TU WIEN Informatics

Anatomy-Driven Layouting for Brain Network Visualization

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Monika Wißmann, MSc. BSc.
Matrikelnummer 00801513

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Ing. Dr.techn. Florian Ganglberger
Dipl.-Math. Dr.techn. Katja Bühler

Wien, 18. November 2022

Monika Wißmann

Eduard Gröller

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Anatomy-Driven Layouting for Brain Network Visualization

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Software Engineering and Internet Computing

by

Monika Wißmann, MSc. BSc.
Registration Number 00801513

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.-Ing. Dr.techn. Florian Ganglberger
Dipl.-Math. Dr.techn. Katja Bühler

Vienna, 18th November, 2022

Monika Wißmann

Eduard Gröller

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Monika Wißmann, MSc. BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. November 2022

Monika Wißmann

Danksagung

Zuallererst möchte ich mich bei meinem Betreuer Florian Ganglberger bedanken, der mich geduldig bei meiner Arbeit angeleitet und mir bei der Auswertung geholfen hat. Ich konnte von ihm viel über neurobiologische Netzwerke und Prozesse lernen.
Katja Bühler möchte ich für ihre Unterstützung und ihr konstruktives Feedback danken, welches mir zu den großartigen Ergebnissen verholfen hat. Ich danke ihr und Eduard Gröller für ihre hilfreichen Ratschläge und Bewertungen beim Schreiben dieser Arbeit.
Gwendolyn Rippberger danke ich für ihren Prototyp, der die Grundlage für diese Arbeit bildete.
Des Weiteren möchte ich mich bei Hsiang-Yun Wu für ihre große Hilfe bei der Durchführung und Auswertung der Nutzerstudien bedanken. In diesem Zusammenhang möchte ich mich auch bei den Teilnehmern der Nutzerstudien bedanken, insbesondere bei der Gruppe Haubensak am Institut für Molekulare Pathologie in Wien und bei Bohringer-Ingelheim in Biberach an der Riß. Ohne ihrem Engagement und ihrem Feedback hätte die Studie nicht durchgeführt werden können.

Ich danke meinem Partner Roland, der mich immer unterstützt und sich unter anderem mit Liebe und leckerem Essen um mich kümmert.
Ich möchte mich bei all meinen tollen Kollegen und Freunden am VRVis bedanken, die mir mit Begeisterung das Leben als Forscherin zeigten und mich mit gemeinsamen Freizeitaktivitäten vom Stress abgelenkt haben. Das hat mir sehr dabei geholfen einen gesunden Verstand zu bewahren, zwischen meiner Arbeit am VRVis, dem Besuch weiterer Veranstaltungen an der TU Wien und dem Schreiben dieser Arbeit.

Diese Arbeit wurde durch die VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH ermöglicht. Die Forschungs-GmbH wird vom BMVIT, BMDW, Land Steiermark, SFG und der Wirtschaftsagentur Wien im Rahmen des COMET-Competence Centers for Excellent Technologies (854174) gefördert, das von der FFG verwaltet wird.

Acknowledgements

First of all, I would like to thank my supervisor Florian Ganglberger, who patiently guided me in my work and greatly helped in its evaluation. I learned a lot from him about neurobiological networks and processes.
I would like to thank Katja Bühler for her support and constructive feedback, which helped to achieve the great results. I thank her and Eduard Gröller for their helpful advice and reviews in the writing of this thesis.
I thank Gwendolyn Rippberger for her prototype, which formed the basis of this work.
Furthermore, I would also like to thank Hsiang-Yun Wu for her great help during the conduction and evaluation of the user studies.
In this regard, I would also like to thank the participants of the user studies, especially the Haubensak group at the Institute of Molecular Pathology in Vienna and Bohringer-Ingelheim in Biberach an der Riß. Without your engagement and feedback, the study could not have been conducted.

I thank my partner Roland, who always supports me and takes care of me by providing love and delicious food, among other things.
I would like to thank all my great colleagues and friends at VRVis who enthusiastically introduced me to research life and distracted me from stress with leisure activities together. This really helped me to keep a sound mind between my work at VRVis, attending further courses at the TU Vienna and writing this thesis.

This thesis was made possible by the VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH. The reaserch company is funded by BMVIT, BMDW, Styria, SFG and Vienna Business Agency in the scope of COMET–Competence Centers for Excellent Technologies (854174) which is managed by FFG.

Kurzfassung

Technologische Fortschritte haben unsere Möglichkeit drastisch erweitert, Daten neuronaler Konnektivität des Gehirns zu sammeln und sie im Bereich der Konnektomik anzuwenden. Der Schwerpunkt der Forschung verlagert sich daher zunehmend auf die Analyse dieser komplexen Daten. Viele Anwendungen visualisieren neurologische Daten im dreidimensionalen Raum. Diese erfordern jedoch Interaktivität, um verdeckte Datenpunkte zu erkennen, und sind daher nicht immer anwendbar. Um die neurowissenschaftliche Forschung in dieser Hinsicht zu unterstützen, stellen wir *Spatial-Data-Driven Layouts* vor, ein neuartiges Web-Tool, welches sich der Visualisierung neuronaler Netzwerke verschiedener Spezies im zweidimensionalen Raum widmet.

Unsere Methode ist datengesteuert und daher unabhängig von der Spezies oder der Perspektive. Wir erstellen Node-Link-Diagramme, welche die Gehirnregionen als Knoten darstellen und die Konnektivität durch die Kanten visualisieren. Für diesen datengesteuerten Ansatz verwenden wir eine aus Hirnatlanten abgeleitete Konnektivität (*Parcellation-derived Connectivity*) in Kombination mit einem Standard-Algorithmus für kraftgesteuertes Graphenlayout.

Um die Orientierung zu erleichtern, wird die zugrunde liegende Gehirn-Hierarchie visualisiert. Farbige Parzellierungen im Hintergrund kapseln und gruppieren Knoten, die gemeinsame Überregionen besitzen. Dieser Hintergrund ähnelt von der Form her dem Gehirn und ist unabhängig von der Vollständigkeit des Netzwerks, wodurch der Vergleich von Teilnetzen untereinander und mit dem gesamten Netzwerk erleichtert wird. Der Detailgrad ist anpassbar, um entweder die anatomische Größe oder die Anzahl der Verbindungen pro Region widerzuspiegeln.

Es wurden Fallstudien für zwei verschiedene Spezies, Mensch und Maus, durchgeführt, um unsere Visualisierungen zu validieren und zu zeigen, dass die räumliche Verteilung der Knoten die Anatomie des Gehirns widerspiegelt. Knoten, welche benachbarte Regionen im Referenzraum darstellen, liegen auch in der Visualisierung nebeneinander.

Die von *Spatial-Data-Driven Layouts* gelieferten Ergebnisse wurden in einer web-basierten Nutzerstudie evaluiert, an der Experten aus den Bereichen Neurowissenschaften, Informatik, Bioinformatik und Computational Biology teilnahmen. Die Auswertung der Studien für die zwei verschiedene Spezies Maus und Mensch zeigt, dass unsere Methodik datenbasiert und Spezies-unabhängig angewendet werden kann. Das Feedback der Experten

zeigt ein deutliches Potenzial. *Spatial-Data-Driven Layouts* stellt schnell und einfach Abbildungen in der Literatur nach, die normalerweise mit viel Aufwand kreiert werden müssen. Das Hinzufügen von Kontext in Teilnetzen, um die Gesamtform des Gehirns zu erhalten und diese Netze miteinander vergleichbar zu machen, wurde als sehr nützlich angesehen. *Spatial-Data-Driven Layouts* ist ein Novum in der Visualisierung neuronaler Schaltkreise des Larvengehirns von *Drosophila melanogaster* und wird als ein erster guter Schritt in diese Richtung angesehen.

Zukünftig wird geplant, die Anwendung mit Interaktivität zu erweitern, um Neurowissenschaftlern eine intuitive Darstellung ihrer Daten zu ermöglichen. Die Anpassung der Gehirnregionen, der Konnektivität sowie der Details des Layouts kann über Parameter an ihre Interessen angepasst werden. Darüber hinaus wollen wir die Visualisierung auf Neuronenebene und die visuelle Kodierung der Graphen des Larven-Netzwerks verbessern, um eine detaillierte Darstellung der neuronalen Schaltkreise zu ermöglichen.

Abstract

Technological advances have dramatically expanded our ability to collect data of neural connectivity in the brain and apply this data in the field of connectomics. The focus of research is thus increasingly shifting towards the analysis of this complex data. Many applications visualize neurological data in three-dimensional space. However, these require interactivity to view hidden data and are not always applicable. To support neuroscientific research we present *Spatial-Data-Driven Layouts*, a novel web-tool to visualize neuronal networks of multiple species in two-dimensional space. Our method is data-driven and is therefore independent of species or perspective. We generate node-link diagrams where nodes represent brain regions, while the edges correspond the connectivity. To realize this data-driven approach we apply *Parcellation-derived Connectivity*, generated from brain atlases in combination with a standard force-directed graph layout algorithm.
We provide further guidance by visually encoding anatomical context of the underlying brain hierarchy. Colored parcellations in the background encapsulate and cluster nodes that belong to the same super-regions. Additionally the background provides an overall shape, similar to the brain and is independent of the graph's completeness, facilitating the comparison of sub-networks with each other as well as with the entire network. The background is customizable in terms of anatomical details to reflect either the anatomical size or the number of connections per region.

We conduct case studies for two species, mouse and human, to validate our visualizations and show that the spatial distribution of nodes reflects the anatomy of the brain. Nodes are adjacent to each other if they also represent neighboring regions in the reference space.
The results provided by *Spatial-Data-Driven Layouts* are evaluated in a web-based user study involving domain experts in neuroscience, computer science, computational science, bioinformatics, and computational biology. Evaluating the studies for two different species, mouse and human, shows that our methodology can be applied data-driven and species-independent. The feedback obtained from the experts indicates clear potential. *Spatial-Data-Driven Layouts* quickly and easily recreate illustrations in literature that usually are created with a great deal of effort. Added context in sub-networks to preserve the overall shape of the brain and to make those networks comparable to each other, was considered very useful. *Spatial-Data-Driven Layouts* is a novelty in the visualization of neuronal circuits of the *Drosophila melanogaster* larval brain and considered a first good step in this direction.

In the future, we plan to extend the application with interactivity to provide neuroscientists with an intuitive representation of their data. The customization of brain regions, connectivity, as well as details of the layout via parameters, can be adapted to their interests. In addition, we aim to improve neuron-level visualization and visual encoding of the *Drosophila* larval network graphs to provide a more detailed representation of circuits.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The bottleneck for progress in most research areas within neuroscience has shifted from the data acquisition to the data analysis stages [FGT+19]. Today's technology provides various ways to gather neuroscientific data that is difficult to fully analyze, due to both their volume and complexity as most brains have millions of neural connections within. Understanding and visualizing these networks is crucial for investigating cognition, memory, and diseases of the brain. Studies, literature, and many others include figures to show the results of experiments carried out or for educational purposes. Visualizations of biological networks, like the brain connectivity, are ubiquitous and quickly provide information [MPB+19]. This is where computer scientists come into play to develop visualization tools that apply methods to convert extensive and complex data into easy to analyze and interpret visualizations. However, creating a good visualization comes with numerous challenges.

1.2 Problem Statement

One possibility to visualize neurological network data is to use abstract visualization methods such as multidimensional scaling and scatter-plots [SSC+05]. These methods lack anatomical context, which could provide neurobiologists with orientation. For this purpose, a common way to visualize brain networks is a 3D node-link diagram, with brain regions rendered as spheres and connections rendered as straight lines [XWH13], [GSF+19]. A major issue with node-link visualizations is the visual clutter that occurs when many edges and nodes overlap due to the flattening of the 3D structure onto a 2D plane. Even though the occluded elements could be discovered via interactive navigation in the 3D space, navigating costs time and is not always possible, as in printed media. It becomes challenging to keep an overview of the global structure while visualizing a high level of detail given a finite display area. The users often lose track of their current position while navigating. Furthermore, most tools are trimmed to visualize data of particular species as the networks vary by size and shape of their brains and their regions [SBS+13].

1.3 Contributions

In this thesis we present *Spatial-Data-Driven Layouts*, a method to visualize neuronal networks in 2-dimensional space. Specifically, following contributions are made:

- Data-driven anatomical layouts are presented that eliminate the need to manually define brain region-related constraints. Their innovative features support neurological networks of multiple species and perspectives.
- The methodology creates reproducible layouts for sub-networks and partial graphs. While the visualization focuses on the partial network it still gives context of the overall brain. This enables the users to keep their orientation and makes visualizations of different parts of the brain comparable.
- Via an qualitative study it is proven that anatomical layouts allow neuroscientists a faster overall understanding of 2D network graphs than with traditional network visualization techniques. The author of this thesis designed the questions and tasks of the user study. The study was conducted and evaluated by Florian Ganglberger and Hsiang-Yun Wu.

1.4 Outline of the Thesis

In Chapter 2 we introduce the reader to the theoretical background that forms the basis of this work. We provide an overview of the biological background with a focus on neuroscience, including terminology and data underlying our work. We also describe the basis of graph drawing, the importance of aesthetics, and how these can be satisfied by means of layout techniques. Finally, we explain techniques of algorithmic geometry applied in our work.
In Chapter 3 we first describe other work that forms the basis of techniques applied in our approach. We then state examples of how neuronal data was visualized in previous works and compare our work with previous approaches.
Chapter 4 describes the underlying data of our work. It includes the hierarchical brain parcellation and the network to visualize.
Chapter 5 states the requirements that *Spatial-Data-Driven Layouts* need to fullfill. We give an overview of our approach and proceed to describe each step in detail, including the layout process, the visualization of parcellations, and giving context for partial networks.
Chapter 6 follows the steps of our methodology and describes the concrete implementation, including the involved technology.

In Chapter 7 we describe the evaluation of *Spatial-Data-Driven Layouts*. This includes a case study regarding the spatial arrangement of the brain network, a user study with domain experts, and their feedback. We conclude with Chapter 8, which discusses the problems as well as the potential of our work.

CHAPTER 2

Background

This chapter presents the theoretical background that forms the basis for the thesis. Selected aspects of neuroscience as well as methodologies of geometry relevant to our work are introduced here. The first part focuses on the biological background. We explain the relevant concepts of neuroscience by providing a brief introduction to biological terminology, followed by a description of how neurobiological connectivity data is obtained. In the second section of this chapter we explain graph drawing concepts, including aesthetics and layouting algorithms. Finally, we state further methods of algorithmic geometry that were applied in the course of our work.

2.1 Biological Background

To illustrate the contribution of this thesis we need to present the challenges that neuroscience has to overcome.

2.1.1 Terminology

Neuroscience is the study of the nervous system. Brains as the central part of the nervous system consist of a vast number of connected nerve cells or *neurons*. The average number of neurons depends on the species and can be about 199,380 cells in a whole adult brain of *Drosophila melanogaster* [RP21] or 86 billion neurons in the human brain [ACG$^+$09]. Neurons receive input via tree-like structures called *dendrites* and send signals via long tubular structures called *axons*. *Synapses* are connections between neurons through which information flows from one neuron to the next. The *connectome* is the complete map of neural connections in the brain and may be thought of as its "wiring diagram". *Connectomics* is the field of science that deals with the compilation, mapping, and analysis of neural connectivity data. The study of connectomics aims to improve our understanding of cognition and mental health by understanding how cells are connected

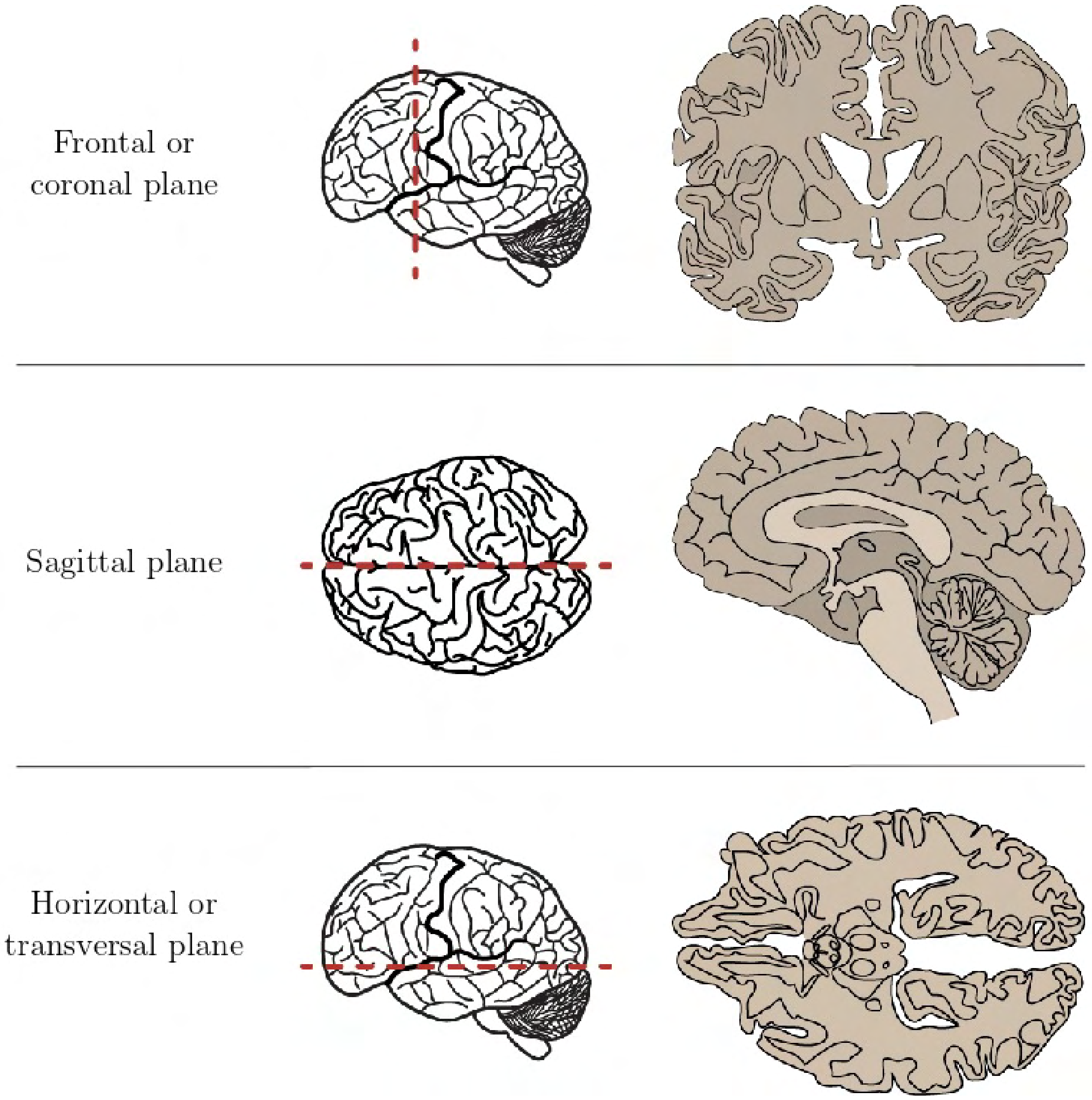


Figure 2.1: Three anatomical planes are used to divide the nervous system to be able to view internal regions and structures. ‘Anatomical Planes’ by Casey Henley is licensed under a Creative Commons Attribution Non-Commercial Share-Alike (CC BY-NC-SA) 4.0 International License [ana].

in the nervous system. Brain *parcellation* defines specific areas in the brain, i.e., areas or networks consisting of several discontinuous but closely connected regions, and is fundamental for the understanding of the function and organization of the brain.

Spatial-Data-Driven Layouts projects these spatial structures of the brain onto a two-dimensional plane. Since a plane is a 2D slice through 3D space, the anatomical planes are different structures used to divide an anatomical body. Using those planes allows for accurate description of a location. There are three commonly used planes (Figure 2.1):

- Coronal plane – a vertical plane which divides a body into a front (anterior) section and back (posterior) section.
- Sagittal plane – a vertical plane which divides a body into a left section and a right section.
- Transversal plane – a horizontal plane which divides a body into an upper (superior) section and a lower (inferior) section.

We describe the viewing directions of our visualisations according to the planes. Here, the sagittal perspective and transversal perspective are of particular relevance.

2.1.2 Data

Understanding the fundamental functioning and organization of the brain is one of the greatest and most important challenges in neurobiology. In order to make progress in research, not only is the acquisition of data crucial, but also the comprehension of this diverse information plays a key role. The field of connectomics uses non-invasive technologies, including resting-state and task-based fMRI, MEG, and EEG (function), as well as diffusion morphometric imaging and tractography, to map brain networks at the macroscopic scale to allow their analysis using graph theory [THZ+13]. The resulting brain data is either non-spatial or spatial. Non-spatial data are for example gene lists related to behavior or the functional association of genes. Spatial data can include structural and functional connections, as well as brain-wide gene expressions. *Spatial-Data-Driven Layouts* focuses on the visualization of spatial data to provide its anatomical context.
Data collections provided by different brain initiatives, as the Allen Institute [All], the Human Brain Project [hum], and the larvalbrain platform [lar], form the basis for our tool. Such atlases exist for various species, e.g., the mouse (Figure 2.2), the human (Figure 2.2), and the *Drosophila melanogaster* larvae (Figure 2.4). They are essential for research and education by annotating organs and creating coherence by defining a nomenclature. The Allen Brain Connectivity Atlases [All, DRS+16] are freely available, foundational resources for structural and functional investigations into the neural circuits that support behavioral and cognitive processes in health and disease. They include a hierarchical definition of the brain regions and parcellations as well as the connectivity.

For data acquisition *Enhanced Green Fluorescent Protein* (EGFP) is utilized to track axonal projections from specific regions and cell types. High-throughput serial two-photon tomography is used to image EGFP-labelled axons throughout the brain. This systematic and standardised approach enables spatial registration of the individual experiments in a common three-dimensional reference space and ultimately generates the connectivity matrix of the whole brain [OHN+14]. One example for such a connectivity is the resting-state fMRI, which measures spontaneous low-frequency fluctuations in the

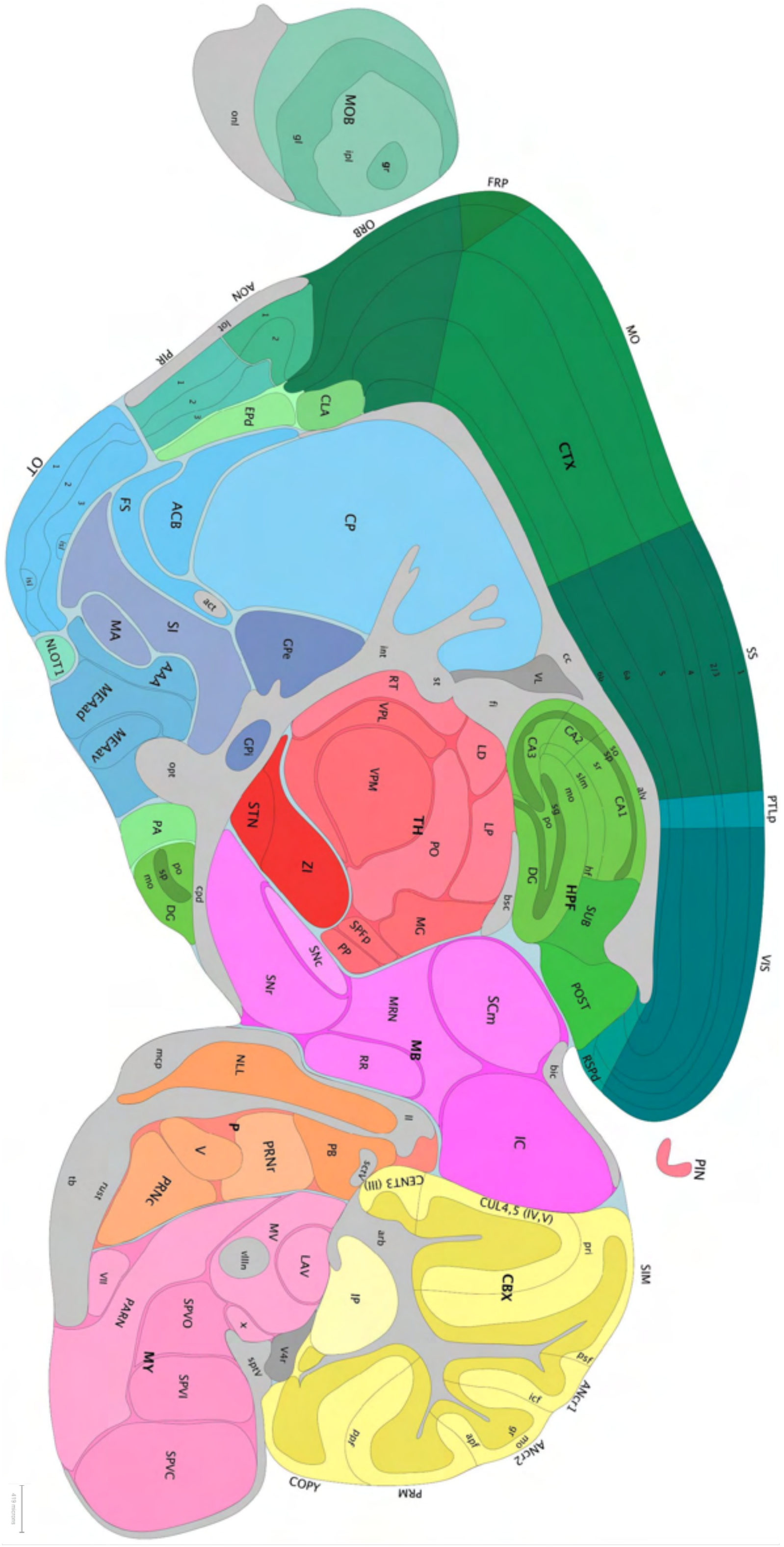


Figure 2.2: Atlas of the adult mouse brain from the transversal perspective [All].

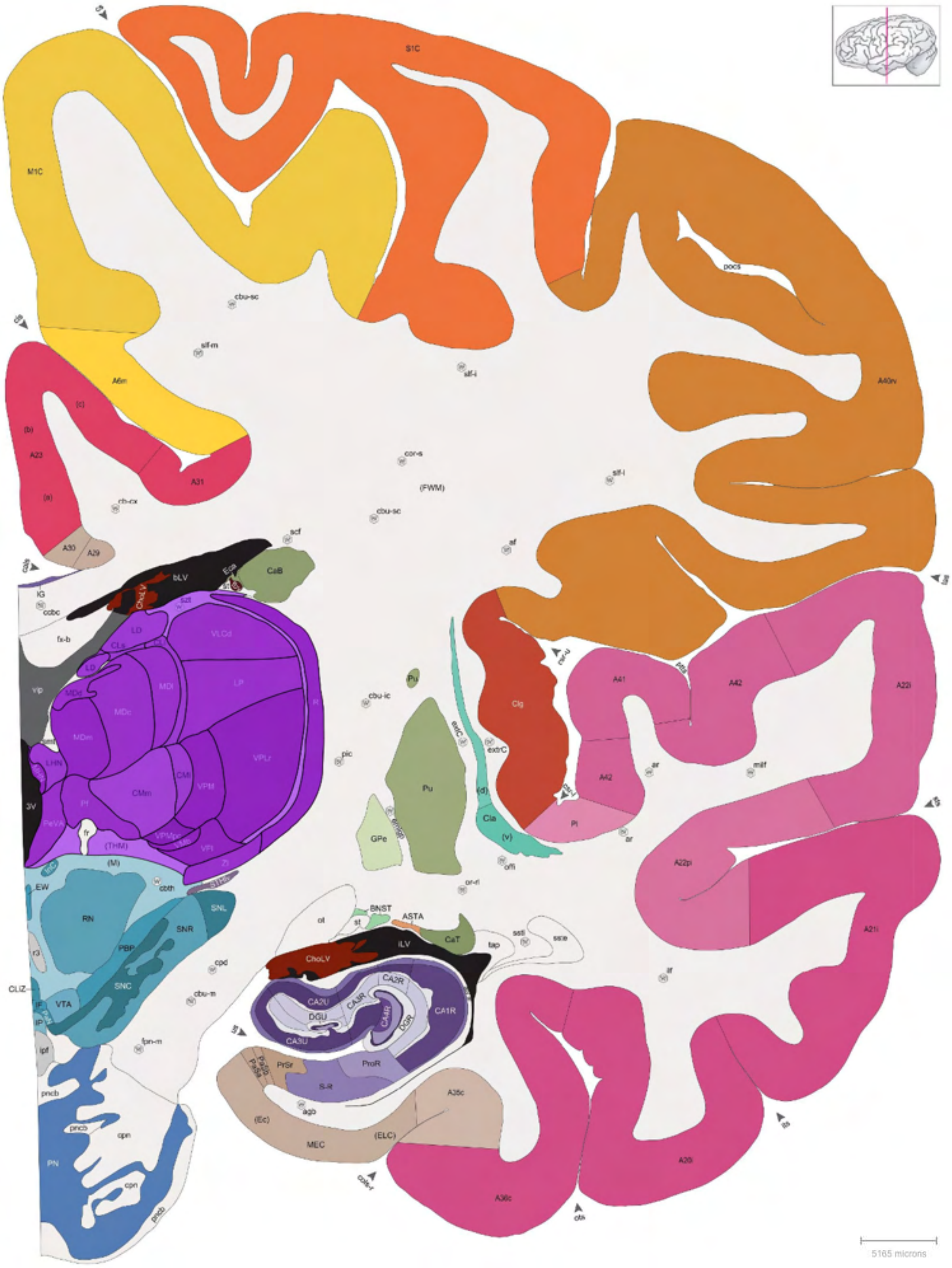


Figure 2.3: Atlas of the human brain from the coronal perspective [All].

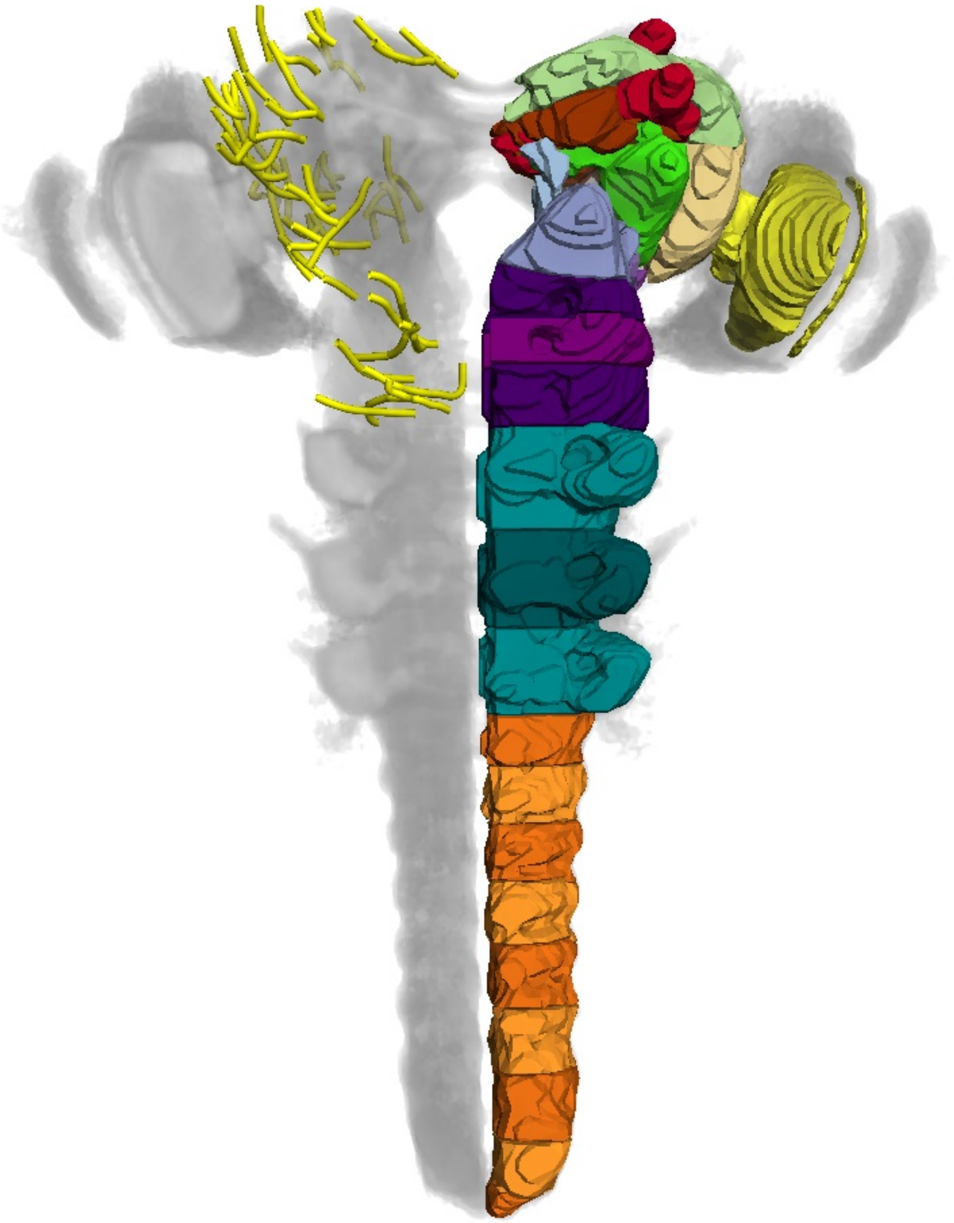


Figure 2.4: Atlas of the *D. melanogaster* larval brain from the transversal perspective [lar].

blood-oxygen-level–dependent signal to investigate the functional architecture of the brain [SBA+13].

The brain atlases are available for a variety of species. Like the species themselves, their brains come in a variety of shapes and sizes. Figure 2.5 displays this differences in brain volume scale. The nematode *Caenorhabditis elegans* is one of the simplest organisms with a nervous system and is as such an important model organism. Its system comprises a total of 302 neurons and has been comprehensively mapped. Its connectome is comparatively simple and has shown to be a small-world network [WS98].
The brain of an adult fruit fly *Drosophila melanogaster* is broader, consisting of about 199,380 cells [RP21]. *D. melanogaster* is one of the most studied organisms in biological research, particularly in genetics and developmental biology. It breeds and matures quickly as females lay up to 400 eggs every 16 days and the development from egg to an adult takes only several days. Keeping and breeding these insects is thus very cost-effective. Genetic modification and experiments are fairly simple which led to a good annotation of its genome.
Another longstanding model organism for human biology and disease is the mouse because of its relative phylogenetic similarity and physiological closeness to the human species. Similar to the fruit fly it is fairly cost-efficient to breed and keep in a laboratory. The total number of neurons in the entire mouse brain was determined to be around 70 million [HHML06]. The human brain includes 86 billion neurons [ACG+09]. It also has about the same number of non-neuronal cells, such as the oligodendrocytes, which insulate neuronal axons with a myelin sheath. The data we use for the human brain is provided by the Human Brain Project [hum]. It proposes a new approach that uses supercomputer technology to integrate existing biological knowledge into multilevel brain models. More details about our data can be found in Chapter 4.

2.2 Graph Drawing

This work aims to depict diverse brain networks in the form of graphs. Graph drawing derives two-dimensional representations of graphs and networks and is motivated by applications where it is crucial to visualize structural information. It combines mathematics and computer science with methods from geometric graph theory and information visualization. In the following section we explain the basic terminology related to graph theory and graph drawing. Following this, we introduce the reader to the theoretical basis of graph layouting, in particular to the force-directed layouting technique that is relevant to our methodology.

2.2.1 Terminology

A drawing of a graph or network is represented by vertices and edges. Directed graphs have edges with a direction. The edges indicate a one-way relationship in that each edge can only be traversed in a single direction. In contrast, undirected graphs have edges with no direction. The edges indicate a two-way relationship and can be traversed in

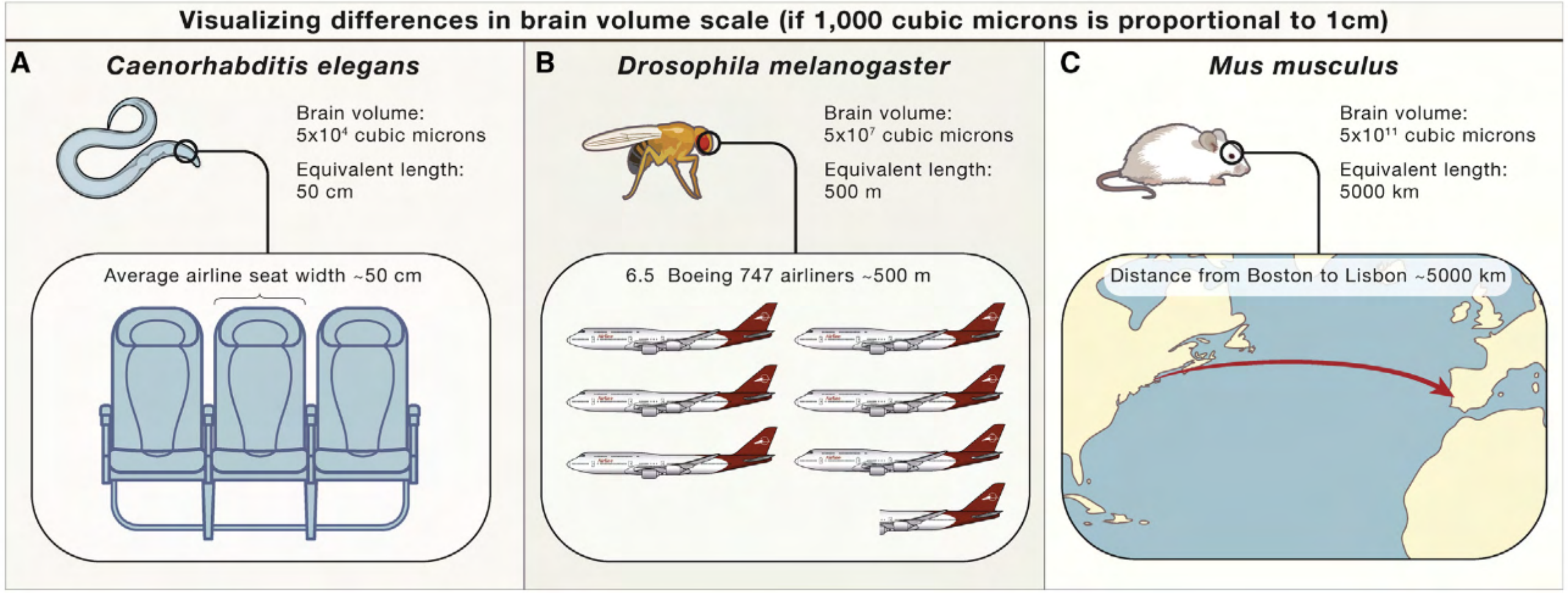


Figure 2.5: Scaling connectomic reconstruction from a worm to a mouse: A 10-million-fold increase in brain volume. Each 1000 cubic microns of brain volume is schematically represented by a 1 cm linear distance [ABC+20].

both ways. For simplicity, we will use the terms nodes/vertices and edges/connections as well as network/graph interchangeably.

2.2.2 Layouting

The drawing of a graph should not be confused with the graph itself as the same graph can be drawn in different layouts. Even though graph drawing is an important subfield of information visualization, making a 'good' drawing becomes more difficult for large and densely-connected graphs. Assuming edges are drawn as straight-line segments, the main goal of graph drawing is assigning appropriate coordinates to graph nodes. This operation is known as graph layouting. The arrangement of nodes and edges is decisive for how comprehensibly information can be perceived by the user. Aesthetic criteria typically refer to the number of crossings of edges, even distribution of nodes, and a similar length of edges.

There are many different graph-layouting strategies, e.g. force-directed methods [Ead84, FR91], dimensionality-reduction methods [KRM$^+$17, Kru80], and spectral methods [Hal70, BP06]. We focus on force-directed algorithms in this thesis, which are applied in *Spatial-Data-Driven Layouts*. Fundamentally, they simulate repulsive and attractive forces between a graph's nodes. Kobourov [Kob13] described them as follows:

> "In general, force-directed methods define an objective function which maps each graph layout into a number in $\mathcal{R}^+$ representing the energy of the layout. This function is defined in such a way that low energies correspond to layouts in which adjacent nodes are near some pre-specified distance from each other, and in which non-adjacent nodes are well-spaced. A layout for a graph is then calculated by finding a (often local) minimum of this objective function."

The nodes are assumed to have a certain "electrical charge" and as the edges work as "springs" with a predefined desired length, force-directed algorithms are also known as spring embedders (see Figure 2.6). Force-directed layouts are widely used and very popular as they provide good-quality results, flexibility, simplicity, are intuitive, and have strong theoretical foundations.

2.3 Algorithmic Geometry

The Voronoi Diagram and the convex hull are essential concepts in computational geometry. We apply these two techniques for the visualization of brain parcellations. The following two sections give a theoretical overview of these concepts.

2.3.1 Voronoi Diagram

The Voronoi diagram (also known as Voronoi tessellation, Voronoi decomposition, or Voronoi partition) was formally introduced by the mathematicians Dirichlet and Voronoi

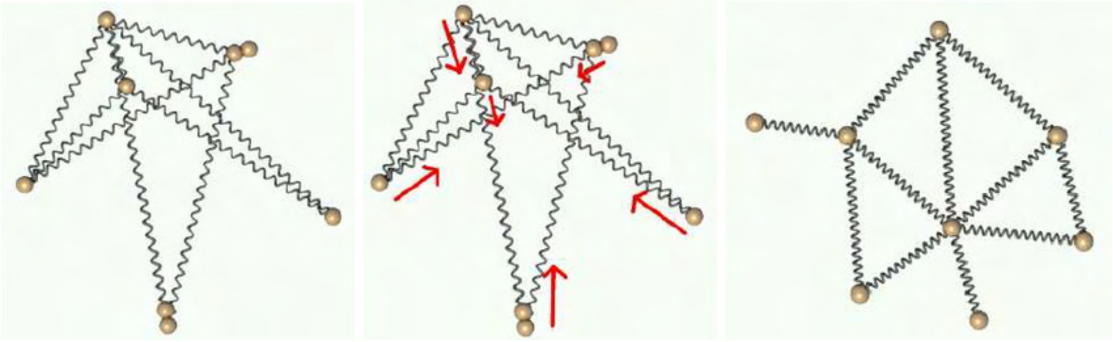


Figure 2.6: Illustration of a generic spring embedder: Starting from random positions of the nodes, the edges are treated as springs [Kob12].

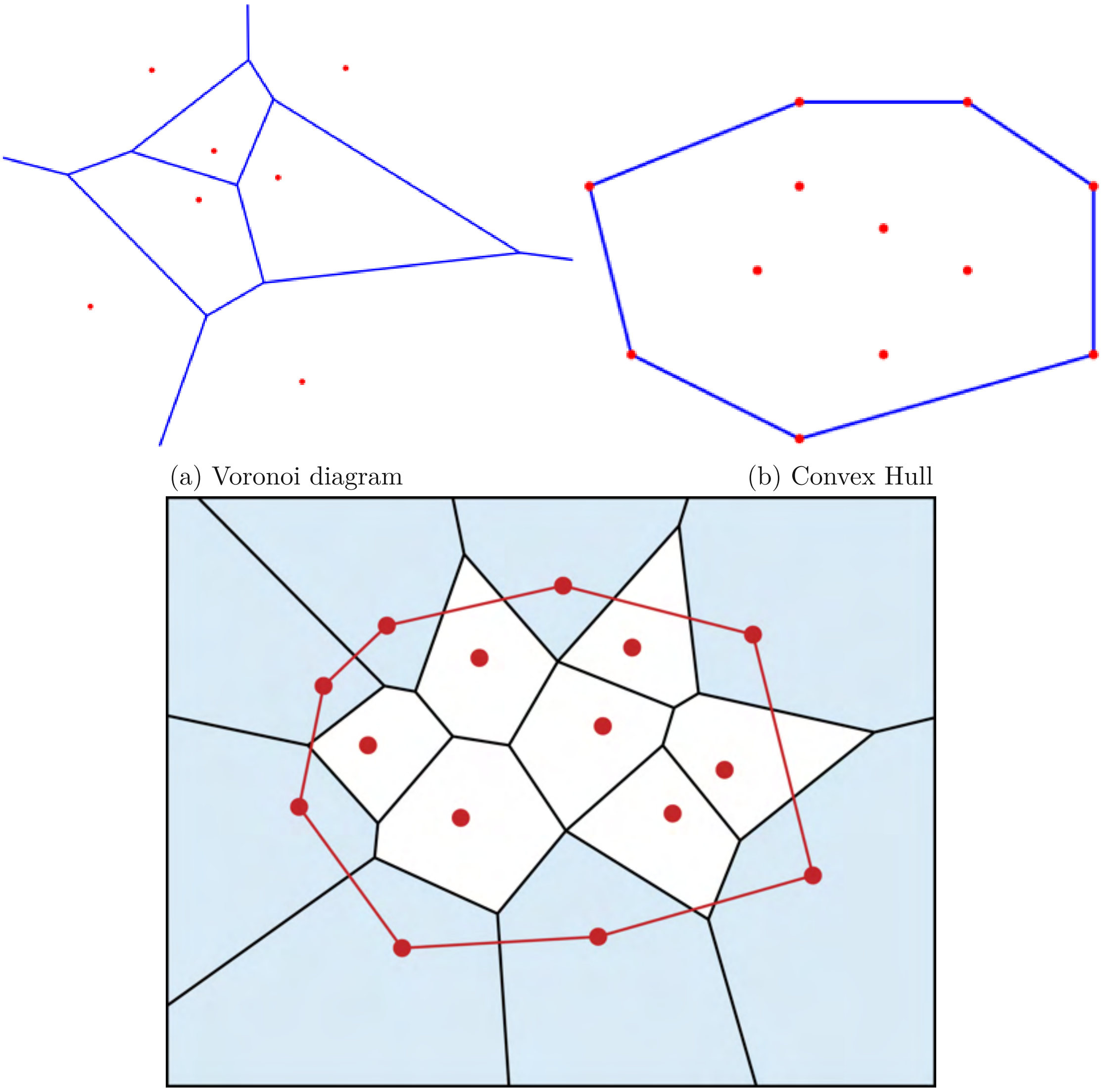
[LD50, Vor08]. A Voronoi diagram is a partition of a plane into regions close to each of a given set of objects (see Figure 2.7(a)). In the simplest case, these objects are just finitely many points in the plane (called seeds, sites, or generators). For each seed there is a corresponding region, called a *Voronoi cell*, consisting of all points of the plane closer to that seed than to any other seed. Points that are exactly the same distance away from two seeds form a line called the *Voronoi line*. It forms the boundary between two Voronoi cells. A point that is equidistant from three or more seeds is therefore located on the boundary between a corresponding number of Voronoi cells and is called a *Voronoi vertex*.

Voronoi diagrams are applied in a variety of domains. One of the most prominent examples is the early work of Snow [Sno55] who applied Voronoi tessellation to trace the source of a cholera outbreak in Soho, London, in 1854. He contained the disease by removing the handle of a water pump. Numerous natural sciences apply Voronoi tessellations. In medicine Voronoi tesselations are applied to understand tissue architecture in development and disease [SGTB+16]. In ecology they are used to study the growth patterns of forests and forest canopies. Meteorologists apply them for verification of precipitation forecasts [KLH18].

2.3.2 Convex Hull

The convex hull (also convex envelope or convex closure) of a set of points in the Euclidean plane is the smallest convex set that contains the points (see Figure 2.7(b)). For a bounded subset of the plane the convex hull can be thought of as the shape enclosed by a rubber band stretched around the subset. It can be defined in two ways: as the set of all convex combinations of points in the subset or as the intersection of all convex sets containing a given subset of a Euclidean space.

Many solutions have been proposed to calculate the convex hull, such as Graham's Scan [Gra72], Jarvis's March [Jar73], and Chan's algorithm [Cha96]. The convex hull has been used in a multitude of scientific fields, especially computer graphics, for applications to find bounding volumes [NW18], pattern matching [DPB+14, GS85], and analysis of



(a) Voronoi diagram

(b) Convex Hull

(c) Duality between Voronoi diagram and convex hull

Figure 2.7: Example of a Voronoi diagram (a) and a convex hull (b). Both techniques are closely related as the duality between Voronoi diagram and convex hull (c) shows [HPS16].

spectrometry data [SPD^{+}17].

Many problems can be reduced to the convex hull problem, including the Voronoi diagram. The Voronoi diagram and the convex hull interrelate with each other as follows: Assume the setting is the Euclidean plane and a group of different points is given. Then two points are adjacent on the convex hull if and only if their Voronoi cells share an infinitely long side (see Figure 2.7(c)).

CHAPTER 3

State of the Art

In this chapter we present related work to put this thesis into a bigger context. We first focus on aspects of graph visualization that are crucial for our results. Subsequently, we present state-of-the-art tools for visualization in neuroscience and compare existing tools with *Spatial-Data-Driven Layouts*.

3.1 2D Graph Visualization

We introduce the importance of aesthetics and explain how force-directed layouts have been used in previous works to satisfy the rules of aesthetics. Furthermore we provide an overview of works that applied hierarchical graph visualization and edge bundling.

3.1.1 Aesthetics

The aim of graph visualization is to create "good" graph drawings. Methodological works that present rules or heuristics for the creation of understandable, aesthetically pleasing graphs, and network figures provide guidelines how to overcome certain issues [BRSG07], [MPB$^+$19]. Several early papers on graph drawing discussed the requirements [TDBB88, STT81]. Tamassia et al. [TDBB88] for example wrote the following:

> *"Aesthetics: We use the term aesthetics to denote the criteria that concern certain aspects of readability. A well-admitted aesthetics, valid independently from the graphic standard, is the minimisation of crossings between edges. Also, in order to avoid unnecessary waste of space, it is usual to keep the area occupied by the drawing reasonably small. When the grid standard is adopted, it is meaningful to minimize the number of bends (turns) along the edges, as well as their total length."*

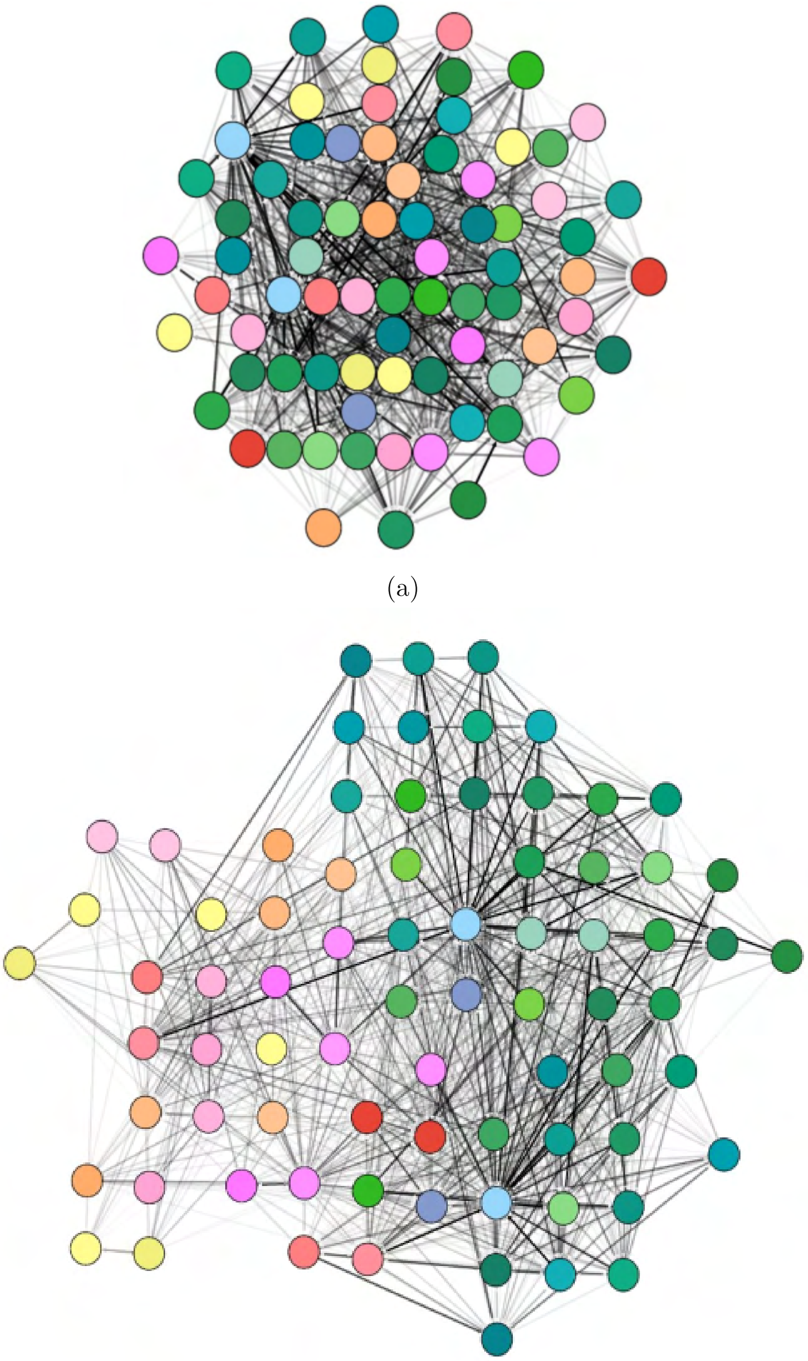
As we can see, aesthetics are not applied to graph drawing in a purely aesthetic sense - it has the practical aim of revealing underlying structures and meaning. Research has led to numerous aesthetic heuristics for drawing graphs. These efforts have been extended to empirically evaluate the effectiveness of these heuristics and to investigate their underlying basis in perceptual processing. Bennett et al. [BRSG07] conducted a survey on visualization heuristics regarding node placement, edge placement, graph layout, and domain-specific heuristics. Balancing potentially conflicting heuristics is an ongoing challenge. The usefulness of traditional quality metrics, such as the number of edge crossings, is recently doubted in the context of increasingly large graphs derived from various application areas including biology [EHNK17].
Understanding why certain data visualization techniques work better than others has psychological roots. Gestalt psychology suggests that human perception does not focus on every small component. Instead, it tends to perceive objects as part of a greater whole, complex system. The principles of grouping are a set of Gestalt principles. They teach us that the spatial arrangement of nodes and edges influences the reader's perception of the network information — even if there is no underlying meaning. Thus, the right layout can effectively enhance features and relations of interest, while the wrong layout could easily lead to misinterpretation. Marai et al. [MPB+19] propose ten simple rules for creating biological network figures. One of the rules points to unintended spatial interpretations that result from the Gestalt principles. Figure 3.1 shows an example. Both graphs represent the same regions of the normalized structural mouse brain connectivity data [MPB+19] as nodes. They apply the same force-directed layouting algorithm. Figure (a) uses connectivity strength as the driving force, positioning strongly connected nodes closely together, but neglecting the spatial context of the network. Figure (b) applies the spatial relation of brain regions resulting in a "flattened" mouse brain with symmetry and spatial positions approximately reproduced. Further rules by Marai et al. [MPB+19] cover aspects of the layout, applying color or other channels to show attributes, and the use of layering and separation.

The importance of the graph shape is also expressed in the work of Eades et al. [EHNK17]. They proposed a new set of quality metrics for graph drawing for larger graphs. Those shape-based metrics state that a drawing is good if the shape of the set of vertex positions is similar to the original graph.

3.1.2 Force-directed Layouts

The purpose of force-directed methods is to place nodes based on their relative positions without overlaps. We have already given an overview of the idea and the functionality of these methods in Section 2.2.2.

Force-directed methods in graph drawing emerged first in the work of Tutte in 1963 [Tut63]. He showed that polyhedral graphs can be drawn in the plane with all faces convex. After fixing the vertices of the outer face of a planar embedding of the graph into convex positions, he placed a spring-like attractive force on each edge and let the system settle into an equilibrium. A force-directed technique for drawing general graphs was first



(a)

(b)

Figure 3.1: Graphs representing mouse brain connectivity data applying a force-directed layouting algorithm. Figure (a) uses connectivity strength as the driving force. Figure (b) applies the spatial relation of brain regions [MPB+19].

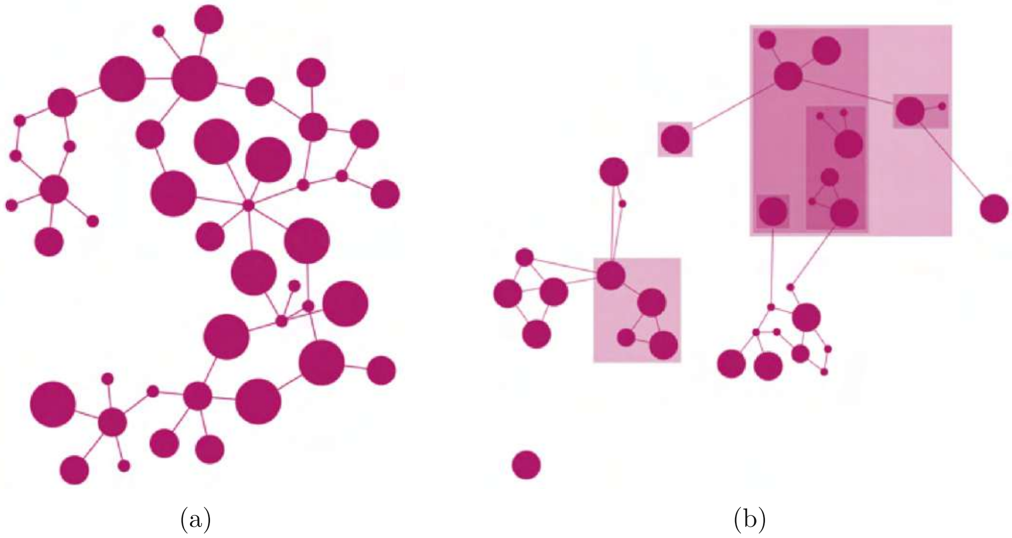


Figure 3.2: CoSE Layout applied to a graph (a) without and (b) with compound nodes.

published by Eades as a spring-embedder in 1984 [Ead84]. The method of Fruchterman and Reingold [FR91] from 1991 is based on Eades work and is used or extended in a significant number of subsequent studies. New variants of force-directed algorithms for drawing graphs are still introduced every year. More recent implementations focus on scalability or can handle large dynamic graphs.

The open-source network analysis and visualization software package Gephi implements ForceAtlas2 [JVHB14] as default layouting algorithm. ForceAtlas2 is a force-directed layout integrating different techniques such as the Barnes Hut simulation, degree-dependent repulsive force, and local and global adaptive temperatures. Another algorithm is CoSE (Compound Spring Embedder), a spring embedder layout developed by i-Vis Lab at the Bilkent University [DGC+09] (see Figure 3.2). It is based on the force-directed layout algorithm of Fruchterman and Reingold [FR91] and additionally handles the following:

- an arbitrary level of nesting
- inter-graph edges that may span multiple levels of nesting
- links to non-leaf nodes in the nesting hierarchy

The CoSE algorithm forms an important part of the methodology of our work.

Due to their property to present graphs organically and aesthetically pleasing, force-directed layouts are applied in numerous tools and frameworks of different domains. FORCE, a layout-based heuristic, handles huge data sets of prokaryotic protein sequences to calculate clusters [WBLR07]. It applies the layout algorithm by Fruchterman and

Reingold [FR91] to partition the data. GMap [GHK10] is another tool that applies the layout algorithm by Fruchterman and Reingold. It visualizes relational data with geographic-like maps. Its implementation couples the layout algorithm with a modularity-based clustering algorithm. Additionally it applies Voronoi diagrams with randomly placed points to recreate map-like boundaries.

3.1.3 Hierarchical Graph Visualization

Compound graphs have been used in the past to represent more complex data structures, which incorporate additional information on relationships, like hierarchy or clusters. Compared to the general graph layout, considerably less work on the layout of compound graphs has been done, probably due to the complex nature of this problem. Simple approaches in a top-down or bottom-up manner fail due to bidirectional dependencies between levels of varying depth.

Sugiyama and Misue [SM95] implemented D-Abductor, a generic compound graph visualizer and manipulator, to support human thought processes. It provides sophisticated techniques for visualizing and manipulating graphs using compound graphs and automatic graph drawing described in a preceding work by Sugiyama and Misue [SM91].

Several papers deal with the visualization of pathways using compounds. For example, Fukuda ad Takagi [FT01] describe a signal transduction-pathway representation-model that is based on a compound graph structure. It is designed to handle the diversity and hierarchical structure of pathways. Patika [DGC+04] is an analysis tool for drawing complicated biological pathways (Figure 3.3). It allows for arbitrary nesting relations to represent molecular complexes and pathway abstractions applying the CoSE algorithm [DGC+09].

Another tool that applies force-directed layout on hierarchical graphs is Chisio, a free, open-source compound graph editing and layout framework [KDBD17]. It comes with a variety of graph layout algorithms, most supporting compound structures, and further algorithms can be easily imported. It serves as a test environment for layout algorithm developers.

3.1.4 Edge Bundling

Edge bundling was first introduced by Holten [Hol06] to reduce visual clutter in hierarchical graphs and was later extended to non-hierarchical graphs using a self-organizing force-directed approach, where edges are modeled as flexible springs that can attract each other. The resulting bundled graphs show significant clutter reduction and visible high-level edge patterns as seen in Figure 3.4. Geometry-based Edge Clustering is introduced by Cui et al. [CZQ+08]. They convert general straight-line graphs into road-map-style graphs by clustering the edges based on a control mesh that reflects the underlying graph structures. They also apply force-directed layouting of nodes to decrease the number of edge crossings and thus reduce visual clutter. Wu et al. [WYY15] presented texture-based edge bundling. It uses textures to encode information about lines and forces and utilizes shaders to conduct an iterative line refinement on the GPU to

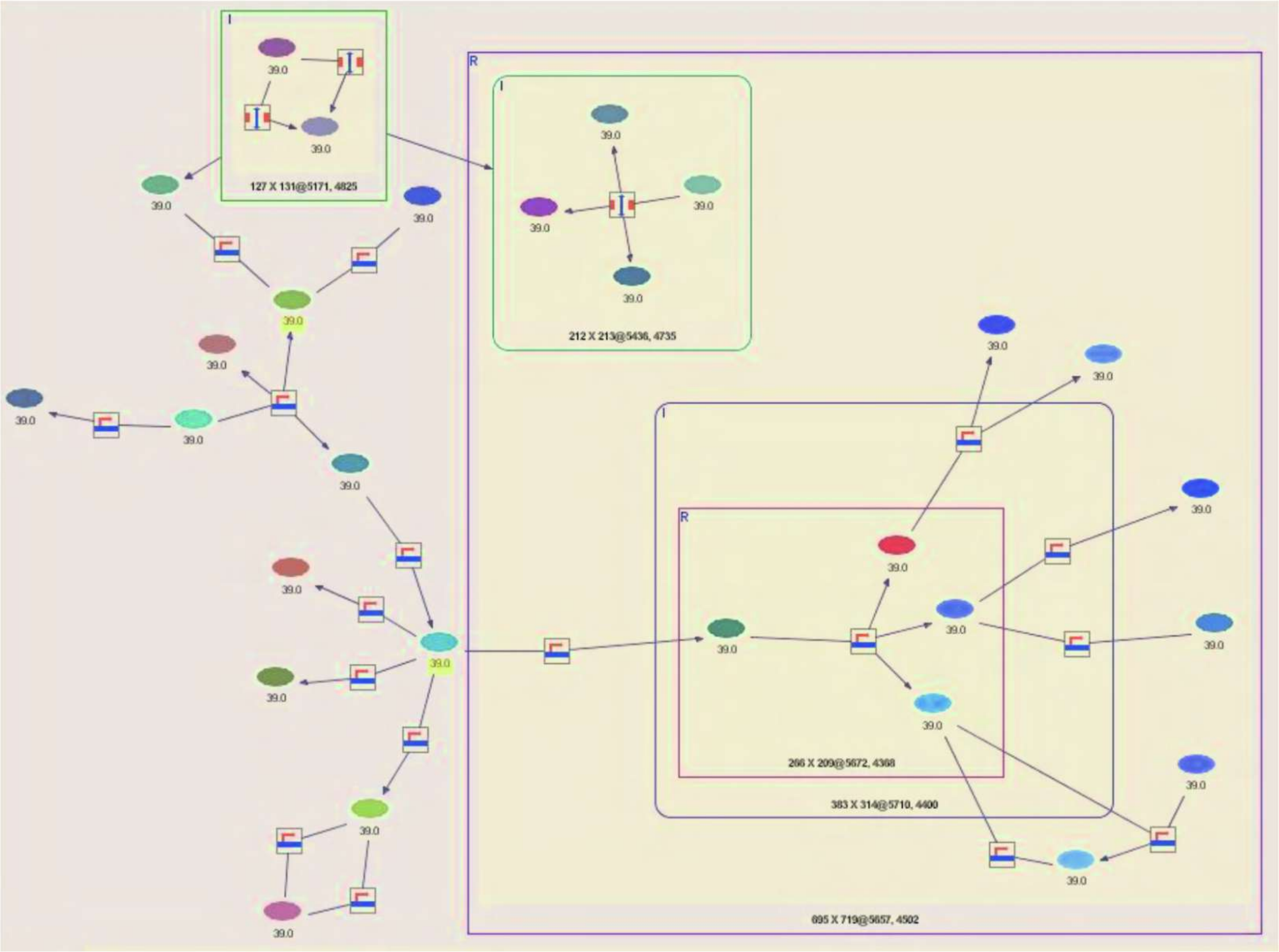


Figure 3.3: Sample pathway from the Patika editor laid out by the CoSE algorithm including compounds [DGC+04].

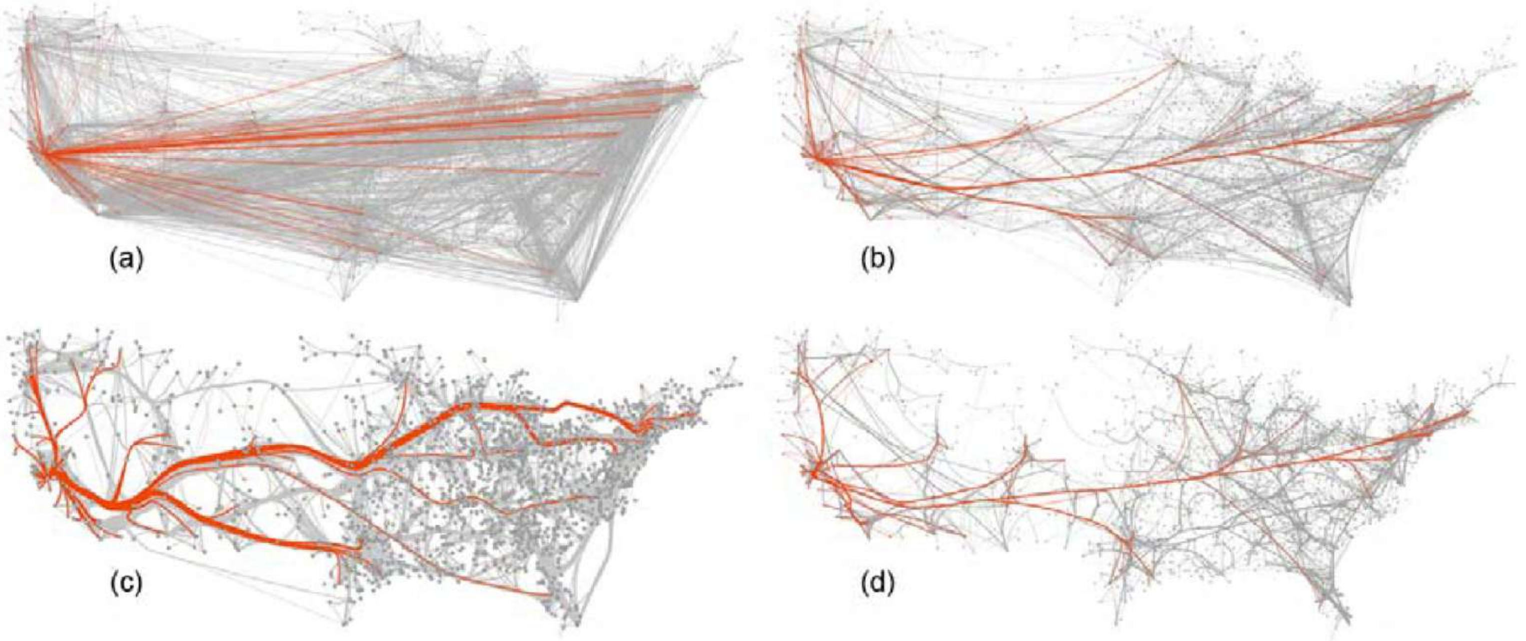


Figure 3.4: Example of the different edge bundling techniques by Holten and Van Wijk [HVW09] on an US migration graph (1715 nodes, 9780 edges) (a) not bundled and bundled using (b) force-directed edge bundling with inverse-linearmodel, (c) geometry-based edge bundling, and (d) force-directed edge bundling with inverse-quadratic model. The same migration flow is highlighted in each graph.

further improve the performance of force-directed edge bundling. Hurter et al. [HET12] described a cost-efficient method by transforming a given graph drawing into a density map using kernel density estimation. They further applied an image sharpening technique to merge local height maxima by moving the convolved graph edges into the height gradient flow. The framework CUBu [VDZCT16] is fully GPU-based. It extends and unifies existing bundling techniques to accelerate edge bundling while offering ways to control bundle shapes, separate bundles by edge direction, and shade bundles to create a level-of-detail visualization.
An example of edge bundling in connection with visualizing a neurological network is given by McGraw [McG15]. He partitions the connectivity matrix into blocks corresponding to brain hemispheres and bundles the graph edges to generate intuitive visualizations to allow investigation on multiple scales (Figure 3.5).

3.2 Related Work in Neuroscience

A very good introduction to connectomics and recent developments is given by Seung [Seu12], who also highlights the advances in high-throughput, high-resolution electronic imaging. Lichtman and Denk [LD11] describe the challenges in achieving the ultimate goal of connectomics - understanding the relationship between function and structure in the brain. Visualizing such networks is a crucial part to accomplish this goal.

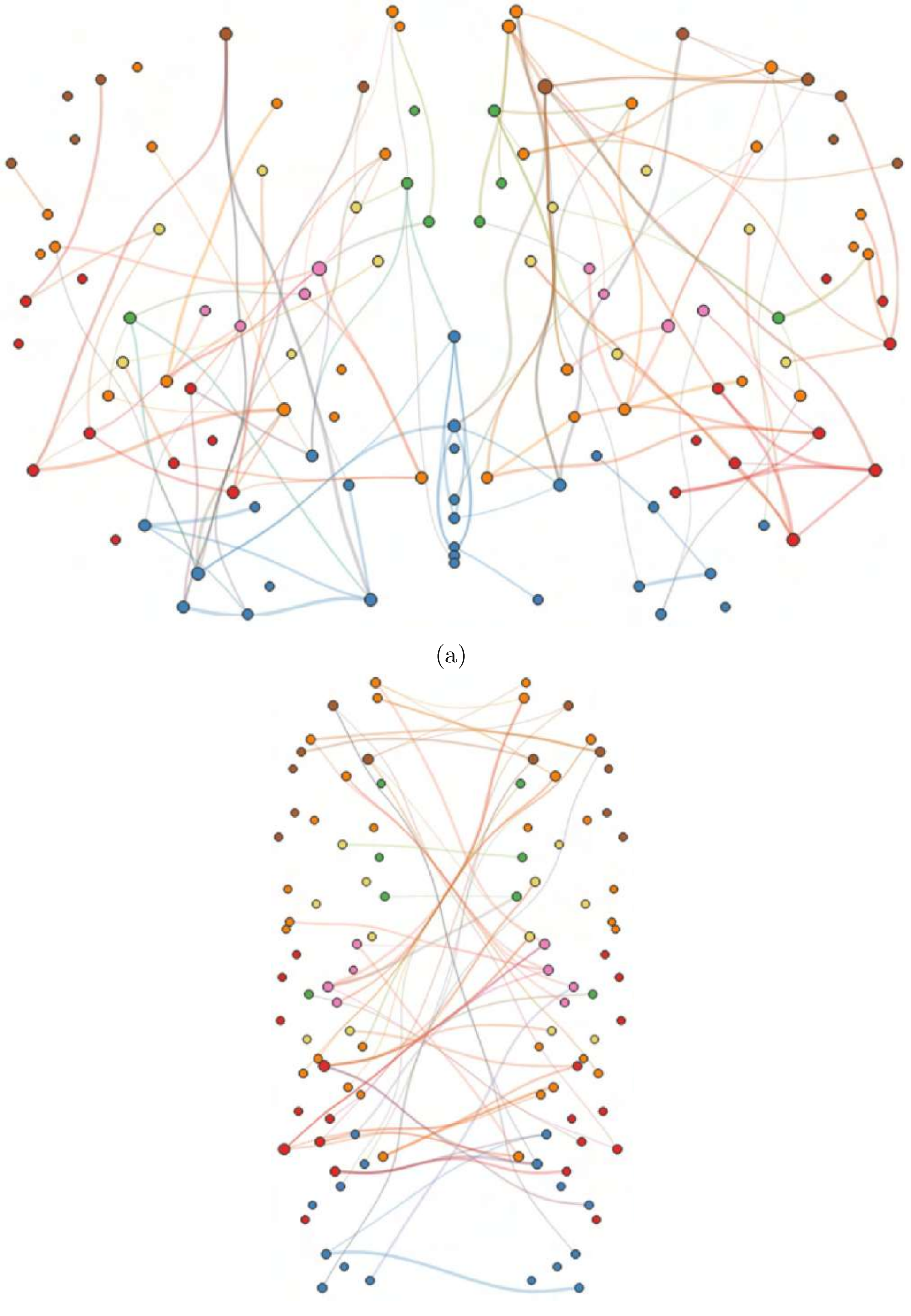
In this section we first describe the sources of data collection. We provide an overview of the various graphical representations of neuronal networks in prior works. Finally, we present some tools and frameworks that enable the visual representation of such networks. We analyze these approaches to delimit them from *Spatial-Data-Driven Layouts*

3.2.1 Visualization in Neuroscience

To visualize brain connectivity data, prior works used abstract, non-spatial approaches or spatial approaches, which take the anatomy of brain regions into account.

Non-spatial Visualization

Non-spatial visualization includes multidimensional scaling, spring embedding, matrix bitmaps, and scatter-plots. These methods aim at identifying structural clusters in the data. An example of such an approach is the work by Deshpande et al. [DSH11] visualizing the functional connectivity as a 2D spring embedded graph using the Kamada-Kawai algorithm [KK$^+$89]. Stefanovski et al. [STS$^+$19] represent structural connectivity using a matrix as a proof of concept to compare the simulated activity of altered molecular pathways concerning neuronal population dynamics (Figure 3.6). Baedea et al. [BWS$^+$19] identified vulnerable brain circuits in mouse models with homozygous targeted replacement of the mouse gene with one of two types of human gene alleles concerned with age and disease-associated challenges. Using tensor-network principal-component-analysis for structural connectomes they inferred the pairwise connections



(a)

(b)

Figure 3.5: Visualization of a human brain network using node-link diagram combined with edge bundling [McG15]. (a) Intrahemispherical connectivity. (b) Interhemispherical connectivity. Nodes representing lobes are colorcoded and connectivity weights are represented by edge thickness.

that best separate the two carriers. They visualized the outcome as scatter-plot of the top principal components (Figure 3.7). These analyses revealed not just genotype, but also sex-specific differences. Raredon et al. [RYG+21] present *Connectome*, a software package for R that facilitates rapid calculation and interactive exploration of cell-cell signaling network topologies contained in single-cell RNA-sequencing data. It allows to display the data as a connectogram (Figure 3.8(a)). Huang et al. [HZZ+20] analyzed and compared the difference in metabolic brain network connectivity among four different age groups from childhood to adolescence. They showed that the topological properties and modular reorganization of the human brain network dramatically changed with age. In an alluvial diagram they illustrate changes in modular assignments of metabolic brain networks across the four groups, where each block represents a module and each line corresponds to a brain region (Figure 3.8(b)).

Spatial Visualization

Spatial techniques such as 3D node-link diagrams help neuroscientists to orient themselves by providing anatomy as well as inspecting the structural patterns. However, these techniques often lead to occluded or scattered visualization. Several approaches have been adopted in the past to solve this problem. McGraw [McG15] integrated spatial relations and anatomical meaning into an abstract visualization directly while avoiding occlusions and clutter at the same time. They positioned the nodes of the graph using the automated anatomical labeling (AAL) brain atlas, discarding one of the three coordinates (Figure 3.5). The nodes are grouped by the hemisphere (left, right) and their corresponding brain lobes. Minimizing the overlap is achieved by using the method described by Misue et al. [MELS95]. The color of a node is determined by the lobe it belongs to, while the radius is proportional to the number of incident edges of the node. Edges are filtered and bundled in a similar approach as described by Holten and Van Wijk [HVW09]. Visualization of inter- and intra-hemispheric connectivity is separated to reduce clutter in inter-hemispheric connectivity.
Tymofiyeva et al. [THZ+13] studied structural brain networks in subjects of different ages, including premature neonates, term-born neonates, six-month-old infants, and adults. They visualized the comparison via anatomic MRI images, tractograms reconstructed based on DTI data, and binary connectivity matrices. Furthermore, they provided a figure from a network-driven segmentation of the cortex into five modules with a similar pattern for subjects from all age groups (Figure 3.9(c)).

3.2.2 Interactive Exploration Tools

In order to cope with the complex field of connectomics, numerous tools have been developed in recent years. An example is BrainNet Viewer [XWH13], a graph-theoretical network visualization toolbox to illustrate macro-scale human brain networks as ball-and-stick models (Figure 3.9(a)). It displays combinations of the brain surface, nodes, and edges from multiple perspectives (sagittal, axial, or coronal), and allows the user to adjust display properties like color and size of the network elements. A more abstract approach

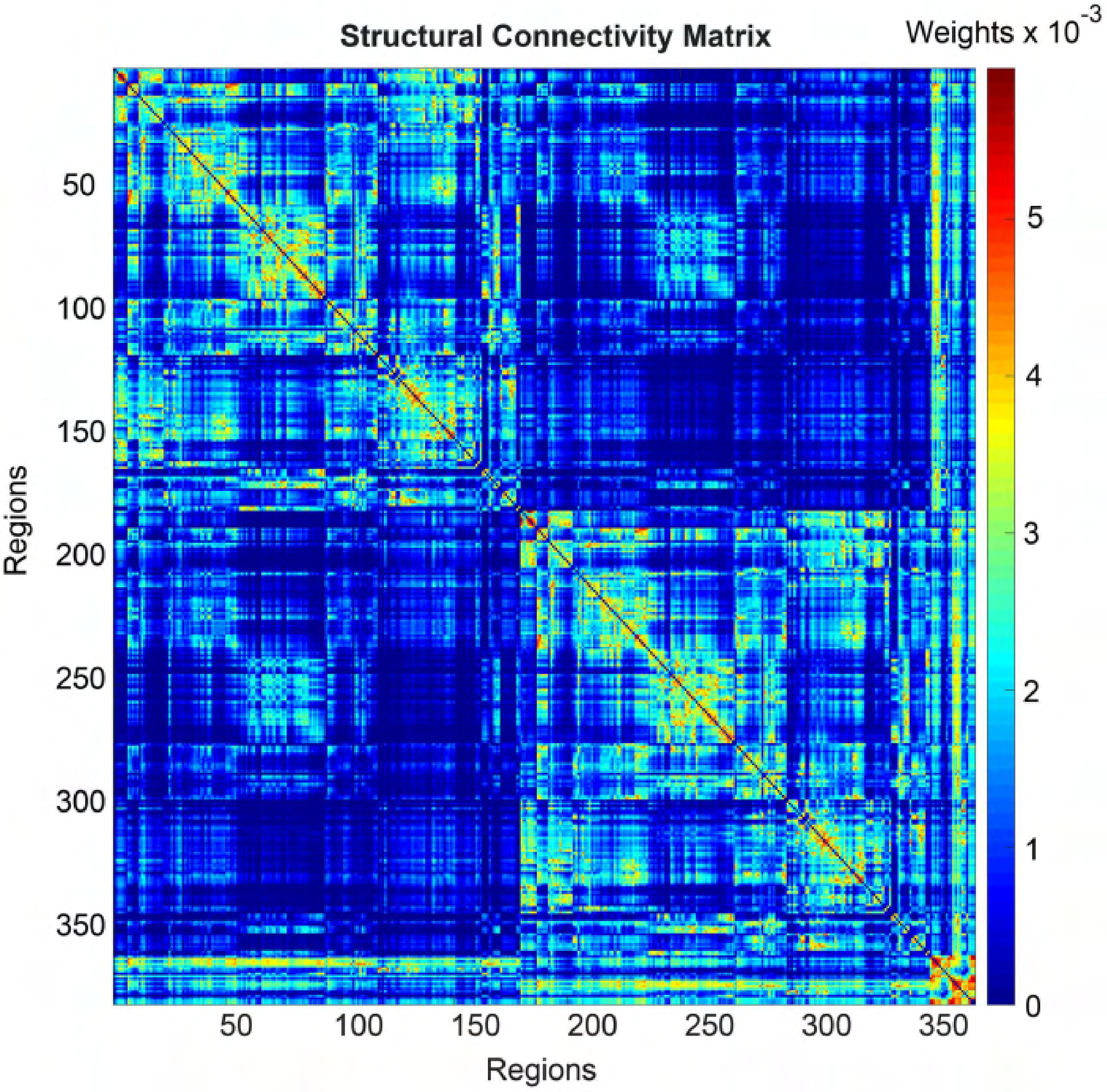


Figure 3.6: Adjacency matrix showing the connections weights of a human connectome [STS+19].

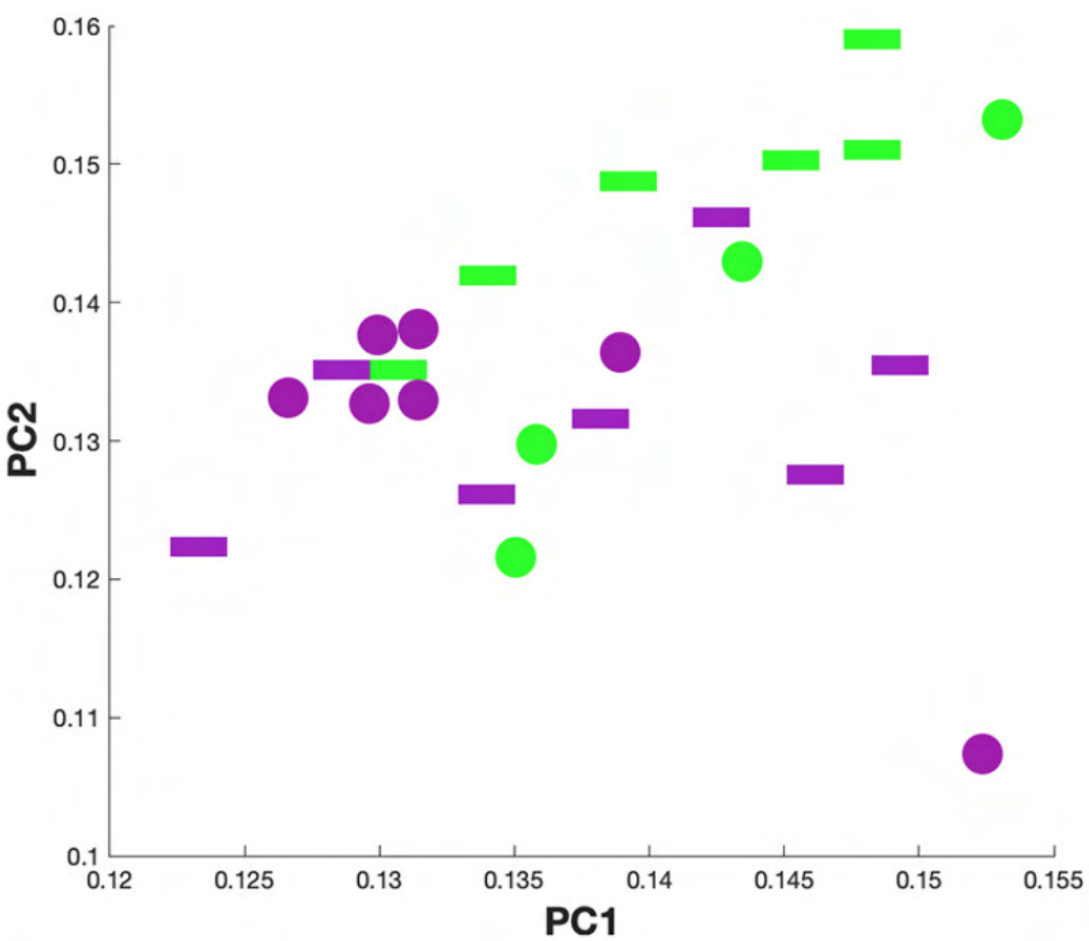
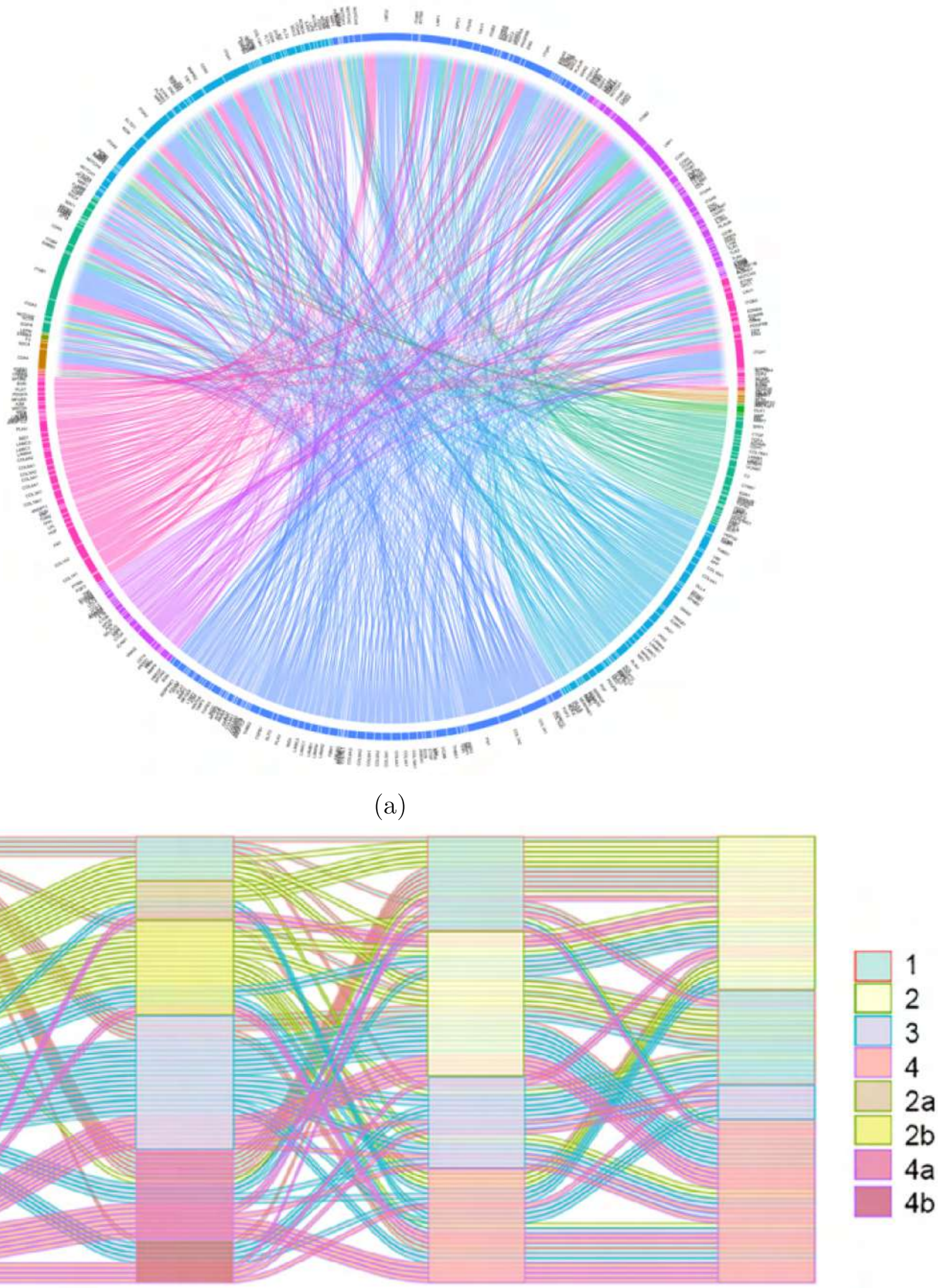


Figure 3.7: Scatter-plot of the top two principal components of a mouse connectome analysis [BWS+19].

is proposed by Jianu et al. [JDL12]. They developed a two-dimensional, representation of axonal tracts of the human brain as tractograms (Figure 3.9(b)). Their visualization is distributed in a traditional stand-alone interactive application and as a web-accessible system. The interactive exploration system facilitates exploration and analysis of brain connectivity with preservation of anatomical context in mind.

The vast resources of brain data, including the relationship between genes, brain circuitry, and behavior, require time-consuming manual aggregation of the data. Several tools handle this issue using queries. Beyer et al. [BAAK+13] presented ConnectomeExplorer, an application for the interactive exploration and query-guided visual analysis of large volumetric connectomic data sets. The query algebra allows neuroscientists to pose domain-specific questions intuitively and to interactively analyze the results. Another data structure is proposed by Ganglberger et al. [GKP+18]. Their *Aggregation Queries* -



(a)

(b)

Figure 3.8: (a) Connectogramm, a circular representations of brain connectomics [RYG+21]. (b) Alluvial diagram representing changes in network structure over time [HZZ+20].

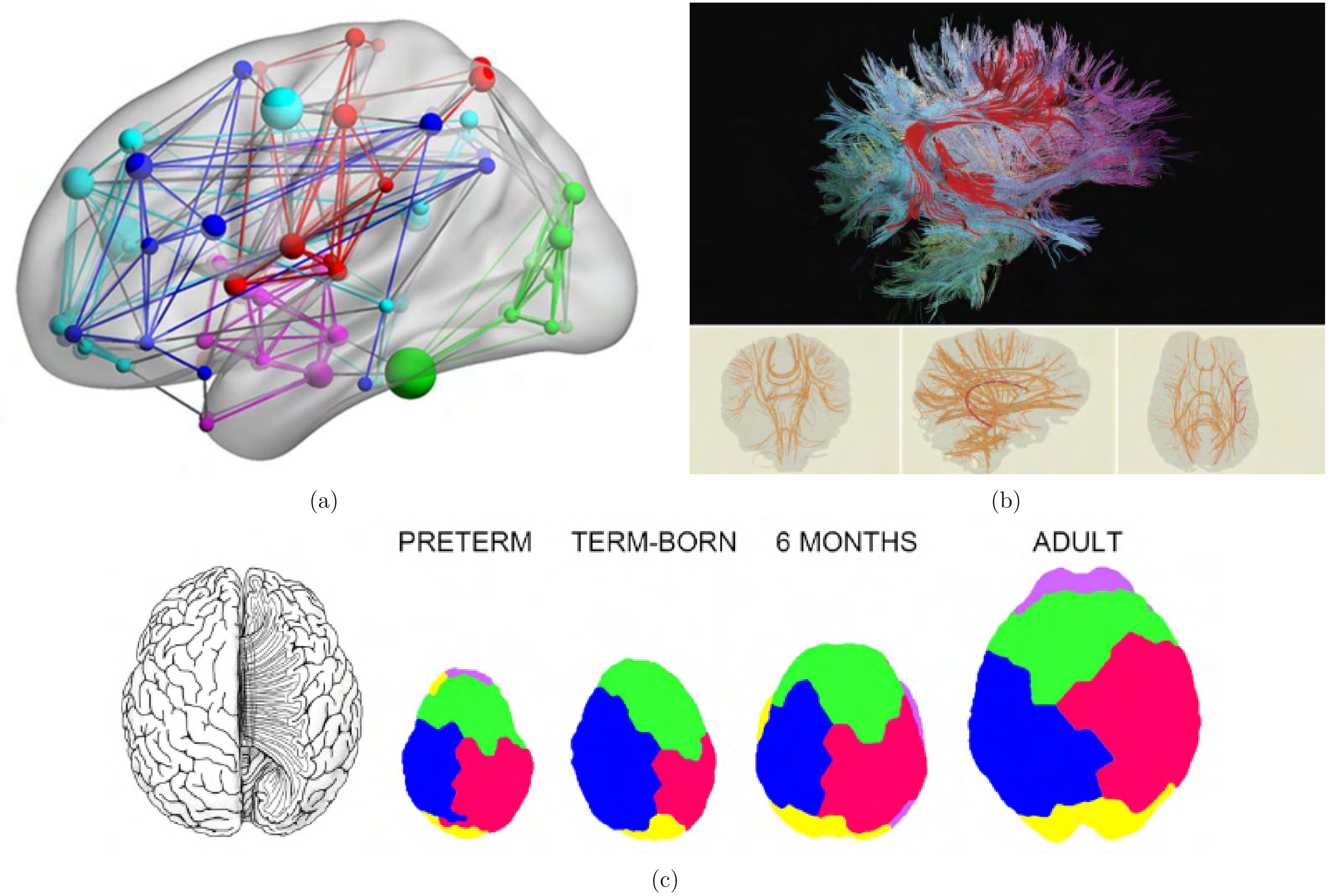


Figure 3.9: Different methods of spatial visualization in connectomics. (a) Node-link diagram of a human connectome from a transversal perspective [XWH13] (b) Tractogram [JDL09]. (c) Segmentation of the cortex [THZ^{+}13].

the aggregated connectivity from, to, or between brain areas - allow experts the comparison of multimodal networks at different levels of hierarchically organized anatomical atlases. *Aggregation Queries* are available in BrainTrawler [GKHB20, GSF$^+$19], a novel task-driven web-framework that incorporates visual analytics methods to explore heterogeneous neurobiological data, including their spatial context. It is the first tool that enables to explore of the genetic and functional characteristics of microcircuits in real-time.

The combination of abstract and spatial approaches preserves the mental map of the neuroscientists and allows them to investigate whether a network topology interacts with the spatial domain. A tool that provides both approaches is Brain Modulyzer [MBB$^+$16], an interactive visualization tool to analyze the correlation between different brain regions when resting or when performing mental tasks. It provides a view for each approach. Anatomical views display spatial information by coloring brain regions based on their correlation strength. Abstract views are used for analyzing patterns in the underlying correlation data while removing anatomical context. InTool [FGT$^+$19] is a web-based tool which provides a user-designed canvas for data visualization and interaction to perform specific exploratory tasks. It permits visualization of the data sets in a dynamic and versatile way using a linked-card approach. Predefined card types offer abstract data representations, a filtering tool, or a set of statistical analysis methods. Another tool is NeuroMap [SBS$^+$13]. It visualizes the brain of the common fruit fly *D. melanogaster* (see Figure 3.10). It renders an interactive two-dimensional graph of the fruit fly's brain and its interconnections in the form of a circuit-style wiring diagram. Anatomical context is provided by partitioning the canvas into compartments that form an abstract representation of actual brain regions. NeuroCave [KZC$^+$17] supports group studies analyses, which are of relevance to examine differences between populations or changes over time within a population. It applies immersive visualization, hierarchical clustering, and dimensionality reduction techniques.

3.2.3 Conclusion

Most previous approaches for visualizing brain connectivity suffer from clutter or loss of anatomical context, while others require multiple views to combine relevant information. Spatial anatomical networks can be visualized as graphs. A simple 3D projection leads to occlusion of nodes, which can be solved in several ways. McGraw's work [McG15] reduces clutter near the midline by representing interhemispheric connectivity separately, resulting in multiple figures. This makes it difficult for the user to see the spatial relationship between the figures. NeuroMap provides solutions to problems such as clutter or loss of anatomical context, but the anatomical layout is created manually. Adaptations to visualize data from other species require extensive reprogramming. This would be very time consuming since each species has a unique hierarchical brain parcellation. Another problem concerning the parcellation is the level of granularity within the hierarchy.

In our work we focus on shape-based metrics [EHNK17] as traditional metrics like edge-crossing minimization are not feasible in large, spatial networks. Shape-based metrics state that a drawing is good if the shape created by the vertex positions is similar to the

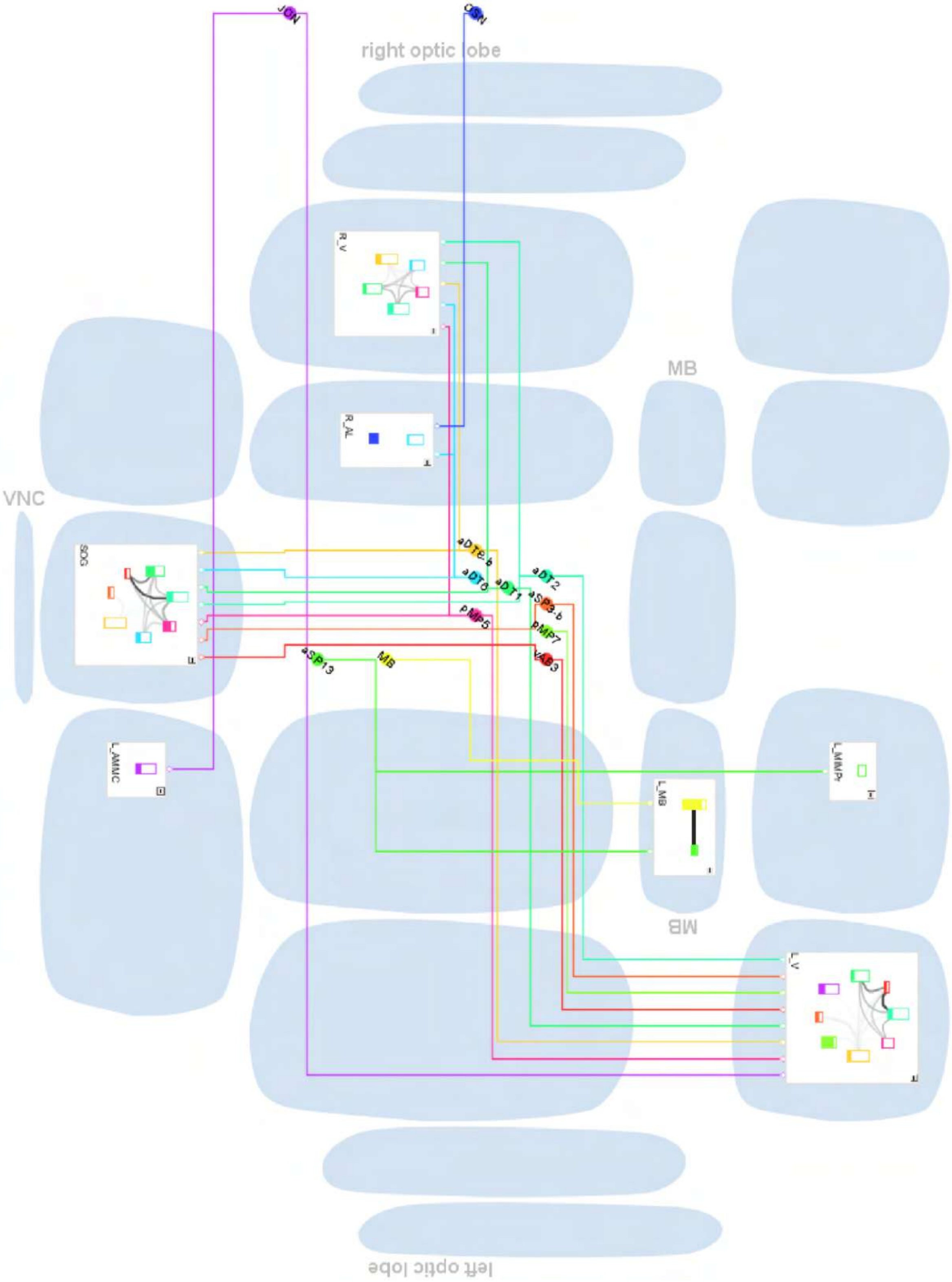


Figure 3.10: Neuromap [SBS+13] emulates an abstract view of the fruit fly's brain.

original graph. The idea of remaining similar to the original graph is naturally essential for spatial layouts. We visualize brain networks in a data-driven manner to reuse the code base for connectivity data from different species (e.g., human, mouse). Our methodology could also be applied in other domains that require graph-based visualizations with spatial context.

CHAPTER 4

Data

To provide a better understanding of our methodology and implementation we start with the description of the data. For loading and visualizing such neuroscientific data we use a specific data format, including naming conventions. The information about the brain hierarchy and the connectivities comes in the form of a human-readable JSON format. Brain regions are labeled by denoting their hemisphere with "R" for right or "L" for left, followed by "_" and an abbreviation of the name as used in the Allen Brain Atlas [All], e.g., L_OLF for left Olfactory areas or R_STR for right Striatum.

To visualize brain networks our tool requires the following two data sets:

- **Hierarchical Brain Parcellation:** It represents the overall information of the species-specific hierarchical partitions of a standard brain. For each region it includes the ID, name, and abbreviation of the name, as well as color, 3D position coordinates, and the volume as voxel count. The hierarchy is organized in a tree-structure. The complete brain represents the top level of the hierarchy, which is subdivided into its sub-regions. This results in each region having a field containing a list of its sub-regions. The data is derived from brain reference atlases, specifically from the *Allen Mouse Brain Common Coordinate Framework* [WDL+20] for the mouse brain, the *Allen Human Reference Atlas* [DRS+16] for the human brain, and the *larval brain platform* [lar] for the *D. melanogaster* larval brain. A scheme of the Hierarchical Representation of Brain Regions comprising the higher hierarchy levels of the *Allen Mouse Brain Common Coordinate Framework* can be seen in Figure 2.2.
- **Brain Network**: This data set contains specific information of the network that should be visualized. It encodes brain regions that are rendered as nodes in the resulting graph. We refer to them as *Network Regions* or *Network Nodes*. Furthermore it encodes connectivity among those regions. Connections are directed

and contain a reference to the ID of their source region and target region, as denoted in the *Hierarchical Brain Parcellation*. Each connection is weighted and has a type. We distinguish between two connectivity types: *Parcellation-derived Connectivity* and *Rendered Connectivity*. The Brain Network data includes connectivities of both types for the same set of *Network Regions*.

- *Rendered Connectivity* contains the brain network that should be visualized. It represents, for example, structural, functional, or genetic networks.

- *Parcellation-derived Connectivity* represents the closeness of brain regions in the anatomical reference space and is used to lay out the graph. Ganglberger [GSF+19] derived this measure from the parcellation of brain regions on a 3D reference space using two different methods:
  - **Neighborhood**: Calculates the number of neighbouring voxels (6-connectivity) between brain regions across all hierarchy levels. The measure is normalized by the total number of voxels of the two brain regions, since otherwise the measure would depend directly on the size of the regions. The localized nature of this connectivity (only neighbouring brain regions are connected) enables graph layouts that retain these local structural relationships between brain regions.
  - **Reciprocal Distance**: Calculates the reciprocal distance between region centers. It is an approximation which can be used if no parcellation is available but yields inferior results.

CHAPTER 5

Methodology

This chapter explains our methodological approach. Section 5.1 describes the requirements that our work has to fulfill. Section 5.2 gives an overview of the individual methodological steps that are carried out to meet the requirements. The remaining sections of this chapter describe these steps in detail.

5.1 Requirements

Based on a long-term collaboration with neuroscientists working on neural networks of humans, mice, and *D. melanogaster* the following requirements for a method to create *Spatial-Data-Driven Layouts* of brain networks have been collectively established together with Ganglberger [GWW+22]:

R1 Anatomically sound *The positioning of the regions in the graph should reflect the anatomical layout, i.e., neighbouring regions of the brain should be represented adjacent in the visualisation.*

R2 Data-driven *The vast number of connections and regions within a brain make manual arrangement of data unfeasible. Our tool should be able to handle data in an automatic and data-driven way.*

R3 Species-independent *Each species has a unique brain parcellation and brain structure that our tool should be able to display regardless of the differences.*

R4 Perspective-independent *Different perspectives, e.g., transversal (from top) and sagittal (from the side) should be supported and stay comparable by maintaining the user's orientation.*

R5 Providing anatomical context *The final visualization should provide sufficient context to facilitate the anatomical localization of a brain network.*

R6 Adaptable with regards to anatomical detail *It should be possible to highlight the anatomical detail of the graph according to information density, i.e., show more anatomical detail for highly connected regions or networks with more than one node per region exceeding the resolution of the hierarchical parcellation, or by the region's anatomical size, i.e., anatomical detail is evenly distributed over regions with equal size.*

R7 Consistent in spatial organization with respect to changes *The layouting should be stable concerning changes in the selection of visualized network nodes and brain regions, and therefore, the mental map of the neuroscientist should be retained.*

R8 Overlap-efficient *Overlap of nodes and edges should be minimized.*

5.2 Approach

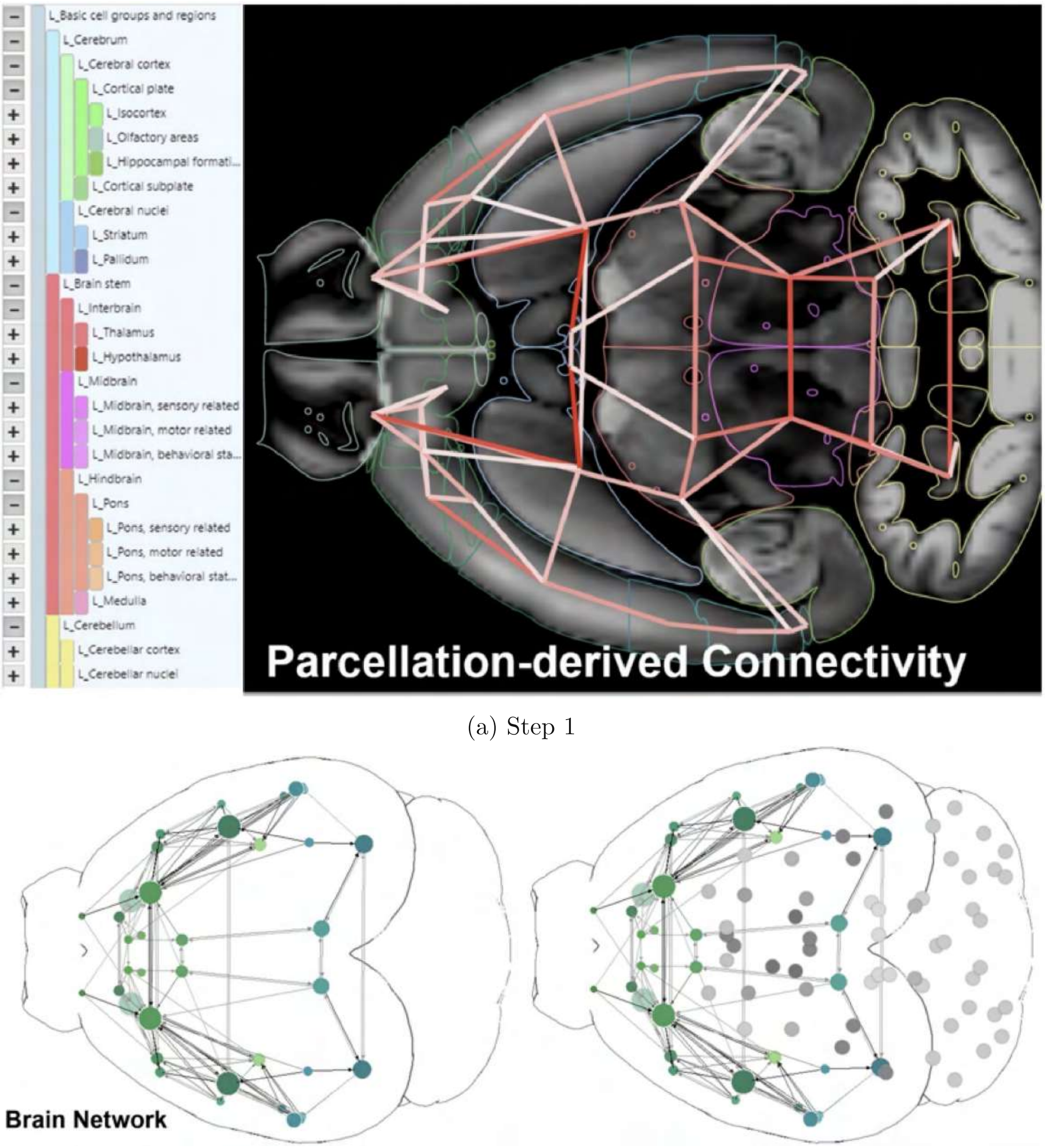
When visualizing three-dimensional structures on a two-dimensional surface, we often struggle with obscuring information that we need to represent. The main idea is to use the *Parcellation-derived Connectivity* that describes the anatomical proximity of each brain region to layout our network. With this approach we can find a layout that reduces the overlap of nodes and achieves a uniform distribution of nodes while preserving the anatomical relationships. After laying out the nodes, the *Parcellation-derived Connectivity* is exchanged for the connections of the *Rendered Connectivity*. For further visual guidance we underlay the nodes with a colored background representing their associated parcellation.
The following approach was developed together with Ganglberger and is also described in our work [GWW$^+$22]:

Step 1: Preprocessing We start by loading and preprocessing the required data. This includes the hierarchical parcellation, the *Rendered Connectivity*, and the *Parcellation-derived Connectivity*, which will be used in later steps to lay out a brain network. Details about the data are described in Chapter 4. Missing regions in the hierarchy can be taken into account.

Step 2: 3D Projection To position the nodes on the plane, two of the three coordinates of their location are selected, depending on the desired perspective (sagittal or transversal) (R4).

Step 3: Providing Context Partial networks can be difficult to compare as they lack common parts. To make all networks comparable and simplify orientation, it is possible to add further context to the partial networks (R5, R7). Partial networks can be extended by further nodes, which we refer to as *Context Nodes*. These nodes are not visualized themselves but are later included in the background described in Step 5.

Step 4: Layouting This step is crucial for the final result as the *Network Nodes* and *Context Nodes* of the graph are positioned. This positioning should reflect the anatomical

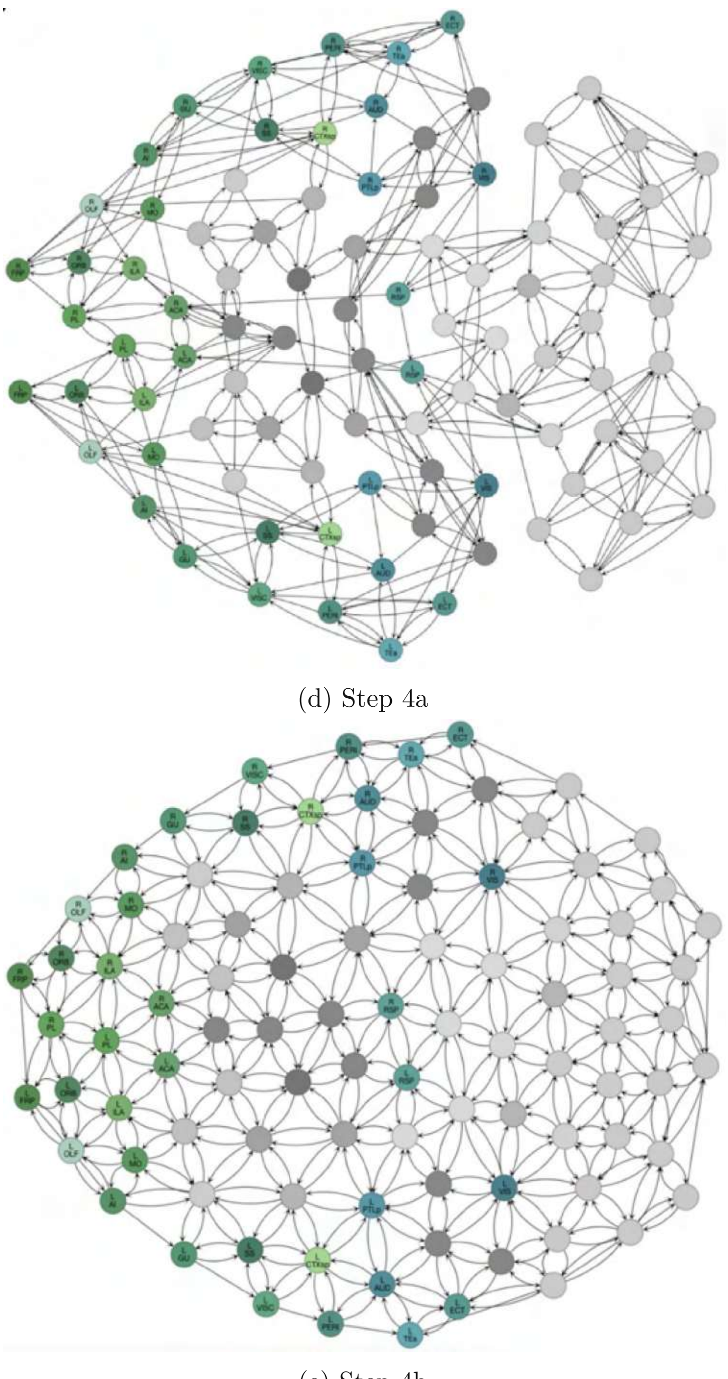


(a) Step 1

(b) Step 2

(c) Step 3

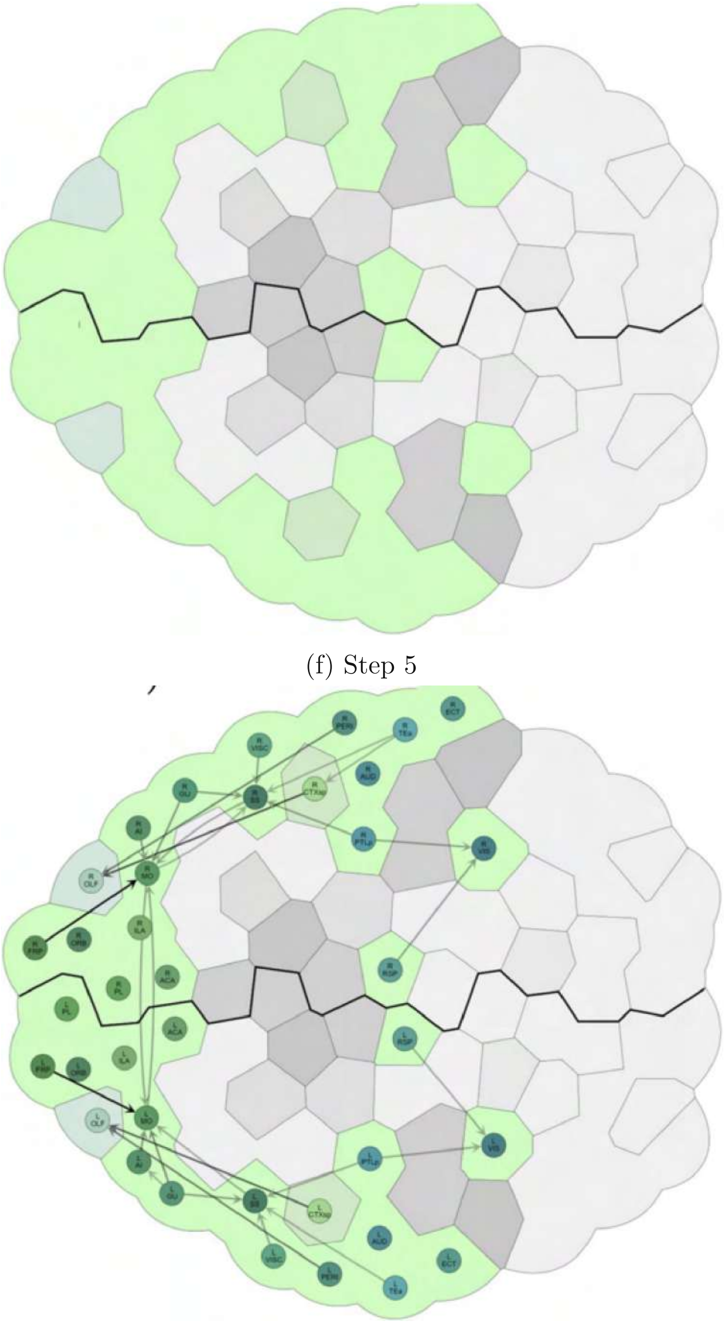
Figure 5.1: Principal steps to generate *Spatial-Data-Driven Layouts*. (a) Step 1: Preprocessing. (b) Step 2: 3D projection (here: transversal perspective). (c) Step 3: Providing context (gray nodes). Figure by Ganglberger et al. [GWW$^+$22].



(d) Step 4a

(e) Step 4b

Figure 5.1: Principal steps to generate *Spatial-Data-Driven Layouts* (cont.). Step 4 - Layouting. (d) *Anatomical Layout* using *Parcellation-derived Connectivity*, (e) *Aesthetic Layout* using triangulation. Figure by Ganglberger et al. [GWW$^+$22].



(f) Step 5

(g) Step 6

Figure 5.1: Principal steps to generate *Spatial-Data-Driven Layouts* (cont.). (f) Step 5 - *Parcellation Background*. (g) Step 6 - Network Rendering. Figure by Ganglberger et al. [GWW+22].

spatial localization of the brain regions as accurately as possible and comply with the aesthetic rules (R1, R8). The graph nodes are arranged based on the *Parcellation-derived Connectivity*, which reflects the anatomical proximity of brain regions using a force-directed layouting algorithm. We call this layout *Anatomical Layout*. In some cases, another layout step using the Delaunay triangulation called *Aesthetic Layout* is applied to achieve an even node distribution. The layout process is applicable to multiple data sets and species (R2, R3).

Step 5: Parcellation Background Nodes representing sub-regions of the same parcellation should be visually clustered. The parcellations are displayed as colored planes, which surround the corresponding nodes. The idea is to facilitate the orientation of the user. The user can control the level of detail of the parcellation (R6).

Step 6: Connectivity This final step renders the edges representing the *Rendered Connectivity*. A set of different edge visualization techniques is applicable.

A visual representation of the approach is seen in Figure 5.1. In the following sections each step is described in detail.

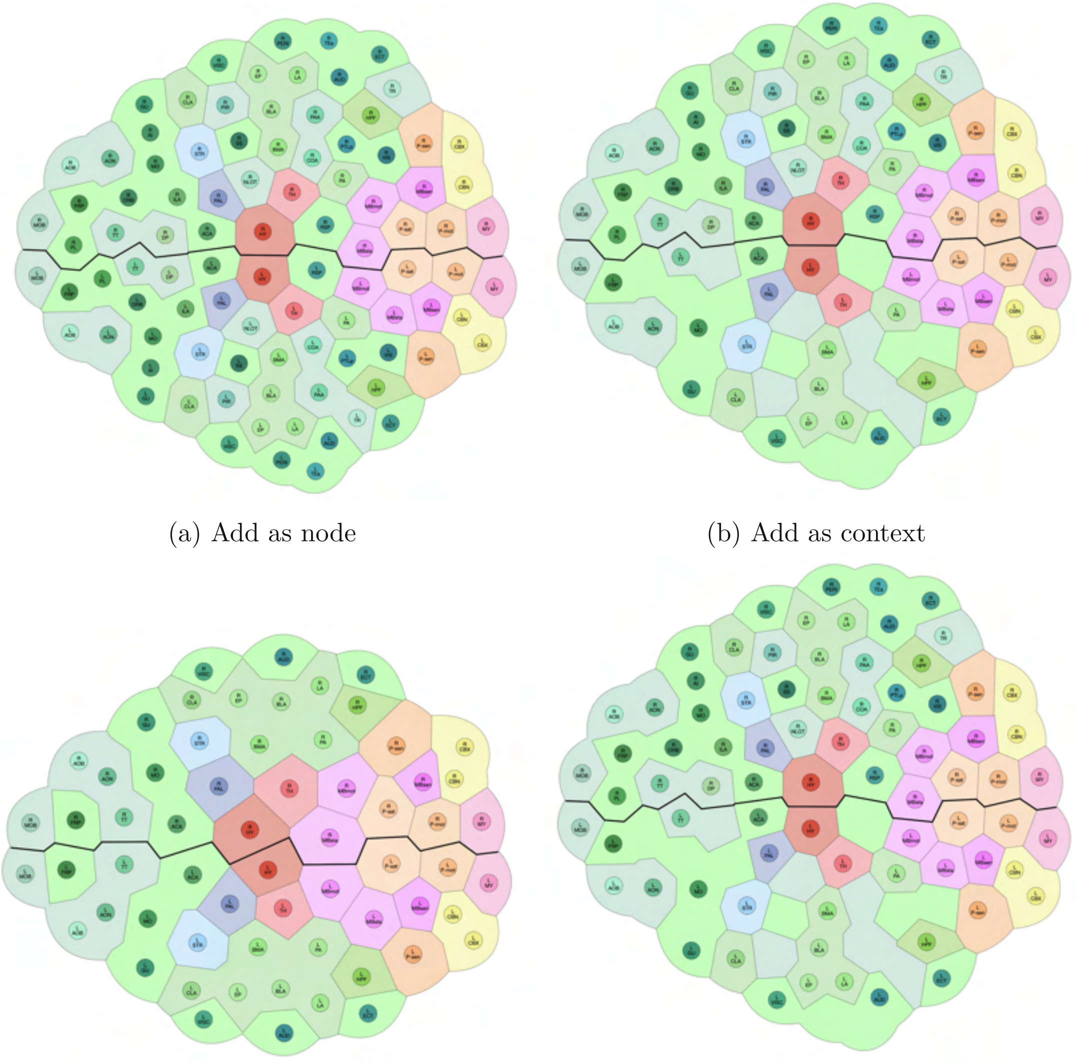
5.3 Preprocessing

First, we load our input data as described in Chapter 4 depending on the species. Nodes encode the brain regions, containing information like the regions' name and color. Edges encode the connectivity, including the reference to the source and target nodes, and are weighted. The anatomical connection used for layouting is derived from the *Parcellation-derived Connectivity*.

Some brain atlases provide unbalanced regions, i.e., corresponding regions of the left and right brain hemispheres are not pairwise available. In this case, the user can decide how to handle such regions having the following possibilities:

- Add as node: The missing counterpart of the region is included in the network as node.
- Add as context: The missing counterpart of the region is included in the *Parcellation Background*.
- Remove: The node which has a missing counterpart is removed from the network.
- Ignore: The network stays as it is.

Ignoring unbalanced regions may naturally lead to an asymmetric appearance of the graph, which is especially visible in the transversal view. An example is given in Figure 5.2.



(a) Add as node

(b) Add as context

(c) Remove

(d) Ignore

Figure 5.2: Different options for unbalanced *Network Regions* displayed on a mouse brain network with 78 regions from the transversal perspective. 10 regions from the left isocortex (light green) and six from the left olfactory regions (olive) were removed.

5.4 3D Projection and Anatomical planes

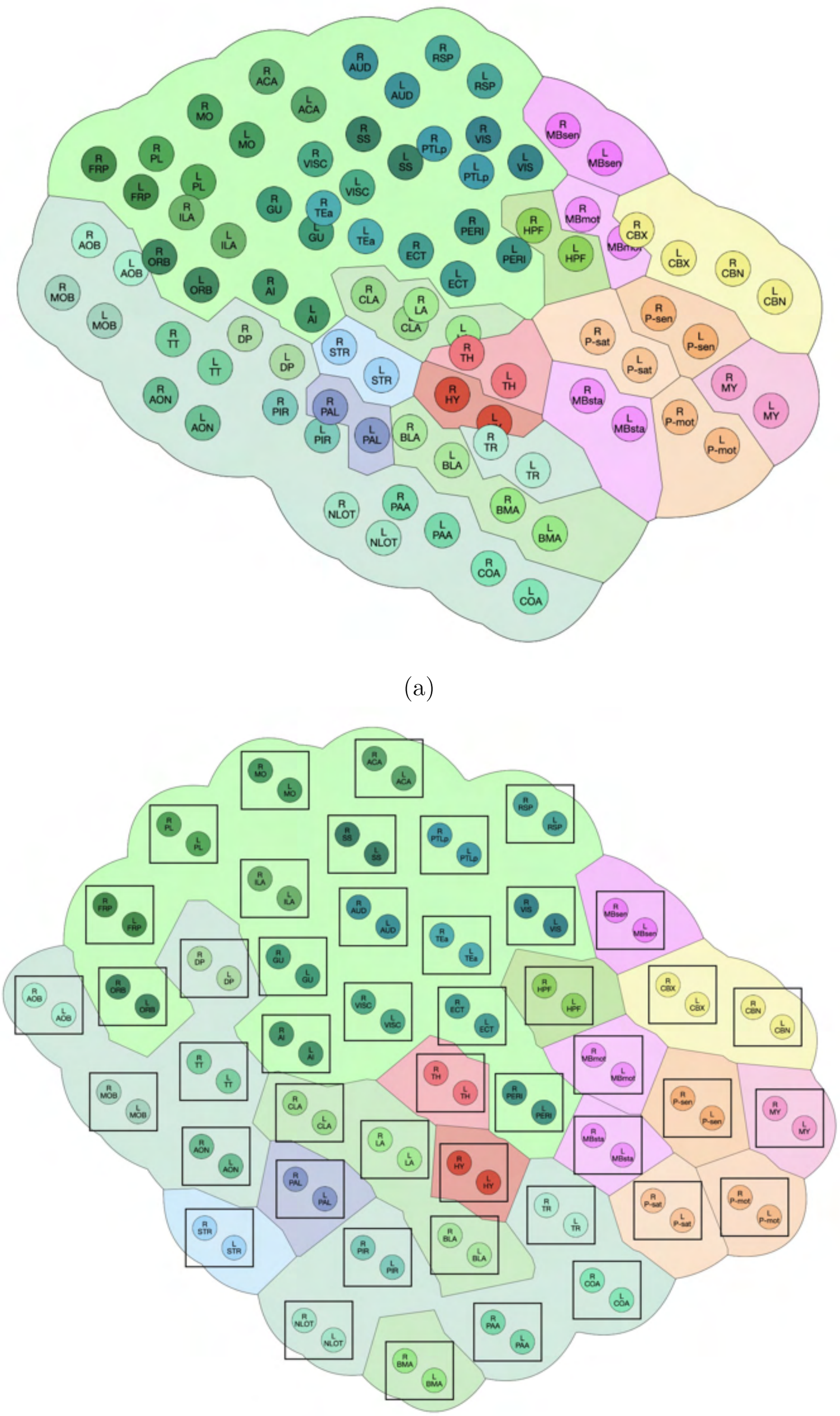
Our application allows to view the network from either the transversal perspective or sagittal perspective (R4). Each region has a 3D coordinate. Based on the desired perspective we select two dimensions (e.g., x- and y-coordinate or y- and z-coordinate) of each region's 3D position to place a node representing the region on the canvas.
From the transversal perspective the two hemispheres are visualized next to each other. In the sagittal perspective nodes that represent corresponding regions of the left and right hemisphere overlap due to the symmetry of the brain (Figure 5.3(a)). To solve this issue we set the two-dimensional coordinate of the position of each node belonging to the right hemisphere to the one of the left side and shift it diagonally by a fixed distance. An advantage of this approach is that two related nodes are evenly distributed and their affiliation is quickly recognized by the viewer.

5.5 Layouting

The *Parcellation-derived Connectivity* represents the local closeness of regions. Applying a force-directed layout algorithm with this connectivity leads to an ordering of nodes that reflects the anatomical and spatial structure of the brain (R1, R7). Another benefit of force-directed layouting is the minimization of node overlap resulting from the 3D projection of the brain network (R8).
Rippberger provided preliminary work for our use case in her bachelor thesis [Rip19]. She evaluated multiple layouting algorithms for graph data concerning properties like symmetry, node overlapping, and anatomical resemblance. The goal of her work was to find an algorithm that regenerates the anatomical structure of the brain visualized without any hard-coded constraints. A user study was conducted to evaluate how the resulting graphs were perceived regarding symmetry, anatomical layout, reproducibility, and visual traceability. Several different selections of regions were tested. The force-directed layouting algorithm by Dogrusoz et al. [DGC+09] called CoSE-Bilkent (Compound Spring Embedder) outperformed the other tested algorithms regarding all properties. Figure 5.4 shows two results generated by Rippberger using the CoSE-Bilkent algorithm. Based on Rippbergers findings, we use the CoSE-Bilkent algorithm that supports compound graphs (nested structures) and varying (non-uniform) node dimensions. Since it accounts for the initial positions of the nodes, it also leads to stable results. We now expand on the work of Rippberger.

5.5.1 Parameters

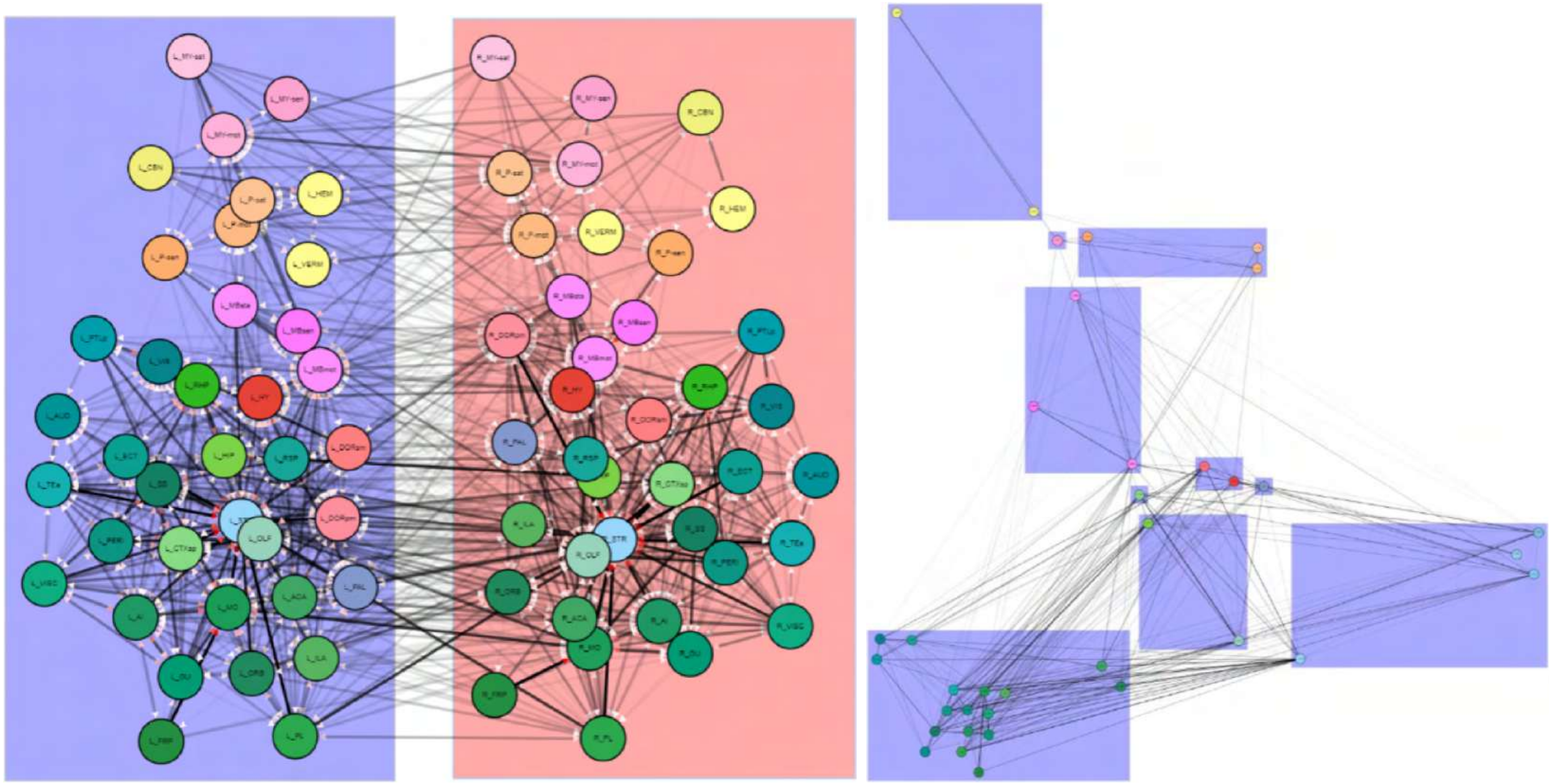
Depending on the specific algorithm and its implementation, force-directed layout algorithms provide a variety of parameters the user can specify to control the applied forces and influence the results. Force-directed layout algorithms reflect forces in a physical system with edges acting as springs and nodes holding electrical charges. Therefore, the parameters usually represent, among other things, node repulsion, spring or edge length,



(a)

(b)

Figure 5.3: Effects of compound nodes on a network of a mouse brain with 90 regions from the sagittal perspective. In both figures the layout algorithm is run first, followed by a shift to align left and right region pairs to each other. (a) Result without compound nodes. This leads to overlap of several nodes. (b) Regions of left and right hemisphere encapsulated in a compound node (black boxes).



(a) Compound nodes splitting the nodes into the left and right side of the brain.

(b) Nodes with the same parent rendered as a compound.

Figure 5.4: Preliminary work by Rippberger [Rip19] applying the CoSE-Bilkent layout.

and edge elasticity. Effects of these parameters can be seen in Figure 5.5. We denote forces as strong if the effect of the force-directed layout changes the result significantly, i.e., changing the spatial adjacency of the *Network Nodes* or changing the overall shape of the graph (see the example where edge elasticity is set to ten in Figure 5.5). Otherwise, we denote forces as weak.

CoSE-Bilkent provides a set of parameters the user can customize to adjust the forces and the resulting layout of the graph. As it is a spring-embedder it simulates the graph as a system of mass particles and tries to minimize the energy of this physical system. The edges act as springs and the nodes hold electrical charges. The following parameters reflect forces in this physical system and are relevant for meeting the requirements of our work:

- *NodeRepulsion* - represents the electrical charge of a node. The higher the value is, the stronger the repulsion between the nodes. A typical effect of increasing the node repulsion is an upscaling of the drawing.
- *EdgeElasticity* - represents the spring's stiffness. This factor is used to determine how strongly edges pull and push at the nodes they connect.
- *IdealEdgeLength* - represents the natural length of non-nested edges, i.e., edges not contained in a compound node. If nodes are connected by an edge, one can treat the edge as a spring that has a natural length. If the nodes are nearer than this length, they are push apart if they are farther away than the length, they are pulled together.
- *InitialEnergyOnIncremental* - represents the reciprocal cooling factor for the incremental layout. The cooling factor helps in controlling how layouts evolve over time. A small positive cooling factor reduces the movement of nodes quickly or abruptly, while a large cooling factor allows for a smoother step by step layout refinement at the cost of more iterations. As the *InitialEnergyOnIncremental* is its reciprocal value, a small value creates the smoother step by step layout refinement.
- *Gravity* - represents a general force field in the center that pulls nodes together, so that nodes are not too loosely scattered.
- *GravityRange* - represents the range of the gravity. compounds.

A fitting set of parameters was heuristically evaluated. For a more efficient testing setup we applied sliders to change individual parameters within reasonable boundaries. The boundaries were roughly defined between a value that produced hardly noticeable differences and a second value that already strongly distorted the layout.
Overall, the parameters *NodeRepulsion* and *IdealEdgeLength* control the distance between the nodes. Increasing those values generates a larger empty area around the nodes and improves the visual quality. The strongest effect has the *EdgeElasticity*. The

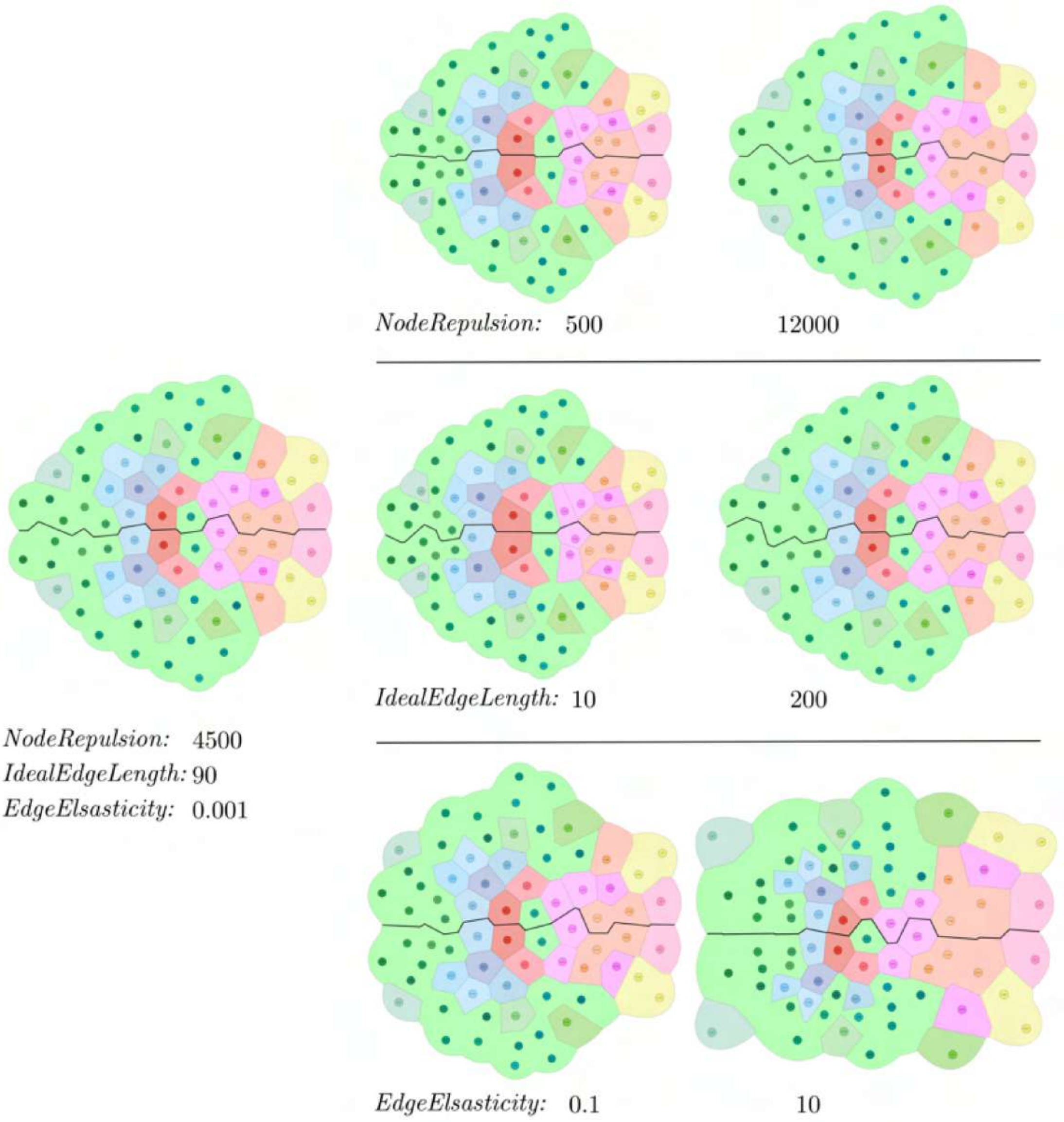


Figure 5.5: The effects of changing the layout parameters. The left most figure shows the reference case. On the right side one of the values was changed in each case. In the top row the node repulsion was changed, in the middle row the ideal edge length was modified, and in the bottom row the effects of changing the edge elasticity can be seen.

increase of its value pulls regions with strong connections of the *Parcellation-derived Connectivity* together. We concluded that high values help to reach an anatomically correct neighborhood of regions but also create node overlaps and alter the shape of the brain as seen in Figure 5.5. Hence, we evaluated the smallest value for the *EdgeElasticity* parameter, that still creates an anatomically correct neighborhood of regions. The effect of the parameters most relevant to us can be seen in Figure 5.5.

Our approach applies the CoSE-Bilkent algorithm twice with different sets of parameters and connections. We therefore distinguish between two layouts:

- *Anatomical Layout*: This is the main layout step. It should produce the anatomical representation as accurately as possible after the 3D projection. Neighboring brain regions should also be adjacent in the graph as nodes. A view that resembles a cross-section is desired (e.g., Figure 2.2 - 2.4).
- *Aesthetic Layout*: Depending on the shape and density of regions in the brain, network forces of the *Anatomical Layout* have to be strong enough to pull nodes of neighboring regions together, while repelling nodes of regions which are not adjacent. Such strong forces, however, can lead to overlaps during the positioning of the nodes. We apply weak forces on a connectivity, that represents the triangulation of the *Network Nodes* to resolve overlaps without changing the without changing the spatial adjacency of the *Network Nodes* that results from the *Anatomical Layout*.

In the following paragraphs we describe the layouts in more detail.

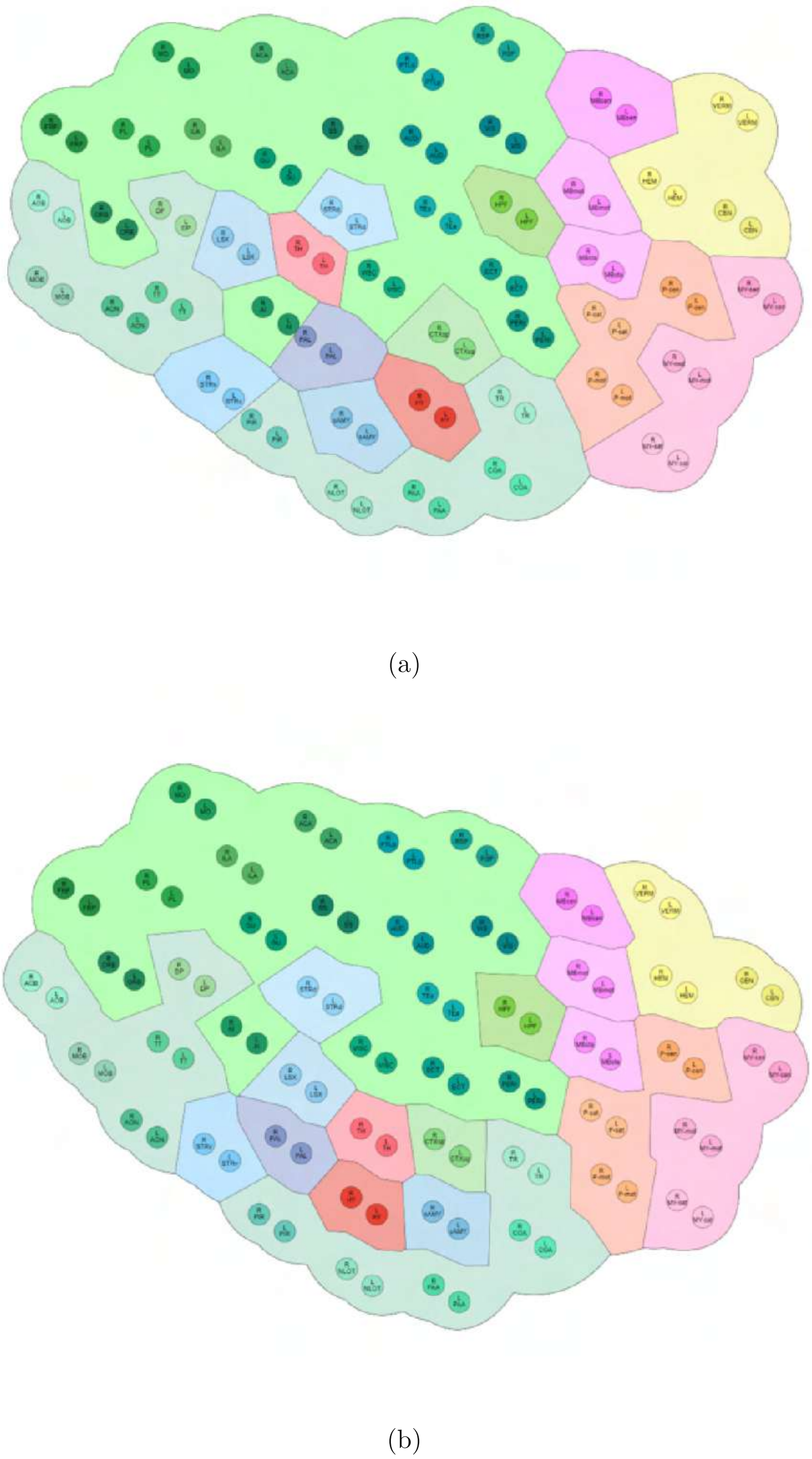
5.5.2 Anatomical Layout

Using sliders, we manipulated the values of each parameter as previously described to observe the effects. We tried to reproduce the subjectively best results, i.e., reflecting the spatial anatomy of the brain. The results were evaluated under the supervision of Florian Ganglberger. We quickly realized, that for different characteristics a distinct set of parameters had to be determined. Depending on the perspective and the species a varying number of overlap occurs due to the overall size and shape of the brain network and its density of nodes. By observation we can provide some guidance how to choose which parameter set in which case.
The parameters for the *D. melanogaster* larval brain need to be set to low values, to maintain the shape. Weak forces in the algorithm are sufficient in this case to layout the graph, such that it reflects the anatomical and spatial structure of the brain and avoids node overlap. *Gravity* was set to a low value for the *D. melanogaster* larval brain network since the brain has a long, T-like shape (see Figure 2.4), which is distorted by the force field that pulls peripheral nodes towards the center. The parameter *InitialEnergyOnIncremental* is set to a low value to keep the graph stable.
Dense brain structures, like the human brain or the mouse brain from sagittal perspective, require strong forces to reach an anatomically correct neighborhood of regions, i.e., to

pull nodes of anatomically adjacent regions together and to repel those that are further apart. This can be achieved by setting a higher value for the *InitialEnergyOnIncremental* or the *EdgeElasticity* parameter for the mouse and human networks compared to the one for the networks of *D. melanogaster*. An example is shown in Figure 5.6(a). In the sagittal perspective of the mouse brain network the peripheral regions of the cerebral cortex (green) as well as the pallidum (purple) are placed in-between the central regions, thalamus (bright red), and hypothalamus (red) due to simple mapping of 3D to 2D. In this case it is desirable to see the nodes positioned similar to a cross-section (see Figure 2.3), where the sub-regions of the cortex are positioned in the periphery of the brain. This requires stronger forces in the layout algorithm. This is achieved by increasing the value for the *EdgeElasticity*. For this reason the forces of the algorithm have to be adjusted accordingly.
The CoSE-Bilkent algorithm we use for layouting does not take into account the weighting of each connection. This means that even insignificant connections are given equal consideration in the layout. To compensate for this shortcoming we decided to filter the *Parcellation-derived Connectivity* to include only the four strongest outgoing connections for each node. This value was determined heuristically.

5.5.3 Aesthetic Layout

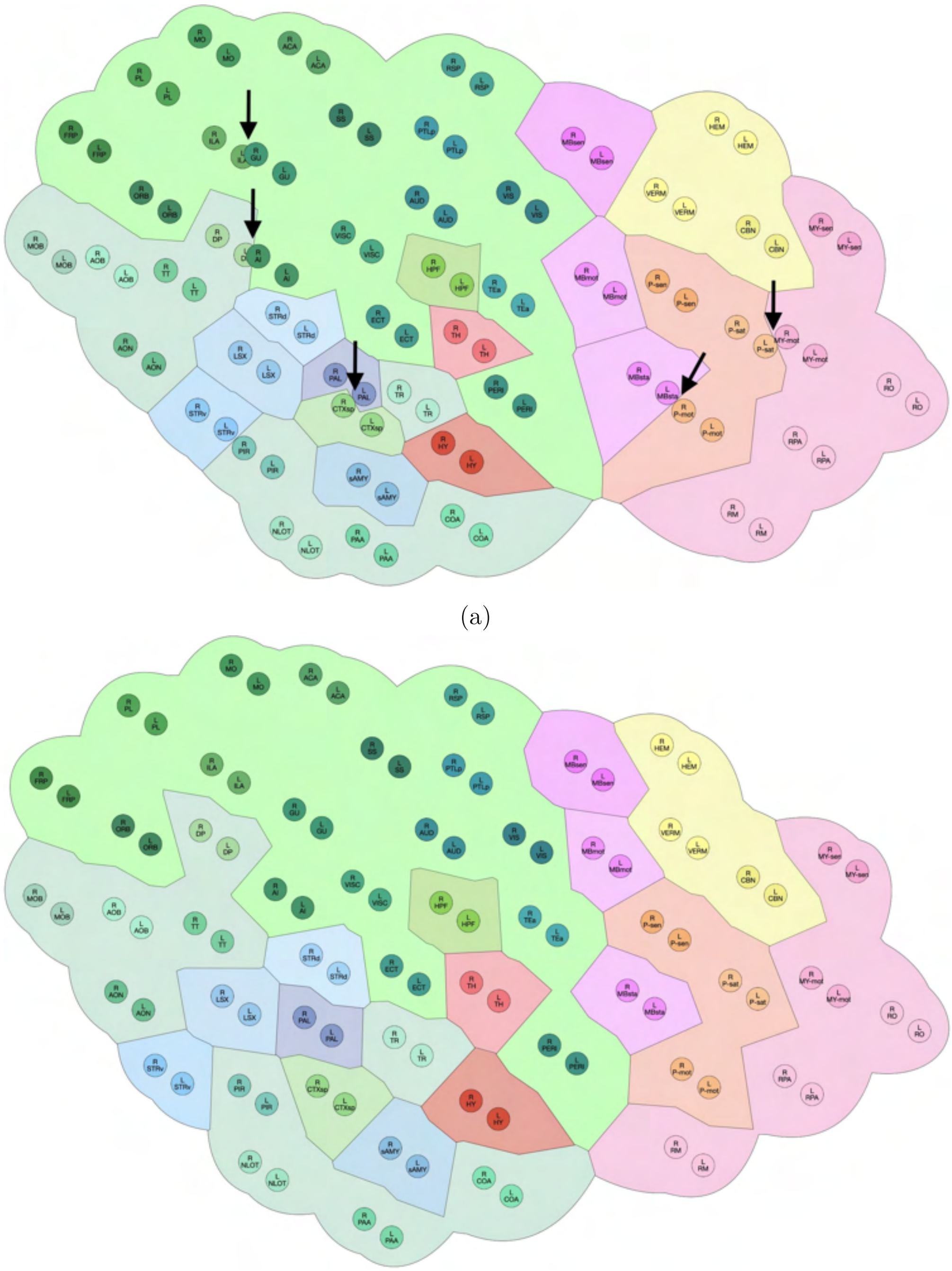
A challenge with dense networks is that regions that are neighbors should be pulled together, while peripheral regions should be pulled away. Applying strong forces resolves this issue. However, one problem that occurs if forces are too strong is that the distribution of the nodes is uneven, eventually leading to overlaps (see Figure 5.7(a)). Our solution for this problem is to introduce another layout step. We denote this step as *Aesthetic Layout* as its aim is to satisfy the aesthetic heuristics of evenly distributing nodes and minimizing node overlap to improve readability and understanding. To achieve an even distribution of nodes while maintaining the spatial relationship of the nodes the *Aesthetic Layout* requires a set of edges connecting nodes that have the shortest distance to each other, as these are most likely to be affected by overlaps. The Delaunay triangulation satisfies this property. We calculate the triangulation for the *Network Nodes* and use it to layout the graph once more with a set of parameters creating weak forces. Due to the properties of the Delaunay triangulation, the spatial anatomy of the brain created by the *Anatomical Layout* is preserved, while an even distribution of the nodes to minimize the overlaps is amplified (Figure 5.7(b)). The *Aesthetic Layout* can be repeated multiple times. Running a high number of iterations or applying strong forces in the *Aesthetic Layout* approximates the shape of the graph to a disc. We heuristically evaluated that three iterations for dense networks, i.e., mouse and human brain networks, are sufficient to reach an even node distribution and to reduce node overlaps, while preserving the shape of the brain. For networks of the *D. melanogaster* larval brain we do not apply the *Aesthetic Layout* at all. As mentioned in Section 5.5.2 this brain does not require strong forces in the *Anatomical Layout*, resulting in satisfactory results concerning node overlaps and distribution. Furthermore, running the *Anatomical Layout* distorts the distinct shape of the *D. melanogaster* larval brain.



(a)

(b)

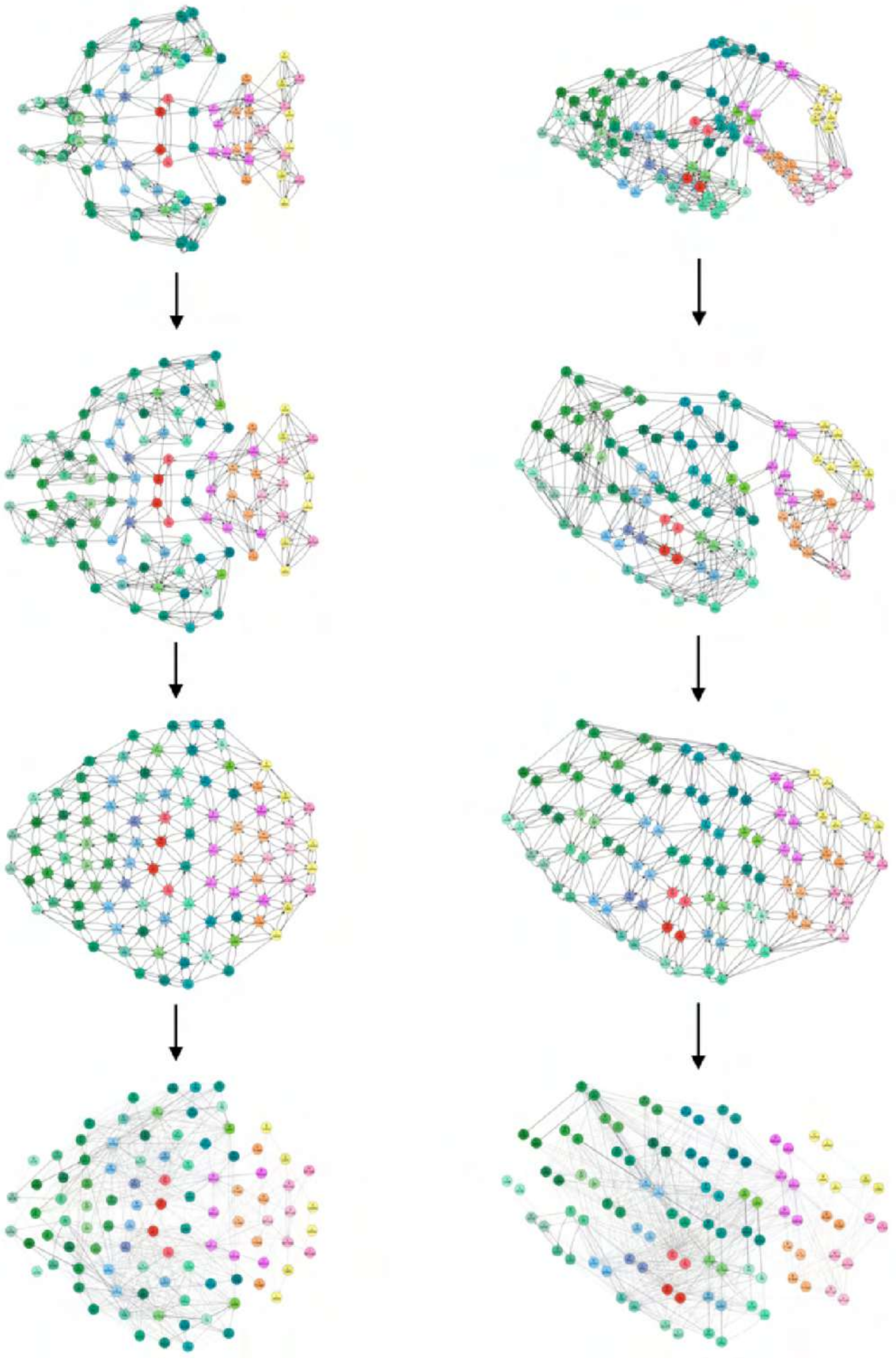
Figure 5.6: (a) Weak parameters for the layouting algorithm are set (*EdgeElasticity* = 0.01). The hypothalamus (red) and thalamus (bright red) should be positioned next to to each other, but are separated by the pallidum (purple) and the peripheral cortex regions (green). (b) A three times higher value for the parameter (*EdgeElasticity* = 0.03) shifts the neighbouring regions towards each other.



(a)

(b)

Figure 5.7: Effects of the *Aesthetic Layout* on a sagittal projected network of the mouse brain with 104 regions. (a) The network after running the *Anatomical Layout*. Overlaps of nodes are marked by black arrows. (b) The network after running the *Anatomical Layout*, followed by iterating the *Aesthetic Layout* three times. The overlaps are eliminated.



(a) Transversal perspective

(b) Sagittal perspective

Figure 5.8: Layouting steps of the mouse brain network from transversal and sagittal perspectives. The first row shows the simple projection from the 3D data to the 2D plane with *Parcellation-derived Connectivity*. The second row shows the result after applying the *Anatomical Layout* with *Parcellation-derived Connectivity*. The third row depicts the result after applying the *Aesthetic Layout* using triangulation. Finally, the last row represents the result after exchanging the triangulation edges with the *Rendered Connectivity*.

Finally, Figure 5.8 shows the layouting steps on a network of a mouse brain for the transversal and sagittal perspective. The parameters for the *Anatomical Layout* are found in the Appendix for networks of the mouse, human, and *D. melanogaster* larval brains from transversal and sagittal perspective. The parameters for the *Aesthetic Layout* are also listed in the Appendix. They are not dependent on the species or perspective.

5.5.4 Compounds

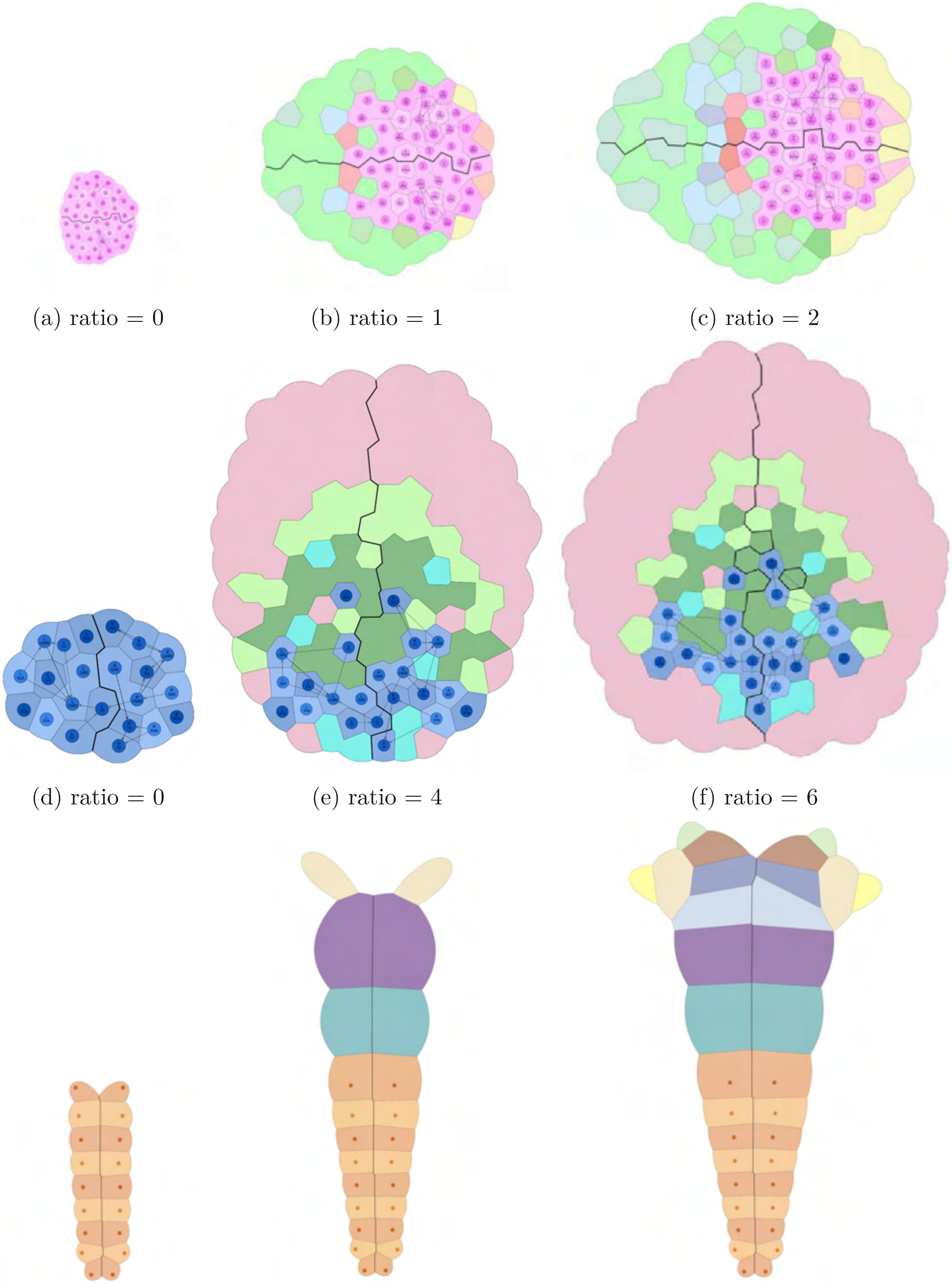
The force-directed layout algorithm CoSE-Bilkent supports compound nodes. The compounds are crucial to group corresponding nodes of the left and right hemisphere in the sagittal perspective, as described in Section 5.4. Running the force-directed layout algorithm would change the new position and we would lose the even distribution of the corresponding left and right *Network Nodes*. Running the layouting algorithm first and then shifting would possibly create new overlaps of nodes (Figure 5.3(a)). To overcome this issue we use the compound nodes to encapsulate the left and right node of each region. The compound nodes are taken into account by the CoSE-Bilkent algorithm, such that overlaps of compound nodes, including their contained *Network Nodes*, are reduced and the shifted positions of the*Network Nodes* are preserved (Figure 5.3(b)). The final result is a balanced appearance, which helps the users to orientate themselves with minimized overlap. The fine-tuning can be done using parameters CoSE-Bilkent specifically provides for compound nodes:

- *NestingFactor* - represents the ideal edge length for nested edges, i.e., that are contained in a compound node.
- *GravityCompound* - represents the gravity for compounds.
- *GravityRangeCompound* - represents the range of the gravity for compounds.

5.6 Context

Neuroscientists may want to visualize only a specific part of the brain. Such sub-networks often lack connections to the entire brain. This makes it difficult to classify their spatial position and relationship. The user can display other regions as context to facilitate comparison between sub-networks (R5). The idea is to embed information in the visualization by positioning regions not contained in the sub-network in the graph without rendering them as nodes. Instead, they are included in the layout process and the *Parcellation Background*, serving as orientation without obscuring the view of the network. We refer to these nodes as *Context Nodes*. The user can control the number and display of context by adjusting two parameters (R6):

- *Context Ratio*: A numeric variable to adjust the amount of context in relation to the number of *Network Nodes* (see Figure 5.9). A value of r will generate a background including r times as many *Context Nodes* as *Network Nodes*, i.e. setting



(a) ratio = 0 (b) ratio = 1 (c) ratio = 2

(d) ratio = 0 (e) ratio = 4 (f) ratio = 6

(g) ratio = 0 (h) ratio = 0.5 (i) ratio = 1

Figure 5.9: Examples of different context ratios for sub-networks. (a)-(c) display a network of a mouse brain, (d)-(f) display a network of a human brain and (g)-(f) show a network of the fruit-fly larval-brain.

this parameter to 0 will only layout and use *Network Nodes* without further context. Setting this parameter to 1 will include as many *Context Nodes* as *Network Nodes*.

- *Gray-scale*: A boolean variable to control the coloring of the *Context Regions*. The user can decide if they want the *Context Regions* depicted in colors corresponding to the Allen Brain Atlas or in gray-scale, to keep the focus on the network.

The pseudocode *CalculateContextRegions* (Algorithm 5.1) describes the main routine to calculate the *Context Regions*. The basic idea of the algorithm is to calculate a set of *Context Regions* C of the size corresponding to the *Context Ratio* r. The *Context Regions* are not part of the sub-network that is being visualized. The value n is the number of *Context Regions* the algorithm calculates and is based on the *Context Ratio* r. As the brain is symmetric we recursively calculate a number of n_l *Context Regions* of the left hemisphere, denoted as C_l. The corresponding regions of the right hemisphere C_r are added afterwards.

The recursive sub-procedure to find the *Context Regions* of the left hemisphere C_l is described by the pseudocode *CalculateLeftContextRegions* (Algorithm 5.2). The main idea is to replace the anatomically largest regions with their sub-regions until the size of C_l reaches n_l.

To achieve this, the parcellations in $\mathcal{C}_l$ are sorted by their size in descending order to set their priority. Starting with the largest parcellation we exclude the *Network Nodes* N from its sub-regions SC. If there are no subregions left the algorithm continues with the next region in C_l. We calculate the updated size of the *Context Node* n_l' if the parcellation c is exchanged with its subregions SC. If n_l' exceeds n_l then only the largest subregions are kept in the set, such that exactly n_l regions in C_l are reached after the exchange. Finally the parent is replaced by its subregions SC in the set C_l. The algorithm recurses with

Algorithm 5.1: CalculateContextRegions

```
input  : An integer r representing the ratio,
         a list N containing Network Regions,
         a map B containing the brain hierarchy.
output : A list C containing the context regions.
n ← |N| * r;
n_l ← ⌊n/2⌋;
if n_l ≤ 0 then
    return ∅;
end
C_l ← CalculateLeftContextRegions(n_l, N, B.subregions[0]) ;
  // B.subregions[0] represents the left hemisphere
C_r ← GetRegionsFromOppositeHemisphere(C_l);
C ← C_l ∪ C_r;
return C;
```

Algorithm 5.2: CalculateLeftContextRegions

```
input  : An integer $\mathsf{n}_l$ representing the number of left *Context Regions*,
         a list $\mathsf{N}$ containing *Network Regions*,
         a list $\mathsf{C}_l$ containing the left *Context Regions*.
output : A list $\mathsf{C}_l$ containing the context regions.
if $|\mathsf{C}_l| \geq \mathsf{n}_l$ then
    return $\mathsf{C}_l$;
end
sort $\mathsf{C}_l$ by size;
foreach $\mathsf{c} \in \mathsf{C}_l$ do
    $\mathsf{SC} \leftarrow \mathsf{c}.subregions$;
    $\mathsf{SC} \leftarrow \mathsf{SC} \backslash \mathsf{N}$ ;        // Remove Network Regions from context
    if $\mathsf{SC} = \emptyset$ then
        continue;
    end
    $\mathsf{n}_l' \leftarrow |\mathsf{SC}| + |\mathsf{C}_l| - 1$ ;
    if $\mathsf{n}_l' > \mathsf{n}_l$ then
        $\mathsf{m} \leftarrow \mathsf{n}_l - |\mathsf{C}_l| + 1$;
        $\mathsf{SC} \leftarrow$ get $\mathsf{m}$ largest regions in $\mathsf{SC}$;
    end
    $\mathsf{C}_l \leftarrow \mathsf{C}_l \backslash \{\mathsf{c}\}$;
    $\mathsf{C}_l \leftarrow \mathsf{C}_l \cup \mathsf{SC}$;
    return CalculateLeftContextRegions($\mathsf{n}_l, \mathsf{N}, \mathsf{C}_l$);
end
return $\mathsf{C}_l$;
```

the updated C_l. The sub-routine terminates in two cases: (1) The size of C_l corresponds n_l or (2) the regions in C_l have no more subregions, excluding *Network Regions*. The *Gray-scale* parameter is later considered while drawing the *Parcellation Background* (see Section 5.7).

After calculating the *Context Regions*, additional connections in-between *Context Regions* as well as connections between *Context* and *Network Regions* are added to the *Parcellation-Derived Connectivity*. By this, the layout algorithm also takes the *Context Nodes* into account, which leads to a more natural appearance of the resulting layout than if the nodes are not considered by the algorithm.

5.7 Parcellation

The networks our tool visualizes are based on the hierarchical anatomy of brain parcellations. Brain parcellations define distinct partitions in the brain, that cover multiple discontinuous but closely interacting regions. We apply distinct colors to distinguish between different parcellations. The coloration is derived from the Allen Brain Atlas

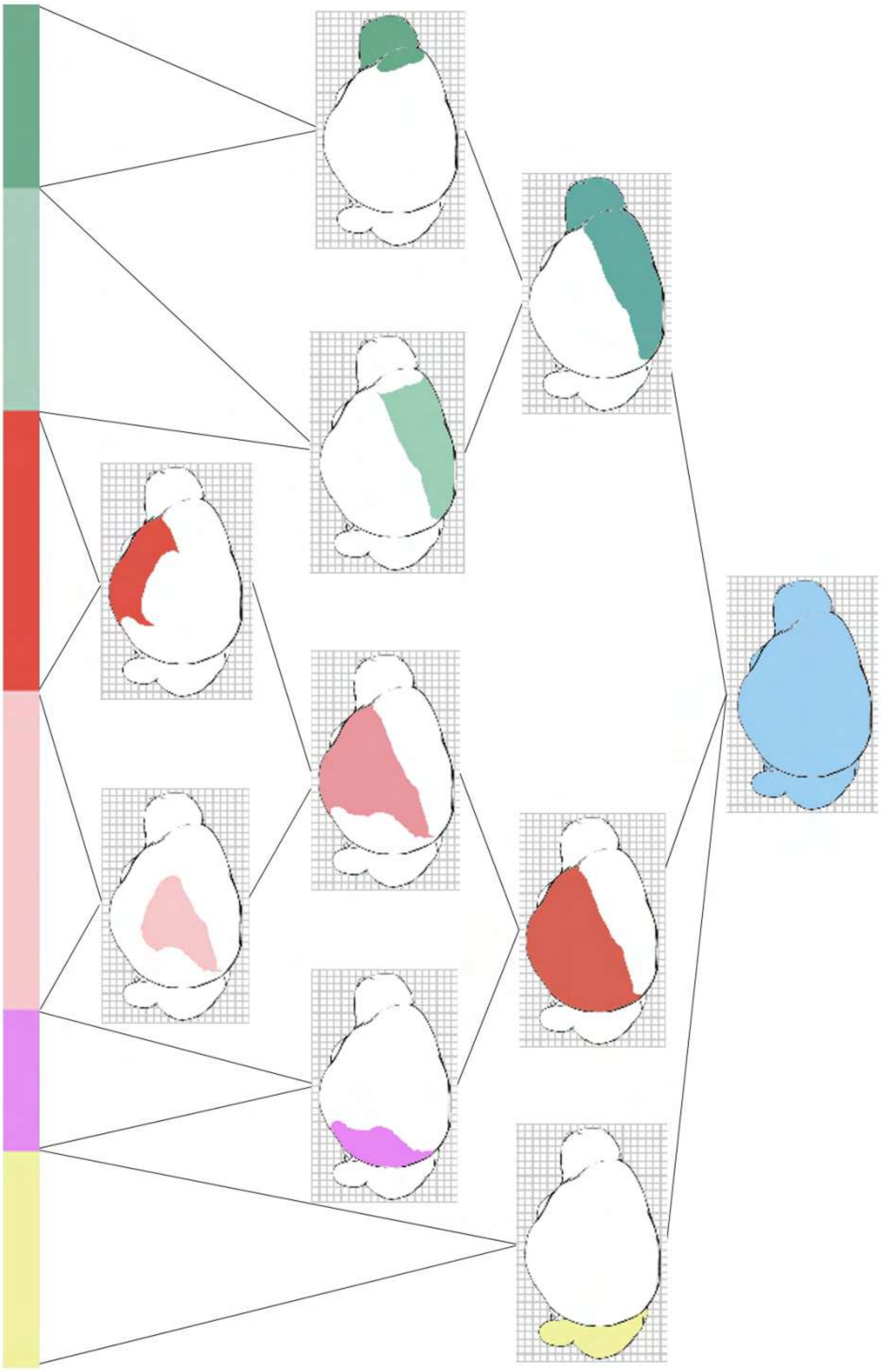


Figure 5.10: Scheme of a hierarchical representation of brain regions of a mouse brain. Figure by Ganglberger et al. [GWW+22].

and assigns colors based on their hierarchical position in the brain. Closely related parcellations can be identified by similar color. Sub-regions are encoded with the same hue but with lighter saturation then their parent regions. Sibling regions further down the hierarchy have a similar and sometimes the same color. Figure 5.10 gives an overview of regions of the mouse brain and their assigned colors.

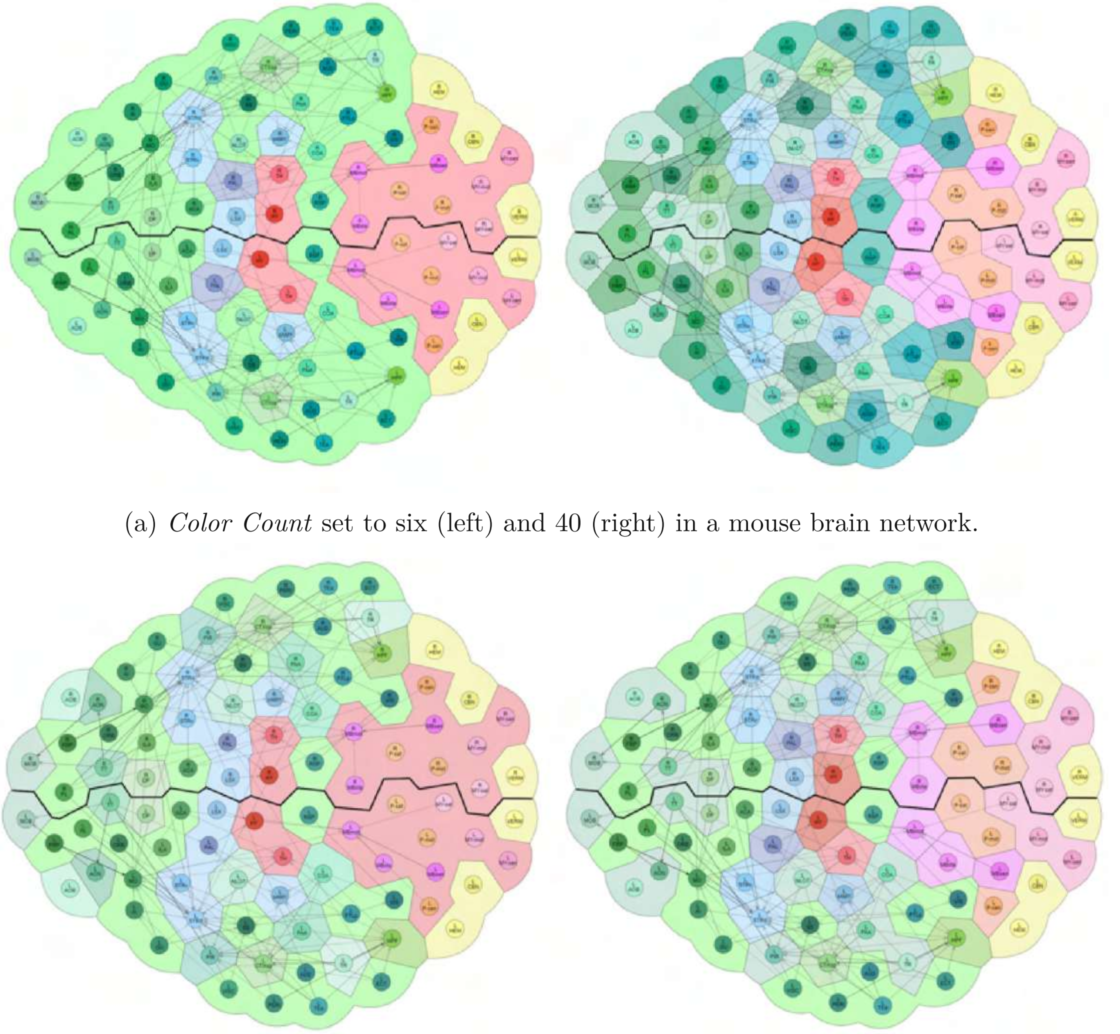
To represent this spatial information and allow the user an easier orientation, we integrate the parcellation in the background of the graph (R6, R7). We denote this background as *Parcellation Background*. This background additionally illustrates the overall shape of the brain. The background is created by applying a Voronoi tessellation to visually group nodes that belong to the same parcellation by color [WNV20]. The Voronoi cells are assigned the color of the parcellation, that the region of the node belongs to. To keep the network and the nodes in the focus, the colors of the Voronoi cells in the background are lightened. The *Parcellation Background* is also used to visualize the context described in Section 5.6 (R5).

To give the user more control over the visualization of the *Parcellation Background* two parameters can be adapted:

- *Color Count*: Represents the number of colors and thus corresponds to the number of visualized parcellations for granularity.
- *Priority*: To select the number of regions corresponding to the *Color Count* we traverse the brain hierarchy and split parcellations into their sub-regions. The algorithm can prioritize, which of these sub-regions should be divided first:
  - Anatomy-driven: Largest regions are divided first.
  - Importance-driven: Regions with the highest count of edges are divided first.

The effects of these parameters are displayed in Figure 5.11. The main goal of the algorithm is to find a set of parcellations that covers all *Network* and *Context Regions*. This means that each *Network* and *Context Region* has a super-region along the brain hierarchy contained in the set of parcellations or is contained in the set itself. The calculation of the parcellations works similar to the algorithm calculating the *Context Regions* (Algorithm 5.1 and 5.2). Again, the algorithm consists of a main routine (Algorithm 5.3) calculating a set of parcellations of the left hemisphere P_l in a recursive sub-routine (Algorithm 5.4). The set P_l is then joined with corresponding parcellations of the right hemisphere P_r for the final set of parcellations P.

There are four main differences to the algorithm calculating the *Context Regions*: (1) The focus of the *Parcellation Background*, that visualizes the parcellations lies on the color coding. The *Color Count* col is used instead of calculating a ratio. This time, not the size of the calculated set is considered. Instead the amount of distinct colors represented by the parcellations in P is checked. Since corresponding regions of the two hemispheres are assigned to the same color, we do not have to halve col. (2) The second difference is that the parcellations P_l in the recursive sub-routine *CalculateLeftParcellations* (Algorithm 5.4) are sorted depending on the *Priority* parameter pr. They are either prioritized by their



(a) *Color Count* set to six (left) and 40 (right) in a mouse brain network.

(b) *Priority* set to importance-driven (left) and anatomy-driven (right) in a mouse brain network. *Color Count* set to 16 in both figures.

Figure 5.11: Effects of different (a) *Color Counts* and (b) *Priorities*. Figure by Ganglberger et al. [GWW+22].

anatomical size or by the number of their connections. (3) As the aim is to find parcellations that cover the *Network* and *Context Regions* along the hierarchy no regions further down the hierarchy should be selected. The algorithm does not traverse further down the hierarchy if a region contained in set R is reached. (4) If the updated number of colors col', i.e., after the exchange of the parent with its sub-regions, exceeds col then we do not calculate a subset of the sub-region SP as in Algorithm 5.2, but continue with the next region in P_l. A subset of SP would possibly not cover all the *Network* and *Context Regions*. The sub-routine again terminates if the amount of distinct colors in P_l corresponds col or if the regions in P_l have no more subregions that are neither *Network Regions* nor *Context Regions*. If the *Color Count col* is higher then the sum of colors of all *Network* and *Context Regions* combined, we reach the second case of termination. The set P results in the combined set of *Network* or *Context Regions*. Hence, each Voronoi cell of a *Network* and *Context Node* in the *Parcellation Background* is assigned the color of the region itself. An example can be seen in Figure 5.11(a) on the right.

After we calculated the set of parcellations P we visualize them in the background of the graph. The background should reflect the anatomy and the shape of the brain abstractly and highlight the relationship between the nodes and the parcellation they belong to. The approach to create such a *Parcellation Background* is to calculate a Voronoi diagram using *Network Nodes* as Voronoi cells for the parcellations. The context for partial networks as described in Section 5.6 is incorporated by including the *Context Nodes* in this calculation. Points along the convex hull of the nodes are incorporated in the Voronoi calculation to approximate the shape of the brain.

Our approach is as follows: We create a set of nodes $\mathcal{S}$ including the *Network Nodes* and *Context Nodes*. The nodes in $\mathcal{S}$ are used to calculate the convex hull. By definition the cells of the Voronoi diagram that are calculated by points lying on the boundary of the convex hull create infinite surfaces (see Figure 5.12(a)). As we want to approximate the

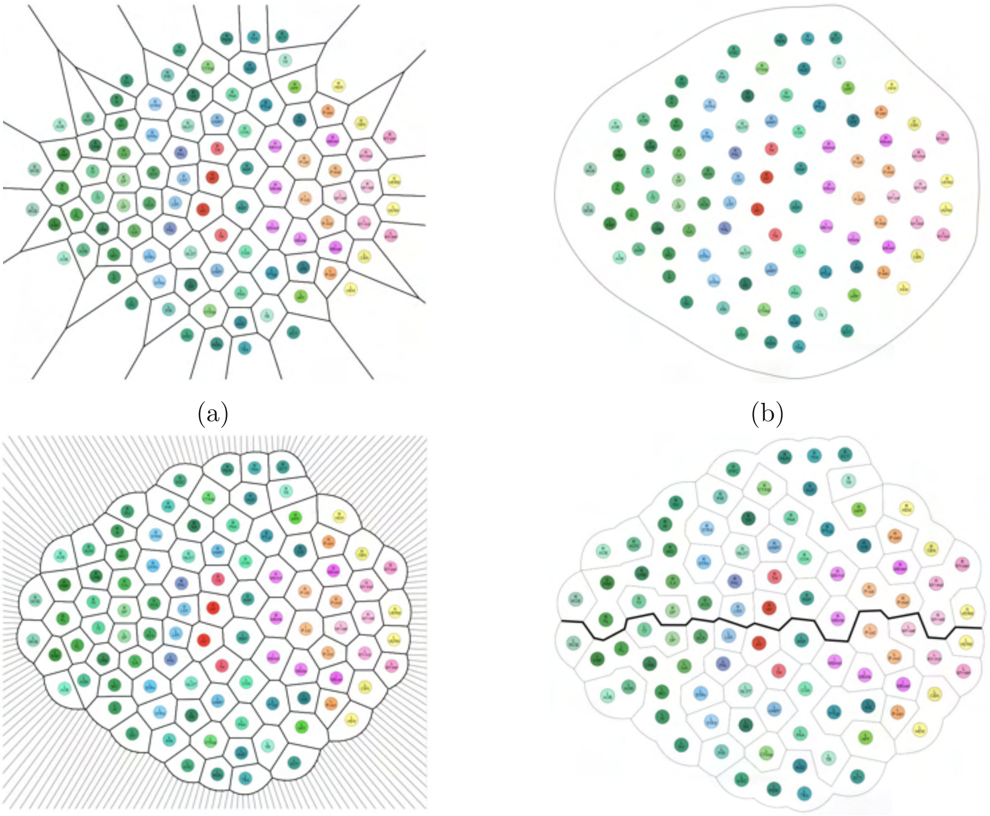
Algorithm 5.3: CalculateParcellations

```
input  : An integer col representing the color number,
         a list R containing Network Regions and Context Regions,
         a map B containing the brain hierarchy
         an enum pr for the priority.
output : A list P containing the parcellations.
if col ≤ 0 then
    return ∅;
end
$P_l \leftarrow$ CalculateLeftParcellations(col, R, B.subregions[0], pr) ;
    // B.subregions[0] represents the left hemisphere
$P_r \leftarrow$ GetRegionsFromOppositeHemisphere($P_l$);
$P \leftarrow P_l \cup P_r$;
return P;
```

Algorithm 5.4: CalculateLeftParcellations

```
input  : An integer $\mathsf{col}$ representing the number of distinct colored left
         parcellations,
         a list $\mathsf{R}$ containing *Network Regions* and *Context Regions*,
         a list $\mathsf{P}_l$ containing the left Parcellations,
         an enum $\mathsf{pr}$ for the priority.
output : A list $\mathsf{P}_l$ containing the context regions.
if $|colors\ in\ \mathsf{P}_l| \geq \mathsf{col}$ then
    return $\mathsf{P}_l$;
end
if $\mathsf{pr} = SIZE$ then
    sort $\mathsf{P}_l$ by size;
end
else
    sort $\mathsf{P}_l$ by connectivity;
end
foreach $\mathsf{p} \in \mathsf{P}_l$ do
    $\mathsf{SP} \leftarrow \mathsf{p}.subregions$;
    $\mathsf{SP} \leftarrow \mathsf{SP} \setminus \mathsf{R}$;
    if $\mathsf{SP} \in \mathsf{R}$ then
        continue;
    end
    $\mathsf{col}' \leftarrow |\text{colors in } \mathsf{SP}| + |\text{colors in } \mathsf{P}_l| - 1$;
    if $\mathsf{col}' > \mathsf{col}$ then
        continue;
    end
    $\mathsf{P}_l \leftarrow \mathsf{P}_l \setminus \{\mathsf{p}\}$;
    $\mathsf{P}_l \leftarrow \mathsf{P}_l \cup \mathsf{SP}$;
    return CalculateLeftParcellations($\mathsf{col}, \mathsf{R}, \mathsf{P}_l, \mathsf{pr}$);
end
return $\mathsf{P}_l$;
```

shape of the brain we use this fact by adding points on the convex hull to the Voronoi calculation. The convex hull is calculated using the *Network* and *Context Nodes* (see Figure 5.12(b)). Along the hull a large number of points at equal distance from each other is calculated. The number of 100 to 500 points yields good results. We denote these points along the convex hull as *Border Points*. To create some space between the outer most *Network Nodes* and the outer shape of the background we add a padding to the convex hull. The effect of the padding becomes clearer in the next step. We calculate the Voronoi diagram based on the points in $\mathcal{S}$ and the *Border Points* (Figure 5.12(c)). To visualize the parcellations, we add a contour by drawing lines between each pair of Voronoi cells belonging to different parcellations. The Voronoi edges generated by a cell



(a)

(b)

(c)

(d)

Figure 5.12: Steps for drawing the contour of the parcellations (black) of a mouse brain. (a) Voronoi diagram. (b) convex hull with padding. (c) Voronoi diagram including 200 *Border Points* evenly spaced along the hull. The Voronoi edges created by a *Border Point* and a node are the outline of the brain. The Voronoi edges of two Voronoi cells of *Border Points* are shown in gray color. (d) Final contours of the parcellations.

of a *Border Point* and a *Network* or *Context Node* create the outer border of the shape of the brain. The padding in the convex hull creates the space between the outer border and the nodes, as otherwise the *Border Points* would lie on the line between the nodes and the resulting outer border would overlap with the nodes. The Voronoi edges resulting from cells of two *Border Points* generate radiating lines (visible in Figure 5.12(c) in gray color). For the final result the radiating edges as well as edges between cells of nodes belonging to the same region of the parcellation are not visualized (Figure 5.12(d)). In the transversal perspective an additional thicker line is drawn between the two hemispheres to emphasize the symmetry. This diagram forms the basis of the *Parcellation Background*.

Finally, each Voronoi cell is colored with the color of the parcellation to which its *Network Node* or *Context Node* belongs. The colors are derived from the Allen Brain Atlases. Depending on the *Gray-scale* parameter for context (see Section 5.6) we calculate the gray-scale of the nodes parcellation color by calculating the weighted average of the three color channels: $0.3R + 0.59G + 0.11B$. The figures for this thesis and the user study were generated with an older application version that used the average method: $\frac{R+G+B}{3}$. The gray-scale option should be used with caution, as different colors can produce the same gray value. The Voronoi cells that belong to the points along the convex hull are rendered transparently.

5.8 Connectivity

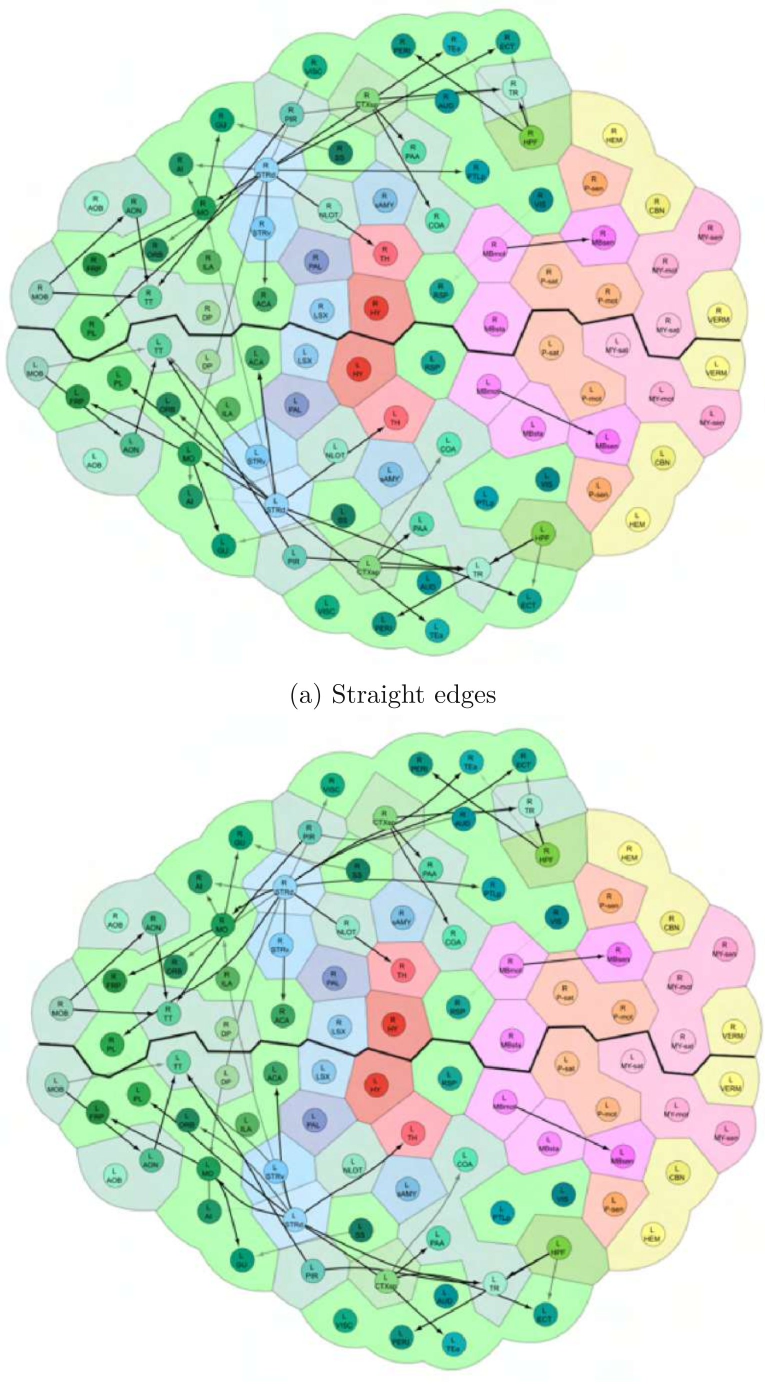
Up until now, we used the *Parcellation-derived Connectivity* to layout the graph using the *Anatomical Layout* (Section 5.5.2) and, depending on the species, applied the triangulation with the *Aesthetic Layout* (Section 5.5.3). After the graph is layouted, those edges are not required anymore. We remove them and instead add the *Rendered Connectivity* we want to visualize as edges in the graph.

Since the edges are numerous, it is expected that they will include many crossings, which is acceptable since we focus on the representation and preservation of spatial relationships between regions. To reduce the crossings a threshold can be applied by setting the percentage of strongest connections that should be rendered. Another option is to apply edge bundling or edge layouting (R8). Edge bundling reduces the visual clutter by visually bundling together similar edges. This bundling can help to highlight important edge patterns and often makes it easier to spot interesting connections or important data. An example is shown in Figure 5.13(b), Figure 3.4 and 3.5. Edge layouting also aims to draw edges in a way that reduces the visual clutter within a graph. One example is the organic edge layouting, a force-directed layout ensuring that edges do not overlap nodes (see Figure 5.13(c)). Another approach is the orthogonal routing, which routes edges using vertical and horizontal line segments only (Figure 5.13(d)).

5.9 Visual Encoding

The generated result of *Spatial-Data-Driven Layouts* as seen in Figure 5.13 is a node-link diagram representing the brain connectivity. It maps neurobiological 3D networks into 2D while maintaining spatial relations and reducing occlusions and clutter. Each node represents a specific brain region. It is labeled with an abbreviation of the region's name, including its brain hemisphere (left or right). The color encoding is derived from the Allen Brain Atlas as described in Section 5.7. We apply this color scheme as it is well known in the neuroscientific research community.

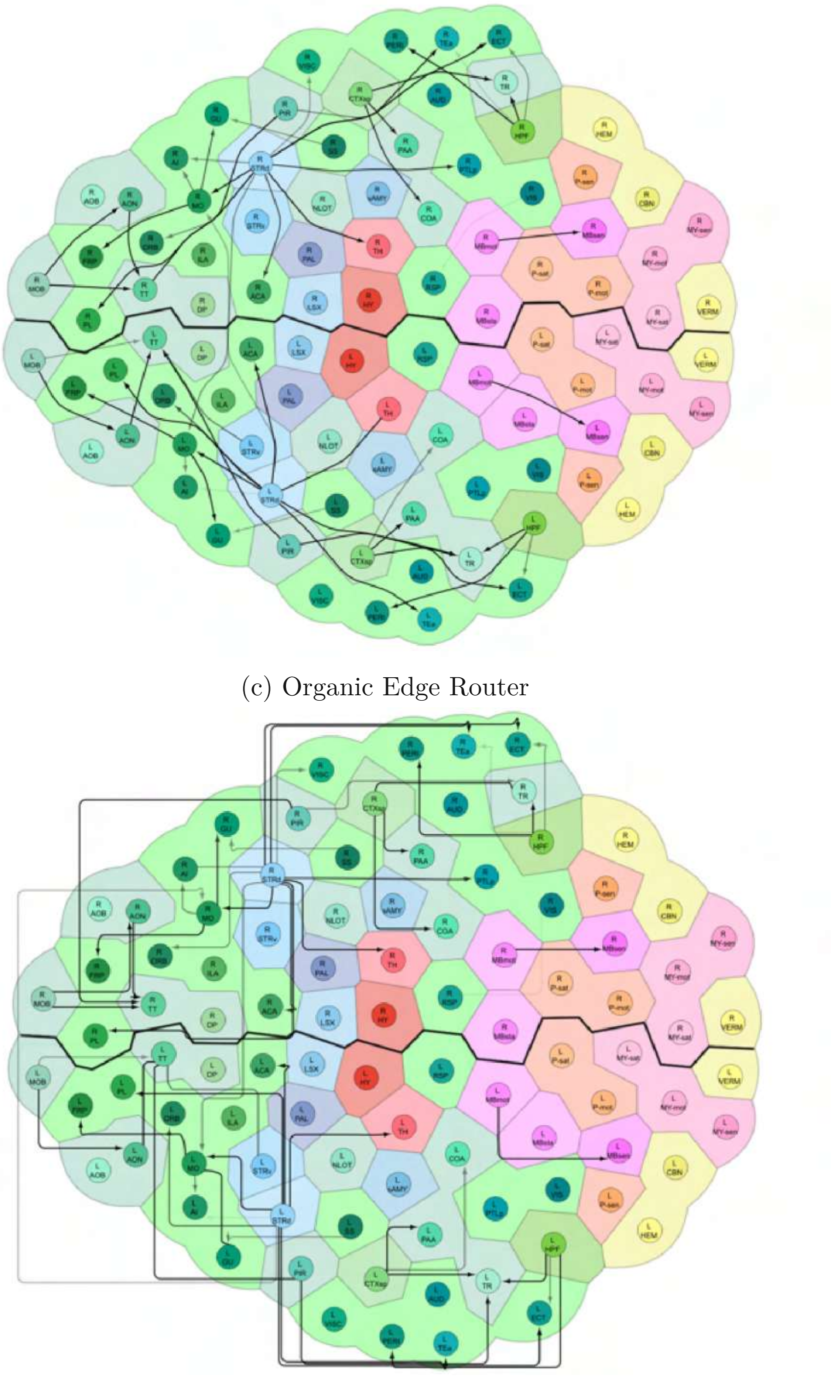
The connectivity holds structural connections between regions, as described by Oh et al. [OHN$^+$14]. The connections of the connectivity are represented by links or edges in the graph. The opacity visualizes the strength of a connection causing weak connections to appear more transparent to keep the focus on stronger and darker connections.



(a) Straight edges

(b) Edge bundling

Figure 5.13: Comparison of different edge routings [GWW+22]. (a) straight, (b) light edge bundling. This figure is an extended version found in the paper by Ganglberger et al. [GWW+22].



(c) Organic Edge Router

(d) Orthogonal Edge Router

Figure 5.13: Comparison of different edge routings [GWW$^+$22] (cont.). (a) organic routing and (b) orthogonal routing. The routers are available in the Cytoscape desktop app via the yFiles extension [yfi]. This figure is an extended version found in the paper by Ganglberger et al. [GWW$^+$22].

CHAPTER 6

Implementation

The implementation follows closely the design we presented in Chapter 5. Additionally we present the technology used to create a tool which is performant and compatible across platforms. We start with the program flow, to give an overview and link components of the implementation with our requirements and methodology. The Section 6.2 describes the used technology stack. Finally we describe each step of our methodology from a technical perspective.

6.1 Program Flow

Our web application is written using the React.js library [rea]. Its setup is rather simple, including two components:

- AppComponent: This is the main component that loads and prepares all required data. It includes mapping our data to nodes and edges.
- CytoscapeContainer: It represents our view component and renders the graph using the CytoscapeComponent library. It also draws the *Parcellation Background*.

An overview of the program is given in Figure 6.1 and the set of parameters and their values can be seen in Table 6.1. The first step in our program is to load the data of the network we want to visualize in JSON format. This information contains the connectivity and the included brain regions as described in Chapter 4. In case of a partial network we then calculate the regions that are not part of the network but are used to show context as described in the Section 5.6. Based on the user-defined *Context Ratio* the appropriate number of *Context Regions* is calculated as described in Algorithm 5.1 and 5.2.
The next step is to compute the parcellations that cover the *Network* and *Context Regions* as described in Section 5.7 using the Algorithms 5.3 and 5.4. After all required

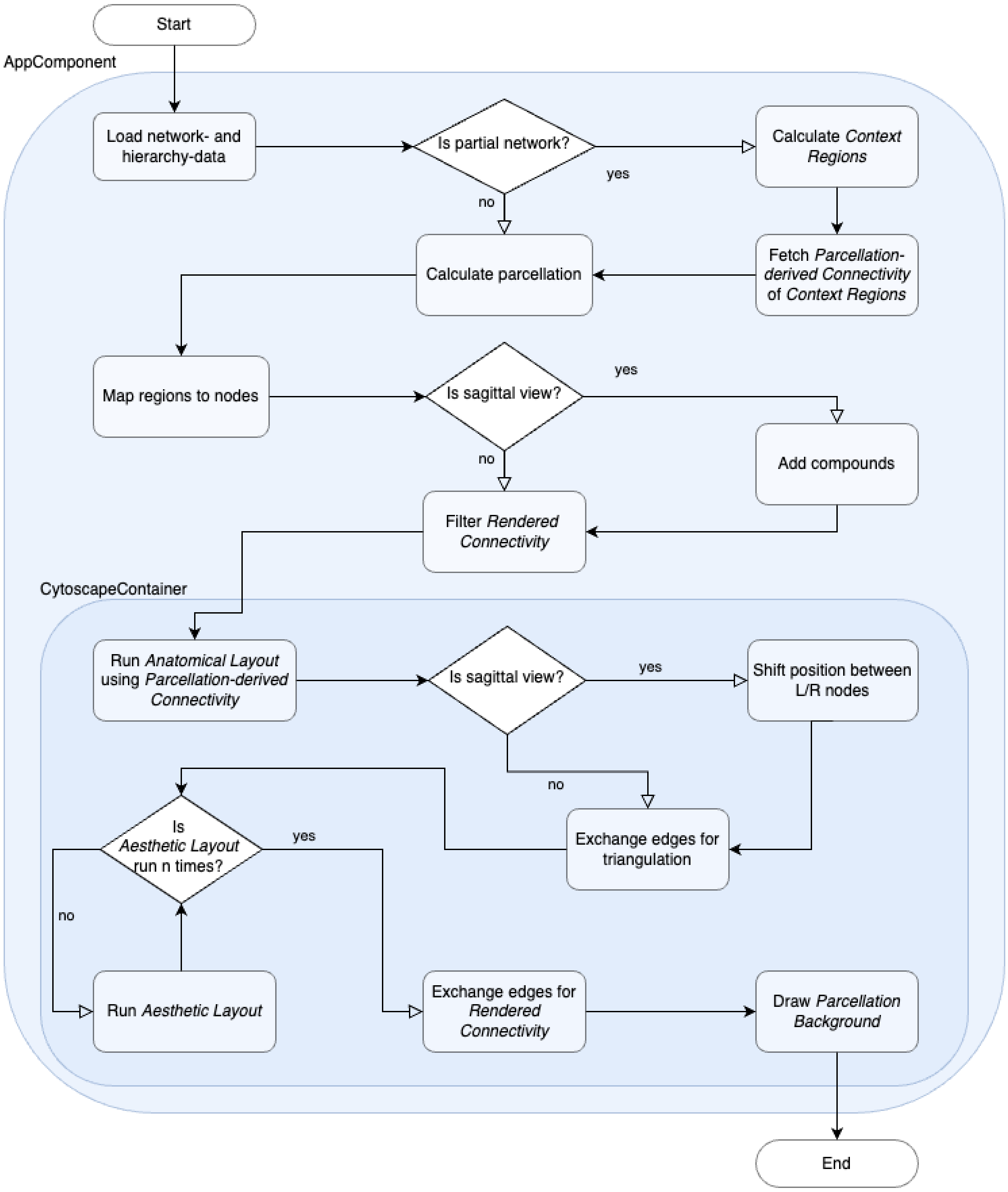

Start
AppComponent
Load network- and hierarchy-data
Is partial network?
yes
no
Calculate *Context Regions*
Fetch *Parcellation-derived Connectivity* of *Context Regions*
Calculate parcellation
Map regions to nodes
Is sagittal view?
yes
no
Add compounds
Filter *Rendered Connectivity*
CytoscapeContainer
Run *Anatomical Layout* using *Parcellation-derived Connectivity*
Is sagittal view?
yes
Shift position between L/R nodes
no
Exchange edges for triangulation
Is *Aesthetic Layout* run n times?
yes
no
Run *Aesthetic Layout*
Exchange edges for *Rendered Connectivity*
Draw *Parcellation Background*
End

Figure 6.1: Program flow and components. The AppComponent prepares the data which is used by the CytoscapeContainer to draw the graph. The number of iteration n of the *Aesthetic Layout* can be adapted, depending on the network. We heuristically evaluated that $n = 3$ works well for the human and mouse brain networks, while we did not apply the *Aesthetic Layout* ($n = 0$) for the fruit fly larval brain.

regions are calculated, we map these regions to Cytoscape.js nodes. We distinguish between *Network Nodes* and *Context Nodes*. The nodes are positioned depending on the perspective, as outlined in Section 5.4. For the sagittal perspective we add compound nodes for each node pair that belongs to corresponding parcellations of the left and right hemisphere.

Finally, we render the graph and the *Parcellation Background*. For the graph we pass the nodes and the *Parcellation-derived Connectivity* to the layout function of Cytoscape.js, which takes care of layouting and visualizing the graph. The *Anatomical Layout* is applied to position the regions to reflect the anatomical and spatial structure of the brain. The methodology for this step is described in Section 5.5.2. In case of the sagittal perspective we shift the position of the nodes representing regions of the right hemisphere in each compound (see Section 5.5.4) to create a consistent distribution between two related nodes. The *Anatomical Layout* can lead to overlaps depending on the overall structure of the brain, as well as the perspective. Additional rounds of layouting can be applied with another set of edges and parameters, which we describe as *Aesthetic Layout* in Section 5.5.3. The edges are replaced by the ones representing the *Rendered Connectivity* (see Chapter 4).

For the *Parcellation Background* we draw the Voronoi diagram as described in Section 5.7 on a HTML canvas. The colors of the parcellations calculated before are applied to colorize the Voronoi cells surrounding the *Network Nodes* and the *Context Nodes*.

6.2 Technology

The application is implemented in JavaScript using the library React.js v16.5.2. We use the open-source graph-drawing library Cytoscape.js v3.14.1. for implementing our visualization of the neuronal networks. The library provides multiple layouting algorithms for graph data. For easier handling of the Cytoscape data with React.js we use react-cytoscapejs v1.1.0. To draw the background underneath the graph, we apply cytoscape-canvas v3.0.1. In the remaining part of this section we give an overview of the graph drawing library Cytoscape.js [FLH+16] and the task-driven, web-based framework BrainTrawler [GSF+19], as both are relevant to our work.

Cytoscape.js

For the rendering of our graph we use the visualization library Cytoscape.js [FLH+16]. Cytoscape.js supports many different graph types such as directed graphs, undirected graphs, mixed graphs, loops, multigraphs, and compound graphs. We describe now the properties and options of Cytoscape.js that are relevant to our work, although more functionality is available.

Cytoscape.js' architecture includes two components - the *Core* (i.e., a graph instance) and the *Collection*. The *Core* is the main entry point to the library. A programmer can use it to alter the viewport and operate on the graph, like running a layouting algorithm. It furthermore provides several functions to access elements of a graph, such as edges

Table 6.1: Overview of parameters that are used for *Spatial-Data-Driven Layouts*.

Parameter	Values	Description
species	mouse human drosophila	the species whose brain network should be visualized
filePath		path to the network data
perspective	sagittal	view from the side
	transversal	top-down view
contextRatio		ratio of context for partial networks
colorCount		number of distinct colors in the *Parcellation Background*
priority	anatomical	largest parcellation is divided first
	importance	parcellation with highest number of connections is divided first
connectivityFilter	0-100 %	percentage of weakest connections, that should be filtered
grayScale	none	nodes and background colorized according to Allen Brain Atlas
	contextGray	parcellation of context regions is rendered in gray-scale
	allGray	all nodes and whole parcellation is rendered in gray-scale
handleUnbalanced	addAsNode	add missing conterpart as node
	addAsContext	add missing conterpart as context
	remove	remove modes without counterpart
	ignore	leave network as it is

and nodes. These functions return a *Collection*, a set of elements of the graph. The *Collection*, in turn, provides functions to view or manipulate nodes and edges.
To create nodes or edges JavaScript objects that follow a specific format can be used. The group field defines the type of an element and can be either nodes or edges. The data field contains metadata for each element. An identifier is mandatory and is assigned automatically if not defined. Label and color can be set for each element. Nodes are positioned by setting coordinates and can have another node as parent, which is rendered as a compound node. In the following a simple example of a node object is given as we use it in our tool:

```
{
    group: 'nodes',
    data: {
        id: 'n_1',
        label: 'label',
        color: '#ff0000',
        parent: 'n_0',
        },
    position: { x: 100, y: 100 },
    classes: [],
},
```

Edges require a reference to their start and end node. We denote these as source and target, respectively, as this is the label of the fields in Cytoscape.js. Edges can be weighted. The following is an example of an edge object:

```
{
    group: 'edges',
    data: {
        id: 'e_0',
        label: 'n_0 to n_1',
        color: '#ff0000',
        source: 'n_0',
        target: 'n_1',
        weight: 0.75,
        },
        classes: [],
}
```

We can apply different styling properties using Cascading Style Sheets (CSS) by setting one or more CSS classes. Cytoscape.js provides additional styling properties including the shape of nodes and a set of different edge types and arrowheads.

Cytoscape.js is a part of the Cytoscape Consortium. They also provide Cytoscape [cyt] an open-source desktop application for visualizing complex networks. As Cytoscape.js is limited regarding the visualization of different edge layouts, we migrate our visualization created with Cytoscape.js to the desktop application to perform further edge layouts.

BrainTrawler

BrainTrawler [GSF+19] is a task-driven, web-based framework developed by VRVis [vrv] in collaboration with the Institute of Molecular Pathology [imp] in Vienna. To create our *Spatial-Data-Driven Layouts*, we use BrainTrawler to select and export our network data and use BrainTrawler's API to query the connectivity data for the *Context Regions*. It allows visualization and iterative exploration of spatial data. This includes imaging data showing brain-wide gene expressions, fMRI, or structural data, with structural, functional, and genetic anatomical relations at different hierarchical anatomical levels. The data includes, among other information, hierarchical parcellations and structural annotations (e.g., Allen Mouse Brain Atlas and Allen Human Brain Region Annotations), region-wise connectomes/relations (e.g., resting-state functional connectivity), and voxel-wise connectivity/relations (e.g., structural connectivity, gene co-expression), all aligned to a common reference space. *Aggregation Queries* [GKHB20] represent the aggregated incoming and outgoing connectivity of volumes of interest and are performed in real-time. This data can be explored interactively on different anatomical scales.
Following steps in the BrainTrawler Framework are required to generate a data set for *Spatial-Data-Driven Layouts*:

1. Browse parcellations: The Parcellation Browser represents the hierarchical organization of the brain parcellation in a tree view, including the name and color code of every brain region (Figure 6.2). Its main purpose is to select the brain regions that should be visualized.

2. Browse connectivities: The database yields the data collections that the user can browse and select (Figure 6.3). This includes the *Parcellation-derived Connectivities* and other connectivities. An example for such a connectivity is the Resting-State Network which is used to identify coherent fluctuations in brain activity when an individual is not actively engaged in a cognitive task.

3. Refine connectivity (optional): The user can perform source/target queries or query areas by directly selecting the areas of interest based on a 3D brain visualization (Figure 6.4). The results of the queries are received instantly.

4. Export the networks: The user gets an overview of the imported connectivities and has the possibility to filter these (Figure 6.5). For the reproducibility of our results, we do not filter the *Parcellation-derived Connectivity* and use all the connections available. Finally, the data set can be exported as a JSON file.

The exported file includes the *Parcellation-derived Connectivity*, the connectivity we want to render, as well as the parcellation of interest.

6.3 Preprocessing

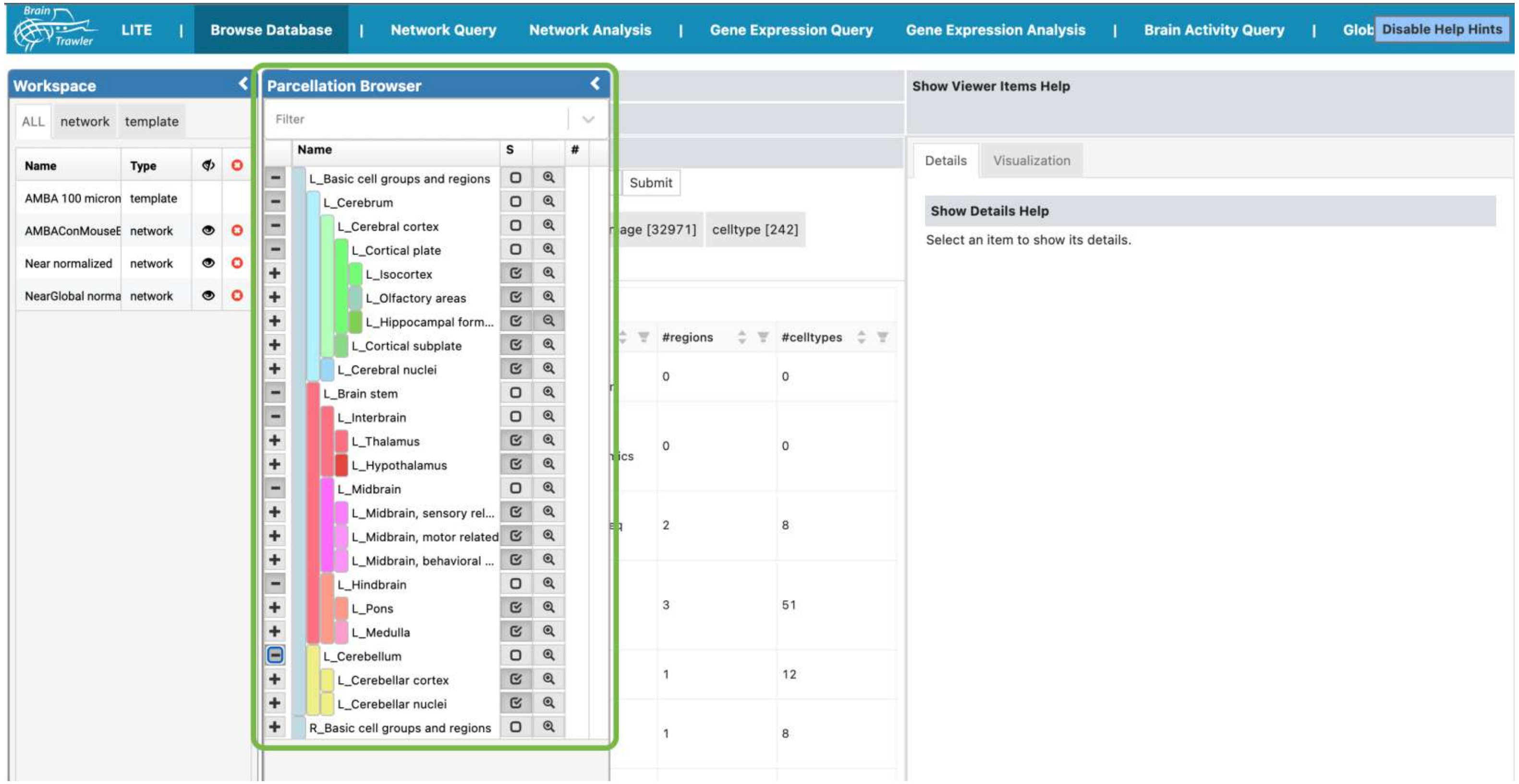
Two files containing the *Hierarchical Brain Parcellation* and the *Brain Network* are generated via BrainTrawler [GSF+19] as described in Section 6.2. The connectivities of the *Brain Network* have to include at least one of the *Parcellation-derived Connectivities* to aid during the layout process and a *Rendered Connectivity* of interest for the final visualization. Both files are imported into our AppComponent.

We often retrieve data of a specific region in the *Hierarchical Brain Parcellation*. For easier access we map the data in the *Hierarchical Brain Parcellation* from its tree structure to key-value pairs, with the key holding the identifier of the region (rid). We denote this collection of key-value pairs as *Parcellation map*. We apply additional information to each region in the *Parcellation map*, which is helpful in later calculations. To prioritize regions by their number of connections during the calculation of the parcellations in Algorithm 5.4, we calculate this number beforehand. The sum of the connections is calculated by accumulating the number of connections of each regions sub-regions in the *Rendered Connectivity*. Only connections where the region is the source are counted. In order to assign each *Network* and *Context Region* more easily to its parcellation, a list of all the super-regions per region is given.

Before we can visualize the imported network, we need to map its data to the node and edge objects provided by Cytoscape.js. as described in Section 6.2. These objects need to have a unique identifier, which we have to parse due to incompatibility to Cytoscape.js. The rids of the regions in our data have the format "#n:m", where n denotes a numeric id of the species and m the id of the parcellation. Cytoscape.js however does not accept the characters '#' and ':' in an identifier, hence we map all the rids to the accepted format "n_m". We map the rid in each region as well as in the source and target reference in each edge representing a connection.

The *Network Regions* can be directly derived from the input file. The *Context Regions* and parcellations for the *Parcellation background* are calculated in the next steps. We handle unbalanced *Network Regions* as described in Section 5.3. If the *handleUnbalanced* parameter is set to 'addAsNode' or 'addAsContext' we look up the missing regions and add their reference to the list of *Network Regions* or *Context Regions*, respectively. In the case of 'remove', we look up all regions from the set of *Network Regions* that do not have a corresponding region from the other hemisphere and remove them. Connections that reference a removed node also need to be excluded as Cytoscape.js will throw an exception due to the missing reference.

Finally, we visualize the *Rendered Connectivity* we chose via BrainTrawler [GSF+19]. The normalized weights of the connections are used to set the opacity of the edges in the final visualization. Edges representing the strong connections are rendered in solid color, while edges with a low weight are rendered transparent. Normalizing the *Parcellation-derived Connectivity* is not required, as the layout algorithm we apply does not account for edge



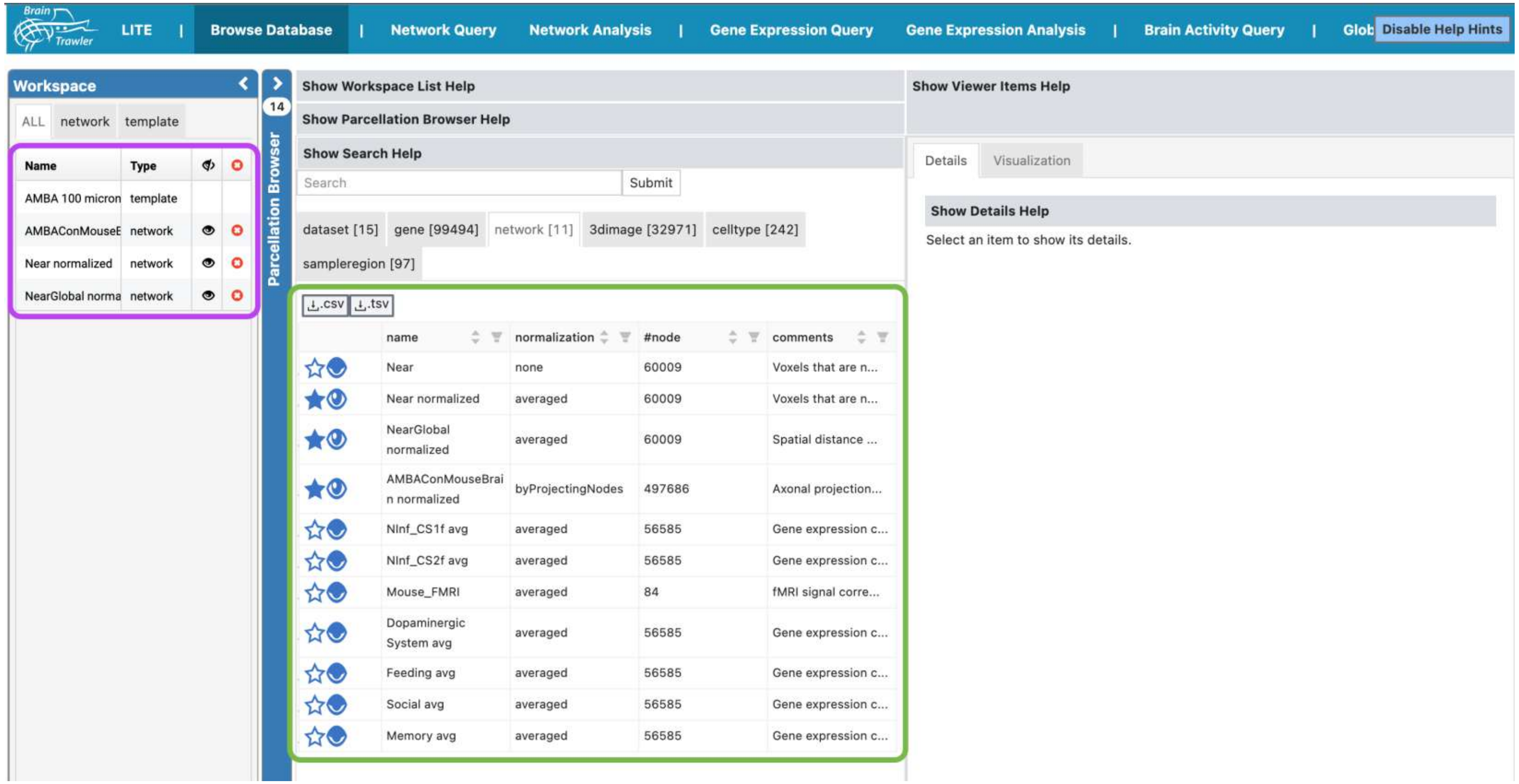
Name	Type
AMBA 100 micron	template
AMBAConMouseE	network
Near normalized	network
NearGlobal norma	network

Name	S
L_Basic cell groups and regions	☐
L_Cerebrum	☐
L_Cerebral cortex	☐
L_Cortical plate	☐
L_Isocortex	☑
L_Olfactory areas	☑
L_Hippocampal form...	☑
L_Cortical subplate	☑
L_Cerebral nuclei	☑
L_Brain stem	☐
L_Interbrain	☐
L_Thalamus	☑
L_Hypothalamus	☑
L_Midbrain	☐
L_Midbrain, sensory rel...	☑
L_Midbrain, motor related	☑
L_Midbrain, behavioral ...	☑
L_Hindbrain	☐
L_Pons	☑
L_Medulla	☑
L_Cerebellum	☐
L_Cerebellar cortex	☑
L_Cerebellar nuclei	☑
R_Basic cell groups and regions	☐

#regions	#celltypes
0	0
0	0
2	8
3	51
1	12
1	8

Select an item to show its details.

Figure 6.2: BrainTrawler workflow. 1) Browse parcellation. Parcellations of the brain hierarchy are selected here (green box).



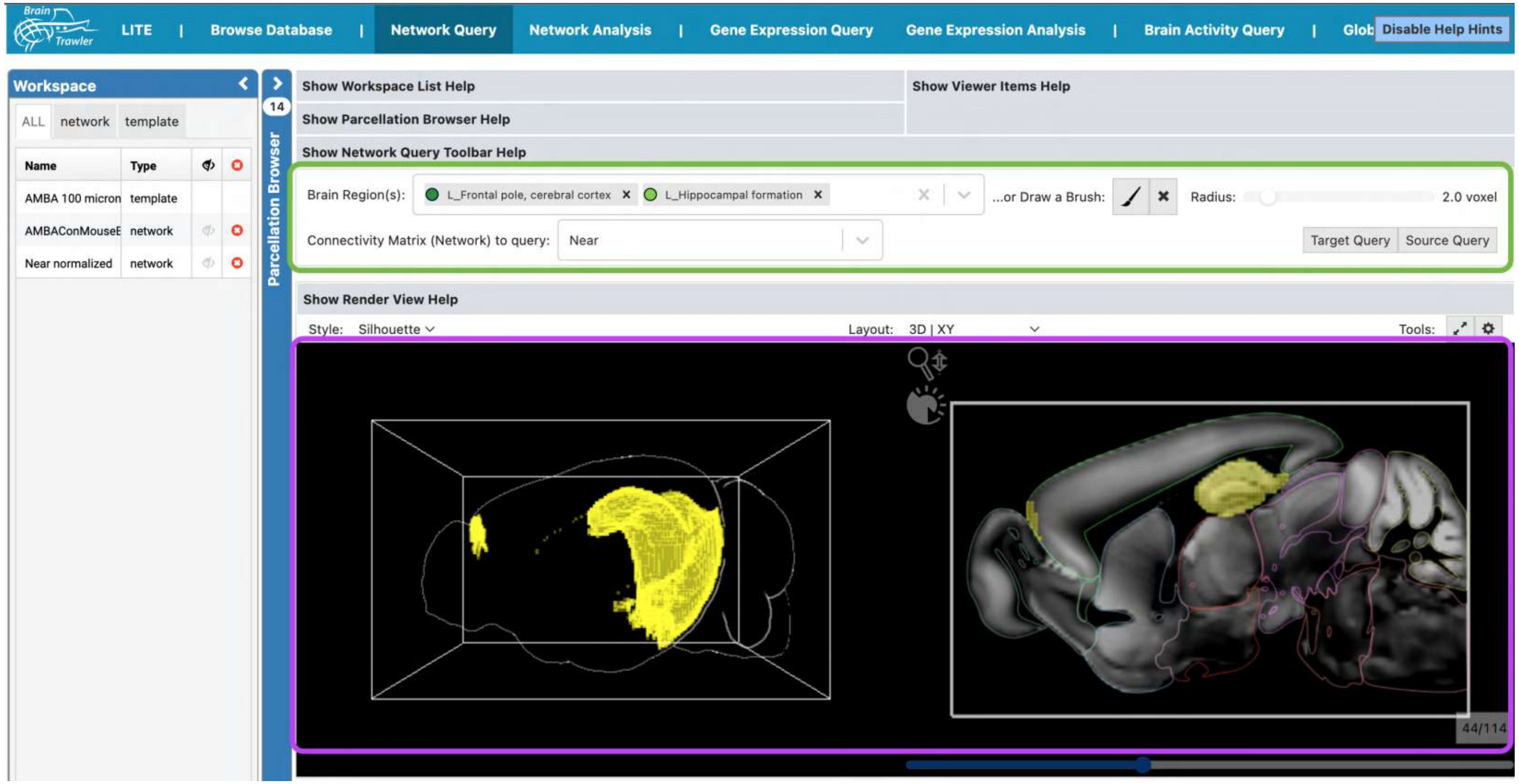
Select an item to show its details.

dataset [15] gene [99494] network [11] 3dimage [32971] celltype [242] sampleregion [97]

Name	Type		
AMBA 100 micron	template		
AMBAConMouseE	network	👁	⊗
Near normalized	network	👁	⊗
NearGlobal norma	network	👁	⊗

	name	normalization	#node	comments
☆	Near	none	60009	Voxels that are n...
★	Near normalized	averaged	60009	Voxels that are n...
★	NearGlobal normalized	averaged	60009	Spatial distance ...
★	AMBAConMouseBrain normalized	byProjectingNodes	497686	Axonal projection...
☆	NInf_CS1f avg	averaged	56585	Gene expression c...
☆	NInf_CS2f avg	averaged	56585	Gene expression c...
☆	Mouse_FMRI	averaged	84	fMRI signal corre...
☆	Dopaminergic System avg	averaged	56585	Gene expression c...
☆	Feeding avg	averaged	56585	Gene expression c...
☆	Social avg	averaged	56585	Gene expression c...
☆	Memory avg	averaged	56585	Gene expression c...

Figure 6.3: BrainTrawler workflow (cont.). 2) Browse connectivities. Connectivities are selected (green box). On overview of the selected connectivities is marked by the purple box.



Brain Trawler
LITE | Browse Database | Network Query | Network Analysis | Gene Expression Query | Gene Expression Analysis | Brain Activity Query | Glob
Disable Help Hints
Workspace
ALL network template
Parcellation Browser
14
Show Workspace List Help
Show Viewer Items Help
Show Parcellation Browser Help
Show Network Query Toolbar Help
Target Query
Source Query
Show Render View Help
Tools:
44/114

Name	Type
AMBA 100 micron	template
AMBAConMouseE	network
Near normalized	network

Brain Region(s): L_Frontal pole, cerebral cortex; L_Hippocampal formation

...or Draw a Brush:

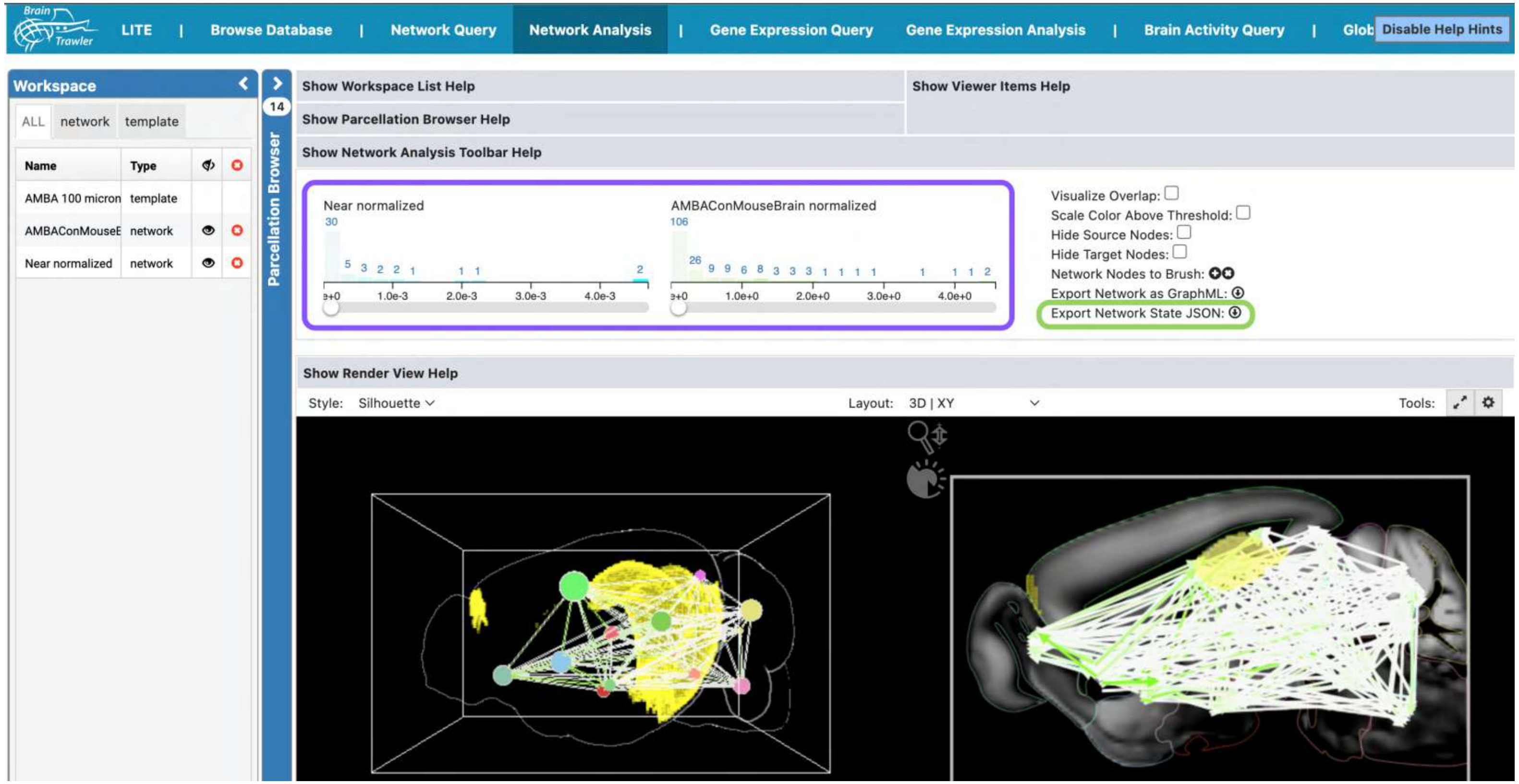
Radius: 2.0 voxel

Connectivity Matrix (Network) to query: Near

Style: Silhouette

Layout: 3D | XY

Figure 6.4: BrainTrawler workflow (cont.). 3) Refine connectivity. Connections can be filtered by specific brain regions (green box) or by applying the brush tool in the render view (purple box).



Name	Type		
AMBA 100 micron	template		
AMBAConMouseE	network	👁	○
Near normalized	network	👁	○

Visualize Overlap: □
Scale Color Above Threshold: □
Hide Source Nodes: □
Hide Target Nodes: □
Network Nodes to Brush:
Export Network as GraphML:
Export Network State JSON:

Style: Silhouette
Layout: 3D | XY

Figure 6.5: BrainTrawler workflow (cont.). 4) Export the networks. The export button is highlighted in a green box. Filtering of connections is possible (purple box).

weights. To overcome the problem that weak and insignificant connections have the same impact on the layout as strong ones we filter the *Parcellation-derived Connectivity* and leave only the four strongest outgoing connections for each node (as described in Section 5.5).

6.4 Context

We calculate the *Context Regions* based on the *Context Ratio* parameter (see Table 6.1) as described by the Algorithms 5.1 and 5.2. We include the connectivity data of the *Context Regions* by fetching missing information via the BrainTrawler API [GSF+19]. This step is required as the *Context Regions* and their connectivity are not part of the input file we created with BrainTrawler (Section 6.2). The *Context Regions* are then included in the layout algorithm. The nodes of the *Context Regions* themselves are set to be opaque and are considered in the later step of drawing the *Parcellation Background*.

6.5 Parcellation

We calculate a set of parcellations that is later used to draw the *Parcellation Background*. The calculation follows the Algorithms 5.3 and 5.4 and depends on the parameters *Color Count* and *Priority* (see Table 6.1). For the anatomical-driven priority we use the anatomical size of each region, which is already included in the files generated by BrainTrawler, while the number of connections for the importance-driven priority is calculated as described in Section 6.3.

6.6 Mapping Regions to Nodes

After we calculated the required *Network* and *Context Regions* we need to map these regions to Cytoscape.js node objects. Data like the color and position of each node is derived from the *Parcellation map*. We add a custom field to the node objects to easily differentiate among the types *Network* and *Context*. We assign each *Network* and *Context Node* to a region from the parcellations. The parcellation for each node can be looked up in its entry in the *Parcellation map*, as we saved a list of all super-regions per regions (see Section 6.3). The *parent* field in the Cytoscape.js node options is reserved for compounds, hence we create a custom field that holds a reference to the parcellation of the node.
In the sagittal perspective *Network Nodes* of corresponding regions of different hemispheres, e.g., left and right hypothalamus, overlap due to the symmetry of the two hemispheres. This issue is solved by applying compounds on the node pairs and by shifting the positions as described in Section 5.5.4. Compound nodes are applied by generating a new Cytoscape.js node object and setting the compounds identifier in the *parent* fields of the two nodes, that should be encapsulated by the compound. The shift to the position of each *Network Node* belonging to the right hemisphere is performed, as described in Section 5.4. The shift creates a uniform relation between each pair. For the

shift we set the position of the right region to the position of the left region and adapt its x and y coordinate. We choose a distance proportional to the diameter of the node and heuristically found that a slightly larger distance for the x coordinate compared to the one of the y coordinate yields good results. In our case the x coordinate is shifted by $1.2 * d$ and the y coordinate by $0.8 * d$, where d is the diameter of the nodes. Overlaps among different node pairs are reduced by the compound nodes (see Figure 5.3). We heuristically determined that applying the shift after the *Anatomical Layout* yields the best result. This is probably due to the forces that are applied during the *Anatomical Layout*, that also affect the shifted position of nodes, if the shift is applied beforehand.

6.7 Layouting

Cytoscape.js provides a set of layout algorithms including circle, concentric, and grid layouts and several force-directed layouts, that can be embedded as an extension for Cytoscape.js. We apply the CoSE (Compound Spring Embedder) implemented in Cytoscape.js. This force-directed algorithm was already mentioned in Sections 3.1.2 and 5.5. Technical details of the algorithm can be found in the work of Dogrusoz et al. [DGC+09]. CoSE's implementation is available as extension to the Cytoscape.js graph visualization library under the name CoSE-Bilkent [cos]. The layouting parameters are found in the Appendix for networks of the mouse, human, and *D. melanogaster* larval brains from transversal and sagittal perspective.

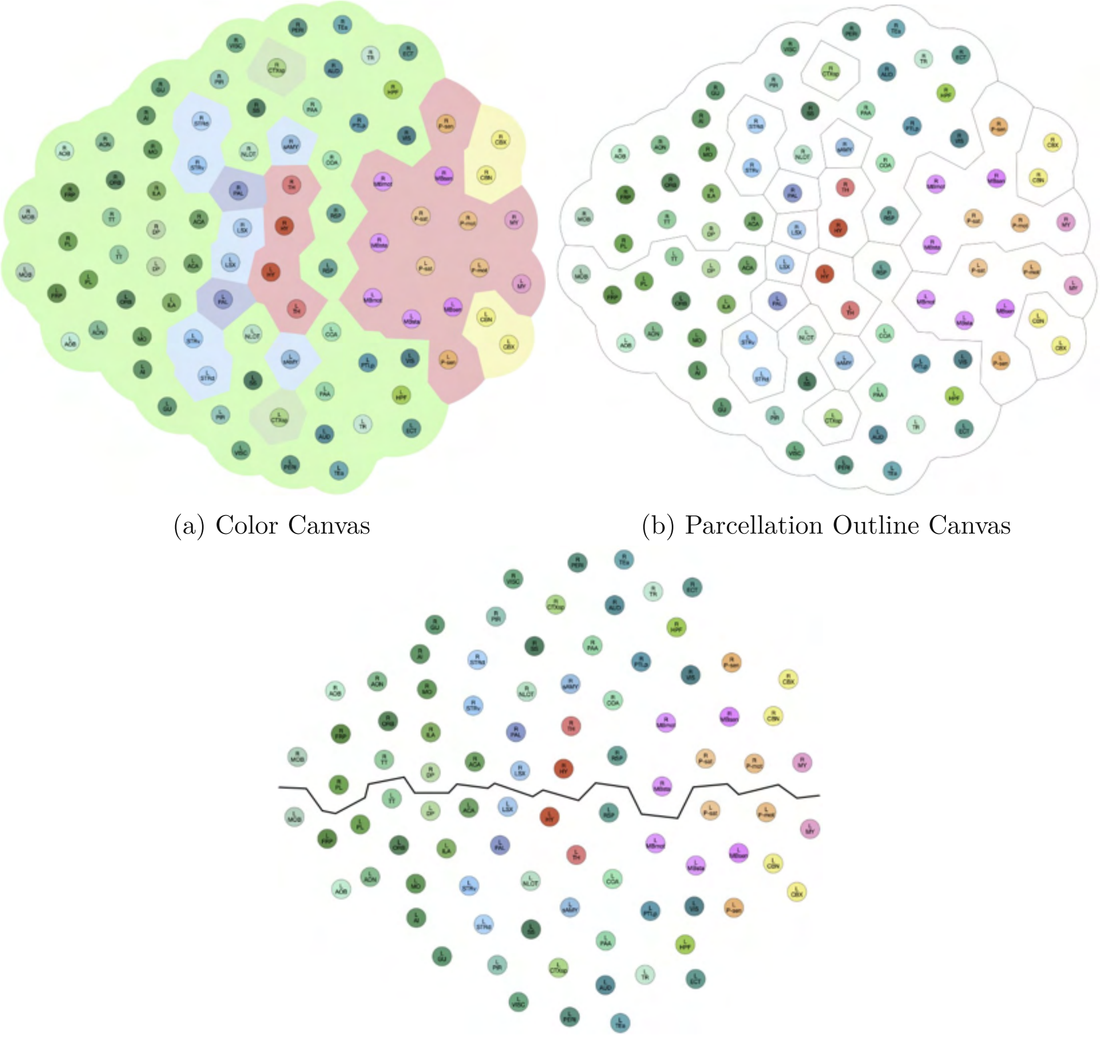
6.8 Parcellation Background

The next step is to draw the background to encode the spatial brain parcellation. As the idea is to group nodes that belong to the same parcellation by color and contour (see Section 5.7) separating lines are drawn between cells belonging to different parcellations. In the transversal perspective another broader line is rendered to visually separate the left and right hemispheres of the brain. To apply the coloring and outlining in the background each node has a reference to its parent in the parcellation.

We draw the *Parcellation Background* using the HTML canvas [htm]. The stroke function is used to set the color of the line drawn around the object, i.e., a Voronoi cell, while the fill function sets the color inside the object.

The first step is the calculation of the convex hull and the Voronoi diagram as described in Section 5.7. To avoid having to implement the algorithms for calculating the Voronoi diagram and the convex hull from scratch, we used existing JavaScript libraries. The library d3-delaunay.js v5.2.1 was used to calculate the Voronoi diagram and the code by Hollasch [hol] was applied to generate the convex hull. The combined set of *Network Nodes* and *Context Nodes* forms the basis for the calculation of the convex hull and the Voronoi diagram.

We first compute the convex hull including some padding around the nodes. We calculate a high number of *Border Points* along the convex hull and include them in the calculation of the Voronoi diagram. The *Border Points* are used to create the outer border and the



(a) Color Canvas

(b) Parcellation Outline Canvas

(c) Hemisphere Outline Canvas

Figure 6.6: Canvas layers for the parcellation of a mouse brain with 90 regions, *Color Count* of six, from transversal perspective. (a) Coloring of regions by the number of colors using the encoding of the Allen Brain Atlas. (b) Contours of the parcellations. (c) The outline (thick black line) between the left and right hemispheres (only applied in transversal perspective).

shape of the brain in the final visualization as depicted in Figure 5.12(c).
To render the background, we use the extension Cytoscape-canvas to create a canvas beneath our Cytoscape.js graph. It provides functionality to synchronize the positioning and interactions of the canvas and the graph, such as dragging and zooming. We apply up to three canvases (see Figure 6.6):

- *Color Canvas*: It includes the color of the parcellation the region belongs to for each Voronoi cell. The parcellation for each region is encoded in its entry in the *Parcellation map*, as it is included in the list of all super-regions per regions (see Section 6.3).
- *Parcellation Outline Canvas*: It shows a black outline between Voronoi cells that belong to regions of different parcellations.
- *Hemisphere Outline Canvas*: It only applies in the transversal perspective and depicts a thicker outline between the left and the right hemispheres of the brain.

For each canvas we iterate over the list of Voronoi cells and apply different colors for filling and outlines. The basis for the colors and the outlines of each canvas are maps with key-value pairs. The parcellation is used as key in the *Color Canvas* and the *Parcellation Outline Canvas*, while the keys for the *Hemisphere Outline Canvas* represent the two hemispheres. The value for each key is a list holding color values for each Voronoi cell in a fixed order. The colors are either hex color codes or the CSS (Cascading Style Sheets) property transparent. For the *Color Canvas*, if a cell belongs to a node that is included in the parcellation the value in the list is set to the color value of the parcellation, else to transparent. The map for the *Parcellation Outline Canvas* looks similar, but includes black as a color value for cells that are included in a specific parcellation. The map for the *Hemisphere Outline Canvas* distinguishes between left and right hemisphere and contains black as value for elements that are contained in this hemisphere and transparent otherwise.

We start with drawing the *Color Canvas*. We iterate over each Voronoi cell and apply the color of its parcellation for the filling (Figure 6.6(a)). To avoid distraction by the colors of the background, the parcelling colors are lightened by 25 %. Three color schemes are available. The default is including all colors as provided in the Allen Brain Atlas. By setting the *Gray-scale* parameter to *contextGray* the colors of Voronoi cells that belong to *Context Nodes* are converted to gray values. The parameter *allGray* does the same for all cells, including the nodes of the graph themselves. We apply the color to gray-scale conversion by calculating the average value of the three color channels as described in Section 5.7.
Rendering the two canvases for the outlines is more complex as it requires the combination of two drawings per parcellation or hemisphere with filling and outlines of two different widths via compositing operation of the Canvas 2D API. As visualized in Figure 6.7 the idea is do first draw a set of Voronoi cells with a solid filling and a thick outline and draw the same cells in a second step with a filling and a thinner outline. By overlapping

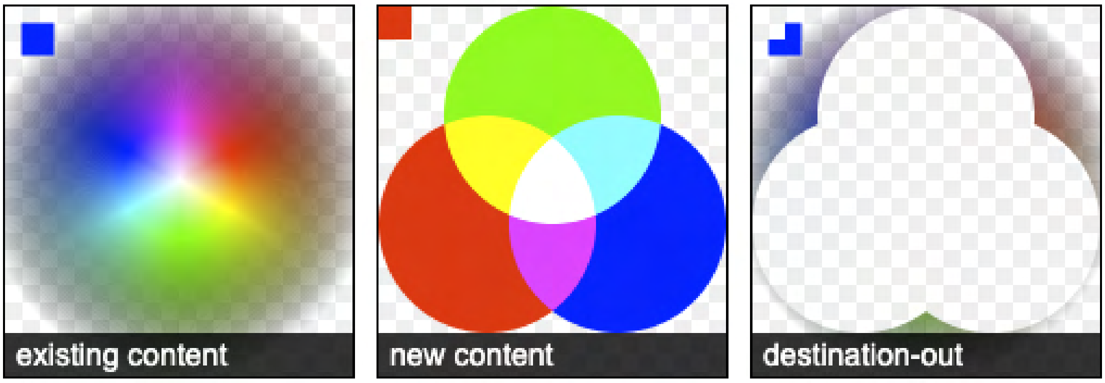


Figure 6.7: Destination-out operation - it keeps the existing content where it does not overlap the new shape [dev].

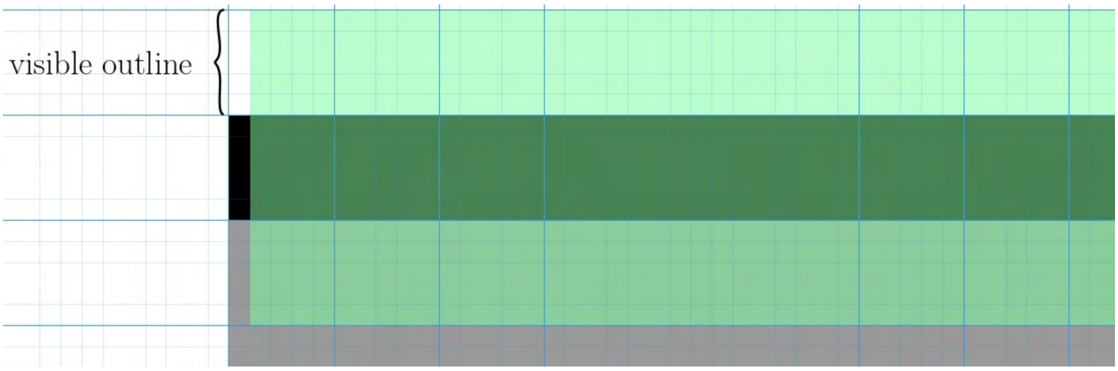


Figure 6.8: The thin outline of width w (black) overlaps the thick outline of width $3 * w$ (green) in its center. The filling is indicated by gray color. Only the outermost part of the thick outline stays visible after the destination-out compositing operation.

the two images and applying the compositing operation only the border of the thicker outline will remain (similar to the third image in Figure 6.7). As the width of the outline grows to both directions equally, the outlines overlap in the center of their widths (see Figure 6.8). If a outline should be drawn in a width of w, we set the width of the thinner outline to w and the width of the thicker outline to $3 * w$, As the thin outline overlaps the thick outline along the middle pixel only the outermost third of it stays visible. In our application we apply values in pixel for the widths. Other units that do not depend on the screen resolution, such as rem or em, can also be used. For our application, however, values in pixel are sufficient. For the *Parcellation Outline Canvas* we iterate over the parcellations and create each time the two drawings filled with black color. In our case we set the width of the outline of the first drawing to a value of of three pixel and the second to one pixel. We apply the *destination-out* operation which keeps the existing content where it does not overlap the new shape. The result is the outline of the parcellation one pixel wide. This step is repeated for each parcellation, which results in the outline of all parcellations (Figure 6.6(b)). By definition two adjacent *Border Points* on the convex hull create Voronoi cells that share an infinitely long side. This is visualized by the gray lines in Figure 5.12(c). As these cells are calculated only to create the outer shape of the brain we do not apply any filling and stroke here. A similar approach is applied for the *Hemisphere Outline Canvas*, but instead of the parcellations we distinguish between left and right hemisphere and set the difference of the line width of both drawings to a higher number, which results in a thicker outline surrounding the two hemispheres. To render this thicker outline only in-between the two hemispheres we set an arbitrary, solid color for the filling and stroke of the second, subtracting shape. This subtracts the outline of the brain, resulting in the single line in-between the two hemispheres (Figure 6.6(c)).

6.9 Edge Routing

Edge routing and edge layouting provides measures to reduce clutter of edges within a graph, as described in Section 5.8. Cytoscape.js only supports simple edge types, such as straight (Figure 5.13(a)) or Bezier. In order to apply more sophisticated edge routing algorithms we first need to transfer the layouted graph into the Cytoscape desktop application [cyt]. Cytoscape.js provides a function for exporting the graph in JSON format, which can be imported into the desktop application. The *Parcellation Background* is also added separately as background image in the Cytoscape desktop application. Cytoscape supports edge bundling using the algorithm by Holten and Van Wijk [HVW09] (see Figure 5.13(b)). Further edge layouting techniques can be imported into Cytoscape via the yFiles diagramming library [yfi]. This includes the *Organic Edge Router* (Figure 5.13(c)), a layout ensuring that edges do not overlap nodes and the *Orthogonal Edge Router* (Figure 5.13(d)), which routes edges using only vertical and horizontal line segments.

Overall, our implementation of *Spatial-Data-Driven Layouts* renders the result in a couple of seconds, depending on the size of the network. The largest part of the time is used for layouting through the cytoscape.js library and drawing the background.

CHAPTER 7

Evaluation

We first describe the aspects that play a vital role in the visualization of brain networks (Section 7.1). Florian Ganglberger, an expert in the field of neuroscientific visualization, conducted a case study to qualitatively evaluate visualizations for different species and anatomical levels based on the positioning of brain regions. Based on his findings [GWW+22], we designed user studies separated by species, which were conducted by domain experts (Section 7.2). Subsequently, the results of these studies are presented, whose analyses were performed by my colleagues Florian Ganglberger and Hsiang-Yun Wu.

7.1 Case Study

Evaluation of the anatomical feasibility was not in the scope of this thesis and has been performed by Ganglberger. A detailed description is provided in related work [GWW+22] and is summarized in this section. Ganglberger conducted case studies on two different species - mouse and human - as those species are of particular importance in neuroscience. His studies form the basis to demonstrate the general applicability of our approach on different brain architectures. Additionally, he evaluated first visualizations of the *Drosophila melanogaster* larval brain. Due to the lack of relevant network data for the fruit fly's circular neurons at the time of the studies, no proper user study has been conducted in this case. However, Ganglberger has already received numerous feedback from the Drosophila neuroscience community and will probably conduct the study at a later time.
To qualitatively evaluate the *Spatial-Data-Driven Layouts* Ganglberger checked if anatomically close regions in 3D are also spatially close in the 2D network graph. Color coding was applied to the nodes and to the *Parcellation Background* to further illustrate this comparison. Brain regions with similar hues are spatially close and should also be positioned close to each other by the layout. Ganglberger used the interactive Atlas

Viewer of the Allen Institute [All] for reference and further evaluation. For the creation of the figures the parameters for the force-directed layout had to be adapted. Based on Ganglbergers case study, a set of parameter values was defined for each species and each perspective (see Section 5.5). The following two sections detail Ganglberger's findings of applying *Spatial-Data-Driven Layouts* combined with *Parcellation-derived Connectivity* to mouse and human brain networks.

7.1.1 Mouse Brain

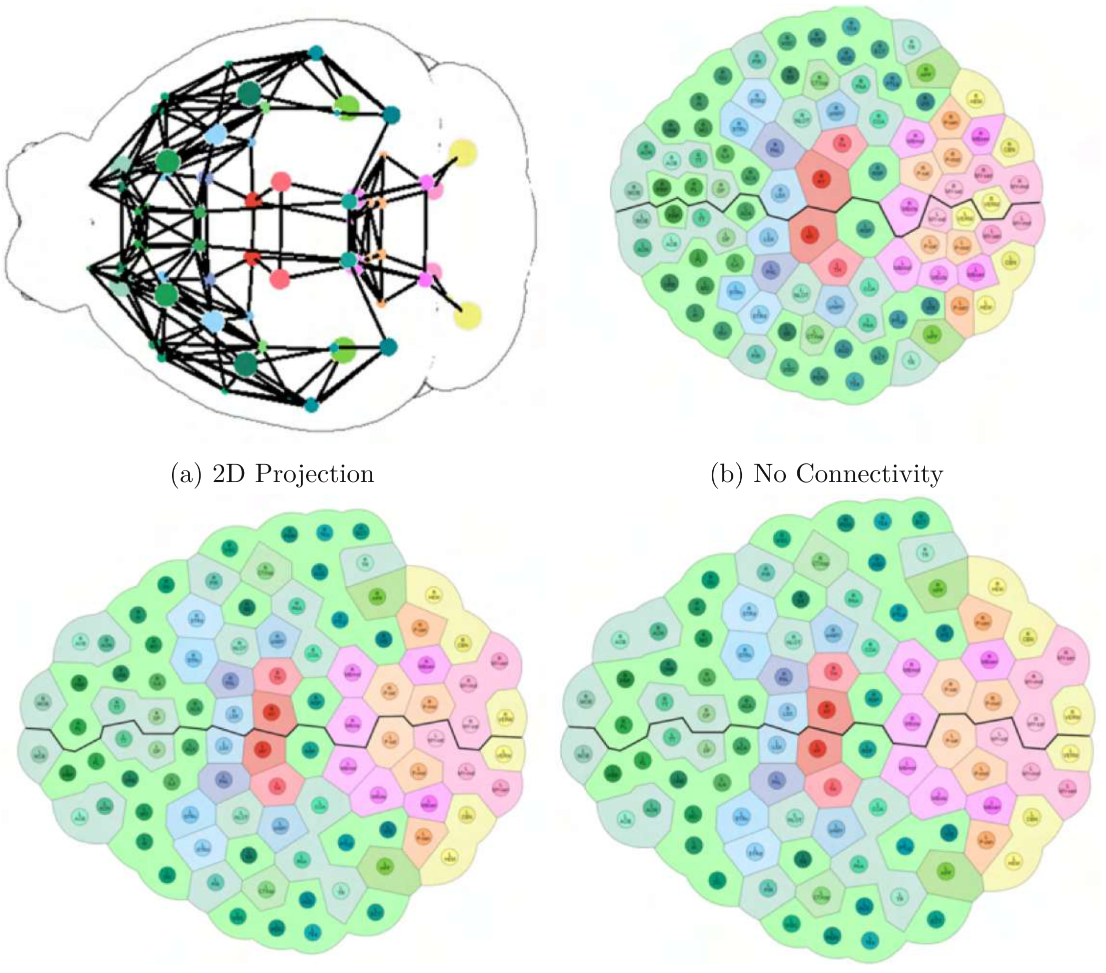
The mouse is a widely used model organism and its brain is commonly used to study neuronal connectivity in neuroscience [FLN+15]. The Allen Institute for Brain Science [WDL+20] has produced several reference atlases for the analysis, visualization, and integration of multimodal and multiscale data sets. It does not only provide a voxel-level representation of brain regions but also a *Hierarchical Representation of Brain Regions* (Figure 5.10).

Ganglberger et al. [GKHB20] used this data to create two types of *Parcellation-derived Connectivities*: one representing the reciprocal distance between the brain regions center-of-gravity (*Reciprocal Distance*) and the second representing the number of neighbouring voxels between the regions (*Neighborhood*). Both connectivities are described in Chapter 4). To demonstrate the effects of the *Parcellation-derived Connectivity* on the *Spatial-Data-Driven Layouts* on a mouse brain network Ganglberger et al. [GWW+22] created Figures 7.1 and 7.2 for the transversal and sagittal perspective, respectively. The edges in the 2D projections represent the *Neighborhood Parcellation-derived Connectivity*. For comparison, images of the network layouted without any connectivity are provided. As seen in the 2D projection of the mouse brain in the transversal perspective it is rather flat as there are not many node overlaps. In the sagittal perspective more regions occlude each other due to the deeper structure. Here the effect of the *Spatial-Data-Driven Layouts* is more apparent than in the transversal perspective. Overall, the effect increases with the size of the network, as can be seen in the distribution of 997 *Network Regions* in Figure 7.3.

Ganglberger stated that the layout without connectivity created a blob-like structure. The layout based on the reciprocal distance (Figure 7.2) pulls together more regions based on their anatomical positions than the layout generated without connectivity, thus resembling the shape of the mouse brain. Ganglberger points out that the layout based on the neighboring voxels yields even better results, as the thalamic and hypothalamic regions (red) and the striatal regions (light and dark blue) are connected in this case (marked in Figure 7.2(d)).

One goal of our method is to reproduce figures from neuroscientific publications. To showcase the applicability of *Spatial-Data-Driven Layouts* for this case, domain experts suggested illustrations of a mouse brain network from neuroscientific publications. We recreate these illustrations using *Spatial-Data-Driven Layouts* for comparison.

The dopaminergic circuitry in the mouse brain as published by Russo and Nestler



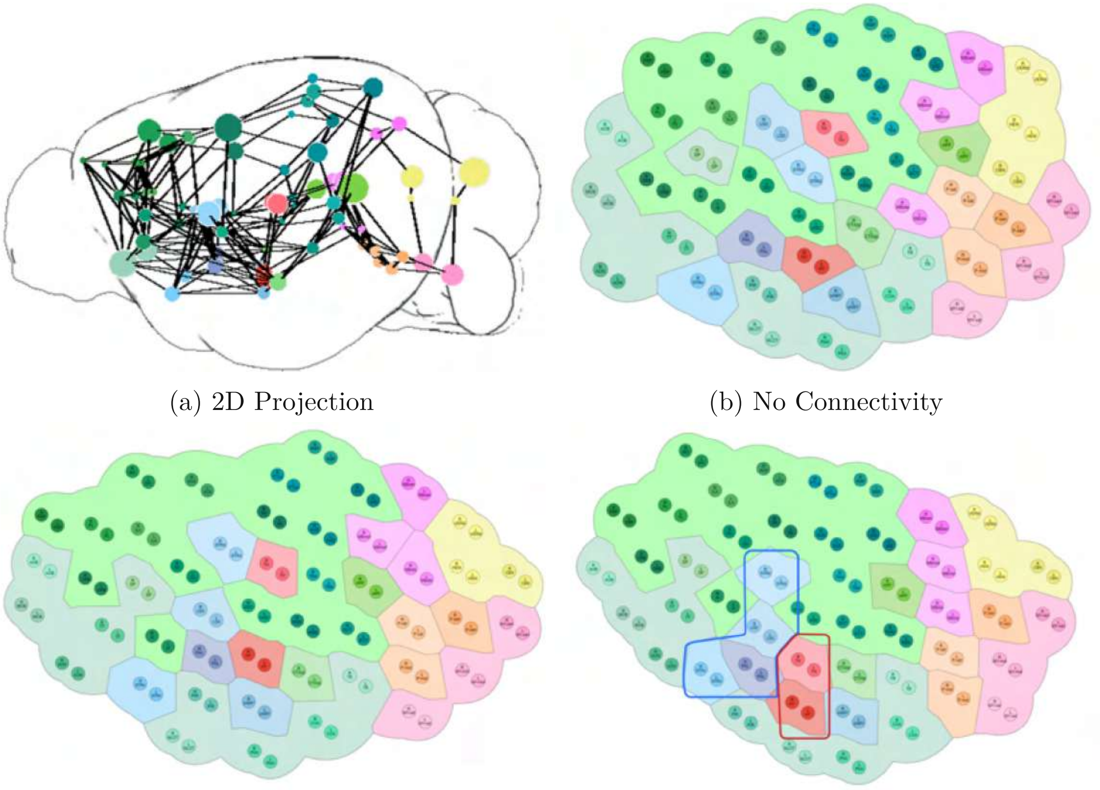
(a) 2D Projection

(b) No Connectivity

(c) Reciprocal Distance

(d) Neighborhood

Figure 7.1: Effects of *Parcellation-derived Connectivitiy* on the *Spatial-Data-Driven Layouts* of the mouse brain from transversal perspective. (a) the 2D projection of *Parcellation-derived Connectivity*. (b) layout of the nodes without connectivity at all. (c) layout with the reciprocal distance as *Parcellation-derived Connectivity*. (d). layout using the number of neighbouring voxels as *Parcellation-derived Connectivity*. The figure is an adaption of a figure created by Ganglberger et al. [GWW$^+$22].



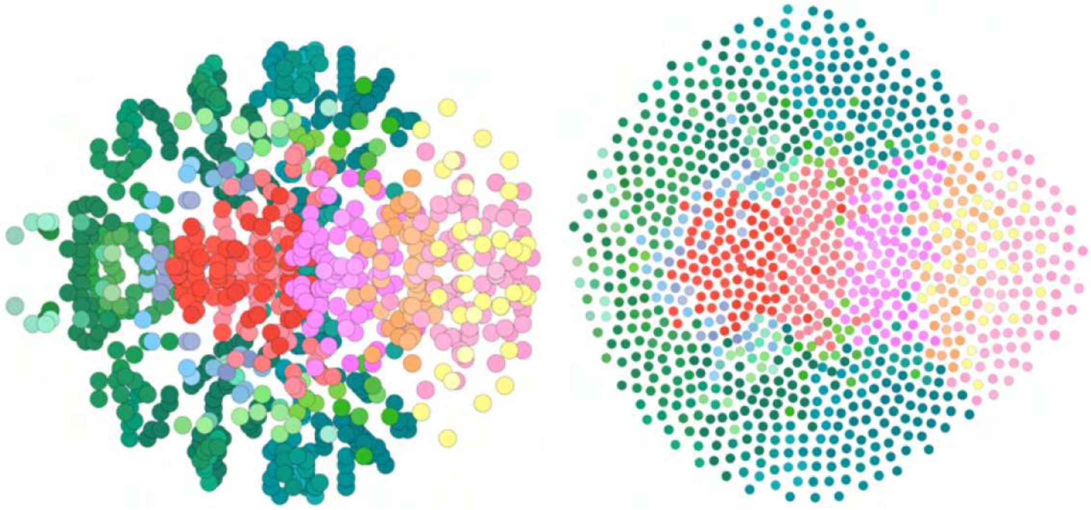
(a) 2D Projection

(b) No Connectivity

(c) Reciprocal Distance

(d) Neighborhood

Figure 7.2: Effects of *Parcellation-derived Connectivitiy* on the *Spatial-Data-Driven Layouts* of the mouse brain from sagittal perspective. (a) the 2D projection of *Parcellation-derived Connectivity*. (b) layout of the nodes without connectivity at all. (c) layout with the reciprocal distance as *Parcellation-derived Connectivity*. (d). layout using the number of neighbouring voxels as *Parcellation-derived Connectivity*. The improvements compared to the other figures are highlighted in red and blue. The figure is an adaption of a figure created by Ganglberger et al. [GWW+22].



(a) 3D Projection (b) Spatial-Data-Driven Layouts

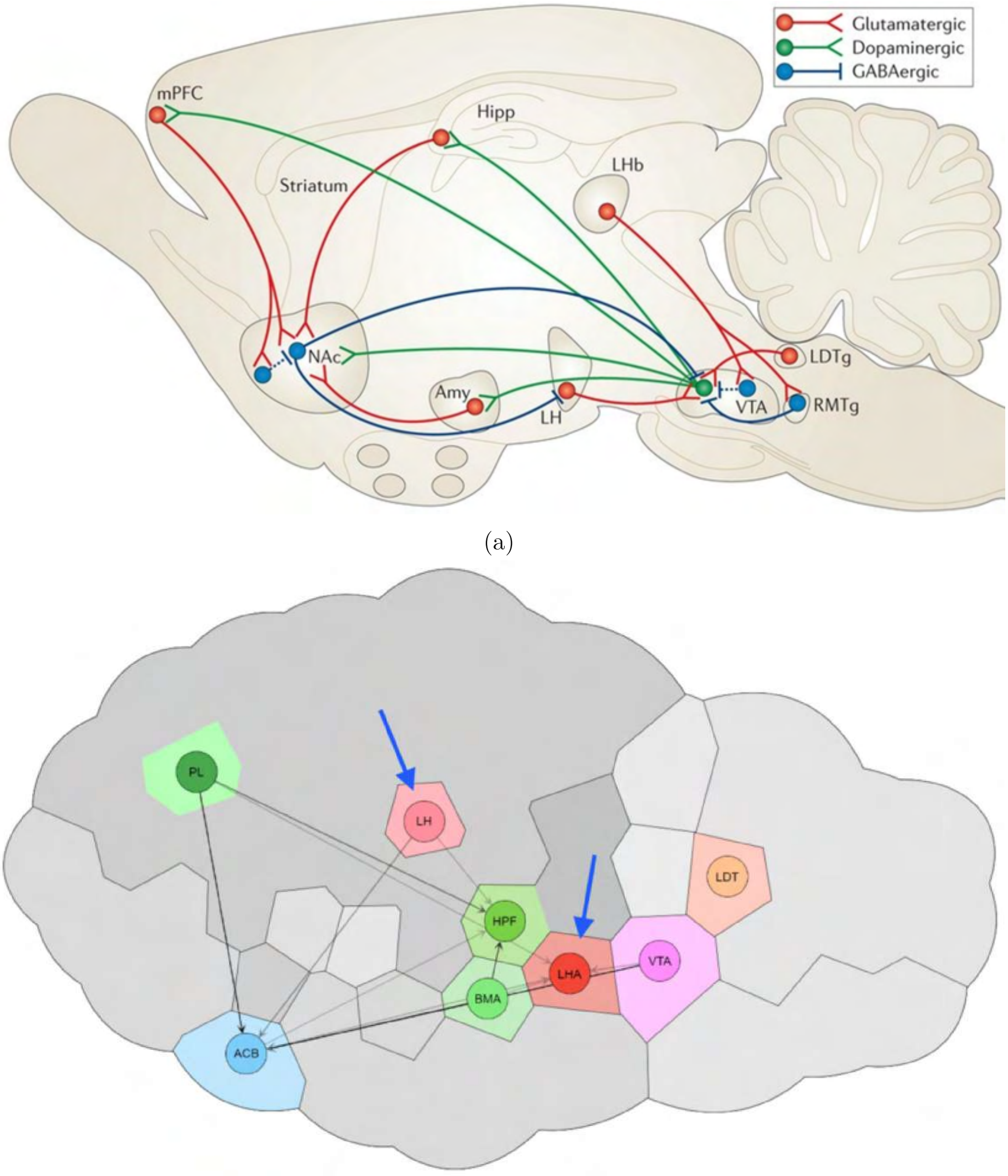
Figure 7.3: Effect of the *Spatial-Data-Driven Layouts* on node distribution for a large mouse brain network with 997 nodes. The left side shows a transversal 2D projection, the right side a *Spatial-Data-Driven Layout* of the same network. Labels, edges, and background are removed for the clarity of the layout. Figure by Ganglberger et al. [GWW+22].

[RN13] was used. The illustration is shown in Figure 7.4(a). We approximated this illustration as seen in Figure 7.4(b) by visualizing a neuronal network consisting of brain regions that correspond to the ones in the paper figure. The structural connectivity and the dopaminergic circuitry do not represent the same modality, hence the structural connectivity can only be seen as an approximation as not all connections are similar or present.

Ganglberger investigated if the spatial positioning of the result is similarly aligned as seen in the interactive Atlas Viewer [int] (Figure 2.2). He pointed out a single alleged inconsistency marked in Figure 7.4(b) in the distance between the lateral habenula (light red, LH) and the lateral hypothalamus (red, LHA). Their super-regions (thalamus and hypothalamus, accordingly) are positioned next to each other, hence the first intuition is that also LH and LHA should be adjacent. His statement is that the LH lies on the superior part of the thalamus, while LHA is positioned on the lateral side of the hypothalamus. Consequently, these regions are not adjacent and therefore correctly positioned.

7.1.2 Human Brain

Similar to the reference atlas of the mouse brain the Allen Institute published the *Allen Human Reference Atlas* [DRS+16]. The human brain atlas provides high-resolution histology 2D slices. In contrast to the reference atlas of the mouse brain it does not



(a)

(b)

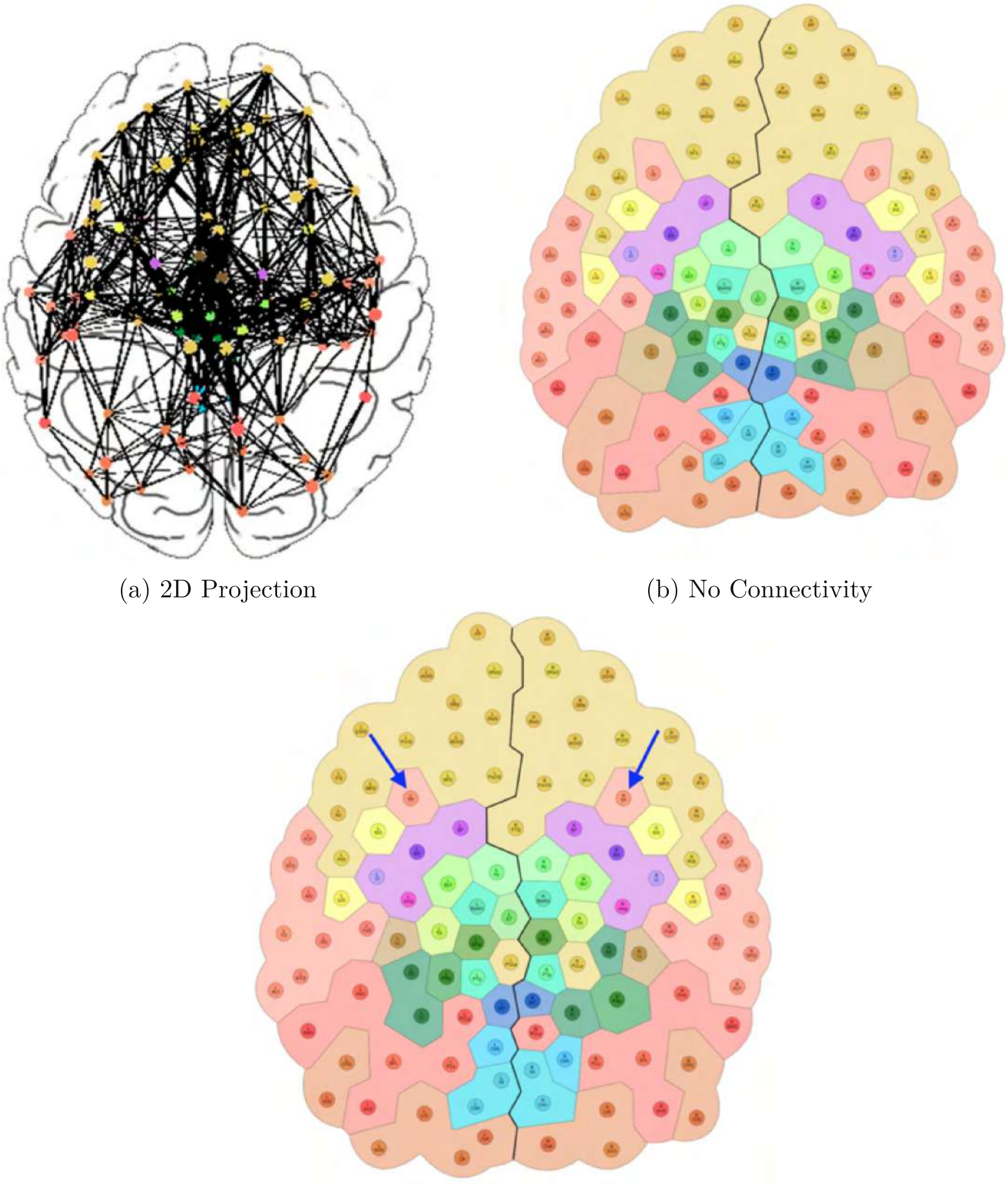
Figure 7.4: The dopaminergic circuitry in the mouse brain (a) by Russo and Nestler [RN13] and (b) its recreation with *Spatial-Data-Driven Layouts*. The two figures do not represent the same modality. The regions of the alleged inconsistency pointed out by Ganglberger are marked by blue arrows.

include a common coordinate framework to derive voxel-level representations of brain regions. Therefore, the neighborhood-based *Parcellation-derived Connectivity* could not be evaluated for the human brain. The brain's reciprocal distance among the regions for the distance-based *Parcellation-derived Connectivity* was calculated by Ganglberger using data from a paper by Hawrylycz et al. [HLGB$^+$12]. It provides 3D positions of samples labeled with *Allen Human Reference Atlas* brain regions.
Ganglberger evaluated the results using a case study analogous to the one of the mouse brain (Figures 7.5 and 7.6). He stated that the transversal perspective already showed promising results after layouting without connectivity. This is mainly because of the humans cortex parcellation in frontal, lateral, and posterior lobes. The sagittal perspective, however, required the *Parcellation-derived Connectivity* to group brain regions by their anatomical relations.
Again he checked if the brain regions were positioned and if they were adjoining each other correctly by comparing the result with the interactive Human Brain Atlas Viewer (Figure 2.3). He found a discrepancy in the position of the temporal pole (TP, light red) that is not connected to the other regions of the temporal lobe (light red). He stated that in relation to the regions of the cerebral nuclei (purple), its position is still correct, as the temporal pole and the cerebral nuclei are actually adjacent to each other. A more detailed examination of the positions of the brain regions using the interactive Human Brain Atlas Viewer [int] showed an overall consistency with the anatomy according to Ganglberger.

Analogous to the figure of the mouse brain, we also recreated a graph from an illustration of the human brain in the literature. The figure by Gotter et al. [GWC$^+$12], was suggested to us by domain experts (Figure 7.7(a)). The figure shows orexinergic neuron projections originating from the hypothalamus in the human brain. For its approximation we applied a data set with 20 % of the strongest connections. They originate from the hypothalamus at a hierarchical brain region level covering most of the figure's brain regions. Since structural connectivity was not available, we used the functional resting-state connectivity from the *Human Connectome Project*. Ganglberger stated that the strongest connections in our visualization using data by Hawrylycz et al. [HLGB$^+$12] cover the ones in the figure by Gotter et al. well. Missing regions were the lateral dorsal tegmental nuclues (LDT) and the locus ceruleus (LC), but their parent region pontine tegmentum (PTg) was still within the strongest 40 % of the connections (not shown in Figure 7.7).

7.2 User Study

The first part of this section gives an overview of the study design, how the study was conducted, and what tasks and questions were posed to the participants. For each species a separate study was designed and conducted in collaboration with Florian Ganglberger and Hsiang-Yun Wu. Following this, an overview of the results evaluated by Ganglberger and Wu is provided. The evaluation of the results was not in the scope of this thesis and is summarized here for completeness. More details can be found in the work of Ganglberger et al. [GWW$^+$22].

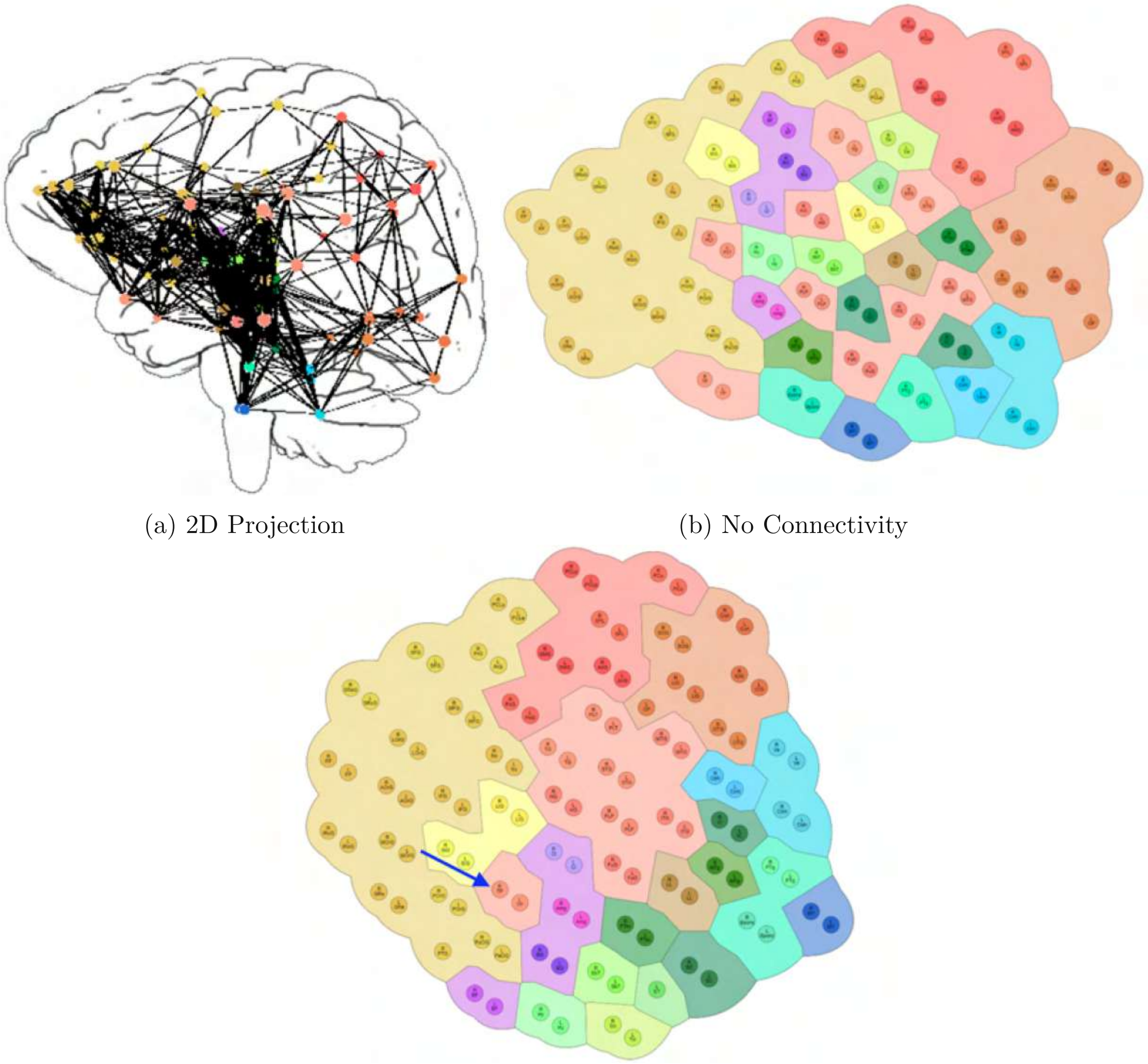


(a) 2D Projection

(b) No Connectivity

(c) Reciprocal Distance

Figure 7.5: Effects of *Parcellation-derived Connectivitiy* on the *Spatial-Data-Driven Layouts* of the human brain from transversal perspective. (a) the 2D projection of *Parcellation-derived Connectivity*. (b) layout of the nodes without connectivity at all. (c) layout with the reciprocal distance as *Parcellation-derived Connectivity*. Regions of the discrepancy described by Ganglberger are marked with blue arrows. The figure is an adaption of a figure created by Ganglberger et al. [GWW+22].

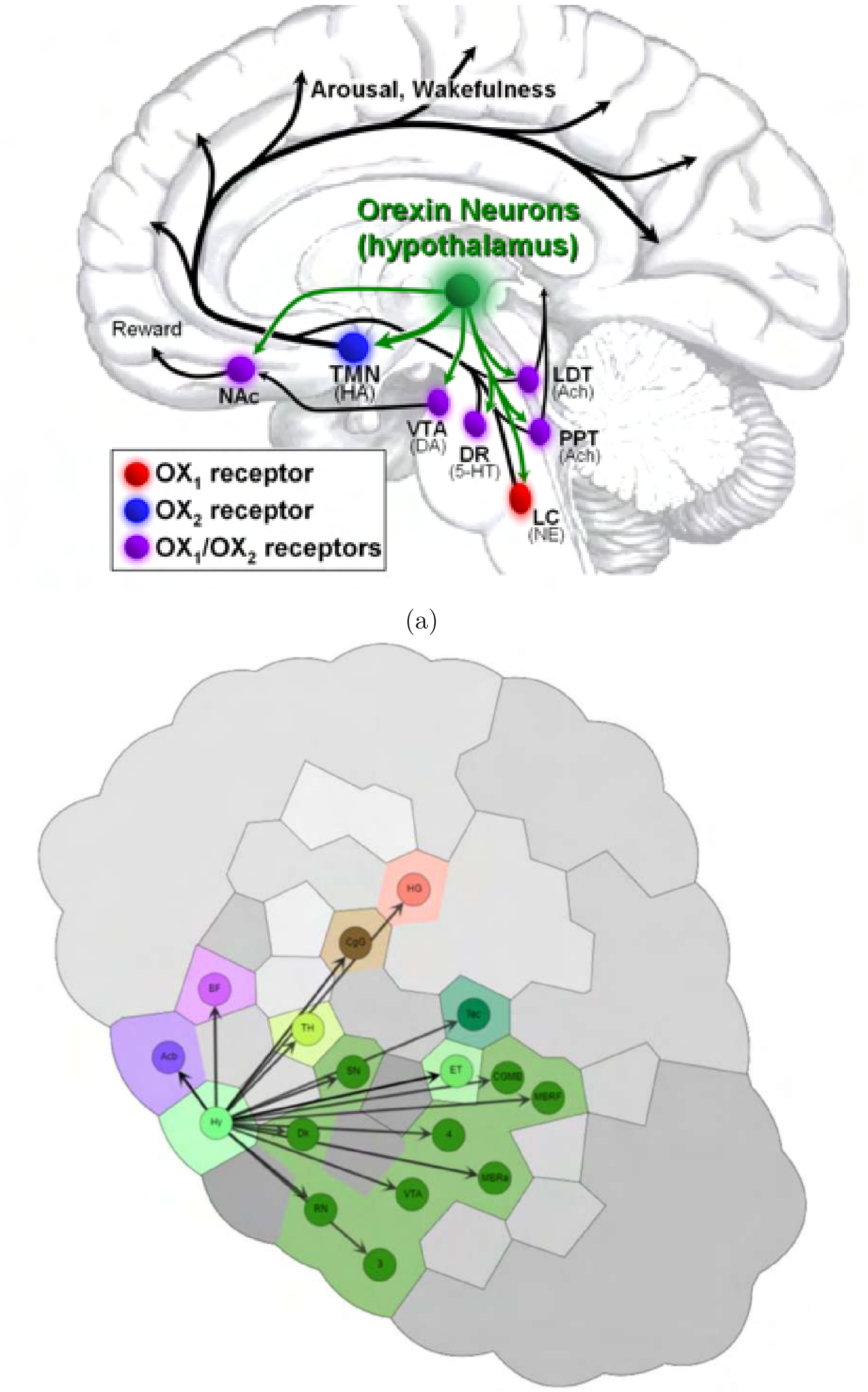


(a) 2D Projection

(b) No Connectivity

(c) Reciprocal Distance

Figure 7.6: Effects of *Parcellation-derived Connectivitiy* on the *Spatial-Data-Driven Layouts* of the human brain from sagittal perspective. (a) the 2D projection of *Parcellation-derived Connectivity*. (b) layout of the nodes without connectivity at all. (c) layout with the reciprocal distance as *Parcellation-derived Connectivity*. Regions of the discrepancy described by Ganglberger are marked with blue arrows. The figure is an adaption of a figure created by Ganglberger et al. [GWW$^+$22].



(a)

(b)

Figure 7.7: The orexinergic neuron projections originating from the hypothalamus in the human brain. (a) Figure by Gotter et al. [GWC$^+$12]. (b) Its recreation with *Spatial-Data-Driven Layouts*. Brain region hierarchy level was selected to cover the majority of brain regions depicted in (a) and the strongest 20 % of outgoing functional resting-state connections of the hypothalamus were used.

7.2.1 Study Design

Separate user studies were created and conducted for the mouse and human brain. Each participant already knows the species in question. We include images of whole and partial networks from sagittal and transversal perspectives. To compare our results we provide multiple visualizations:

- Directly projected layout: No layout algorithm applied
- *Spatial-Data-Driven Layouts* without background
- *Spatial-Data-Driven Layouts* with background: This represents our final result. In this case we additionally evaluate the coloring:
  - Grayscale: The whole visualization - *Network Nodes* and *Parcellation Background* - are displayed in gray-scale.
  - Context Grayscale: Only the parcellations of context in partial networks are displayed in gray-scale.

The created *Spatial-Data-Driven Layouts* represent perspectives commonly used in neuroscience - sagittal and transversal. For the creation of the visualizations we took care to choose a set of layout parameters (described in Sections 5.5 and 6.7) per species and perspective that best reflects the findings of Ganglbergers case studies (Section 7.1).

The results produced by *Spatial-Data-Driven Layouts* of different networks, including the reproduction of the figures from the literature were evaluated by experts using a questionnaire. The questionnaire consists of four parts:

(S1) **Identifying Nodes/Connections:** The first part evaluates the efficiency of our layout. Using click-tasks, we asked the participants to find the target-node that yields the strongest connection to a given region. We measured the time the user needed to click on the found region. The surveyed region pairs were carefully selected so that the strongest connection per node was distinct and overlaps of the relevant edges did not lead to impairments. The visualizations include the whole brain from the transversal perspective in three representations: directly projected layout, *Spatial-Data-Driven Layouts* without background and *Spatial-Data-Driven Layouts* with background. We then compared the times of the three representations to test the efficiency of *Spatial-Data-Driven Layouts*.

(S2) **Visualization of Anatomical Context:** In this part we focus on the visualization of the anatomical context that includes figures of complete and partial brain networks. The tasks are to rank the figures by clarity and answer questions about suitability for a figure in a paper or for educational purposes. The networks include different numbers of regions and the partial brain networks are visualized with a varying amount of context. Differences in coloring based on connectivity or size of regions, as described in Section 5.5 are also assessed. For the partial networks, we want to examine how helpful the context is. In addition, the comparison between the left and right hemispheres of the brain will be evaluated. Furthermore, we compare artistically drawn figures from neuroscientific publications [GWC+12, SWT+20] with the figures generated by our tool (Figures 7.4 and 7.7).

(S3) **Edge Visualization:** This part evaluates the preference among multiple edge layouts. Participants are asked to rank figures including different numbers of edges (top 10 %, 20 %, or 30 % of connections). Additionally, we examine different edge routing layouts (straight arrows, organic edge routing, and orthogonal edge routing) based on suitability for publications.

(S4) **Demographic Data:** The last part includes personal questions like the current position and gender of the participant. We asked if the user is already familiar with the Allen Brain Atlas as we use its color scheme and abbreviations of region names. As the perception of color plays a crucial role in visualizations, we also check for color blindness.

The recruitment of participants was conducted by my colleagues Ganglberger and Wu. To reach as many participants as possible they sent out the user study in the form of a web survey. It was implemented with *Google Forms* by Wu.

7.2.2 Results

The evaluation of the user study was carried out by Ganglberger and Wu. To complete this work the results are summarised in the following sections. An overview of the results can be seen in Table 7.1. Again, details of the results are given in the work of Ganglberger et al. [GWW+22].

Evaluation Results of the Mouse Brain Network Visualization

For the mouse user study three female participants and five male participants were recruited by Ganglberger and Wu to investigate the effectiveness of our visualization. All participants are at a senior level - five postdoctoral researchers and three principal investigators - with certain domain knowledge. Among the participants are four neuroscientists, one bioinformatics scientist, one computational biology scientist, and two computer scientists. Six participants are familiar with the *Allen Mouse Brain Common Coordinate Framework* [WDL+20]. As a side note, two neuroscientists have a red-green deficiency, so it was also investigated whether this affects task performance with our visualization.

Part (S1) consists of three configuration settings of a network covering structural connectivity over the whole brain, including (a) a directly projected layout, (b) *Spatial-Data-Driven Layouts* without background, and (c) *Spatial-Data-Driven Layouts* with the background. There are two click tasks per layout to find a particular node, resulting in a total of six

tasks.
All participants did an excellent job as only one participant made one mistake, which occurred when creating the graph directly from the projection (a). For configuration (a) the median completion time was 31 seconds, for (b) 24 seconds, and for (c) 30.5 seconds, respectively. The processing time for (a) is longer than in (b), which is explained by Ganglberger with the occlusions in the direct projection. He stated that in case (c) the time increase compared to (b) could be caused by the colored background, which increases visual complexity. One participant also pointed out that the multitude of strong colors makes it difficult to read the connectivity of entities in the diagram.

In Part (S2) different settings for the visualisation were tested. It consists of four questions about different levels of the hierarchy, two questions about varaying levels of background detail (number of parcellations, determined by the *Color Count* parameter), three questions about different *Context Ratios* for sub-networks, and additional questions about their coloring. For the different hierarchy levels, sagittal and transversal perspectives, and the three configuration settings were tested.
On average, six participants thought our approach made the most visual sense at the coarse, medium, and detailed levels of the hierarchy, respectively. There was no clear outcome regarding the number of parcellations, as participants tended to prefer the extremes ,i.e., the lowest or highest level of background detail. Half of the participants preferred to read sub-networks with the highest proportion of background, i.e., with the highest *Context Ratio*. The two neuroscientists with color weaknesses preferred simple sub-networks without any background. Seven out of eight participants preferred a mixture of gray and colored backgrounds compared to full color images. The usefulness of our visualizations was rated on a scale from one (poor) to five (good). The background for spatial orientation was rated with a score of 4.05. When visualizing the graph in Figure 7.4 to replicate the hand-crafted image from an existing publication [RN13] an average score of 3.36 was received, which Ganglberger considered as quite promising. In addition to these tasks participants were also asked for subjective feedback. Seven participants are convinced that our visualisations are helpful for educational purposes, while they are undecided about their use as figures in a paper.

In Part (S3) both different numbers of edges and different types of edge routings are compared. The participants preferred a smaller number of edges for an increased clarity. Five out of eight participants preferred the organic edge routing, as this layout allows for easy tracking of the edges in a graph drawing due to the avoided overlap between edges and nodes. The remaining two participants preferred the straight edges.

Evaluation Results of the Human Brain Network Visualization

The user study concerning the human brain networks was similarly designed to that of the mouse brain. In total, Ganglberger and Wu recruited three female and three male participants for this experiment. Among the participants are five postdoctoral researchers and one principal investigator. They include two neuroscientists, one bioinformatics scientist, one computational biology scientist, and two computer scientists. Half of the participants have already worked with the *Allen Human Reference Atlas* [DRS$^+$16] and are familiar with it. One of the participants has a red-green color weakness.

In Part (S1) all participants managed to identify the correct nodes. For each layout (a) - (c) two tasks were specified and the time to solve them was measured. Configuration (a) took a median of 43 seconds, configuration (b) 32 seconds, and (c) 30 seconds. Again configuration (b) took less time than configuration (a). However, this time configuration (c) provided a slightly quicker response than (b). Ganglberger stated that this could be argued with the more spherical shape of the human brain compared to the one of the mouse. Due to the higher density of regions, the displacement of the nodes is greater. The contextual background supports the participant's orientation to allocate specific regions and connections rather than distracting them.

In Part (S2) the results were consistent with the mouse user study. Again, the usefulness of our results was rated on a scale from one (poor) to five (good). Regarding the usefulness of the contextual background, a score of 4.58 was received, which is even higher than the one for the mouse brain network visualisation. Again, an approximated hand-crafted image from a publication [GWC$^+$12], shown in Figure 7.7, was evaluated. It received an average score of 3.33. The score is a bit lower than for the mouse, which could have two causes according to Ganglberger: (1) The number of participants was small, which leads to no significant difference or (2) the complexity is higher in terms of the number of nodes and edges in the human figure.

The results of Part (S3) are again consistent with those of the mouse user study. The participants like visualisations with fewer edges and thus less visual clutter. Again, most of the participants preferred organic edge routing.

General Feedback

In the user studies, participants also had the opportunity to give general feedback [GWW$^+$22]. According to Ganglberger, one participant stated "*Good work with the nice, comprehensive visualizations.*". Another participant mentioned that "*the honeycomb parcellation is very nice, the visibility of the edges in the long-range is quite tricky.*". Another participant suggested to "*summarize these arrows into one arrow, pointing to some meaningful position in the target hierarchy, and only then branching out to each target area separately.*". Therefore an improvement would be to bundle edges of nodes that project between two brain regions on a higher hierarchy level.

Although a user study for the *D. melanogaster* larval brain network was not conducted, Ganglberger already received positive feedback from the neuroscientific community. This encourages us to further develop our approach for the neural circuits, to be able to apply it in the future to species that are neurobiologically very distinct from mammals.

	Mouse	**Human**
Participants	8 (3 female, 5 male)	6 (3 female, 3 male)
Part (S1): Median Task Completion Time		
(a) directly projected layout	31s	43s
(b) SDD[a] layout without background	24s	32s
(c) SDD[a] layout with background	30.5s	30s
Part (S2): Anatomical Context		
preferred our approach over 2D projection on different hierarchy levels (votes)	6	5
Number of Background Regions least | middle | most (votes)	5 | 2 | 1	0 | 6 | 0
Number of *Context Nodes* least | most (votes)	2 | 6	1 | 5
Context Nodes background colored | gray (votes)	1 | 7	1 | 5
helpfulness of background scores[b]	4.05	4.58
visual appealing of re-imagined figure[b]	3.36	3.33
Part (S3): Preferred Edge Routing (votes)		
straight (clarity | paper | education)	3 | 3 | 3	2 | 2 | 2
organic (clarity | paper | education)	5 | 5 | 5	4 | 4 | 4
orthogonal (clarity | paper | education)	0 | 0 | 0	0 | 0 | 0
Part (S4): Demographics		
female | male	3 | 5	3 | 3
postdoc | principal investigators	5 | 3	5 | 1
neurosci. | bioinf.| comp. sci.	4 | 2 | 2	2 | 2 | 2
red-green color weakness	2	1

[a]SDD = *Spatial-Data-Driven* [b]1 (poor) to 5 (good)

Table 7.1: Results of the user study of Part (S1) - Identifying Nodes/Connections; Part (S2) - Visualization of Anatomical Context; Part (S3) - Edge Visualization; Part (S4) - Demographic Data

CHAPTER 8

Discussion

In the following section we recap how our requirements in Section 5.1 could be reached with the help of the evaluation carried out. We also consider the usefulness of the visual encoding, limitations, and possible further improvements of our approach.
The case studies in Section 7.1 show the relevance of our work in neuroscientific research on different species. Their potential is highlighted by the results of our user studies in Section 7.2. By evaluating the studies for two different species, mouse and human, we can show that our methodology can be applied in a data-driven way and thus species-independent (**R2**, **R3**).

8.1 Visual Encoding

Node-link diagrams are typical visualization approaches for neural networks in two-dimensional space, but require an appropriate layout to display information. Surveyed experts found our representations during the case studies intuitive for perceiving arrangements of spatial relationships (**R1**).
In the studies our *Spatial-Data-Driven Layouts* show a positive effect on the perception of brain networks. The experts were able to solve network-orientation tasks faster with the help of our layouts than with a simple 2D projection of the 3D networks. Searching nodes was simplified by the placement of the nodes in relation to each other, which is necessary for the orientation within a graph (**R4**).
To further improve the representation of the anatomical context, we generate a convex hull around the graph and compute a Voronoi diagram to divide the background into major anatomical regions of the brain. These represent hierarchically higher regions and visually group nodes belonging to the same region.
The user study showed that there is no clear preference for the coarseness of the background regions (determined by the *Color Count* parameter). The participants preferred either the highest or the lowest number of regions, i.e., number of colors (**R6**). One issue

resulted from the background obscuring the edges, which was identified in part (S1) of the mouse user study. The users needed on average a little more time to find the regions of interest if the visualization included the background.

For the visualization of sub-networks we created the option to include brain regions that are not part of the sub-network in the background (defined by the *Context Ratio* parameter). This preserves the overall shape of the brain and allows the user comparison of different sub-networks and the complete network (**R5**, **R7**). The feature was considered very useful, with a score of 4.05 for the mouse brain and 4.58 for the human brain, on a scale from one (poor) to five (good). A higher ratio of contextual nodes was preferred, as it resembles the shape of the network covering the entire brain. Displaying this additional context in shades of gray (*Gray-scale* parameter set to *contextGray*) is an option to keep the viewer's focus on the actual sub-network. This partially colored version was preferred by the majority of participants.

8.2 Limitations

Our approach is fully spatial-data-driven and does not require manual readjustment of nodes. The parameters, such as the *Context Ratio* and the *Color Count*, influence the anatomical context, but hardly the arrangement of nodes per se.
Our approach to lay out the network follows the idea of shape-based metrics by Eades et al. [EHNK17]. They describe the design of experiments to fully validate shape-based metrics to remain an open problem, as testing shape-based metrics is difficult. A significant problem is to determine which tasks are appropriate for large-graph visualization. Further difficulties arise because the results of such an experiment could be highly dependent on the specific tasks used. This is also the case for *Spatial-Data-Driven Layouts*.
The arrangement of the nodes is influenced by the parameters of the force-directed layout algorithm. During the development of our methodology, we realized that the parameters strongly depend on the *Parcellation-derived Connectivity* and the size and density of the graph. Choosing the set of parameters is an essential part of this work and is crucial for the resulting visualizations. Since the networks differ strongly between the species and the perspectives they have to be chosen for each species-perspective combination to achieve good results. This is rather time-consuming, since the multitude of parameters has to be combined and the adjustment of individual parameters is not always intuitive. Furthermore, due to the fixed set of layouting parameters and the data-driven approach, sub-optimal results may be produced for certain networks. A straightforward solution, that we also applied to facilitate the search for a suitable set of parameters, is to interactively control the parameters with the help of sliders, as seen in most applications for force-directed layouts. This is easily implemented allowing the user to adjust the parameters if the result is indeed not satisfactory.
One disadvantage of the CoSE-Bilkent implementation is that it does not consider the weighting of the edges. Although the data is available we had to work around this drawback in our approach by limiting the number of connections per node to a small fraction by choosing the strongest ones. This, obviously, leads to a loss of information

that could affect the result negatively.
The application of the *Aesthetic Layout* is only helpful if the underlying network is dense and requires strong forces in the main layout step. The *Aesthetic Layout* can even disturb some arrangements, for example very flat and distinctive shapes like in the *D. melanogaster* larval brain. The number of iterations or if it should be applied at all has to be evaluated for each new species or perspective, which could be added in the future. Another requirement for applying *Spatial-Data-Driven Layouts* is the availability of brain hierarchy data as well as the *Parcellation-derived Connectivity*, which needs to be calculated for a new species.
The performance of the algorithm that draws the *Parcellation Background* (as described in Section 6.8) has potential to be improved. We decided to use only three canvases, but this requires relatively large data structures, since a list of all the Voronoi cells for each parcellation in the *Color Canvas* and *Parcellation Outline Canvas*, and for each hemisphere in the *Hemisphere Outline Canvas* was used to create the drawing. A more performant solution could be to save per parcellation and hemisphere only the cells that are actually in it and then draw these cells on a separate canvas. This would result in a higher number of canvases (one per parcellation or hemisphere), but with far fewer iterations for drawing the cells themselves.

8.3 Outlook

Our studies have shown that the generated figures by *Spatial-Data-Driven Layouts* were well received by the community. *Spatial-Data-Driven Layouts* can be used to approximate hand-crafted illustrations. Compared to creating such illustrations by hand, rendering a network with *Spatial-Data-Driven Layouts* takes only a couple of seconds. Because our work is considered suitable for educational purposes, it is valuable not only to domain experts, but also to their audiences to gain an understanding of brain networks that would otherwise be difficult to grasp.
Integration of interactivity could bring further benefits in direct neuroscience research beyond print media. By giving the user the possibility to change the layouting parameters to a certain extent, they can further customize the visualization to their needs. Another interactive approach could include extending or collapsing regions and parcellations. This functionality is already available as Cytoscape.js extension [DKS+18]. Another step could be to interactively change the hierarchy level to determine the level of detail of the network visualization. In terms of connectivity, one could interactively filter certain connections or bundle connections between hierarchy levels to focus on the essentials and reduce clutter. Highlighting the flow of information to and from a node could also be a helpful feature to support brain-network research. Other data, such as gene expression, could be overlaid to provide additional information to enable novel visual-analysis workflows. As the scientific field of connectomics is developing rapidly, the data quality is also increasing. The data used in our tool can be easily updated to provide even better results in the future. At the time of the user study, no data on neighborhood-based *Parcellation-derived Connectivity* for the human-brain network was available, so it was not possible to evaluate the results.

Meanwhile, recent publications released voxel-level common coordinate frameworks with region-level annotations for the human brain [AMBZ20, WDL$^+$20], which would be suitable for our application in the future.
Regarding *D. melanogaster* larval-brain networks, our presented visualizations of the neural circuits are the first step in this direction. Further developments of our methodology to include markers for input and output locations or other encodings of neurons spanning multiple brain regions could make our approach a valuable addition to the currently used circuit diagrams.

Finally, our approach can be used not only for the application to spatial brain networks. In principle, three-dimensional networks from different domains can be flattened to two dimensions. Even without a hierarchy within the network and thus without rendering any background as context, nodes can be layered with respect to their spatial relationships to provide spatial orientation.

8.4 Conclusion

We believe that *Spatial-Data-Driven Layouts* are helpful in neuroscience, not only to visualize neuronal brain networks, but also to give new insights for varying analysis goals. It might replaces the tedious, artistic creation of graphics and can be applied in only a couple of seconds on brain networks of different species. In the future the tool can be used for other species by generating a single set of base parameters for the layout process. Using interactive elements, details could be additionally adapted to the needs of neurobiologists

List of Figures

List of Tables

Appendix

Anatomical Layout Parameters

Mouse Brain Network

Transversal Perspective

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: true,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 4000,
        idealEdgeLength: 45,
        edgeElasticity: 0.0001,
        gravity: 2,
        gravityRange: 5,
        initialEnergyOnIncremental: 0.2,
}
```

Sagittal Perspective

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: false,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 1000,
        idealEdgeLength: 10,
        edgeElasticity: 0.03,
        gravity: 0.5,
        gravityRange: 3,
        initialEnergyOnIncremental: 0.2,
        nestingFactor: 2,
        gravityCompound: 1.5,
        gravityRangeCompound: 1.2,
}
```

Human Brain Network

Transversal Perspective

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: false,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 4000,
        idealEdgeLength: 60,
        edgeElasticity: 0.001,
        gravity: 0,
        gravityRange: 0,
        initialEnergyOnIncremental: 0.3
}
```

Sagittal Perspective

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: false,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 800,
        idealEdgeLength: 12,
        edgeElasticity: 0.1,
        gravity: 2,
        gravityRange: 1,
        initialEnergyOnIncremental: 0.5,
        nestingFactor: 1,
        gravityCompound: 1.5,
        gravityRangeCompound: 1.2
}
```

D. Melanogaster Larval Brain Network

Transversal Perspective

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: false,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 10000,
        idealEdgeLength: 500,
        edgeElasticity: 0.1,
        gravity: 0.01,
        gravityRange: 0.1,
        initialEnergyOnIncremental: 0.01,
}
```

Sagittal Perspective

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: false,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 10000,
        idealEdgeLength: 200,
        edgeElasticity: 0.01,
        gravity: 0.01,
        gravityRange: 0.1,
        initialEnergyOnIncremental: 0.06,
        nestingFactor: 0.5,
        gravityCompound: 1,
        gravityRangeCompound: 1,
}
```

Aesthetic Layout Parameters

```
{
        name: "cose-bilkent",
        nodeDimensionsIncludeLabels: false,
        randomize: false,
        numIter: 2500,
        tile: false,
        animate: false,
        nodeRepulsion: 1000,
        idealEdgeLength: 20,
        edgeElasticity: 0.03,
        gravity: 2,
        gravityRange: 1,
        initialEnergyOnIncremental: 0.05,
        nestingFactor: 1.5,
        gravityCompound: 0,
        gravityRangeCompound: 0,
}
```

Bibliography

[ABC+20] Larry F Abbott, Davi D Bock, Edward M Callaway, Winfried Denk, Catherine Dulac, Adrienne L Fairhall, Ila Fiete, Kristen M Harris, Moritz Helmstaedter, Viren Jain, et al. The mind of a mouse. *Cell*, 182(6):1372–1376, 2020.

[ACG+09] Frederico AC Azevedo, Ludmila RB Carvalho, Lea T Grinberg, José Marcelo Farfel, Renata EL Ferretti, Renata EP Leite, Wilson Jacob Filho, Roberto Lent, and Suzana Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541, 2009.

[All] Allen brain atlas. `http://atlas.brain-map.org/`. Accessed: 18.11.2022.

[AMBZ20] Katrin Amunts, Hartmut Mohlberg, Sebastian Bludau, and Karl Zilles. Julich-brain: A 3d probabilistic atlas of the human brain's cytoarchitecture. *Science*, 369(6506):988–992, 2020.

[ana] Libraries Michigan State University foundation of neuroscience. `https://openbooks.lib.msu.edu/neuroscience/chapter/anatomical-terminology/`. Accessed: 18.11.2022.

[BAAK+13] Johanna Beyer, Ali Al-Awami, Narayanan Kasthuri, Jeff W Lichtman, Hanspeter Pfister, and Markus Hadwiger. ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2868–2877, 2013.

[BP06] Ulrik Brandes and Christian Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *International Symposium on Graph Drawing*, pages 42–53. Springer, 2006.

[BRSG07] Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The aesthetics of graph visualization. *Proceedings of Computational Aesthetics in Graphics, Visualization, and Imaging*, pages 57–64, 01 2007.

[BWS+19] Alexandra Badea, Wenlin Wu, Jordan Shuff, Michele Wang, Robert J Anderson, Yi Qi, G Allan Johnson, Joan G Wilson, Serge Koudoro, Eleftherios Garyfallidis, et al. Identifying vulnerable brain networks in mouse models of genetic risk factors for late onset Alzheimer's disease. *Frontiers in neuroinformatics*, 13:72, 2019.

[Cha96] Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.

[cos] Cytoscape - cose-bilkent. `https://github.com/cytoscape/cytoscape.js-cose-bilkent`. Accessed: 18.11.2022.

[cyt] Cytoscape - an open source software platform for visualizing complex networks. `https://cytoscape.org/`. Accessed: 18.11.2022.

[CZQ+08] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1277–1284, 2008.

[dev] MND Web Docs canvasrenderingcontext2d.globalcompositeoperation. `https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/globalCompositeOperation`. Accessed: 18.11.2022.

[DGC+04] Ugur Dogrusoz, Erhan Giral, Ahmet Cetintas, Ali Civril, and Emek Demir. A compound graph layout algorithm for biological pathways. In *International Symposium on Graph Drawing*, pages 442–447. Springer, 2004.

[DGC+09] Ugur Dogrusoz, Erhan Giral, Ahmet Cetintas, Ali Civril, and Emek Demir. A layout algorithm for undirected compound graphs. *Information Sciences*, 179(7):980–994, 2009.

[DKS+18] Ugur Dogrusoz, Alper Karacelik, Ilkin Safarli, Hasan Balci, Leonard Dervishi, and Metin Can Siper. Efficient methods and readily customizable libraries for managing complexity of large networks. *Plos one*, 13(5):e0197238, 2018.

[DPB+14] Nibaran Das, Sandip Pramanik, Subhadip Basu, Punam Kumar Saha, Ram Sarkar, Mahantapas Kundu, and Mita Nasipuri. Recognition of handwritten bangla basic characters and digits using convex hull based feature set. *arXiv preprint arXiv:1410.0478*, 2014.

[DRS+16] Song-Lin Ding, Joshua J Royall, Susan M Sunkin, Lydia Ng, Benjamin AC Facer, Phil Lesnar, Angie Guillozet-Bongaarts, Bergen McMurray, Aaron Szafer, Tim A Dolbeare, et al. Comprehensive cellular-resolution atlas of the adult human brain. *Journal of Comparative Neurology*, 524(16):3127–3481, 2016.

[DSH11] Gopikrishna Deshpande, Priya Santhanam, and Xiaoping Hu. Instantaneous and causal connectivity in resting state brain networks derived from functional MRI data. *Neuroimage*, 54(2):1043–1052, 2011.

[Ead84] Peter Eades. A heuristic for graph drawing. *Congressus numerantium*, 42:149–160, 1984.

[EHNK17] Peter Eades, Seok-Hee Hong, An Nguyen, and Karsten Klein. Shape-based quality metrics for large graph visualization. *J. Graph Algorithms Appl.*, 21(1):29–53, 2017.

[FGT$^+$19] Diana Furcila, Marcos García, Cosmin Toader, Juan Morales, Antonio LaTorre, Ángel Rodríguez, Luis Pastor, Javier DeFelipe, and Lidia Alonso-Nanclares. Intool explorer: an interactive exploratory analysis tool for versatile visualizations of neuroscientific data. *Frontiers in neuroanatomy*, 13:28, 2019.

[FLH$^+$16] Max Franz, Christian T Lopes, Gerardo Huck, Yue Dong, Onur Sumer, and Gary D Bader. Cytoscape. js: a graph theory library for visualisation and analysis. *Bioinformatics*, 32(2):309–311, 2016.

[FLN$^+$15] David Feng, Chris Lau, Lydia Ng, Yang Li, Leonard Kuan, Susan M Sunkin, Chinh Dang, and Michael Hawrylycz. Exploration and visualization of connectivity in the adult mouse brain. *Methods*, 73:90–97, 2015.

[FR91] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[FT01] Ken-ichiro Fukuda and Toshihisa Takagi. Knowledge representation of signal transduction pathways. *Bioinformatics*, 17(9):829–837, 2001.

[GHK10] Emden R Gansner, Yifan Hu, and Stephen Kobourov. Gmap: Visualizing graphs and clusters as maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208. IEEE, 2010.

[GKHB20] Florian Ganglberger, Joanna Kaczanowska, Wulf Haubensak, and Katja Bühler. A data structure for real-time aggregation queries of big brain networks. *Neuroinformatics*, 18(1):131–149, 2020.

[GKP$^+$18] Florian Ganglberger, Joanna Kaczanowska, Josef M Penninger, Andreas Hess, Katja Bühler, and Wulf Haubensak. Predicting functional neuroanatomical maps from fusing brain networks with genetic information. *Neuroimage*, 170:113–120, 2018.

[Gra72] Ronald Lewis Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Info. Pro. Lett.*, 1:132–133, 1972.

[GS85] Ardeshir Goshtasby and George C Stockman. Point pattern matching using convex hull edges. *IEEE Transactions on Systems, Man, and Cybernetics*, (5):631–637, 1985.

[GSF+19] Florian Ganglberger, Nicolas Swoboda, Lisa Frauenstein, Joanna Kaczanowska, Wulf Haubensak, and Katja Bühler. Braintrawler: A visual analytics framework for iterative exploration of heterogeneous big brain data. *Computers & Graphics*, 82:304–320, 2019.

[GWC+12] Anthony L. Gotter, Andrea L. Webber, Paul J. Coleman, John J. Renger, and Christopher J. Winrow. International union of basic and clinical pharmacology. LXXXVI. Orexin receptor function, nomenclature and pharmacology. *Pharmacological Reviews*, 64(3):389–420, 2012.

[GWW+22] Florian Ganglberger, Monika Wißmann, Hsiang-Yun Wu, Nicolas Swoboda, Andreas Thum, Wulf Haubensak, and Katja Bühler. Spatial-data-driven layouting for brain network visualization. *Computers & Graphics*, 2022.

[Hal70] Kenneth M Hall. An r-dimensional quadratic placement algorithm. *Management science*, 17(3):219–229, 1970.

[HET12] Christophe Hurter, Ozan Ersoy, and Alexandru Telea. Graph bundling by kernel density estimation. In *Computer Graphics Forum*, volume 31, pages 865–874. Wiley Online Library, 2012.

[HHML06] Suzana Herculano-Houzel, Bruno Mota, and Roberto Lent. Cellular scaling rules for rodent brains. *Proceedings of the National Academy of Sciences*, 103(32):12138–12143, 2006.

[HLGB+12] Michael J. Hawrylycz, Ed S. Lein, Angela L. Guillozet-Bongaarts, Elaine H. Shen, Lydia Ng, Jeremy A. Miller, Louie N. Van De Lagemaat, Kimberly A. Smith, Amanda Ebbert, Zackery L. Riley, et al. An anatomically comprehensive atlas of the adult human brain transcriptome. *Nature*, 489(7416):391–399, 2012.

[hol] Smooth polygon convex hull. `http://bl.ocks.org/hollasch/9d3c098022f5524220bd84aae7623478`. Accessed: 18.11.2022.

[Hol06] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[HPS16] George T. Heineman, Gary Pollice, and Stanley Selkow. *Algorithms in a nutshell: A practical guide*. O'Reilly Media, Inc., 2016.

[htm] w3schools - html canvas graphics. `https://www.w3schools.com/html/html5_canvas.asp`. Accessed: 18.11.2022.

[hum] Human brain project. https://www.humanbrainproject.eu. Accessed: 18.11.2022.

[HVW09] Danny Holten and Jarke J Van Wijk. Force-directed edge bundling for graph visualization. In *Computer graphics forum*, volume 28, pages 983–990. Wiley Online Library, 2009.

[HZZ+20] Qi Huang, Jian Zhang, Tianhao Zhang, Hui Wang, and Jianhua Yan. Age-associated reorganization of metabolic brain connectivity in chinese children. *European journal of nuclear medicine and molecular imaging*, 47(2):235–246, 2020.

[imp] IMP - research institute of molecular pathology. https://www.imp.ac.at/. Accessed: 18.11.2022.

[int] Interactive Atlas Viewer. http://atlas.brain-map.org/. Accessed: 18.11.2022.

[Jar73] Ray A Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information processing letters*, 2(1):18–21, 1973.

[JDL09] Radu Jianu, Cagatay Demiralp, and David Laidlaw. Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1449–1456, 2009.

[JDL12] Radu Jianu, Cagatay Demiralp, and David H Laidlaw. Exploring brain connectivity with two-dimensional neural maps. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):978–987, 2012.

[JVHB14] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6):e98679, 2014.

[KDBD17] Cihan Küçükkeçeci, Ugur Dogrusoz, Esat Belviranli, and Alptug Dilek. Chisio: A compound graph editing and layout framework. *CoRR*, abs/1708.07762, 2017.

[KK+89] Tomihisa Kamada, Satoru Kawai, et al. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.

[KLH18] Ki-Hwan Kim, Eun-Hee Lee, and Song-You Hong. Potential of voronoi diagram for the conserved remapping of precipitation. *Monthly Weather Review*, 146(7):2237–2246, 2018.

[Kob12] Stephen G. Kobourov. Spring embedders and force directed graph drawing algorithms. *ArXiv*, abs/1201.3011, 2012.

[Kob13] Stephen G. Kobourov. Force-directed drawing algorithms. In *Handbook of Graph Drawing and Visualization*, 2013.

[KRM+17] Johannes F. Kruiger, Paulo E. Rauber, Rafael Messias Martins, Andreas Kerren, Stephen Kobourov, and Alexandru C. Telea. Graph layouts by t-SNE. In *Computer Graphics Forum*, volume 36, pages 283–294. Wiley Online Library, 2017.

[Kru80] Joseph B. Kruskal. Designing network diagrams. In *Proc. 1st General Conf. On Social Graphics, 1980*, pages 22–50. US Dept. of the Census, 1980.

[KZC+17] Johnson JG Keiriz, Liang Zhan, Morris Chukhman, Olu Ajilore, Alex D Leow, and Angus G Forbes. Exploring the human connectome topology in group studies. *ArXiv*, abs/1706.10297, 2017.

[lar] Larvalbrain. `http://www.larvalbrain.org/`. Accessed: 18.11.2022.

[LD50] Peter Gustav Lejeune Dirichlet. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1850(40):209–227, 1850.

[LD11] Jeff W. Lichtman and Winfried Denk. The big and the small: challenges of imaging the brain's circuits. *Science*, 334(6056):618–623, 2011.

[MBB+16] Sugeerth Murugesan, Kristopher Bouchard, Jesse A Brown, Bernd Hamann, William W Seeley, Andrew Trujillo, and Gunther H Weber. Brain modulyzer: interactive visual analysis of functional brain connectivity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(4):805–818, 2016.

[McG15] Tim McGraw. Graph-based visualization of neuronal connectivity using matrix block partitioning and edge bundling. In *International Symposium on Visual Computing*, pages 3–13. Springer, 2015.

[MELS95] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210, 1995.

[MPB+19] G. Elisabeta Marai, Bruno Pinaud, Katja Bühler, Alexander Lex, and John H. Morris. Ten simple rules to create biological network figures for communication. *Plos computational biology*, 15(9), 2019.

[NW18] Benjamin Naujoks and Hans-Joachim Wünsche. An orientation corrected bounding box fit based on the convex hull under real time constraints. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2018.

[OHN+14] Seung Wook Oh, Julie A Harris, Lydia Ng, Brent Winslow, Nicholas Cain, Stefan Mihalas, Quanxin Wang, Chris Lau, Leonard Kuan, Alex M Henry, et al. A mesoscale connectome of the mouse brain. *Nature*, 508(7495):207–214, 2014.

[rea] React - a javascript library for building user interfaces. `https://reactjs.org/`. Accessed: 18.11.2022.

[Rip19] Gwendolyn Rippberger. Data-driven anatomical layouting of brain network graphs. *Institute of Computer Graphics and Algorithms, TU Wien*, April 2019.

[RN13] Scott J. Russo and Eric J. Nestler. The brain reward circuitry in mood disorders. *Nature Reviews Neuroscience*, 14(9):609–625, 2013.

[RP21] Joshua I Raji and Christopher J Potter. The number of neurons in drosophila and mosquito brains. *Plos one*, 16(5):e0250381, 2021.

[RYG+21] Micha Sam Brickman Raredon, Junchen Yang, James Garritano, Meng Wang, Dan Kushnir, Jonas Christian Schupp, Taylor Sterling Adams, Allison M Greaney, Katherine L Leiby, Naftali Kaminski, et al. Connectome: computation and visualization of cell-cell signaling topologies in single-cell systems data. *bioRxiv*, 2021.

[SBA+13] Stephen M. Smith, Christian F. Beckmann, Jesper Andersson, Edward J. Auerbach, Janine Bijsterbosch, Gwenaëlle Douaud, Eugene Duff, David A. Feinberg, Ludovica Griffanti, Michael P. Harms, et al. Resting-state fMRI in the human connectome project. *Neuroimage*, 80:144–168, 2013.

[SBS+13] Johannes Sorger, Katja Bühler, Florian Schulze, Tianxiao Liu, and Barry Dickson. neuromap—interactive graph-visualization of the fruit fly's neural circuit. In *2013 IEEE Symposium on Biological Data Visualization (BioVis)*, pages 73–80. IEEE, 2013.

[Seu12] Sebastian Seung. *Connectome: How the Brain's Wiring Makes Us who We are*. Mariner Book. Houghton Mifflin Harcourt, 2012. ISBN: 9780547508184.

[SGTB+16] Daniel Sánchez-Gutiérrez, Melda Tozluoglu, Joseph D Barry, Alberto Pascual, Yanlan Mao, and Luis M Escudero. Fundamental physical cellular constraints drive self-organization of tissues. *The EMBO journal*, 35(1):77–88, 2016.

[SM91] Kozo Sugiyama and Kazuo Misue. Visualization of structural information: Automatic drawing of compound digraphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(4):876–892, 1991.

[SM95] Kozo Sugiyama and Kazuo Misue. A generic compound graph visualizer/manipulator: D-abductor. In *International Symposium on Graph Drawing*, pages 500–503. Springer, 1995.

[Sno55] John Snow. On the mode of communication of cholera (2nd ed.). *John Churchill, New Burlington Street, London*, 1855.

[SPD+17] Pierre-Hugues Stefanuto, Katelynn A Perrault, Lena M Dubois, Benjamin L'Homme, Catherine Allen, Caitriona Loughnane, Nobuo Ochiai, and Jean-François Focant. Advanced method optimization for volatile aroma profiling of beer using two-dimensional gas chromatography time-of-flight mass spectrometry. *Journal of Chromatography A*, 1507:45–52, 2017.

[SSC+05] Raymond Salvador, John Suckling, Martin R Coleman, John D Pickard, David Menon, and ED Bullmore. Neurophysiological architecture of functional magnetic resonance images of human brain. *Cerebral cortex*, 15(9):1332–1342, 2005.

[STS+19] Leon Stefanovski, Paul Triebkorn, Andreas Spiegler, Margarita-Arimatea Diaz-Cortes, Ana Solodkin, Viktor Jirsa, Anthony Randal McIntosh, Petra Ritter, Alzheimer's Disease Neuroimaging Initiative, et al. Linking molecular pathways and large-scale computational modeling to assess candidate disease mechanisms and pharmacodynamics in Alzheimer's disease. *Frontiers in computational neuroscience*, 13:54, 2019.

[STT81] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.

[SWT+20] Michael Schleyer, Alićе Weiglein, Juliane Thoener, Martin Strauch, Volker Hartenstein, Melisa Kantar Weigelt, Sarah Schuller, Timo Saumweber, Katharina Eichler, Astrid Rohwedder, et al. Identification of dopaminergic neurons that can both establish associative memory and acutely terminate its behavioral expression. *Journal of Neuroscience*, 40(31):5990–6006, 2020.

[TDBB88] Roberto Tamassia, Giuseppe Di Battista, and Carlo Batini. Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):61–79, 1988.

[THZ+13] Olga Tymofiyeva, Christopher P. Hess, Etay Ziv, Patricia N. Lee, Hannah C. Glass, Donna M. Ferriero, A. James Barkovich, and Duan Xu. A DTI-based template-free cortical connectome study of brain maturation. *PloS one*, 8(5):e63310, 2013.

[Tut63] William Thomas Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3(1):743–767, 1963.

[VDZCT16] Matthew Van Der Zwan, Valeriu Codreanu, and Alexandru Telea. Cubu: Universal real-time bundling for large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 22(12):2550–2563, 2016.

[Vor08] Georges Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. premier mémoire. sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(133):97–102, 1908.

[vrv] VRVis - zentrum für virtual reality und visualisierung. `https://vrvis.at/`. Accessed: 18.11.2022.

[WBLR07] Tobias Wittkop, Jan Baumbach, Francisco P. Lobo, and Sven Rahmann. Large scale clustering of protein sequences with FORCE - a layout based heuristic for weighted cluster editing. *BMC bioinformatics*, 8(1):1–12, 2007.

[WDL+20] Quanxin Wang, Song-Lin Ding, Yang Li, Josh Royall, David Feng, Phil Lesnar, Nile Graddis, Maitham Naeemi, Benjamin Facer, Anh Ho, et al. The Allen mouse brain common coordinate framework: a 3D reference atlas. *Cell*, 181(4):936–953, 2020.

[WNV20] Hsiang-Yun Wu, Martin Nöllenburg, and Ivan Viola. Multi-level area balancing of clustered graphs. *IEEE Transactions on Visualization and Computer Graphics*, 28:2682–2696, 2020.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[WYY15] Jieting Wu, Lina Yu, and Hongfeng Yu. Texture-based edge bundling: A web-based approach for interactively visualizing large graphs. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2501–2508, 2015.

[XWH13] Mingrui Xia, Jinhui Wang, and Yong He. Brainnet viewer: a network visualization tool for human brain connectomics. *PloS one*, 8(7), 2013.

[yfi] Cytoscape App Store - files layout algorithms. `https://apps.cytoscape.org/apps/yfileslayoutalgorithms`. Accessed: 18.11.2022.