrieved December 22, 2009, from http://pythoncard.sourceforge.net/what_is_python.html
- [2] Koslow S, Subramaniam S. Databasing the Brain: From Data to Knowledge. *Wiley*, 2005.
- [3] Johansen-Berg H, Behrens T. Diffusion MRI: From Quantitative Measurement to In-vivo Neuroanatomy. *Academic Press*, 2009.
- [4] Denk W. Department of Biomedical Optics. Max Planck Institute for Medical Research. Retrieved December 22, 2009, from <http://wbmo.mpimf-heidelberg.mpg.de/>
- [5] Markram, H. The blue brain project. *Nat Rev Neurosci*. 7, 153-160, 2006.
- [6] Hagmann P, Cammoun L, Gigandet X, Meuli R, Honey CJ, et al. Mapping the Structural Core of Human Cerebral Cortex. *PLoS Biol* 6(7): e159, 2008.
- [7] Sporns O, Tononi G, Kötter R. The Human Connectome: A Structural Description of the Human Brain. *PLoS Comput Biol* 1(4): e42, 2005.
- [8] Meunier D, Lambiotte R, Fornito A, Ersche KD and Bullmore ET. Hierarchical modularity in human brain functional networks. *Front. Neuroinform*. 3:37, 2009.
- [9] Python in neuroscience (Special Topic). (2009). In *Frontiers of Neuroinformatics*. Retrieved December 22, 2009, from <http://frontiersin.org/neuroinformatics/specialtopics/8/>
- [10] Open format. (2009, December 11). In *Wikipedia, the free encyclopedia*. Retrieved December 22, 2009, from http://en.wikipedia.org/wiki/Open_format
- [11] ZIP (file format). (2009, December 14). In *Wikipedia, the free encyclopedia*. Retrieved December 22, 2009, from [http://en.wikipedia.org/wiki/ZIP_\(file_format\)](http://en.wikipedia.org/wiki/ZIP_(file_format))
- [12] XML-Based Clinical Experiment Data Exchange Schema (2009). Retrieved December 22, 2009, from <http://www.xcede.org/XCEDE.html>
- [13] INCF - Minimal Metadata Standards. Retrieved January 4, 2010, from <http://incf.org/about/programs/metadata>
- [14] PyNIfTI - Pythonic access to NIfTI and ANALYZE files (2009). Retrieved December 22, 2009, from <http://niftilib.sourceforge.net/pynifti/>
- [15] Geometry format under the Neuroimaging Informatics Technology Initiative (2009). Retrieved December 22, 2009, from <http://www.nitrc.org/projects/gifti/>
- [16] Hagberg A, Schult D, and Swart P. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Aug 2008.
- [17] Ruopeng Wang, Van J. Wedeen, TrackVis.org, Martinos Center for Biomedical Imaging, Massachusetts General Hospital
- [18] Kitware. (2009). VTK - The Visualization Toolkit. Retrieved December 22, 2009, from <http://www.vtk.org/>

- [19] OpenGL - The Industry Standard for High Performance Graphics (2009). Retrieved December 22, 2009, from <http://www.opengl.org/>
- [20] Mayavi Project - 3D Scientific Data Visualization and Plotting (2009). Retrieved January 5, 2009, from <http://code.enthought.com/projects/mayavi/>
- [21] IPython Tutorial (2009). Retrieved December 22, 2009, from <http://ipython.scipy.org>
- [22] matplotlib (2009). Retrieved December 22, 2009, from <http://matplotlib.sourceforge.net/>
- [23] Pascual-Marqui R. Instantaneous and lagged measurements of linear and nonlinear dependence between groups of multivariate time series: frequency decomposition. 2007. Retrieved December 22, 2009, from <http://arxiv.org/abs/0711.1455>
- [24] Gardner D (2007). Datasharing.Net. Retrieved December 22, 2009, from <http://datasharing.net/>
- [25] fMRI Data Center (2009). Retrieved December 22, 2009, from <http://www.fmridc.org/>
- [26] The GraphML File Format (2004). Retrieved December 22, 2009, from <http://graphml.graphdrawing.org/>
- [27] Pittsburgh Brain Competition. Retrieved December 22, 2009, from <http://pbc.lrdc.pitt.edu/>
- [28] Scalable BLOB Streaming Infrastructure for MySQL (2009). Retrieved December 22, 2009, from <http://www.blobstreaming.org/>
- [29] Thalamic Connectivity Atlas (2003). Retrieved December 22, 2009, from <http://www.fmrib.ox.ac.uk/connect/>
- [30] LaunchPad ConnectomeViewer Development Platform (2009). Retrieved December 22, 2009, from <https://launchpad.net/connectomeviewer>
- [31] Holten D, van Wijk J. Force-Directed Edge Bundling for Graph Visualization. In *11th Eurographics/IEEE-VGTC Symposium on Visualization*, Pages 983 - 990, 2009.
- [32] Rubinov M, Sporns O. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage*, 2009. Retrieved December 22, 2009, from <http://www.brain-connectivity-toolbox.net/>
- [33] Scientific Tools for Python. Retrieved December 22, 2009, from <http://www.scipy.org/>
- [34] Numerical Python. Retrieved December 22, 2009, from <http://numpy.scipy.org/>
- [35] Traits Technology for Python. Retrieved December 22, 2009, from <http://code.enthought.com/projects/traits/>
- [36] Millman KJ, Brett M. Analysis of Functional Magnetic Resonance Imaging in Python. *Computing in Science and Engineering*, vol. 9, no. 3, pp. 52-55, 2007.
- [37] enthought blog. Retrieved December 22, 2009, from <http://blog.enthought.com/>
- [38] Envisage. Extensible Application Framework. Retrieved December 22, 2009, from <http://code.enthought.com/projects/envisage/>
- [39] SPL-PNL Brain Atlas 2008. Retrieved December 22, 2009, from <http://www.spl.harvard.edu/publications/item/view/1265>
- [40] The MNI brain and the Talairach atlas (2006). Retrieved December 22, 2009, from <http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach>

Chapter 3

ConnectomeWiki

3.1 Introduction

Dealing with the complexity of human neuroanatomy, it is clear that there are many levels of investigations. For instance, traditional neuroanatomy research dealing with brain sections, tracer injections and neural tract tracing on one side. On the other side, there is Diffusion MRI research dealing with the reconstruction of virtual fiber pathways. It was clear to me that people from both camps need to talk to each other in some constructive way.

In the beginning, I was looking for web resources where I could ask question such as: What are all the afferent and efferent connections of brain region X in a given species? What are differences in this brain region and connectivity in different species? What are the studies doing electrophysiology in this region? What is the internal organisation, i.e. sub-regions, of this region and how they are differentiated? In what superregion is this region contained in? What cell types exist in this region? What are functions with putative involvement of this region?

What I found is that such questions can be answered, but each of them needs long-lasting literature searches, the information is widespread. Inspired by the CoCoMac project [4], the Neuroscience Lexicon project [2] and recent synthetic efforts such as the Neuroscience Information Framework [1], I was thinking about building a platform that can be integrated with ConnectomeViewer, yet being as separate and useful to represent neuroanatomical knowledge from the literature in a very accessible way. Only later, when the ConnectomeWiki took form, the paper *A Proposal for a Coordinated Effort for the Determination of Brainwide Neuroanatomical Connectivity in Model Organisms at a Mesoscopic Scale* by 37 neuroscientists [3] drew my attention with a very similar idea.

Using recent technologies available in the internet communities, especially the idea of a linked web of data [5], the setup of a wiki with semantic extensions seemed like the only logical and sustainable way to go. There are several reasons for this:

- A wiki provides an asynchronous "forum" where experts in their respective field can write down their particular knowledge.
- A good virtual place for discussing conflicting standpoints is implicit in the wiki architecture.
- There should be no predefined set of properties one can express for brain regions and brain connections. Users should be able to create their own when needed.
- A wiki can continuously grow with time, building up a community of researchers.
- It can well integrate with other platforms and databases or other data sharing platforms.
- Semantic queries are easy and powerful, and become very useful once there is enough data. (See Section 3.3)

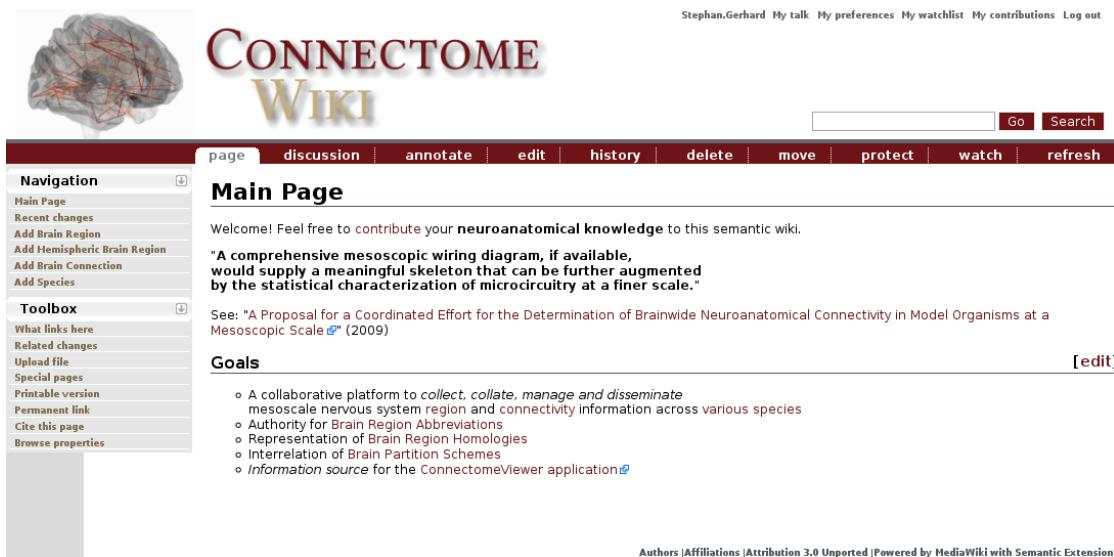


Figure 3.1: A screenshot of the start page <http://connectome.ch/wiki/>

3.2 Basic Ontology

Basically, the ontology employed for the semantic wiki is kept as simple as possible allowing people to quickly understand it and get involved. The three basic concepts are: species, brain region and brain region connection.

Wiki pages represent instances of this basic concepts and are explained in more details in the following sections. Using a templating mechanism, adding instances of one of these three basic types automatically annotates the page as belonging to the appropriate category: brain region, brain connection or species. The templates predefine specific properties (see following sections), but it is also possible to add normal (wiki-style) text. Additionally, so called semantic content can be annotated, such as categories and/or semantic properties. Such kind of annotations lie at the heart of semantic wiki technology and offer powerful capabilities for sophisticated data querying.

3.2.1 Brain Region

It is possible to tag a Brain Region as belonging to a particular *Brain Region Partition Scheme*. The main partition schemes, which are based on well-agreed anatomical subdivision of the brain, are called *Foundational Partition*.

Adding a special Partition Scheme, a special naming convention for the page names is adopted (see next section) and the Brain Region page is tagged as belonging to this special Partition Scheme, e.g. *Category:Brodmann_(1909)* for Brodmann's cytoarchitectonically defined regions [6] with names BA1, BA2, etc. or the partition from Economo and Koskinas relabeled from EK1 until EK107 [7].

Brain Region Naming Conventions

The abbreviation of a Brain Region serves as a label and unique ID together with the species binomial name to identify the Brain Region. Using species binomial name [8] in parenthesis in the page name opens up the possibility to use the ConnectomeWiki for any number of species, since the basic scheme is essentially the same.

Property:	Description
Abbreviation	The abbreviation conforming to the conventions. It is used also as part of the page title.
Species	Uniquely identifying the species this brain region belongs to with its binomial name.
English Name	The most commonly used English name for this brain region.
Latin Name	The old Latin name used when existing.
Other Name(s)	Any other name that is existing to describe this brain region.
Defining Criteria	It is possible to select the delineation criteria that was used for this brain region. Since this wiki concerns structural data, functional data is excluded. Criteria are: anatomy, boundaries, chemoarchitecture, connectivity, cytoarchitecture, dendroarchitecture, developmental, imaging, myeloarchitecture, pigmentarchitecture or other ROI. You can look up the corresponding property details for more information.
Definition	A short definition of this brain region, probably containing information contingent upon the defining criteria.
Parcellation Protocol	If there exists an official protocol on how to delineate this brain region, this can be described here. For instance descriptions of the boundaries for anatomical delineations.
Function Tag(s)	Single words expressing some functional involvement of this brain region.
Wikipedia Link	Whenever there is a Wikipedia page, link to it.
Neurolex Link	Linking to the Neuroscience Lexikon
BraininfoID	Link to the well-known and extensive database of brain regions.
Bredewiki Link	A direct link to a wiki containing functional MRI studies.
Free text	Most importantly, references to journal articles and books are made here. Additionally, one can add images, information about neuronal populations in this region, putative functions, links to electrophysiology data and more.
Part of	Here we specify superregions, thereby building a partonomy from bottom to top.

Table 3.1: Brain Region Properties included in the Template

Of course, these abbreviations do not emerge from the void, but are whenever possible, derived from well-established and accepted nomenclatures and standardized terminology in the field of neuroanatomy. For example for the human brain, the partonomy of the brain is mainly derived from NeuroLex [2] and the Foundational Model of Anatomy [9]. Table 3.2 exemplifies the naming scheme for brain region pages in the ConnectomeWiki.

Construction principles of abbreviations

The conventions are partly derived from works of George Paxinos [10]. A driving principle is that one should easily remember the structure name when seeing the abbreviation.

- The abbreviations are usually based on the English name, rather than on their Latin name.
- Certain patterns for specific kinds of structures: *N: for nuclei, *trc: for tracts, *seg: for segments of the spinal cord
- Anatomical specifications are put after the abbreviation of the main structure:
e.g. Dorsal tegmental nucleus into TNd
e.g. Opercular part of inferior frontal gyrus into IFGOp
- If structures contain the same name, they are almost always given the same abbreviation:
e.g. TH for Thalamus, MTH for Metathalamus
- Arabic numerals are used instead of Roman numerals e.g. for cranial nerves, layers of the cortex:
e.g. CNVIII into CN8 for the Vestibulocochlear nerve
- Well established abbreviations are taken despite of these conventions.

Brain Region Levels

Three principle levels of description are distinguished in the ConnectomeWiki with increasing resolution, but still describe the mesoscale connectivity. The main level is the top-most level (*Category:Brain Region*) where most of the data resides. The other two levels were introduced to take into account the potential expression of interhemispheric connections (*Category:Brain Region Hemisphere*), and the layer level (*Category:Brain Region Layer*) to express even more details. Because the layer level is already departing from a mesoscale level description quite a bit, it is not really used yet.

Category:	Naming convention	Example page
Brain Region	BrainRegionAbbrev_(SpeciesName)	V1_(Homo_sapiens)
Brain Region Hemisphere	BrainRegionAbbrev_LH_(SpeciesName) or BrainRegionAbbrev_RH_(SpeciesName)	amgRes83_RH_(Homo_sapiens)
Brain Region Layer	BrainRegionAbbrev_LH_L1_(SpeciesName) or BrainRegionAbbrev_L1_(SpeciesName)	A1_LH_L2_(Homo sapiens)

Table 3.2: Three levels with the naming convention employed and an example

Important guiding principles to interrelate these levels:

- The *Property:Part of* should only be used on the Brain Region level!
- Linking the Brain Region Hemispheric Level to Brain Region Level, the *Property:Hemispheric_part_of* is used.
- Linking the Brain Region Layer Level to Brain Region (Hemispheric) Level, the *Property:Layer_part_of* is used.

Literatur search and other databases

Hyperlinks to many other relevant resources were added, thereby weaving the semantic web. Currently, efforts such as ScienceCommons [27] and NeuroCommons [28] are building Open Source knowledge management platforms, which will become more important in the future as Semantic Web technologies advance. A quick description of the resources that were linked from the ConnectomeWiki follows.



Figure 3.2: Links to other resources

Wikipedia Of course, Wikipedia provides some information about brain regions and brain anatomy, mainly for *Homo sapiens*.

BrainInfo BrainInfo, formerly NeuroNames has a long history (starting 1991). BrainInfo helps in identifying structures in the brain of many species and provides a partonomy.

Neurolex A recent effort to build a Lexikon with well-defined terms for the neurosciences. Concerning the brain regions, it is derived mainly from BrainInfo. We are in the process of backlinking ConnectomeWiki pages from Neurolex.

Bredewiki A recent effort to gather information about published peer-reviewed neuroimaging articles (mainly fMRI). It already links with ConnectomeWiki brain regions and connectivity.

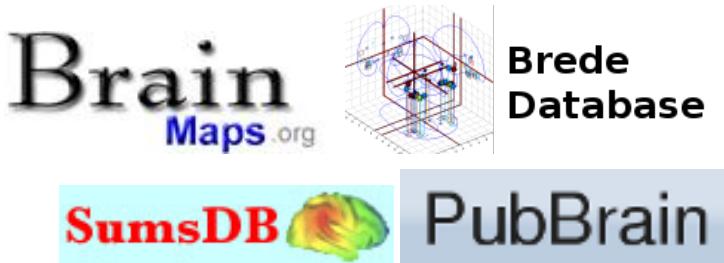


Figure 3.3: Query external databases

BrainMaps An interactive multi-resolution atlas. It contains very high resolution, annotated serial sections of both primate and non-primate brains.

BredeDatabase This database contains findings from functional MRI study activations which are searchable by Talairach coordinates or anatomy.

SumsDB The Surface Management System DataBase contains among others searchable stereotaxic foci and fMRI activation maps.

PubBrain Allows to make Pubmed queries, thereby displaying found articles results containing anatomical terms or activations in a dynamically colored brain.



Figure 3.4: Literature search and Text Mining databases

NIF The Neuroscience Information Framework is a dynamic inventory of Web-based neuroscience resources. Its goal is to serve as a gateway to all the neuroscience resources in the web.

GoPubMed Provides a very nice and supportive interface to make literature searches in the Pubmed database, using semantically annotated text intensively.

Lucene and Textpresso These are two interfaces to a text mining system concerned with brain architecture including brain structure. It allows to search a mined text-corpus of about 55000 full-text journal articles.

3.2.2 Brain Region Connection

The convention for brain region connection page titles is to hyphenate two brain regions and append the binomial species name in parenthesis. This means that such connections are implicitly directed.

Category:	Brain Region Connection
Naming convention	BrainRegionAbbreviationFrom-BrainRegionAbbreviationTo_(Species_name)
Example page	LGN-V1_(Homo_sapiens)

Table 3.3: The naming convention employed for brain region connections

Property:	Description
Abbreviation	The abbreviation as seen in the page name (e.g. LGN-V1).
Species	The binomial name for the species where this connection was found.
Description	A quick description of this brain region connection characteristics.
From	From which brain region this connection starts (is implied by the page title).
To	To which brain region this connection leads to.
Mediating NT	What is the mediating neurotransmitter found for this connection? Allowed values so far: Acetylcholine, GABA, Glutamate, Norepinephrine, Dopamine, Serotonin
Synaptic Mechanism	Allowed values so far are: Excitatory, Inhibitory, Neuromodulatory, Unknown
Reciprocal?	Is this connection reciprocal. This allows to decrease the number of pages generated if only the reciprocity property is known but not more.
Free Text	Adding images, electrophysiology data or others information, but most importantly, the references to journal paper or books where this connection is investigated or mentioned.

Table 3.4: Brain Region Connection Properties included in the Template

Category:Brain Region	Category:Brain Region Connection
Levels for structural delineations	Connections found between brain regions.
[–] Brain Region (1)	[–] Brain Region Connection (7)
[–] Brain Region Hemisphere (1)	[+] Cerebelloafferent fibers (0)
[x] Brain Region Layer (0)	[+] Cerebelloefferent fibers (0)
	[+] Corticoafferent fibers (0)
	[+] Corticoefferent fibers (0)
	[+] Spinoafferent fibers (0)
	[+] Spinoefferent fibers (0)
	[+] Thalamoafferent fibers (0)
Category:Cortex	Category:Anatomy
Tags a Brain Region as cortical, with subclasses	Express more anatomical knowledge about brain regions
[–] Cortex (3)	[–] Anatomy (7)
[–] Allocortex (2)	[+] Ganglia (0)
[+] Archicortex (0)	[+] Grey Matter (0)
[x] Paleocortex (0)	[+] Gyrus (0)
[+] Isocortex (0)	[+] Nucleus (0)
[+] Mesocortex (0)	[+] Pole (0)
	[+] Sulcus (0)
	[–] White Matter (1)
	[–] White Matter Tract (5)
	[–] Association fiber (2)
	[+] Long Association fiber (0)
	[+] Short Association fiber (0)
	[+] Brainstem fiber (0)
	[+] Commissural fiber (0)
	[+] Cranial nerve (0)
	[+] Projection fiber (0)

Figure 3.5: Category hierarchies to specifically annotate brain regions and connections.

3.2.3 Main Categories

The two main entities in this wiki, namely Brain Regions and Brain Region Connections, apart from having any number of properties, can be annotated (jargon for: classified). A set of brain regions that belong to the same class or category, should have the same annotation. For this, one can use the Category Tag. This becomes clearer with an example:

The human primary auditory cortex is represented with the page title **A1_(Homo_sapiens)**. This brain region is a cortical brain region. So we annotate it with *Category:Cortex*. Because we know, that this areas is a special kind of cortex, namely Isocortex, we want to express this. Thus, we annotate it with *Category:Isocortex* by directly typing in the free text field when editing, or using the annotate-menu on top of the page. Later, using semantic search, we can display all the cortical brain regions that are considered Isocortex. The primary auditory cortex should be visible in the result set. Similarly, other categorial hierarchies already exist in the ConnectomeWiki for brain regions, brain region connections and species. They are shown in Figure 3.5.

3.3 Example Applications

For semantic wikis, there are sophisticated query mechanism. A special Query Interface helps in creating this queries and outputs a syntactically correct query that can be embedded in wiki pages. I will present here some example queries together with some explanation. More details about query processing can be found here [11].

3.3.1 Efferent connections of zebra finch robust nucleus

Question: Display all the efferent connections for the robust nucleus RA in zebrafinch.

Query:

```
 {{#ask: [[Category:Brain Region Connection]]
[[Species::+]]
[[Species::Taeniopygia guttata]]
[[From::+]]
[[From::RA (Taeniopygia guttata)]]
| format=table
| link=all
| }}
```

Each connection contains a list of reference to papers where this connection is mentioned or investigated. If available, also information about mediating neurotransmitter and synaptic mechanisms and more can be shown. A similar query is used within brain region pages in the wiki to automatically display the afferent and efferent connections of the particular brain region.

Output:

```
RA-DLM (Taeniopygia guttata)
RA-DM (Taeniopygia guttata)
RA-DMP (Taeniopygia guttata)
RA-NAM (Taeniopygia guttata)
RA-NRA (Taeniopygia guttata)
RA-nXIIts (Taeniopygia guttata)
```

3.3.2 Subregions of a brain region STG in Homo sapiens

Question: Display the english name of all the stored subregions of the Superior Temporal Gyrus of Homo sapiens.

Query:

```
 {{#ask: [[Part of:: <q>[[STG_(Homo_sapiens)]] </q> ]]
| ?NameEnglish=English Name
| mainlabel=Abbreviation (Species)
| link=all
| limit = 10
| default=No subregions found.
| format=table
| }}
```

The partonomy of brain regions is constructed from bottom to top with the Part-of property. Nevertheless, semantic wiki provides a way to automatically query the inverse property, i.e. has parts, in this case.

Output:

Abbreviation (Species)	English Name
FST (Homo sapiens)	Floor of superior temporal sulcus
MSTD (Homo sapiens)	Medial superior temporal cortex dorsal
MSTl (Homo sapiens)	Medial superior temporal cortex ventral
STGa (Homo sapiens)	Anterior superior temporal gyrus
STGp (Homo sapiens)	Posterior superior temporal gyrus
STP (Homo sapiens)	Superior temporal plane
STPa (Homo sapiens)	Superior temporal polysensory cortex anterior
STPp (Homo sapiens)	Superior temporal polysensory cortex posterior

3.3.3 Topographical zebra finch nuclei

Question: I want to know which nuclei of the zebra finch (*Taeniopygia guttata*) were found to have a topographical organization of neuronal response properties.

Query:

```
 {{#ask: [[Category:Brain Region]]
[[Species::+]]
[[Species::Taeniopygia guttata]]
[[Topography::+]]
[[Topography::Yes]]
| format=table
| link=all
| }}
```

This query outputs a table of zebra finch brain regions where the Topography property is set to Yes. The information about topography was in this case derived from Iyengar et al. [12]. *Output:*

AreaX (Taeniopygia guttata)
FieldL (Taeniopygia guttata)
LMAN (Taeniopygia guttata)
RA (Taeniopygia guttata)
nXIIts (Taeniopygia guttata)

3.4 Initial Datasets

Adding new brain regions and connections can be achieved by creating them over using semantic web forms. If you have thousands of them, one might prefer to write a script [13] that does these imports automatically. I used the latter for many initial imports. These data import sources are quickly described in the following. As of January 2010, the statistics are shown in Table 3.5.

Species	Brain Regions	Brain Region Connections
Homo sapiens		
<i>Foundational Partition</i>	876	65
<i>Lausanne Partition</i>	2006	0
Macaca mulatta	317	1
Taeniopygia guttata	40	40
Rattus norvegicus	20	120
Felis catus	5	0

Table 3.5: Current statistics (January 2010) of the content of the ConnectomeWiki

Homo sapiens Foundational Partition As one might imagine, there is no universally accepted scheme to partition a generic human brain. When one searches the internet and consults relevant literature, one is confronted with many more or less up-to-date, conflicting resources. Only recently, and also with the advent of new computer technologies to build ontologies, there are international efforts underway to close this gap [14, 15]. Most notably, the NeuroLex [2] based upon NIF Standard Ontology tries to provide a scaffold for projects in need for such data.

Zebra finch nuclei and connections The Songbird research group at the Institute of Neuroinformatics provided data for zebra finch (*Taeniopygia guttata*) brain nuclei and connections with many references. The scripts for generating the pages and importing them into the ConnectomeWiki are available [18]. Additionally, I added some more data from connectivity diagrams from the literature and updated and annotated the connections where possible.

Rat hippocampal-parahippocampal connectivity Niels van Strien and his research group from the Norwegian University of Science and Technology allowed to include a subset of their connectivity data about [19] hippocampal-parahippocampal connectivity. The full dataset is available from their review paper [20].

Rhesus monkey Partition Scheme Brain regions are derived from the well-known atlas by Saalem and Logothetis [21]. Additionally, some connectivity data with references were added.

Lausanne Partition Schemes Derived from the Connectome Mapping Pipeline, cortical partition in five different resolutions including 17 subcortical nuclei were available. The resolution are a) 83 b) 150 c) 258 d) 500 e) 1015 cortical regions. Each hemispheric region for each resolution has an unique label and was then added to the ConnectomeWiki as being part of a well-known anatomical structure from the foundational partition. Of course, these labels are compatible with the converter from the Pipeline to the Connectome File Format. This allows to directly access the appropriate wiki-pages from within the ConnectomeViewer for a given network processed with the Connectome Mapping Pipeline.

Brodmann Partition Scheme Of course, the Brodmann areas [6] are well-known and are widely used. I used information derived from Wikipedia [22] and this site [23] for the description of where they are located.

brainCOLOR: Collaborative Open Labeling Online Resource Jason Tourville and Ruth Carper have developed a cortical parcellation to be used to label hundreds of brains in the next years. They kindly allowed to include the delineation protocol for brain regions into the ConnectomeWiki. For more information, see their homepage [24]. The labeled brains will then be made available to the public.

3.5 Conclusion

The field of neuroanatomy is quite old, first testimonies date back to egypt period [25], with some revivals in the Renaissance and then with a modern expansion and better documented investigations beginning in 17th century [26]. Bringing the vast body of knowledge aquired over the centuries to the information technology age with the internet and semantic wikis, one must take a broader perspective.

Contemplating the vast complexity of the systems under investigation, one might hope to have a comprehensive map at the mesoscale level within the next decades or so. This will need persistent, collaborative efforts, no one-man show. So the ConnectomeWiki could be an important platform, employing the 21st century IT technology, in this endeavor of describing the (coarse) structural details of nervous systems.

The integration of the ConnectomeWiki within the Neuroscience Information Framework [1] is underway and is an important step in making the resources accessible from a centralized neuroscience portal.

The combination of the ConnectomeWiki with the ConnectomeViewer might as well induce some synergy effect. For instance, the wiki provides well-established neuroanatomical knowledge from the literature, which can be used to validate the reconstructed networks from Diffusion MRI-derived Connectome data against it. On the other hand, the ConnectomeWiki might serve as an interface to query a ConnectomeDatabase which stores data from Diffusion MRI studies (as described in Section 2.5).

Issues of quality control and scaling-up have to be addressed in the future too.

Bibliography

- [1] Martone M, et al. The Neuroscience Information Framework (NIF): accessing diverse neuroscience resources through a single interface. *Frontiers in Neuroinformatics*. Conference Abstract: Neuroinformatics 2008.
- [2] Gardner D, et al. The neuroscience information framework: a data and knowledge environment for neuroscience. *Neuroinformatics* 6 (3): 149-60, 2008.
- [3] Bohland J, et al. A Proposal for a Coordinated Effort for the Determination of Brainwide Neuroanatomical Connectivity in Model Organisms at a Mesoscopic Scale. *PLoS Comput Biol*. 5(3): e1000334, 2009
- [4] Stephan KE, Kamper L, Bozkurt A, Burns GA, Young MP, et al. Advanced database methodology for the Collation of Connectivity data on the Macaque brain (CoCoMac). *Philos Trans R Soc Lond B Biol Sci*;356:1159–1186, 2001.
- [5] Linked Data - Connect Distributed Data across the Web. Retrieved December 21, 2009, from <http://linkeddata.org/>
- [6] Brodmann, K. Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues. Barth, Leipzig, Germany, 1909.
- [7] Triarhou LC. A Proposed Number System for the 107 Cortical Areas of Economo and Koskinas, and Brodmann Area Correlations. *Stereotact Funct Neurosurg*;85:204-215, 2007.
- [8] Binomial nomenclature. (2009, December 21). In *Wikipedia, the free encyclopedia*. Retrieved December 21, 2009, from http://en.wikipedia.org/wiki/Binomial_nomenclature
- [9] Rosse C, Mejino JVL. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *J Biomed Inform*. 36:478-500, 2003.
- [10] Paxinos G, Watson C. The Rat Brain in Stereotaxic Coordinates. Academic Press, 1998.
- [11] Semantic Mediawiki (2009). Help:Inline queries. Retrieved January 6, 2009, from http://semantic-mediawiki.org/wiki/Help:Inline_queries
- [12] Iyengar S, Viswanathan S, Bottjer S W. Development of topography within song control circuitry of zebra finches during the sensitive period for song learning. *J Neurosci*, Vol. 19 (14): 6037-57, 1999.
- [13] Pywikipediabot. (2009, November 13). *Meta, discussion about Wikimedia projects*. Retrieved December 21, 2009, from <http://meta.wikimedia.org/wiki/Pywikipediabot>
- [14] Program on Ontologies of Neural Structures. Retrieved December 21, 2009, from <http://www.incf.org/about/programs/p ons>
- [15] 1st INCF Workshop on Neuroanatomical Nomenclature and Taxonomy. Retrieved December 21, 2009, from <http://www.incf.org/documents/workshop-reports/ incfworkshop-1st-NeuroanatomicalNomenclature.pdf>
- [16] BrainInfo. Neuroscience Division, National Primate Research Center, University of Washington. Retrieved December 21, 2009, from <http://www.braininfo.org>

- [17] Paxinos G, Mai K, The Human Nervous System (Second Edition). *Elsevier Inc.*, 2004.
- [18] How To. (December 2009). In *ConnectomeWiki*. Retrieved December 8, 2009, from http://connectome.ch/wiki/How_To
- [19] Temporal-Lobe.com. Retrieved December 21, 2009, from <http://www.temporal-lobe.com/>
- [20] van Strien NM, Cappaert NLM, Witter MP. The anatomy of memory: an interactive overview of the parahippocampal–hippocampal network. *Nature Reviews Neuroscience* 10, 272-282, 2009.
- [21] Saleem K, Logothetis N. A combined MRI and histology atlas of the rhesus monkey brain in stereotaxic coordinates. *Academic Press*, 2007.
- [22] Brodmann area. (2009, November 29). In *Wikipedia, the free encyclopedia*. Retrieved December 21, 2009, from http://en.wikipedia.org/wiki/Brodmann_area
- [23] Dubin M. Brodmann Areas. Retrieved December 21, 2009, from <http://spot.colorado.edu/~dubin/talks/brodmann/brodmann.html>
- [24] brainCOLOR - Collaborative Open Labeling Online Resource. Retrieved December 21, 2009, from <http://www.braincolor.org/>
- [25] Edwin Smith Papyrus. (2009, November 19). In *Wikipedia, the free encyclopedia*. Retrieved December 21, 2009 from http://en.wikipedia.org/wiki/Edwin_Smith_Papyrus
- [26] History of Neuroanatomy Lecture. Retrieved Decembe4r 22, 2009, from <http://anatomy.yonsei.ac.kr/neuro/History.htm>
- [27] Science Commons. Retrieved December 21, 2009, from <http://sciencecommons.org/>
- [28] The Neurocommons. Retrieved December 21, 2009, from <http://sciencecommons.org/projects/data/>

Part II

Applications

Chapter 4

ConnectomeViewer Applications

4.1 Introduction

The ConnectomeViewer was applied to analyze and visualize four datasets: (1) Developmental Diffusion MRI data from 34 subjects (1.5 - 18 years), (2) Average Human Connectivity data, Diffusion MRI data averaged from 20 subjects, (3) sLORETA EEG functional connectivity data, and (4) reconstructed cat visual cortex neurons. The processing of each dataset is described in one subsection.

Most elaboration is devoted to the analysis and visualisation of the developmental project described in subsection 4.2. However, each application described shows possible ways the ConnectomeViewer can be used in the study of brain structural or functional connectivity. The section of the developmental project is structured as follows: Description of Dataset Acquisition, Question / Hypothesis, Acknowledgement, References, Analysis and/or Visualization Protocol (Method), Discussion and Conclusion. The Analysis Protocol consist of a description of the procedures taken and partly of the corresponding executable source code. The other sections adopt this structure only partly.

4.2 Human Developmental project

Method The first datasets contains Diffusion MRI data from 30 subjects aged 1.5 to 18 years (15 male, 19 female). Diffusion MRI datasets were processed according to the Connectome Mapping Pipeline [6]. Diffusion MRI data was aquired with high-resolution T1-weighted imaging, and Q-ball imaging. Connectome Mapping Pipeline processing yielded structural connectivity matrices with mean Apparent Diffusion Coefficient (ADC), and density measures between a) 83, b) 150, c) 258, d) 500 and e) 1015 nodes (Regions of Interest) for each subject.

Question / Hypothesis

- Is the speed of myelinisation assessed by Diffusion MRI using ADC and density measures spatially non-uniform? We expect primary sensory cortices to show a slower age-dependent increase than association cortices and subcortical regions.

Acknowledgement Patric Hagmann, Olaf Sporns

Reference Mapping the development of the human connectome, P. Hagmann, O. Sporns, S. Gerhard, R. Pienaar, J-P. Thiran, L. Cammoun, N. Madan, and P. E. Grant, International Society for Magnetic Resonance in Medicine 2010 conference. (*Submitted*)

4.2.1 Analysis Protocol

Previous studies have shown mean diffusivity of water molecules (as measured with Apparent Diffusion Coefficient in Diffusion MRI) to reflect age-related myelination changes [1]. Regional age-related changes in cortical thickness and white matter volume and microstructure were recently shown [3].

Afferent and efferent white matter connections to and from primary cortical areas such as the primary visual, auditory and motor cortex are expected to be already well-myelinated by the age of two years. In subsequent development, connections to and from associative areas are expected to get myelinated [4].

Densely packaged bundles lead to small ADC values. Furthermore the ADC index is modulated by myelinisation strength [5]. The stronger the myelinisation, the smaller the ADC index. Formula 4.1 is used to compute the age-dependent inverse myelination strength $M(t)$ for one brain region. It takes into consideration the mean ADC value of adjacent fiber bundles. Furthermore, we use the fiber density to take into account the relative influence of the number of adjacent fibers.

$$M(t) = \frac{1}{N} \sum_i ADC_{mean_i}(t) \frac{d_i(t)}{\sum_j d_j(t)} \quad (4.1)$$

M is the age-dependent strength for one brain region. t represents time, i.e. age. The edges are indexed by i . The ADC_{mean} is computed for each edge in the Connectome Mapping Pipeline. It represents the mean of all ADC values for all fibers connecting two Regions of Interest (ROIs). The calculation of the density value d is based on the Formula 4.2 (for detailed description see [6]). It is used for a relative weighting. The larger the number of incoming or outgoing fibers represented by the density d value is, the stronger is the weight attributed to the ADC_{mean} value.

$$d(i) = \frac{2}{S_u + S_v} \sum_{f \in E_f} \frac{1}{l(f)} \quad (4.2)$$

S_u and S_v denotes the surfaces of ROIs for edges from node u to node v . $l(f)$ denotes the length of fiber f . This term is introduced to correct for biases introduced by the tractography algorithms.

Linear and exponential regression analysis were performed on the inverse myelination strength for each node. Average cortical surface were colorcoded according to resulting slope values of the exponential models.

The relevant calculations are listed in appendix B. Preliminary results are visualized and described. Then, found results are discussed.

4.2.2 Results

In Figure 4.1 the computed inverse myelination strength values M are shown colorcoded for each subject in resolution 83 and 150. The Linear regression curves for all 83 nodes are shown in Figure 4.2. The gradient of each curve is shown in Figure 4.5 colorcoded on an average brain surface.

As an example, the M values of the superior parietal cortex which is part of the association cortex are depicted in Figure 4.3 (Node 18 in Resolution 83). The linear regression curve shows a negative slope. Pearson's product-moment correlation coefficient reveal a significant correlation of $r = -0.72$ ($p = 1.15e - 6$, $N = 34$). For the exponential regression, the correlation is significant with $r = -0.73$ ($p = 6e - 7$, $N = 34$).

The overall distribution of Pearson's product-moment correlation coefficient in resolution 83 and 150 are shown in histograms in Figure 4.4.

In Figure 4.8, two relatively large clusters of nodes can be distinguished: (a) which prolonged high M value as compared to region (b) with similar inverse myelination strength across subjects. This two sets of nodes a) and b) are highlighted in red on an average surface mesh in Figure 4.9 and in Figure 4.10.

The nodes highlighted as a) in Figure 4.8 are: Supramarginal gyrus, Superior parietal cortex, Inferior parietal cortex, Precuneus cortex (both hemispheres). The mean and the standard deviation of their r-value is -0.66 ± 0.05 .

The nodes highlighted as b) in Figure 4.8 are: Thalamus-Proper, Caudate, Putamen, Pallidum, Accumbens-area (one hemisphere). These are all subcortical nuclei are shown in Figure 4.10. The mean and standard deviation of their r-value is -0.49 ± 0.06 .

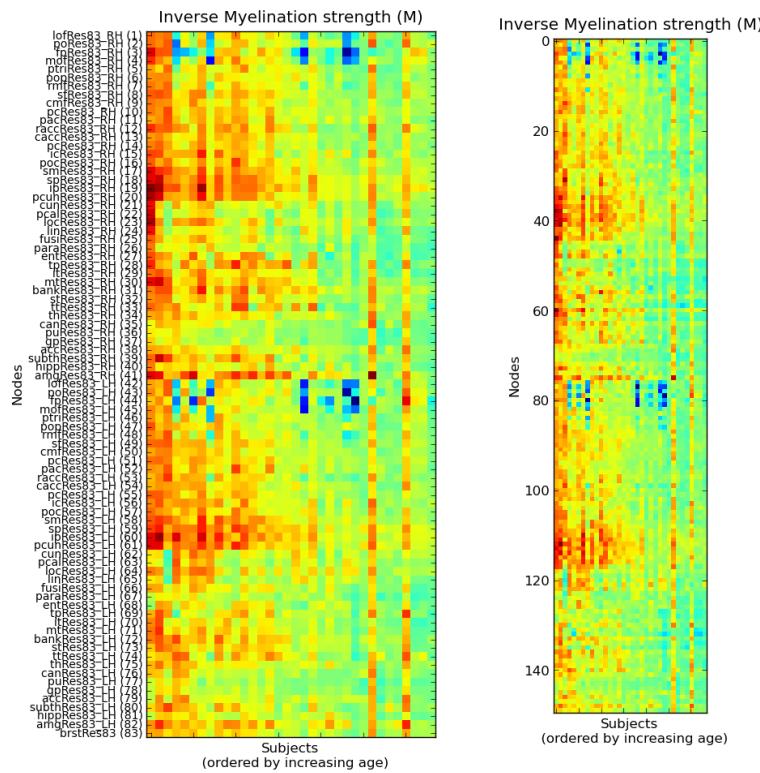


Figure 4.1: The calculated M values for all nodes and subject in Resolution 83 (left) and for Resolution 150 (right). A marked decrease in the inverse myelination strength is visible.

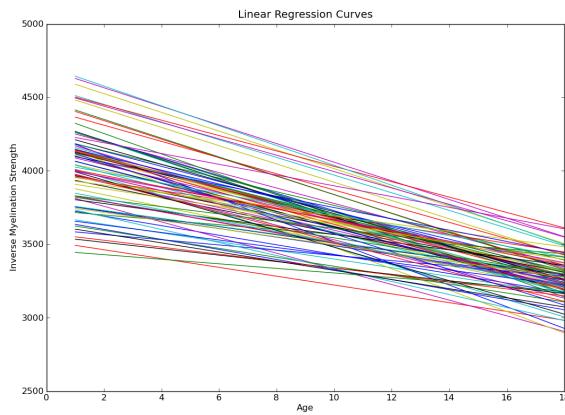


Figure 4.2: Linear regression curves for all nodes in Resolution 83.

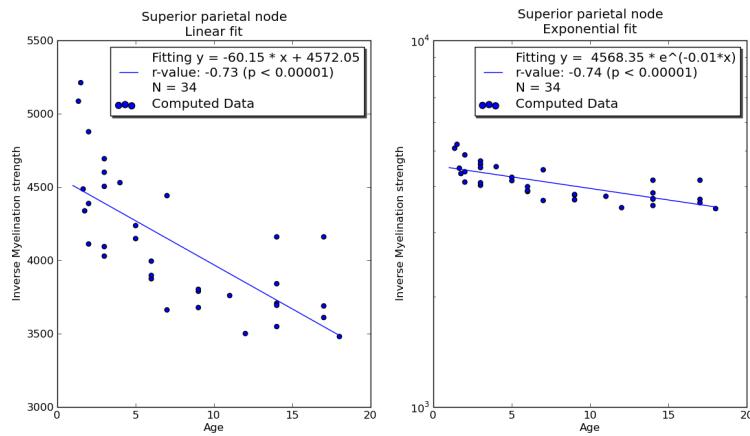


Figure 4.3: Scatterplot of a superior parietal node with fitted linear (left) and exponential (right) regression curves.

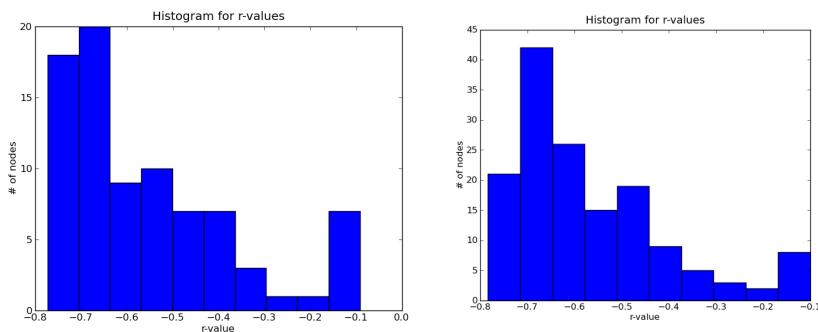


Figure 4.4: Histograms of the linear regression r-values are shown for resolution 83 (left) and 150 (right).

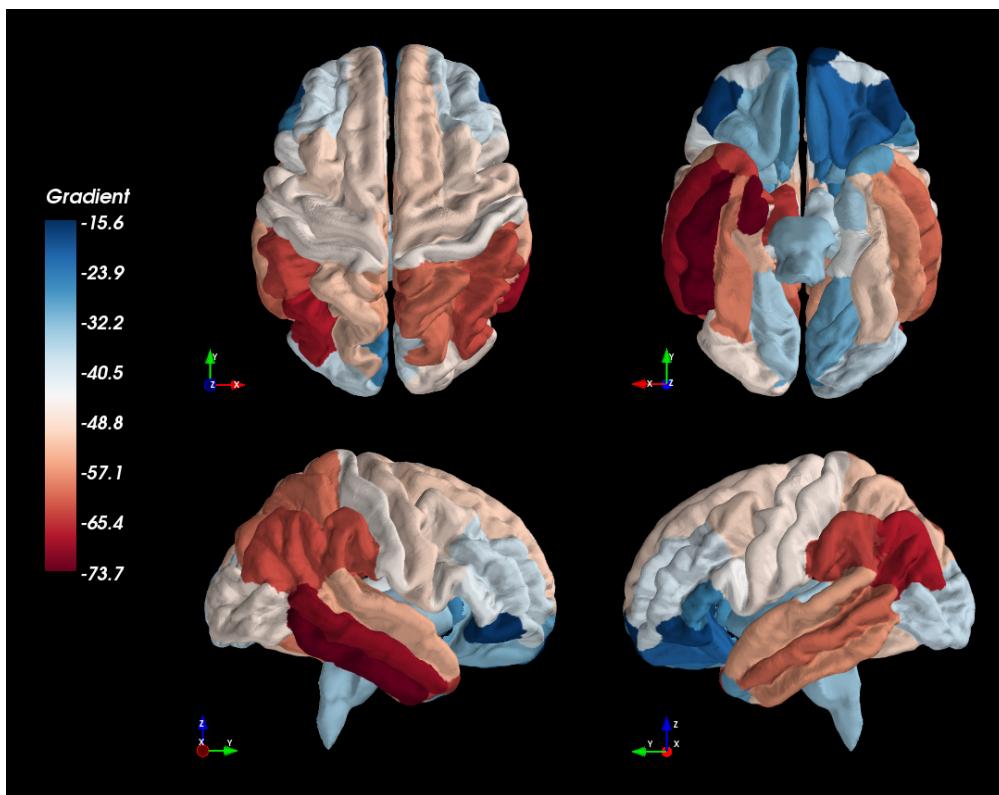


Figure 4.5: The average surface mesh colorcoded by the gradient values of the linear regression for ROIs in resolution 83. For $p < .05$, the absolute r-value must exceed 0.35.

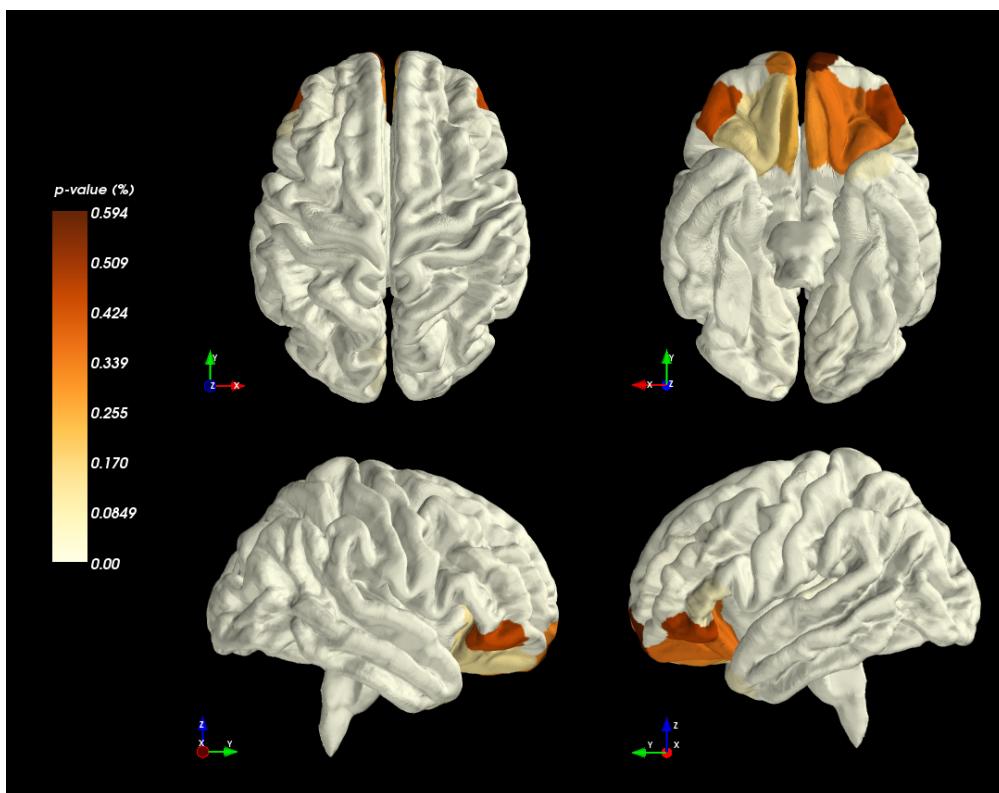


Figure 4.6: The average surface mesh colorcoded by the p-values reflecting the confidence of the slope in the linear regression analysis for ROIs in resolution 83.

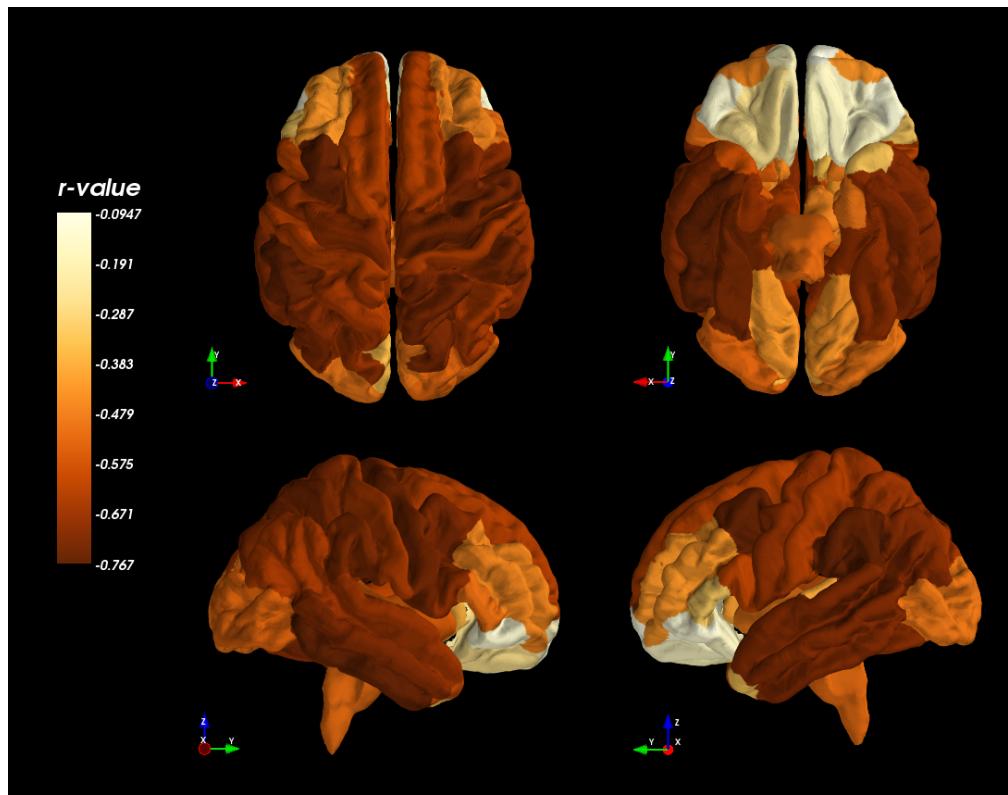


Figure 4.7: The average surface mesh colorcoded by the r-values for the linear regression analysis.

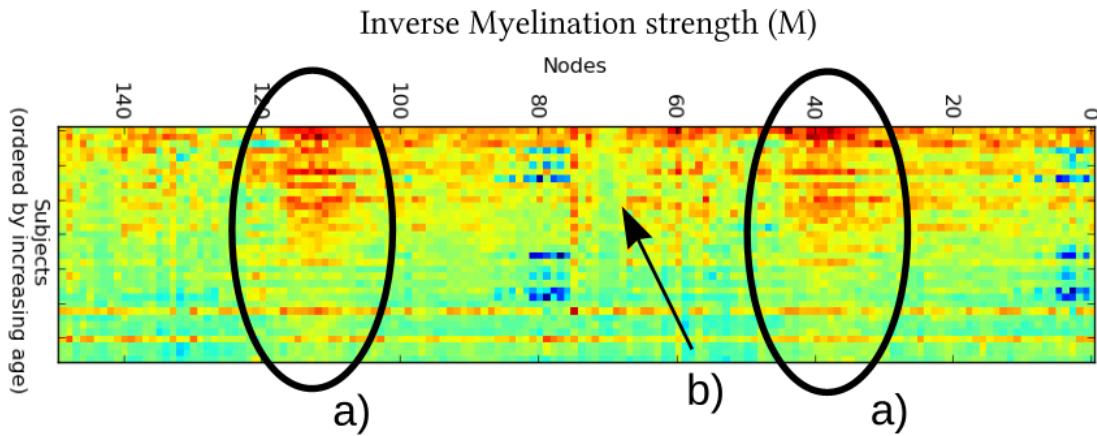


Figure 4.8: Two red cluster are salient in a). No change visible for nodes in b).

4.2.3 Discussion

Figure 4.1 suggests a reliable decrease in the Inverse Myelination Strength index defined here. Furthermore, the interpretation of this Figure suggests spatial differences for nodes. Namely, some nodes appear to have a relatively high Inverse Myelination Strength value for young subjects as compared to older subjects, whereas other nodes' value seems to show no age dependency (Figure 4.8).

Resolution 150 shows a similar result (Figure 4.1) as for resolution 83. A general decrease in the Inverse Myelination Strength is apparent with spatial differences. This could be confirmed up to resolution 500 (data not shown).

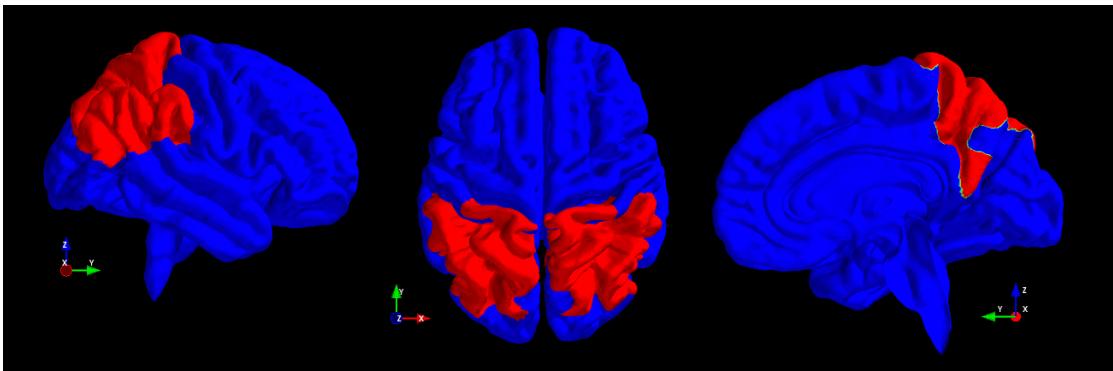


Figure 4.9: The nodes representing the red clusters a) in Figure 4.8 are highlighted with red (parietal region). Side, top and medial view.

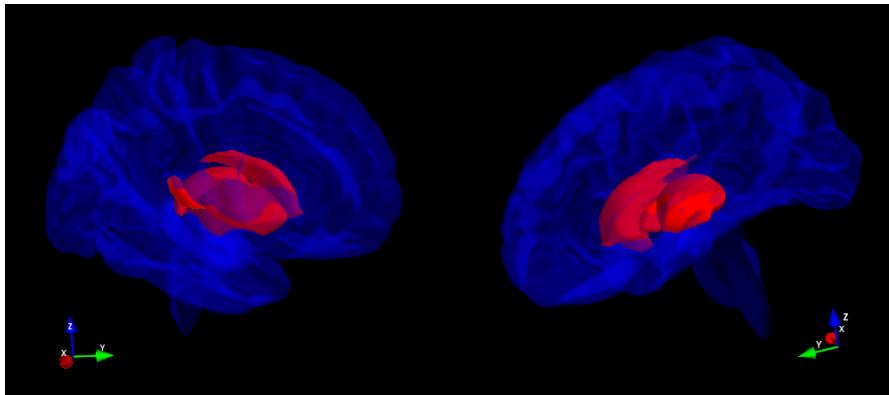


Figure 4.10: Subcortical nuclei corresponding to group b) in Figure 4.8 are shown in red embedded in a transparent cortical surface rendering.

The result of the linear regression analysis in Figure 4.2 for resolution 83 supports our expectation that differences in speed, i.e. slope of the linear regression curve, of myelination as assessed by the Inverse Myelination Strength per node can be observed.

Figure 4.3 displays the computed Inverse Myelination Strength values for the superior parietal node over all subjects with a linear regression fit. Very strong correlation $r = -0.72$ was found and is compatible with our hypothesis of this association cortex node.

In the same Figure, the exponential fit on the right hand side shows only a slightly better correlation ($r = -0.72$) with higher significance ($p = 6e - 7$). It seems that no improvement can be gained from using a exponential regression model, so further discussion is limited to the linear case. Confidence in the linear regression model is increased by showing that the mass of nodes show a significant correlation. This can be seen from the histogram plots in 4.4 for both resolution 83 and 150.

To evaluate the extent of the spatial non-uniformity, one could create lists with node labels that show a significant age-dependend correlations. A better solution is to colorcode the brain surface patches corresponding to nodes/ROIs with the gradient of the linear regressions. The ConnectomeViewer allows to do this and create renderings. Figure 4.5 shows this results for resolution 83.

The parietal cortical areas show the strongest age-dependent changes in myelination. This agrees well with the hypothesis because the parietal areas belong to the association cortices. Occipital areas display only slow change in myelination, as expected due to being primary visual areas, and compatible with the hypothesis.

However, the orbitofrontal cortex displays very weak age-dependent changes in myelination in disagreement with the hypothesis. Figure 4.6 shows the cortical surface colorcoded with the p-values stating the confidence into the estimated linear regression slope being correct. The orbitofrontal areas stand out exclusively with very high p-values. As seen in Figure 4.7, the

corresponding r-values for these areas are very weak. This leads to the conclusion that these areas show very big interindividual differences. No age-dependent effect can reliably be postulated for these areas.

For other primary areas, such as pre- and postcentral gyrus with primary motor and sensory areas, Figure 4.5 suggests also slow changes in myelination with high correlation (Figure 4.7) and high confidence (Figure 4.6). This finding is also consistent with the hypothesis.

No strong lateralization effect are discernible except for the inferior temporal gyrus. This might be attributable to interhemispheric fiber bundles (e.g. corpus callosum) that are usually well-reconstructed with the Diffusion MRI technique employed here. Thus, a strong density value of edges connecting corresponding interhemispheric ROIs might bias the relative weighting of the Inverse Myelination Strength and lead to the symmetric picture in Figure 4.5.

Critical issues regarding the interpretation of the results are, firstly, the fact that the data was collected from different individuals at different ages. Thus, it is not completely clear how much of the age-dependent differences could be attributed to interindividual differences. A longitudinal study measuring the same individuals might resolve this issue in the future. Secondly, the volume of the brain in very young subjects is significantly smaller than in young adults. Since the data acquisition uses the same voxel size for all subjects, there is a smaller number of voxels usable in young children, thus the tractography results have to be interpreted with special care.

4.2.4 Conclusion

The ConnectomeViewer was successfully employed to perform the analysis and visualization requirements of research questions based on Diffusion MRI data. A Connectome File which contains study data of 34 subjects was converted from the Connectome Mapping Pipeline. The application is scriptable using Python to carry out analysis on the datasets. Visualizations of cortical surfaces and plotting was done using the ConnectomeViewer.

4.3 sLORETA EEG Attention Study

Departing from a pure tool for structural neuroimaging, in this section the application of the ConnectomeViewer with its scripting capabilities are shown using an sLORETA processed dataset of functional EEG data.

To process sLORETA data, I developed a LORETA to Connectome File Format converter. Using this converter various LORETA data types (EEG functional connectivity, data sets containing statistical values: r, t, F) can be converted and therefore visualized or further analyzed in the ConnectomeViewer.

Study design, data acquisition and analysis were performed by Patricia Milz in the scope of her master thesis. The dataset was kindly provided by the KEY Institute for Brain-Mind Research.

Study Description 23 male, right-handed students between 19 and 27 years (mean=23.22; stdD=1.91), recruited at the ETH, took part in an EEG attention study. Data was aquired under three eyes-closed conditions (relaxing, breath counting, arithmetic operation) in pseudo-randomized order. Several questionnaires were used to assess behavioural variables such as basic personality traits, well-being, momentary and general anxiety levels and vividness of visual imagery. For details of the study design, see Milz (2010).

Data processing EEG data was processed according to established standards (artifact rejection, filtering, re-referencing) and prepared to be analyzed with the sLORETA software (REF). Using standard frequency bands (1.5 - 6: Delta; 6.5 - 8: Theta; 6.5 - 10 Alpha 1; 10.5 - 12 Alpha 2; 12.5 - 18 Beta 1; 18.5 - 21 Beta 2; 21.5 - 30 Beta 3; 35 - 44 Gamma), sLORETA intracerebral current densities were computed. 19 ROIs were defined as hemispheres below 19 electrodes, each defining a the set of intracerebral voxels. As a measure of functional connectivity between these ROIs, lagged linear connectivity was computed using sLORETA. For ease of visualization, the electrode positions were used as a representative of the ROIs in 3D space.

Additionally, correlation statistics were computed for behavioural variables and lagged linear connectivity averaged across repeated conditions. These statistics are represented again as networks for the correlation coefficients for each individual variable, thus edge values depict r-values for all analyzed frequency bands between the ROIs.

Aims

- Converter of sLORETA connectivity output and "statistical" networks into the Connectome File Format
- Visualize this data in the ConnectomeViewer
- Write a script to visualize a large number of networks, thresholded by appropriate values and projected to 2D, for better visual inspection and analysis

Acknowledgement Patricia Milz, KEY Institute of Brain-Mind Science

Reference Milz, P. (2010). Hirnelektrische Mechanismen der Aufmerksamkeit - Eine Kohärenzanalyse intrazerebraler Zeitreihen. Unveröffentlichte Lizentiatsarbeit, Universitaet Zürich, Psychologisches Institut, Klinische Psychologie, Psychotherapie und Psychoanalyse.

4.3.1 Visualization Protocol

Before starting with the visualization of the EEG data, a conversion from the sLORETA datasets to the Connectome File Format has to be made. After two sessions with Roberto Pascual-Marqui, the inventor and developer of sLORETA, it was straightforward to write a Python tool which makes this conversion. You see the GUI in Figure 4.11. Because sLORETA and its extension eLORETA, computes measures of functional brain "connectivity" based on coherence and phase synchronization between electrodes or intracerebral ROIs (low-resolution tomography)

[12], these data are represented as connectivity matrices and are thus apt to be converted to networks representations (section 2.2.4) as used in the Connectome File Format.

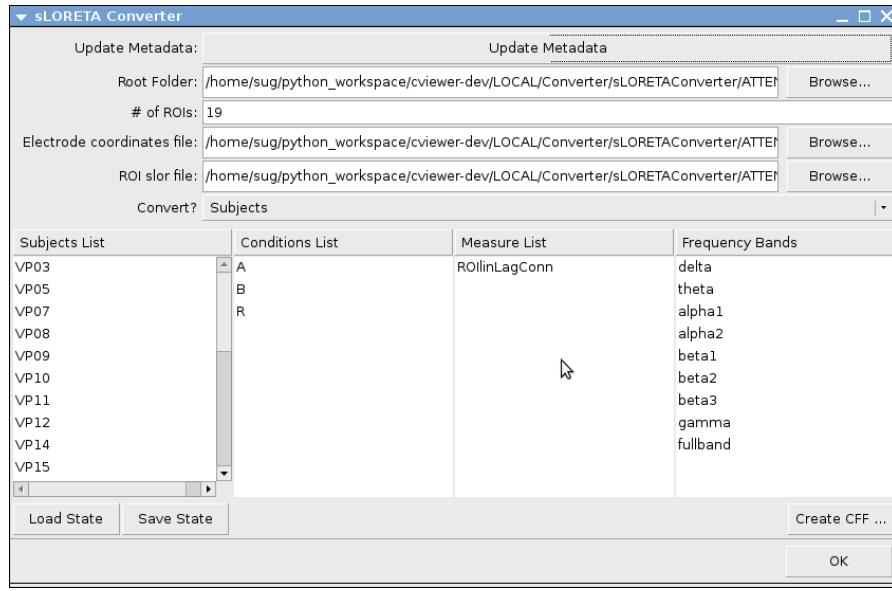


Figure 4.11: The GUI of the sLORETA Converter with mask to specify the data parameters and for entering metadata. State saving allows to retrieve already entered information.

For the dataset at hand, it was convenient to use a set of EEG electrodes location as representatives for the network node locations, even though the ROIs were defined as intracerebral hemisphere under the actual electrodes.

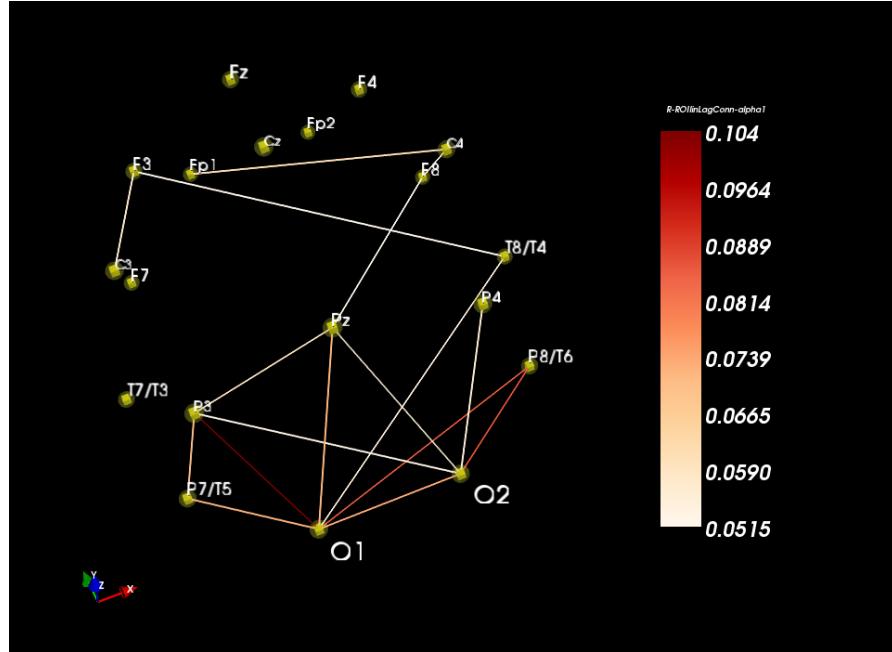


Figure 4.12: For a single subject, the lagged linear connectivity information is shown colorcode for the alpha1 band in the breathing condition.

Interesting enough, in Figure 4.12, one can readily see the increased lagged linear connectivity in the alpha1 frequency range in the occipital lobe. The increased occurrence of alpha rhythms during relaxed, wakefull, eyes-closed condition is a long-known fact. It is also called Berger rythm [10] after its discoverer.

Note here, that the lagged linear connectivity measure discards zero-phase coherences, thus the 'lagged' in its name. For details, see [12]. Speculating here, it could well be that this lagged linear connectivity is related to underlying structural connectivity, as reconstructed over the occipital lobe (Figure 2.7 in Section 2.4.7). Since lagged linear connectivity is related to "lagged" coherences, this lag or time delay might be related to signal propagation time across these fibers. Additionally, and as a critique of this speculation, it must be remarked that for other subjects, the overall picture of the alpha1 frequency network looks completely different.

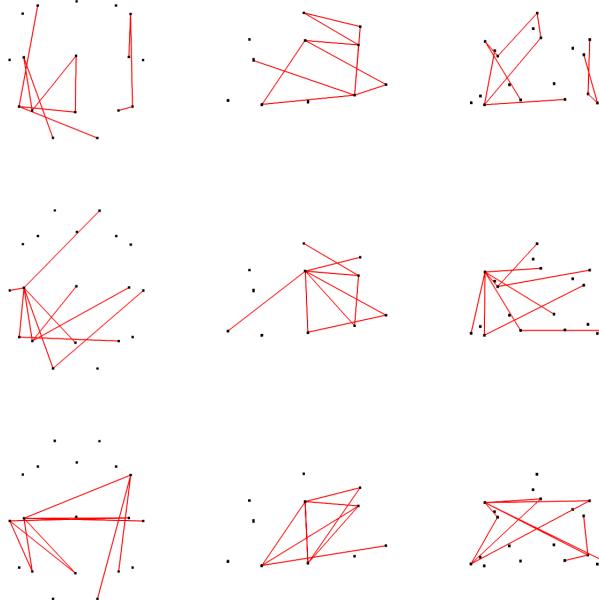


Figure 4.13: Section from the "statistical" networks produced by the visualization script. Red indicates positive correlation. Top to bottom row are conditions breathing, arithmetic and rest. The columns are top, left and back view. The networks depict the correlation of the linear lagged connectivity in the beta1 frequency range with the behavioural variable age. A statistically significant increase in beta1 linear lagged connectivity is thus observed, although not spatially consistent.

It is tedious to look at all these networks by hand. By visually inspecting thresholded networks, one might be able to see patterns that are otherwise hidden in the data, which can lead to new hypothesis. But inspecting a large number of networks by hand is tedious. For instance in the case of the "statistical" networks, there are 18, 17 and 15 behavioural variables that were correlated (in each condition). Each network depicts correlation coefficients for 8 frequency bands, and one might want to look at these networks from the three standardized directions (top, sagittal, coronal). This yields $15(\text{behav.vars}) * 3(\text{conditions}) * 8(\text{freq.bands}) * 3(\text{views}) = 1080$ pictures minimally. The devised visualization script automates this task and yields two large pictures with all the networks, thresholded by the statistically significant r-value and only showing the ten strongest (most negative or positive) correlations, in a systematic way. Figure 4.13 shows a section from this big picture.

4.3.2 Conclusion

It has been shown that the ConnectomeViewer is usable to visualize and analyze functional EEG data after conversion into the Connectome File Format from sLORETA. Even comparatively large amounts of data are readily visualized. Scripting allows to manipulate all visualization parameters and output automatically.

4.4 Average Human Connectivity

Description 20 healthy subjects aged between 22 and 33 years, males and females were scanned with Diffusion MRI (DSI 258 diffusion gradients over a hemisphere with $b_{max}=8000$ s/mm², Voxel size 2.2 x 2.2 x 3 mm³) and processed with the Connectome Mapping Pipeline. The output connectomes were then averaged across all subjects yielding average connectivity matrices at five resolutions. As an atlas for visualization, the 'Desikan-Killiany' Freesurfer cortical atlas [13] was used.

Aims

- Calculate basic network measures (node degree, betweenness centrality, clustering coefficient) on the average dataset.
- Visualize the results using the ConnectomeViewer.

Acknowledgement Patric Hagmann

Reference The dataset is online available ¹. Data processed according to Hagmann P, Cammoun L, Gigandet X, Meuli R, Honey CJ, et al. Mapping the Structural Core of Human Cerebral Cortex. *PLoS Biol* 6(7): e159, 2008.

4.4.1 Analysis and Visualization Protocol

The connectivity matrices for two resolutions (83 and 258 nodes) are shown colorcoded in Figure 4.14 and 4.16. The value of their elements depict the number of subjects that actually have the particular connection. The name *probability of connection* is used to denote this. The matrix was normalized by the number of subjects, yielding edge values from zero to one.

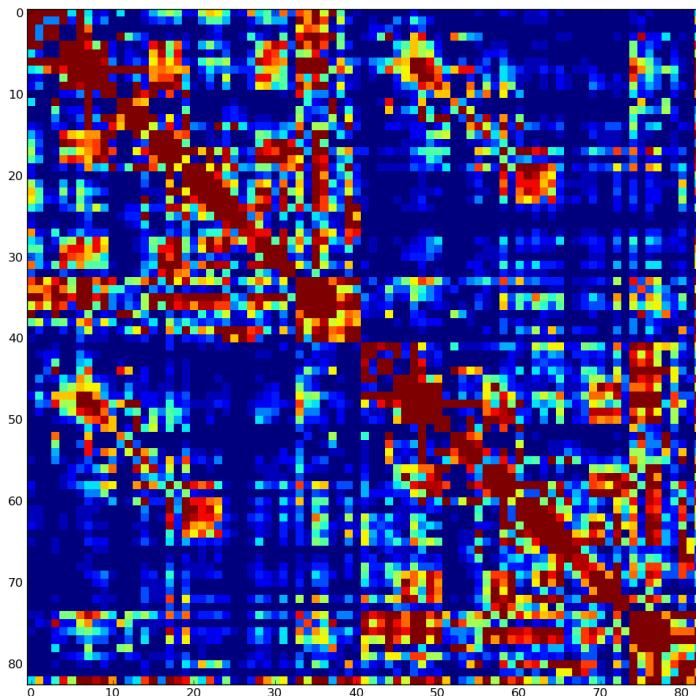


Figure 4.14: Probability of connection matrix (resolution 83) showing colorcode the probability of subjects having this connection (red: 1.0, blue: 0)

¹<http://www.connectomeviewer.org/>

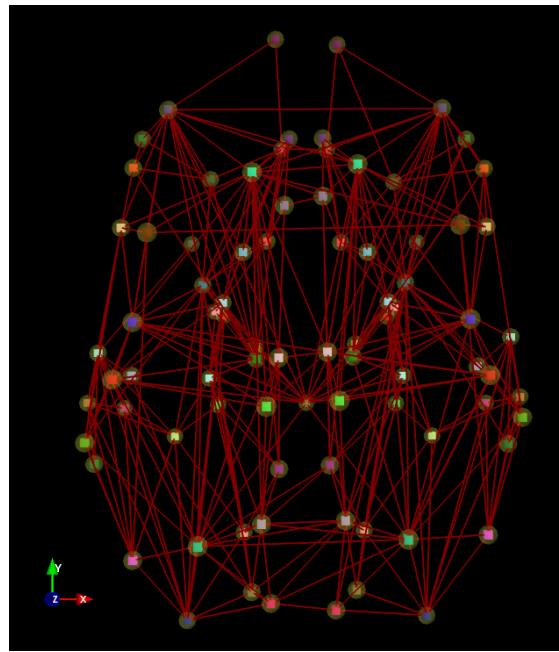


Figure 4.15: Resolution 83 network embedded in 3D is shown from top view. Nodes are positioned at the center of gravity of their corresponding ROI in the atlas mesh. Only edges with a probability of connection equals 1.0 (edge occurs in all subjects) are shown.

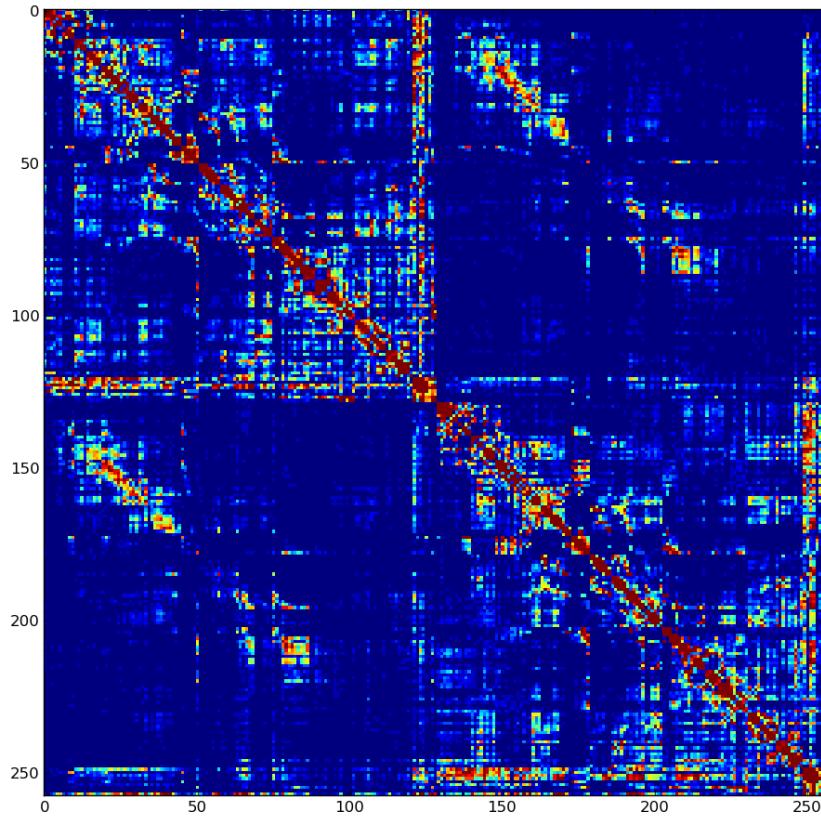


Figure 4.16: Probability of connection matrix shown for resolution 258.

A recent paper by Mikail Rubinov and Olaf Sporns [8] depicts a wide range of network measures that are applied to brain networks. A subset of these measures are computed here, namely node degree, centrality and clustering coefficient indices. Their computation is quickly

introduced.

Node degree (Figure 4.17):
 a_{ij} is the connection between i and j . N is the set of all nodes. Degree of node i is k_i .

$$k_i = \sum_{j \in N} a_{ij} \quad (4.3)$$

Betweenness centrality (Figure 4.18):
Betweenness centrality of a node is the fraction of all shortest paths that pass through that node [7]. One can also compute the measure for weighted networks using weighted shortest paths. ρ_{hj} is the number of shortest paths between h and j , and $\rho_{hj}^{(i)}$ is the number of shortest paths between h and j that pass through i .

$$b_i = \frac{1}{(n-1)(n-2)} \sum_{h,j \in N, h \neq j, h \neq i, j \neq i} \frac{\rho_{hj}^{(i)}}{\rho_{hj}} \quad (4.4)$$

Clustering coefficient (Figure 4.19):
 t_i is the number of triangles around a node i : $\frac{1}{2} \sum_{j,h \in N} a_{ij} a_{ih} a_{jh}$

$$C_i = \frac{2t_i}{k_i * (k_i - 1)} \quad (4.5)$$

The Python NetworkX library [7] was used to compute these measures. More advanced network measures are going to be implemented in the ConnectomeViewer in the near future, based on [8] and the Brain Connectivity Toolbox ².

Since the computed measures yield scalar values per node, it is convenient to code this information graphically into the size of individual nodes embedded in 3D. The ConnectomeViewer allows to do this easily. The following Figures display the results done on the dataset with the highest resolution of 1015 nodes.

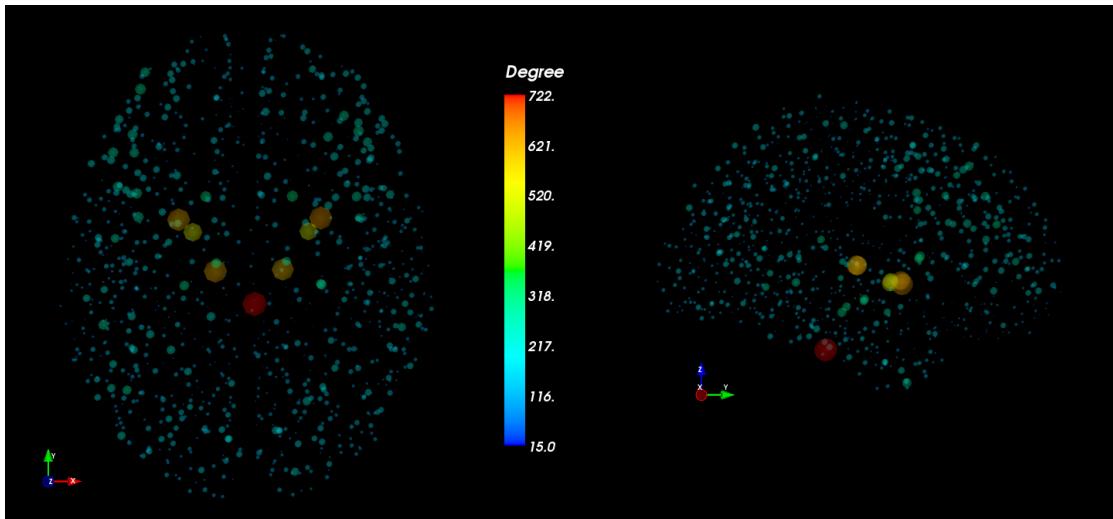


Figure 4.17: Top- and side view of nodes colorcoded and scaled by their node degree. The subcortical nuclei stand out as compared to the cortical nodes.

²<http://www.brain-connectivity-toolbox.net/>

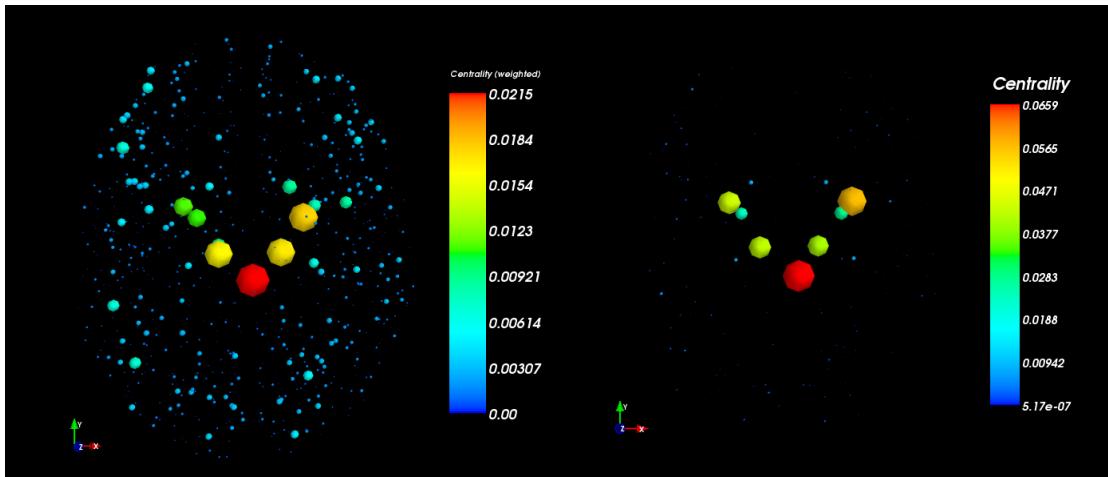


Figure 4.18: Left topview of centrality measures in the weighted case. Weights are the probability of connection. On the right side, the unweighted case. The subcortical nuclei stand out, especially the brain stem. The picture is similar as for the node degrees.

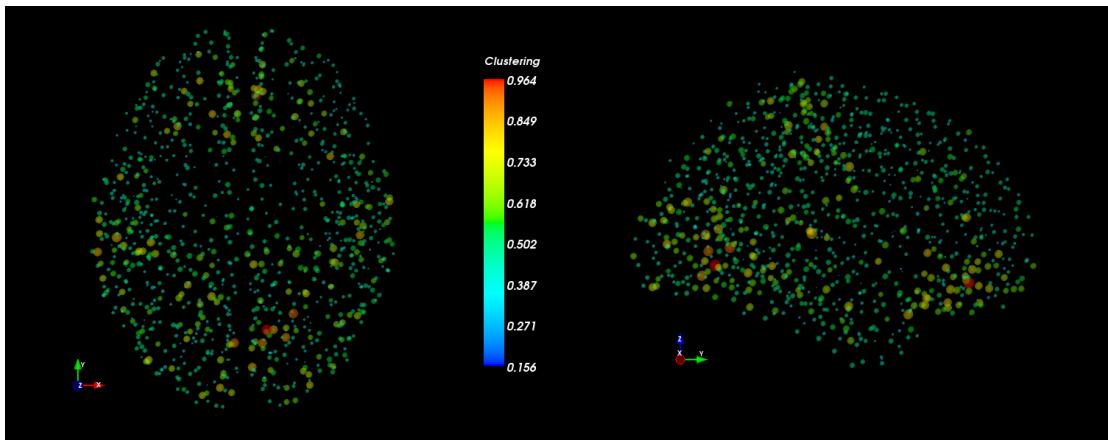


Figure 4.19: Top- and side view of transparent nodes depicting their clustering coefficient with a colorcode and scaling. No obvious pattern is visible, except for a minimally pronounced increase in anterior and posterior medial sites.

4.4.2 Conclusion

Basic network measures were computed on an average human connectome of 20 subjects. The usability of the ConnectomeViewer in computing and visualizing basic brain network analysis has been shown. More sophisticated brain network measures will be implemented in a future release.

4.5 Cat area 17 cortex

Departing from whole-brain connectivity analysis, this section zooms into the cellular levels of investigations of neural tissue. Neuronal reconstruction and quantitative analysis has a long tradition at the Institute of Neuroinformatics³, especially in the cat visual cortex. Through asiduous work, much is known about the cortical microcircuitry of this part of neocortex.

The first aim is simply to visualize the cortical microcircuit network. The data is derived from publication of Binzegger et al. as shown in Figure 4.20.

The second aim is to show a connectomic application on the cellular level. Three reconstructed neurons of cat visual cortex were used. Data by courtesy of Elisha Ruesch and Kevan Martin. It is shown that the Connectome File Format is general enough to represent data, and the ConnectomeViewer is apt for microscale investigations. This is due to the analogy of MRI data to optical imaging data:

- volumes: represent serial (histological, stained) sections from microscopy rather than MRI voxels
- tracks: axonal and dendritic trees are followed by hand rather than white matter tractography
- surfaces: depict the reconstructed neurite and soma of neurons (including diameter information) rather than cortical surface patches
- networks: nodes represent neurons or axonal/dendritic trees rather than ROIs. edges might represent synaptic connection strength or synapse count rather than

Aims

- Convert the estimated cortical microcircuit into Connectome File Format and visualize it.
- Generate a network from light-microscope reconstructed neurons based on a synapse proximity criterion for edges.

Acknowledgement Kevan Martin, Elisha Ruesch, Stefan Roth, German Koestinger

Reference Binzegger T, Douglas RJ, Martin KA. A Quantitative Map of the Circuit of Cat Primary Visual Cortex. *J Neurosci*. Sep 29;24(39):8441-53, 2004.

4.5.1 Analysis Protocol Aim 1

The relevant data and parameters to build a cortical microcircuit was extracted from Binzegger et al. [11] and is reproduced in Figure 4.20. The Python script in the Appendix C recreates the microcircuit data from Figure 4.20.

After the execution of the script, the microcircuit data is available in the Python programming environment. Another script (not shown) was used to write the networks in GraphML format (Section 2.2.4) and eventually produce the connectome file. The nodes were positioned schematically, adjusted to the space available for each layer. Inhibitory and excitatory nodes were separated. They represent all existing inhibitory or excitatory neurons in a layer respectively. Figure 4.21 shows a thresholded rendering using the ConnectomeViewer and labeled nodes. Only three strongest edges in terms of probability of connection are shown.

³<http://www.ini.uzh.ch/>

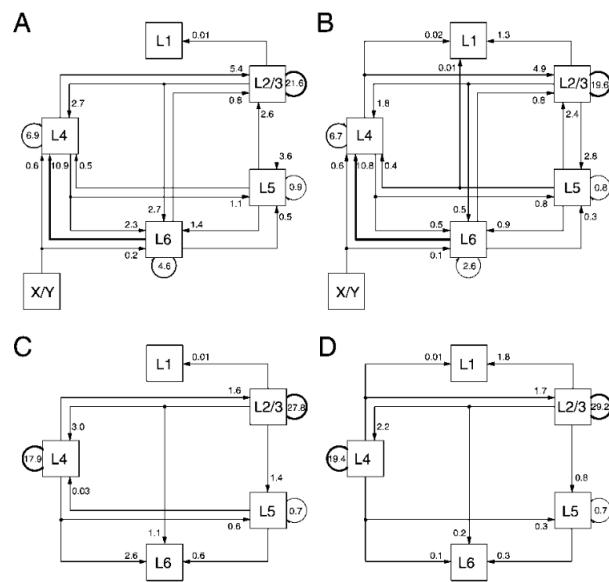


Figure 4.20: Percent of number of synapses involved in the projections between excitatory and inhibitory neurons between layers. A) between excitatory B) from excitatory onto inhibitory, C) from inhibitory onto excitatory, D) between inhibitory neurones.

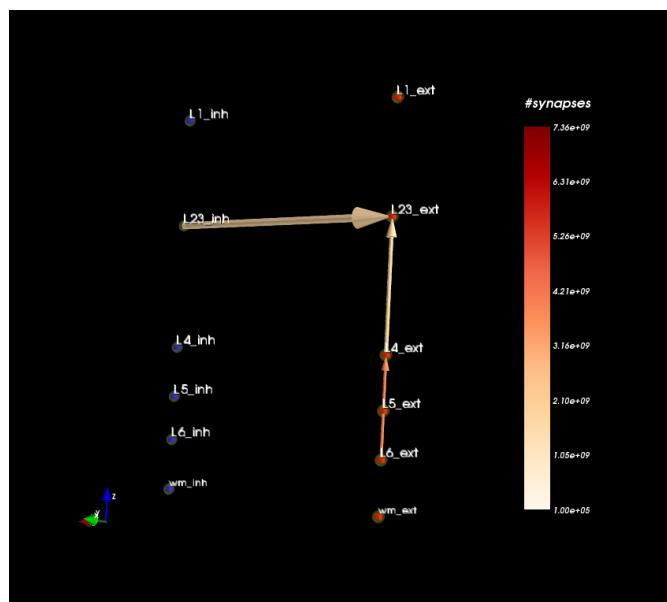


Figure 4.21: This rendering shows the thresholded microcircuit as described in the text. Several drawbacks of this directed graph layout are visible, e.g. the arrow from L6_ext to L4_ext goes through L5_ext (so-called edge crossing), self-loops are not shown, and bidirectional arrows occlude themselves. More issues with regard to directed graphs have to be addressed in the future.

4.5.2 Analysis Protocol Aim 2

The Institute of Neuroinformatics provided me with three reconstructed layer 2/3 pyramidal cell (cell body are in layer 2/3 with axons in layer 2/3 and layer 5, and dendrites confined to layer 2/3) in Neurolucida format (data by courtesy of Kevan Martin). Custom written Python scripts were used to parse the format, create the connectome file and perform the data analysis. Surface meshes of the neurons were constructed using the diameter information. Figure 4.22 shows the three loaded reconstructed neurons.

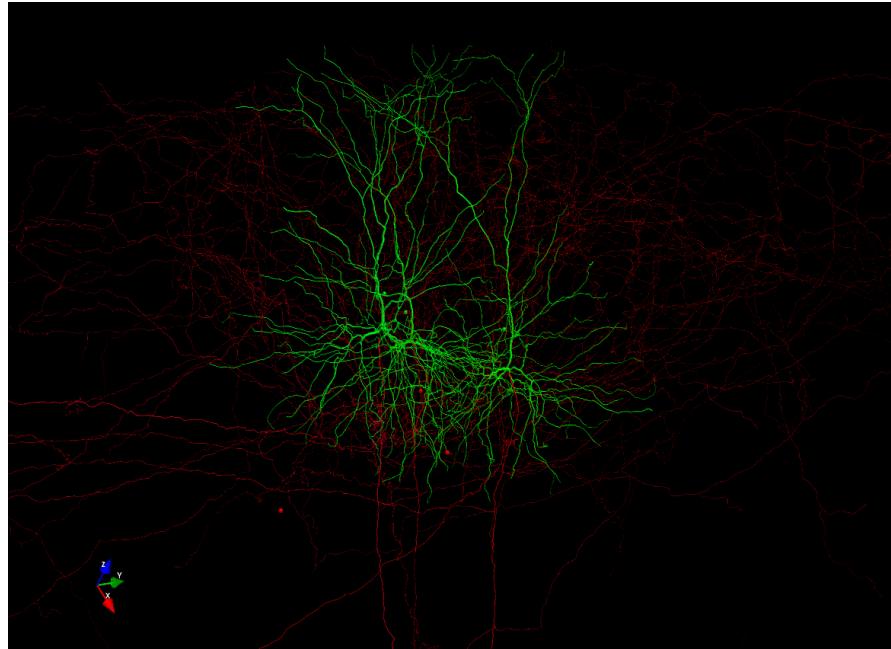


Figure 4.22: An intricate net even for only three reconstructed cortical neurons. (Axons: red, Dendrite: green)

Then, a script was set in place to find places where the distance between axonal branches and dendrites was closest. These places serve as a putative place for potential synapses (See Figure 4.25). Similar analysis were performed by [9]. In particular, their paper demonstrates that individual synapse formation between neurons is not independent of the other synapses between the same neurons. This is why a synapse count between neurons can be seen as a very crude weight approximation, as employed to estimate the networks.

From this estimated networks, it is now possible to construct a network with nodes representing individual neurons and edges representing the synapse count between individual neurons, and generate a Connectome File thereof. Of course, it could be thought of having a different construction of networks, e.g. by representing parts of a dendrite as a nodes and so forth, which would yield to networks readily usable for biologically more realistic simulations. But we are not concerned here with simulation. I have chosen for nodes to represent either the axonal part or the dendritic parts of a neuron.

The results are depicted graphically in Figure 4.23 and in Table 4.1. The script to parse the Neurolucida reconstructions and estimate the synapses is approximately 150 lines of code and is available upon request.

4.5.3 Conclusion

The ConnectomeViewer is usable to perform data visualization and analysis on single cell and cellular network level. This was shown using data of the putative neocortical microcircuit and reconstructed single cells from cat visual cortex.

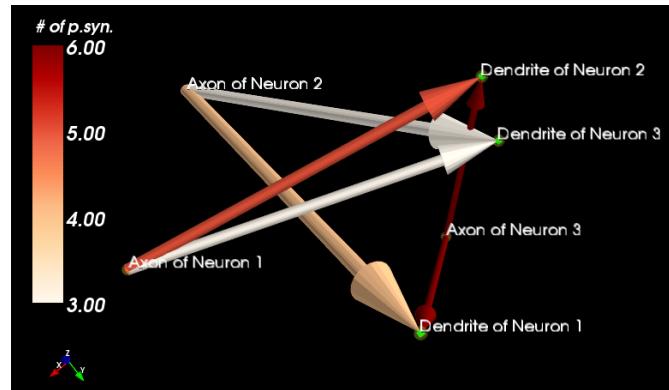


Figure 4.23: The estimated directed network for these three neurons based on potential synaptic count with a proximity criterium. The node position is the center of gravity of the corresponding surface meshes.

	Neuron 1	Neuron 2	Neuron 3
Neuron 1		4	3
Neuron 2	5		3
Neuron 3	6	6	

Table 4.1: Number of estimated synapse based on a proximity criterion

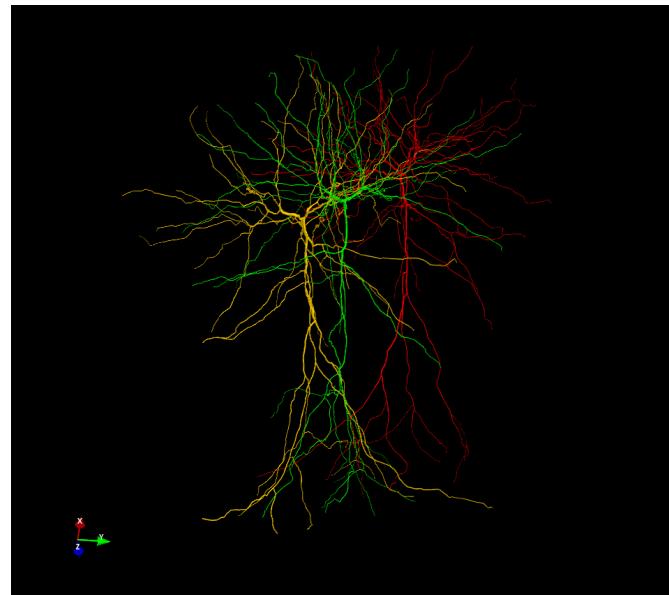


Figure 4.24: It is possible to select only a subset of surfaces to display, such as in this case, only the dendritic arbors.

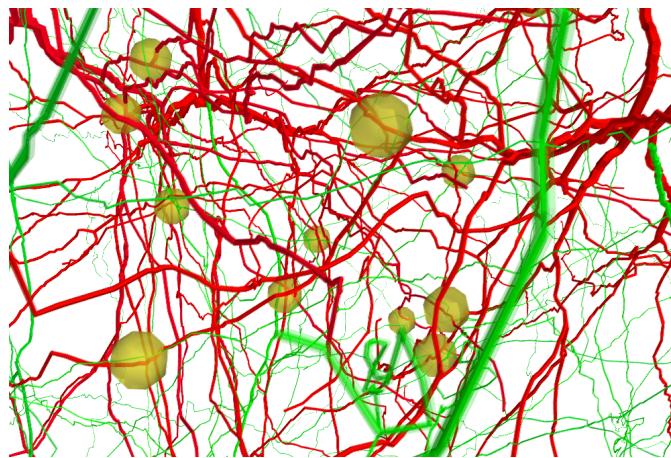


Figure 4.25: The yellow spheres are located where potential synapses are expected, based on the proximity of an axonal arbor to a dendritic arbor.

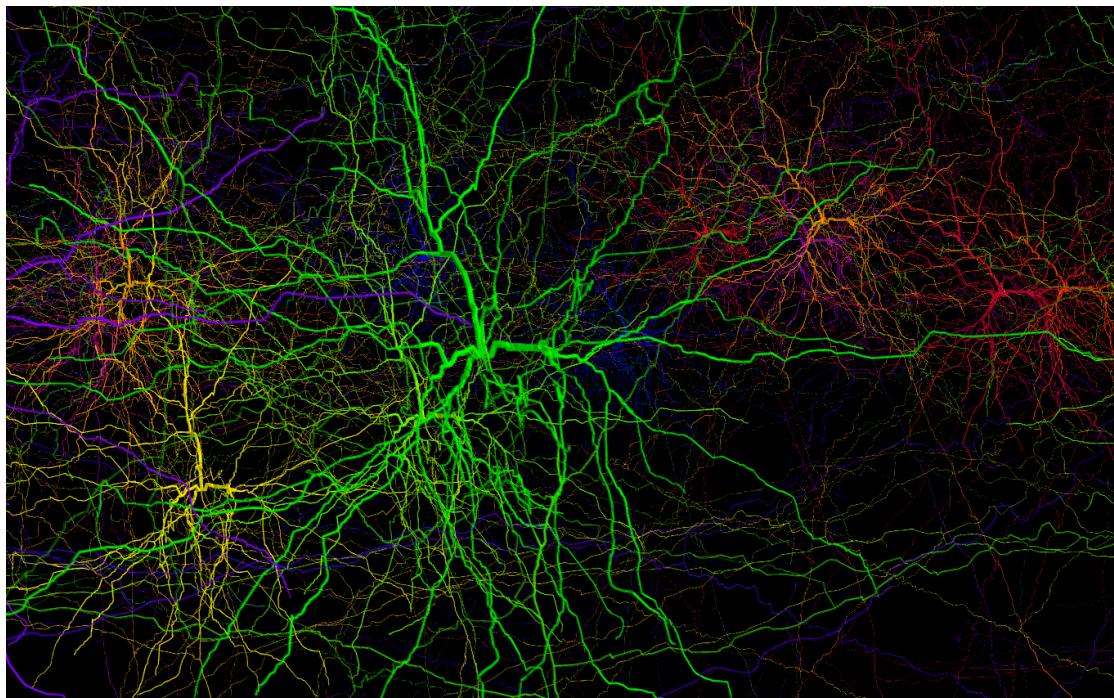


Figure 4.26: One reconstructed neuron surface was duplicated and placed in the same layer 20 times with random distribution. How many virtual neurons one can display at the same time depends mainly on working memory and level of detail of the surface reconstructions.

Bibliography

- [1] Barnea-Goraly, N., Menon, V., Eckert, M., Tamm, L., Bammer, R., Karmeskiy, A., Dant, C. and Reiss, A.L. White Matter Development During Childhood and Adolescence: A Cross-Sectional Diffusion Tensor MR Imaging Study. *Cereb Cortex*. 15(12):1848-54, 2005.
- [2] Johansen-Berg H, Behrens T. Diffusion MRI: From Quantitative Measurement to In-vivo Neuroanatomy. *Academic Press*, 2009.
- [3] Tamnes C, Østby Y, Fjell A, Westlye L, Due-Tønnessen P, Walhovd K. Brain Maturation in Adolescence and Young Adulthood: Regional Age-Related Changes in Cortical Thickness and White Matter Volume and Microstructure. *Cerebral Cortex*, June 11, 2009.
- [4] Volpe J. Neurology of the newborn. *Elsevier Health Sciences*, 2008.
- [5] The Biological Basis of Diffusion Anisotropy In Johansen-Berg H, Behrens T. Diffusion MRI: From Quantitative Measurement to In-vivo Neuroanatomy. *Academic Press*, 2009.
- [6] Hagmann P, Cammoun L, Gigandet X, Meuli R, Honey CJ, et al. Mapping the Structural Core of Human Cerebral Cortex. *PLoS Biol* 6(7): e159, 2008.
- [7] Hagberg A, Schult D, and Swart P. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Aug 2008.
- [8] Rubinov M, Sporns O. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage*, in press, 2009.
- [9] Fares T, Stepanyants A. Cooperative synapse formation in the neocortex. *PNAS* Vol. 106 No. 38 16463-16468, 2009.
- [10] The Berger Rhythm. *The British Medical Journal*, Vol. 1, No. 3920 p. 372, 1936.
- [11] Binzegger T, Douglas RJ, Martin KA. A Quantitative Map of the Circuit of Cat Primary Visual Cortex. *J Neurosci*. Sep 29;24(39):8441-53, 2004.
- [12] Pascual-Marqui R. Instantaneous and lagged measurements of linear and nonlinear dependence between groups of multivariate time series: frequency decomposition. 2007. Retrieved December 21, 2009, from <http://arxiv.org/abs/0711.1455>
- [13] Desikan et al. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, 31(3):968-80, 2006.

Appendix A

Homepage and Source Code

The ConnectomeViewer Homepage (Figure A.1) serves as starting point to user registration, documentation and the developer zone ¹.

The screenshot shows the ConnectomeViewer homepage. At the top, there is a large image of a brain with a network of connections. Below it, the title "ConnectomeViewer" is displayed in a large, bold font, followed by the subtitle "Multi-Modal Multi-Level Network Visualization & Analysis". A navigation bar at the bottom includes links for "Home", "Documentation", "index", and "Support". The main content area contains several sections: "The ConnectomeViewer is a software application for the visualization and analysis of structural neuroimaging data.", a list of features (including usage of Connectome File Format, support for NetworkX, Nifti, TrackVis, Mayavi, Python, Envisage, and Brain Connectivity Toolbox plugin), "Preview and Screenshots" (with a link to "Getting a first impression of the application."), "User Registration" (with a link to "Help us in getting to know you. After registration, you are redirected to the download page."), "Login and Download Page" (with a link to "Login first before getting to the download page of the ConnectomeViewer BETA"), "Documentation" (with a link to "Here you will find primarily the installation instructions at the moment."), and "Developer Zone" (with a link to "Source code hosting, bug tracking, feature requests and more."). On the right side, there are two boxes: "Support" (with a link to "Requests for help should be directed to us per email.") and "License" (with a link to "The ConnectomeViewer is released under the GPLv3. See the license page for details."). A "Quick search" input field is also present.

Figure A.1: ConnectomeViewer Homepage <http://www.connectomeviewer.org/>

The complete source code is available under a GPLv3 license ². I omit inclusion of any code to save the forest. For the ConnectomeViewer core functionality, approximately 4500 lines of code including comments were written.

¹<https://launchpad.net/connectomeviewer>

²<http://gplv3.fsf.org/>

Appendix B

Developmental Project Script

The relevant calculations to compute the linear regression models of the developmental data are listed here.

Initialization

```
1 # import plotting library matplotlib
2 from pylab import *
3 # import numerical library
4 from numpy import *
5 # importing statistics module from scientific python
6 from scipy import stats

8 # the connectome file with the subject data has to be loaded
9 # initialize variables
10 # the number of subjects is equal to the number of networks
nr_networks = len(cfile.networks)
12 # extract the number of nodes for the resolution given in the connectome file
nr_nodes = len(cfile.networks[0].graph.nodes())
14 # prepare a 3D matrix to store ADC mean values for all edges for all subjects
cmatrix_adc_unsorted = zeros((nr_nodes,nr_nodes, nr_networks))
16 cmatrix_adc = zeros((nr_nodes,nr_nodes, nr_networks))

18 # similarly, prepare a 3D matrix for density values
cmatrix_density_unsorted = zeros((nr_nodes,nr_nodes, nr_networks))
20 cmatrix_density = zeros((nr_nodes,nr_nodes, nr_networks))

22 # matrix to store the computed values for the inverse myelination index
M = zeros( (nr_nodes, nr_networks) )
24

26 # data structures to store statistics
regress = zeros( (nr_nodes, 5) )
regress_log = zeros( (nr_nodes, 5) )
28 # storage of fitted curves
regress_plot = zeros( (nr_nodes, x_steps.shape[0]) )
exp_plog = zeros( (nr_nodes, x_steps.shape[0]) )
30 x_steps = linspace(1,18,50)
```

Data extraction

```
1 for i, network in enumerate(cfile.networks):
2     age_unsorted[i] = float(network.metadata['age'])
3     # get the subjects connectivit matrices
4     cmatrix_adc_unsorted[:, :, i] = network.get_matrix(weight_key = 'de_adc_mean')
      cmatrix_density_unsorted[:, :, i] = network.get_matrix(weight_key = 'de_density')
```

Reorder data by age

```
1 # order matrix by age
2 ind = age_unsorted.argsort()
3 age = age_unsorted[ind]
4 cmatrix_adc[:, :, :] = cmatrix_adc_unsorted[:, :, ind]
      cmatrix_density[:, :, :] = cmatrix_density_unsorted[:, :, ind]
```

```

8 | # discarding self-loops
9 | index = arange(min(nr_nodes, nr_nodes))
10| cmatrix_adc[index, index, :] = 0
11| cmatrix_density[index, index, :] = 0

```

Calculate Inverse myelination index

```

# define the M function as in formula 4.1 calculate node values
2 def calc_node_M(adc_vect, density_vect):
    """ Applies the myelin strength formula for a single node """
4     N = adc_vect.shape[0]
5     density_sum = sum(density_vect)
6     retVal = 0.0
7     for i in range(N):
8         retVal = retVal + adc_vect[i] * density_vect[i] / density_sum
9     return retVal / N
10
11 for network_idx in range(nr_networks):
12     for node_idx in range(nr_nodes):
13         M[node_idx, network_idx] = calc_node_M( cmatrix_adc[node_idx, :, network_idx], \
14                                         cmatrix_density[node_idx, :, network_idx])

```

Regression analysis

```

1 # calculate linear regression curves
2 # for every node and store result
3
4 for i in range(nr_nodes):
5
6     # compute the linear regression using the statistics library
7     # output columns are: gradient, intercept, r_value, p_value, std_err
8     regress[i, :] = stats.linregress(age, M[i, :])
9     # compute the linear regression for the exponential fit
10    regress_log[i, :] = stats.linregress(age, log(M[i, :]))
11
12    # sample the regression curves for plotting and later thresholding
13    regress_plot[i, :] = regress[i, 0] * x_steps + regress[i, 1]
14
15    # and sample the regression curve
16    exp_plog[i, :] = e**regress_log[i, 1] * (e**(regress_log[i, 0]*x_steps))

```

Appendix C

Cat area 17 cortical microcircuit

This script recreates the microcircuit data from Binzegger et al. (2004) and makes it available for further processing.

Initialization

```
1 # Purpose: Estimated cortical network of cat area 17.
# November 2009, Stephan Gerhard
3 # Comment: not using symmetric unassigned synapses data
5 # Data
# ----
7 # A Quantitative Map of the Circuit of Cat Primary Visual Cortex
# http://www.jneurosci.org/cgi/content/abstract/24/39/8441
9
11 import numpy as np
import networkx as nx
```

Layer dimension / Number of neurons

```
1 # a cortical "column" (box)
# dimension: 1 [mm] x 1 [mm] x 1.675 [mm]
3 # values from: Neocortex by R. Douglas, H. Markram, K. Martin
# Fig 12.2 B (macaque monkey Area 18)
5 L1_h = 125E-6
L23_h = 750E-6
7 L4 = 250E-6
L5 = 150E-6
9 L6 = 200E-6
wm = 200E-6
11
13 # Number of neurons in each layer (Beaulieu and Colonnier 1983)
14 # for 399 [mm^2] (Anderson et al. 1988)
scale = 399
15
17 L1_c_tot = 0.50E6 / scale # (of which 97% are GABAergic) for layer 1
L1_c_ext = (0.03 * L1_c_tot)
L1_c_inh = (0.97 * L1_c_tot)
19 L23_c_tot = 10.58E6 / scale # (22% GABAergic) for layer 2/3
L23_c_ext = (0.78 * L23_c_tot)
L23_c_inh = (0.22 * L23_c_tot)
21 L4_c_tot = 10.93E6 / scale # (20% GABAergic) for layer 4
L4_c_ext = (0.8 * L4_c_tot)
L4_c_inh = (0.2 * L4_c_tot)
23 L5_c_tot = 2.36E6 / scale # (18% GABAergic) for layer 5
L5_c_ext = (0.82 * L5_c_tot)
L5_c_inh = (0.18 * L5_c_tot)
25 L6_c_tot = 6.92E6 / scale # (17% GABAergic) for layer 6
L6_c_ext = (0.83 * L6_c_tot)
L6_c_inh = (0.17 * L6_c_tot)
27 neuro_tot_sheet = L1_c_tot + L23_c_tot + L4_c_tot + L5_c_tot + L6_c_tot
31
```

Probabilistic cell type connectivity

```

1 # Probabilistic network of cell type connectivity
2 # Figure 12 (The numbers were calculated based on Figure 7)
3
4 # Case A: between excitatory neurons
5 A_tot_syn = 13.6E10 / scale
6 # The proportion of asymmetric unassigned synapses that the excitatory neurons in each layer receive is
7 # 0.1% (layer 1), 6% (layer 2/3), 10% (layer 4), 2% (layer 5), and 12% (layer 6).
8 # These synapses are presumably formed by the afferents originating outside area 17.
9
10 # Mapping to labels
11 mapid2name = {0: 'L1', 1: 'L23', 2: 'L4', 3: 'L5', 4: 'L6', 5: 'wm'}
12 mapname2id = {'L1': 0, 'L23': 1, 'L4': 2, 'L5': 3, 'L6': 4, 'wm': 5}
13
14 # probabilistic connectivity matrix in order: L1, L23, L4, L5, L6, XY
15 A = np.array([
16     [0, 0, 0, 0, 0, 0],
17     [0.01, 21.6, 2.7, 3.6, 2.7, 0],
18     [0, 5.4, 6.9, 1.1, 2.3, 0],
19     [0, 2.6, 0.5, 0.9, 1.4, 0],
20     [0, 0.8, 10.9, 0.5, 4.6, 0],
21     [0, 0, 0.6, 0, 0.2, 0]])
22 A_graph = nx.relabel_nodes(nx.from_numpy_matrix(A.T), mapid2name)
23
24 # Case B: from excitatory onto inhibitory neurons
25 B_tot_syn = 2.1E10 / scale
26 # The proportion of asymmetric unassigned synapses that the inhibitory neurons in each layer receive is
27 # 17% (layer 1), 5% (layer 2/3), 9% (layer 4), 0.5% (layer 5), and 11% (layer 6).
28 B = np.array([
29     [0, 0, 0, 0, 0, 0],
30     [1.3, 19.6, 1.8, 2.8, 0.5, 0],
31     [0.02, 4.9, 6.7, 0.8, 0.5, 0],
32     [0.01, 2.4, 0.4, 0.8, 0.9, 0],
33     [0, 0.8, 10.8, 0.3, 2.6, 0],
34     [0, 0, 0.6, 0, 0.1, 0]])
35 B_graph = nx.relabel_nodes(nx.from_numpy_matrix(B.T), mapid2name)
36
37 # Case C: from inhibitory onto excitatory neurons
38 C_tot_syn = 2.4E10 / scale
39 # The proportion of symmetric unassigned synapses that the excitatory neurons in each layer receive is
40 # 0.1% (layer 1), 6% (layer 2/3), 12% (layer 4), 6% (layer 5), and 19% (layer 6).
41 C = np.array([
42     [0, 0, 0, 0, 0, 0],
43     [0.01, 27.8, 3.0, 1.4, 1.1, 0],
44     [0, 1.6, 17.6, 0.6, 2.6, 0],
45     [0, 0, 0.03, 0.7, 0.6, 0],
46     [0, 0, 0, 0, 0, 0],
47     [0, 0, 0, 0, 0, 0]])
48 C_graph = nx.relabel_nodes(nx.from_numpy_matrix(C.T), mapid2name)
49
50 # Case D: between inhibitory neurons
51 D_tot_syn = 0.4E10 / scale
52 # The proportion of symmetric unassigned synapses that the inhibitory neurons in each layer receive is
53 # 11% (layer 1), 5% (layer 2/3), 10% (layer 4), 4% (layer 5), and 15% (layer 6).
54 D = np.array([
55     [0, 0, 0, 0, 0, 0],
56     [1.8, 29.2, 2.2, 0.8, 0.2, 0],
57     [0.01, 1.7, 19.4, 0.3, 0.1, 0],
58     [0, 0, 0, 0.7, 0.3, 0],
59     [0, 0, 0, 0, 0, 0],
60     [0, 0, 0, 0, 0, 0]])
61 D_graph = nx.relabel_nodes(nx.from_numpy_matrix(D.T), mapid2name)

```

Summary statistics

```

1 # total amount of synapses in the whole cortical sheet
2 syn_tot_sheet = A_tot_syn + B_tot_syn + C_tot_syn + D_tot_syn
3 print '# Total number of synapses', syn_tot_sheet
4 print '# Excitatory cells:', L1_c_ext, L23_c_ext, L4_c_ext, L5_c_ext, L6_c_ext
5 print '# Inhibitory cells:', L1_c_inh, L23_c_inh, L4_c_inh, L5_c_inh, L6_c_inh
6 print 'Total number of cells', neuro_tot_sheet

```

Appendix D

Connectome File XML Schema

Listing D.1: The XML Schema Definition for the Connectome File

```
<xss: schema xmlns:xss="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.connectome.ch/2009/Connectome/xmlns" xmlns="http://www.connectome.ch/2009/Connectome/xmlns" xmlns:graphml="http://www.connectome.ch/Network" xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified">

    <xss:annotation xml:lang="en">
        <xss:documentation>
            Stephan Gerhard, Version 1.0
        </xss:documentation>
    </xss:annotation>

    <xss:complexType name="labelstype">
        <xss:attribute name="labelid" type="xs:string">
            <xss:annotation xml:lang="en">
                <xss:documentation>
                    The labelid uniquely defining the set of vertices labels in the given Gifiti file.
                </xss:documentation>
            </xss:annotation>
        </xss:attribute>

        <xss:attribute name="src" type="xs:string">
            <xss:annotation xml:lang="en">
                <xss:documentation>
                    The source of the surface (in the zipped CFF file), including the relative pathname to the Gifiti/ folder
                </xss:documentation>
            </xss:annotation>
        </xss:attribute>

        <xss:attribute name="fileformat" type="xs:string">
            <xss:annotation xml:lang="en">
                <xss:documentation>
                    So far, only Gifiti formats (ending .gii) are supported. Write here "gifiti"
                </xss:documentation>
            </xss:annotation>
        </xss:attribute>
    </xss:complexType>

    <xss:complexType name="surfacetype">
        <xss:attribute name="name" type="xs:string">
            <xss:annotation xml:lang="en">
                <xss:documentation>
                    Descriptive name of the surface container, seen as a node in the Connectome Viewer TreeView
                </xss:documentation>
            </xss:annotation>
        </xss:attribute>

        <xss:attribute name="src" type="xs:string">
            <xss:annotation xml:lang="en">
                <xss:documentation>
                    The source of the surface (in the zipped CFF file), including the relative pathname to the Gifiti/ folder
                </xss:documentation>
            </xss:annotation>
        </xss:attribute>

        <xss:attribute name="fileformat" type="xs:string">
            <xss:annotation xml:lang="en">
                <xss:documentation>
                    So far, only Gifiti formats (ending .gii) are supported. Write here "gifiti"
                </xss:documentation>
            </xss:annotation>
        </xss:attribute>
    </xss:complexType>
</xss: schema>
```

```

59   </xs:attribute>
60
61   <xs:attribute name="addatlas" type="xs:string">
62     <xsd:annotation xml:lang="en">
63       <xsd:documentation>
64         This indicates, that an additional set of surfaces from this atlas is added
65         to each network, e.g. for standardized visualization. ConnectomeViewer has
66         the following atlases for species included:
67         Homo sapiens:
68           - template_atlas_homo_sapiens_01
69             Extracted atlas from Freesurfer using fsaverage mesh
70           - template_atlas_homo_sapiens_02 (not yet included)
71             Possibly the SPL_PNL_Brain_Atlas2008
72           </xsd:documentation>
73         </xsd:annotation>
74       </xs:attribute>
75
76   <xs:all>
77     <xs:element name="surface-label" type="labelstype" minOccurs="0">
78       <xsd:annotation xml:lang="en">
79         <xsd:documentation>
80           Reference the to Gifti file containing the labels. In this Gifti file, the labelid for a ←
81             particular NIFTI_INTENT_LABEL
82             should be the same as the labelid as attribute of this element for the ConnectomeViewer to ←
83               parse correctly
84             </xsd:documentation>
85           </xsd:annotation>
86         </xs:element>
87       </xs:all>
88     </xs:complexType>
89
90   <xs:complexType name="volumetype">
91     <xs:attribute name="name" type="xs:string">
92       <xsd:annotation xml:lang="en">
93         <xsd:documentation>
94           Name of the volume
95         </xsd:documentation>
96       </xsd:annotation>
97     </xs:attribute>
98
99     <xs:attribute name="src" type="xs:string">
100       <xsd:annotation xml:lang="en">
101         <xsd:documentation>
102           Including the relative pathname to the Nifti/ folder
103         </xsd:documentation>
104       </xsd:annotation>
105     </xs:attribute>
106
107     <xs:attribute name="fileformat" type="xs:string">
108       <xsd:annotation xml:lang="en">
109         <xsd:documentation>
110           Roughly with what we have it to do, conforms to filename ending.
111         </xsd:documentation>
112       </xsd:annotation>
113     </xs:attribute>
114
115     <xs:attribute name="segmentation" type="xs:boolean">
116       <xsd:annotation xml:lang="en">
117         <xsd:documentation>
118           If set to true, this means that the volume contains a segmentation labels, potentially ←
119             used in conjunction with trackvis.
120             Otherwise, this could be any other volume information, like raw MRI data.
121           </xsd:documentation>
122         </xsd:annotation>
123       </xs:attribute>
124
125     <xs:all>
126       <xs:element name="description" type="xs:string" minOccurs="0">
127         <xsd:annotation xml:lang="en">
128           <xsd:documentation>
129             Description of the content of the file, for example whether it is raw MRI data or a ←
130               segmentation.
131             </xsd:documentation>
132           </xsd:annotation>
133         </xs:element>
134       </xs:all>
135     </xs:complexType>
136
137   <xs:complexType name="tracktype">
138
139   <xs:complexType name="datakeyvaluetype">
140     <xs:attribute name="key" type="xs:string">
141       <xsd:annotation xml:lang="en">
142         <xsd:documentation>
143           The key to be used later in the dictionary
144         </xsd:documentation>
145       </xsd:annotation>
146     </xs:attribute>
147   </xs:complexType>
148
149 
```

```

        </xs:attribute>
    </xs:complexType>

144    <xs:attribute name="name" type="xs:string">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                Name of the track
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

        <xs:attribute name="src" type="xs:string">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                Including the relative pathname to the Tracks/ folder
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

159    <xs:attribute name="fileformat" type="xs:string">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                So far, only track files from the TrackVis suite with ending .trk are supported
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

164    <xs:all>
        <xs:element name="description" type="xs:string" minOccurs="0">
            <xsd:annotation xml:lang="en">
                <xsd:documentation>
                    Description of the content of the file, e.g tractography algorithms
                </xsd:documentation>
            </xsd:annotation>
        </xs:element>

        </xs:all>
    </xs:complexType>
<xs:complexType name="nwmetadatatype">

    <xs:all>
        <xs:element name="data" type="datakeyvaluetype" minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation xml:lang="en">
                <xsd:documentation>
                    A data element containing the key as attribute and the value as a tag
                </xsd:documentation>
            </xsd:annotation>
        </xs:element>
    </xs:all>
</xs:complexType>

<xs:complexType name="networkstype">

194    <xs:attribute name="name" type="xs:string">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                The short name of the network as it appears in the ConnectomeViewer TreeView as node
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

        <xs:attribute name="src" type="xs:string">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                The name of the network.graphml file in the package
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

209    <xs:attribute name="hierarchical" type="xs:boolean">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                Is the graph hierarchical, i.e. do nodes contain subgraphs (not yet supported)
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

214    <xs:attribute name="hypergraph" type="xs:boolean">
        <xsd:annotation xml:lang="en">
            <xsd:documentation>
                Is the graph a hypergraph? (not yet supported)
            </xsd:documentation>
        </xsd:annotation>
    </xs:attribute>

219    <xs:attribute name="directed" type="xs:boolean">
        <xsd:annotation xml:lang="en">

```

```

229   <xsd:documentation>
230     Is the graph directed? Should be the same value as in the corresponding GraphML file.
231   </xsd:documentation>
232   </xsd:annotation>
233   </xs:attribute>

234 <xs:all>

235   <xs:element name="network-surface" type="surfacetype" minOccurs="0" maxOccurs="unbounded">
236     <xsd:annotation xml:lang="en">
237       <xsd:documentation>
238         All the surfaces corresponding to this network are stored in one Gifiti file in the Gifti/
239         folder. Since the vertices id should be corresponding over all surfaces, the nodes in the GraphML file have an unique intensityvalue integer corresponding to a set of surface vertices. This information is stored in the label Gifiti file.
240       </xsd:documentation>
241     </xsd:annotation>
242   </xs:element>

243 <xs:element name="network-metadata" type="nwmetadatatype" minOccurs="0" maxOccurs="1">
244   <xsd:annotation xml:lang="en">
245     <xsd:documentation>
246       Often, there is particular metadata for networks one wants to use for later processing. For example one can annotate networks that represent subjects with
247       information about being member of the control or test group. The very flexible schema of a dictionary (key/value pairs) is adopted to represent any kind of
248       metadata. This is not typed yet and just uses strings for key and value.
249     </xsd:documentation>
250   </xsd:annotation>
251 </xs:element>

252 <xs:element name="network-volume" type="volumetype" minOccurs="0" maxOccurs="unbounded">
253   <xsd:annotation xml:lang="en">
254     <xsd:documentation>
255       All the volumes found in the Nifti directory for this specific network. It is not yet clear how exactly they will be used by the ConnectomeViewer application
256     </xsd:documentation>
257   </xsd:annotation>
258 </xs:element>

259 <xs:element name="network-track" type="tracktype" minOccurs="0" maxOccurs="1">
260   <xsd:annotation xml:lang="en">
261     <xsd:documentation>
262       The track file in the Tracks/ directory that were the basis for the networks, if any
263       .
264     </xsd:documentation>
265   </xsd:annotation>
266 </xs:element>

267 </xs:all>
268 </xs:complexType>

269 <xs:complexType name="metadata">
270   <xs:all>
271     <xs:element name="generator" type="xs:string" minOccurs="0">
272       <xsd:annotation xml:lang="en">
273         <xsd:documentation>
274           Software version/Converter etc. this file was generated
275         </xsd:documentation>
276       </xsd:annotation>
277     </xs:element>

278   <xs:element name="initial-creator" type="xs:string" minOccurs="0">
279     <xsd:annotation xml:lang="en">
280       <xsd:documentation>
281         Author(s) of the data
282       </xsd:documentation>
283     </xsd:annotation>
284   </xs:element>

285   <xs:element name="creation-date" type="xs:date" minOccurs="0">
286     <xsd:annotation xml:lang="en">
287       <xsd:documentation>
288         When was the file created?
289       </xsd:documentation>
290     </xsd:annotation>
291   </xs:element>

292   <xs:element name="modification-date" type="xs:date" minOccurs="0">
293     <xsd:annotation xml:lang="en">
294       <xsd:documentation>
295         When was the file modified?
296       </xsd:documentation>
297     </xsd:annotation>
298   </xs:element>

299 </xs:all>
300 </xs:complexType>

```

```

        </xs:element>

309      <xs:element name="name" type="xs:string" minOccurs="0">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            What is the short name of this Connectome File?
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

314      <xs:element name="species" type="xs:string" minOccurs="0">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            Which species is dealt with? According to the ConnectomeWiki Category:Species taxonomy ( ←
              case-sensitive!)
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

324      <xs:element name="legal-notice" type="xs:string" minOccurs="0">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            Some legal notice concerning usage rights for the data.
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

329      <xs:element name="reference" type="xs:string" minOccurs="0">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            References to paper where the data has been published.
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

334      <xs:element name="url" type="xs:string" minOccurs="0">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            References to URL with further information or download location.
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

344      <xs:element name="description" type="xs:string" minOccurs="0">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            Description of the dataset.
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

349      <xs:element name="nr_of_networks" type="xs:integer" minOccurs="1" maxOccurs="1">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            The numbers of existing network files in the current Connectome File.
          </xsd:documentation>
        </xsd:annotation>
      </xs:element>

359      </xs:all>
      <xs:attribute name="version" type="xs:string">
        <xsd:annotation xml:lang="en">
          <xsd:documentation>
            Defines the version of the Connectome Schema Definition the current Connectome File is ←
              compatible with. Should be 1.0
          </xsd:documentation>
        </xsd:annotation>
      </xs:attribute>

364    </xs:complexType>

374    <xs:element name="viewer">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="viewer-meta" type="metadata" minOccurs="1" maxOccurs="1">
            <xsd:annotation xml:lang="en">
              <xsd:documentation>
                All meta data information that has nothing particularly to do with one instance of a ←
                  network
              </xsd:documentation>
            </xsd:annotation>
          </xs:element>

          <xs:element name="viewer-network" type="networkstype" minOccurs="0" maxOccurs="unbounded" ←
            "gt;
            <xsd:annotation xml:lang="en">
              <xsd:documentation>
```

The name of the network file in the current Connectome File. This defines a data source in the ConnectomeViewer.

```
394     </xsd:documentation>
  </xsd:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
399 </xs:element>
</xs:schema>
```

Appendix E

ConnectomeViewer Object Model

Depicted are the main classes developed for the ConnectomeViewer. No inheritance or composite structure is shown graphically. Methods beginning with an underscore are considered to be private. The complete code with many comments is released under GPLv3 and is available from Launchpad.net.



Figure E.1: The main classes for the Connectome File

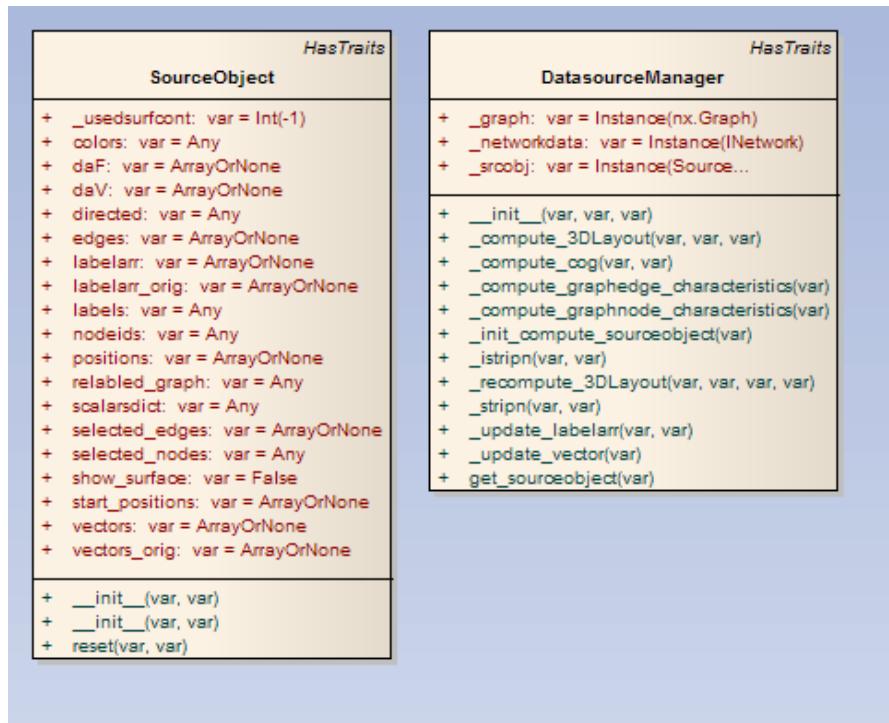


Figure E.2: The DataSource classes. Represent data for visualization and interaction (e.g. selection).

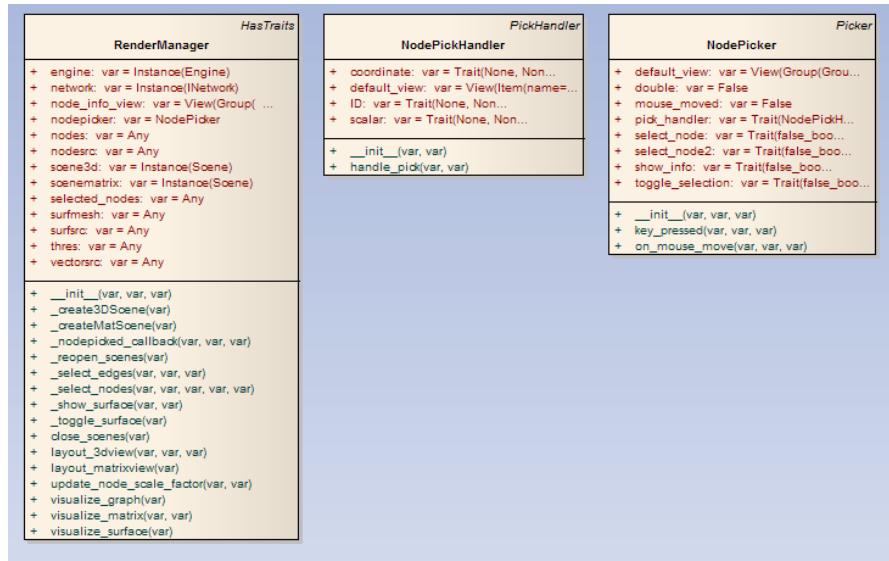


Figure E.3: The RenderManager deals with interfacing the Connectome File data with Mayavi2 using the DataSource object.



Figure E.4: The AnalysisPlugin.

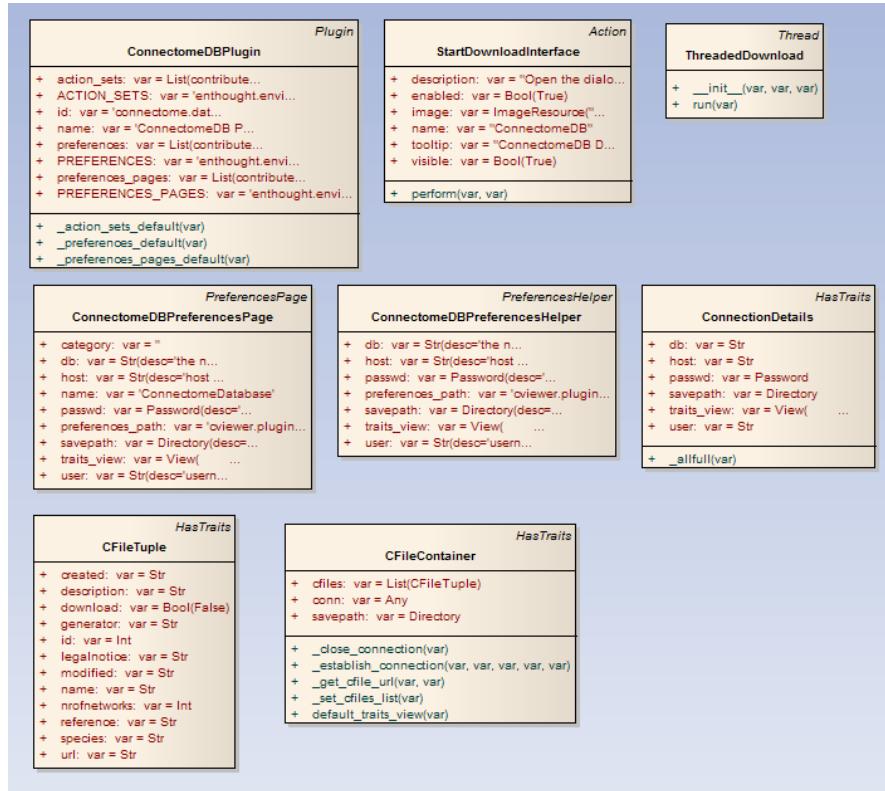


Figure E.5: The ConnectomeDatabase plugin

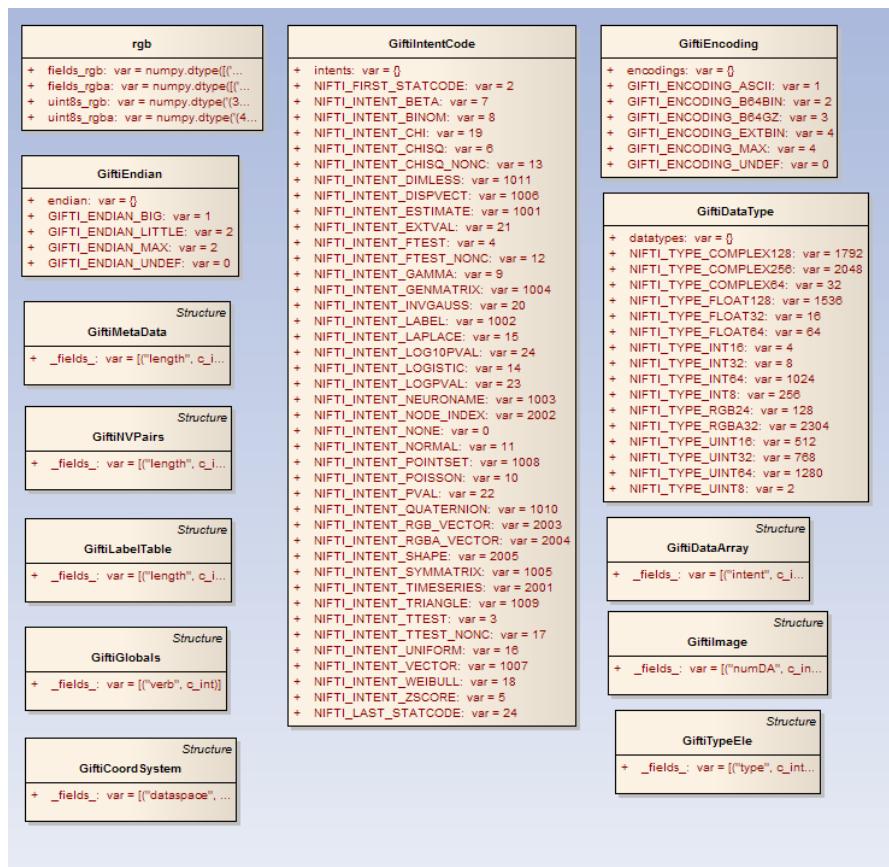


Figure E.6: The Gifti classes for surfaces. This is an external class wrapping fast C-functions. See 2.2.3

List of Figures

1.1	Overview of the Connectome Mapping Pipeline developed at EPFL LTS5. Necessary steps to process MRI data into Connectomes.	9
2.1	This Volume Slicer helper tool was adapted from a Mayavi Example. It allows to have a quick look at the segmentation of volumes.	20
2.2	The bottom part depicts the ROIs as surfaces of the left and right cortical hemispheres as spheres and subcortical nuclei. The center of gravity of these ROI patches define the position of the nodes for the network layout.	21
2.3	Software Application Layers [37]	23
2.4	Connectome File View (green), Mayavi Visualization Tree and Object (green-blue), 3D View (blue), IPython Shell (orange)	24
2.5	Topview of the average EEG attention dataset generated for eye-closed, resting condition in the beta1 frequency range.	27
2.6	Two nodes are selected nodes in the 3D View over the occipital lobe. Adjacent edges are shown. The edges are colorcoded and represent the mean FA (mean fractional anisotropy) in this case.	28
2.7	Automatically generated occipital ROIs in TrackVis from selected nodes in Figure 2.6. Fibers are filtered by these two ROIs.	28
2.8	The Node Information box appearing after key pressing 'i' in the 3D View.	28
2.9	The ConnectomeDatabase plugin view from within the ConnectomeViewer provides access to a relational database.	30
3.1	A screenshot of the start page http://connectome.ch/wiki/	34
3.2	Links to other resources	36
3.3	Query external databases	37
3.4	Literature search and Text Mining databases	37
3.5	Category hierarchies to specifically annotate brain regions and connections.	38
4.1	The calculated M values for all nodes and subject in Resolution 83 (left) and for Resolution 150 (right). A marked descrease in the inverse myelination strength is visible.	49
4.2	Linear regression curves for all nodes in Resolution 83.	50
4.3	Scatterplot of a superior parietal node with fitted linear (left) and exponential (right) regression curves.	50
4.4	Histograms of the linear regression r-values are shown for resolution 83 (left) and 150 (right).	50
4.5	The average surface mesh colorcoded by the gradient values of the linear regression for ROIs in resolution 83. For $p < .05$, the absolute r-value must exceed 0.35.	51
4.6	The average surface mesh colorcoded by the p-values reflecting the confidence of the slope in the linear regression analysis for ROIs in resolution 83.	51
4.7	The average surface mesh colorcoded by the r-values for the linear regression analysis.	52
4.8	Two red cluster are salient in a). No change visible for nodes in b).	52
4.9	The nodes representing the red clusters a) in Figure 4.8 are highlighted with red (parietal region). Side, top and medial view.	53

4.10 Subcortical nuclei corresponding to group b) in Figure 4.8 are shown in red embedded in a transparent cortical surface rendering.	53
4.11 The GUI of the sLORETA Converter with mask to specify the data parameters and for entering metadata. State saving allows to retrieve already entered information.	56
4.12 For a single subject, the lagged linear connectivity information is shown colorcode for the alpha1 band in the breathing condition.	56
4.13 Section from the "statistical" networks produced by the visualization script. Red indicates positive correlation. Top to bottom row are conditions breathing, arithmetic and rest. The columns are top, left and back view. The networks depict the correlation of the linear lagged connectivity in the beta1 frequency range with the behavioural variable age. A statistically significant increase in beta1 linear lagged connectivity is thus observed, although not spatially consistent.	57
4.14 Probability of connection matrix (resolution 83) showing colorcode the probability of subjects having this connection (red: 1.0, blue: 0)	58
4.15 Resolution 83 network embedded in 3D is shown from top view. Nodes are positioned at the center of gravity of their corresponding ROI in the atlas mesh. Only edges with a probability of connection equals 1.0 (edge occurs in all subjects) are shown.	59
4.16 Probability of connection matrix shown for resolution 258.	59
4.17 Top- and side view of nodes colorcoded and scaled by their node degree. The subcortical nuclei stand out as compared to the cortical nodes.	60
4.18 Left topview of centrality measures in the weighted case. Weights are the probability of connection. On the right side, the unweighted case. The subcortical nuclei stand out, especially the brain stem. The picture is similar as for the node degrees.	61
4.19 Top- and side view of transparent nodes depicting their clustering coefficient with a colorcode and scaling. No obvious pattern is visible, except for a minimally pronounced increase in anterior and posterior medial sites.	61
4.20 Percent of number of synapses involved in the projections between excitatory and inhibitory neurons between layers. A) between excitatory B) from excitatory onto inhibitory, C) from inhibitory onto excitatory, D) between inhibitory neurones.	63
4.21 This rendering shows the thresholded microcircuit as described in the text. Several drawbacks of this directed graph layout are visible, e.g. the arrow from L6_ext to L4_ext goes through L5_ext (so-called edge crossing), self-loops are not shown, and bidirectional arrows occlude themselves. More issues with regard to directed graphs have to be addressed in the future.	63
4.22 An intricate net even for only three reconstructed cortical neurons. (Axons: red, Dendrite: green)	64
4.23 The estimated directed network for these three neurons based on potential synaptic count with a proximity criterium. The node position is the center of gravity of the corresponding surface meshes.	65
4.24 It is possible to select only a subset of surfaces to display, such as in this case, only the dendritic arbors.	65
4.25 The yellow spheres are located where potential synapses are expected, based on the proximity of an axonal arbor to a dendritic arbor.	66
4.26 One reconstructed neuron surface was duplicated and placed in the same layer 20 times with random distribution. How many virtual neurons one can display at the same time depends mainly on working memory and level of detail of the surface reconstructions.	66
A.1 ConnectomeViewer Homepage http://www.connectomeviewer.org/	69
E.1 The main classes for the Connectome File	81
E.2 The DataSource classes. Represent data for visualization and interaction (e.g. selection).	82
E.3 The RenderManager deals with interfacing the Connectome File data with Mayavi2 using the DataSource object.	82
E.4 The AnalysisPlugin.	83
E.5 The ConnectomeDatabase plugin	83

E.6 The Gifti classes for surfaces. This is an external class wrapping fast C-functions. See 2.2.3	84
---	----

List of Tables

2.1	Key Bindings available in the 3D View	25
2.2	Default objects available in the IPython shell.	25
2.3	The fields in the SQL Table correspond to the metadata in the Connectome File (see Appendix D)	29
3.1	Brain Region Properties included in the Template	35
3.2	Three levels with the naming convention employed and an example	36
3.3	The naming convention employed for brain region connections	38
3.4	Brain Region Connection Properties included in the Template	38
3.5	Current statistics (January 2010) of the content of the ConnectomeWiki	40
4.1	Number of estimated synapse based on a proximity criterion	65