Neuroinformatics
https://doi.org/10.1007/s12021-019-09428-9

ORIGINAL ARTICLE

A Data Structure for Real-Time Aggregation Queries of Big Brain Networks

Florian Johann Ganglberger[1] · Joanna Kaczanowska[2] · Wulf Haubensak[2] · Katja Bühler[1]

Abstract

Recent advances in neuro-imaging allowed big brain-initiatives and consortia to create vast resources of brain data that can be mined by researchers for their individual projects. Exploring the relationship between genes, brain circuitry, and behavior is one of the key elements of neuroscience research. This requires fusion of spatial connectivity data at varying scales, such as whole brain correlated gene expression, structural and functional connectivity. With ever-increasing resolution, these tend to exceed the past state-of-the art in size and complexity by several orders of magnitude. Since current analytical workflows in neuroscience involve time-consuming manual data-aggregation, incorporating efficient techniques for handling big connectivity data is a necessity. We propose a novel data structure enabling the interactive exploration of heterogeneous neurobiological connectivity data with billions of edges. Based on this data structure we realized *Aggregation Queries*, i.e. the aggregated connectivity from, to or between brain areas allows experts to compare the multimodal networks residing at different scales, or levels of hierarchically organized anatomical atlases. Executed on-demand on volumetric gene expression and connectivity data, they allow an interactive dissection of networks in real-time and based on their spatial context. The data structure is optimized in order to be accessible directly from the hard disk, since connectivity of large-scale networks typically exceeds the memory size of current consumer level PCs. This allows experts to embed and explore their own experimental data in the framework of public data resources without the need for their own large-scale infrastructure. Our data structure outperforms state-of-the-art graph engines in retrieving connectivity of arbitrary user defined local brain areas. We demonstrate the feasibility of our approach by analyzing fear-related functional neuroanatomy in mice. Further, we show its versatility by comparing multimodal brain networks linked to autism. Importantly, we achieve cross-species congruence in retrieving human psychiatric traits networks, which facilitates the selection of neural substrates to be further studied in mouse models.

Keywords Brain networks · Spatial data structures · Aggregation queries · Structural connectivity · Functional connectivity · Large networks · Hierarchical Parcellation · Big data · Interactive data mining

Introduction

Recent brain initiatives, such as the Allen Institute (Oh et al. 2014; Hawrylycz et al. 2012; Lein et al. 2007), the Human Brain Project (Markram et al. 2011), the WU-Minn Human Connectome Project (Van Essen et al. 2013), and the China Brain Project (Poo et al. 2016), have accumulated large sets of brain data for neuroscience research. Visual analytics emerges as a promising tool to mine this multimodal neurobiological data for insight into the functional organization of the brain (K. Li et al. 2012). Such technologies allow the direct exploration of relations between genes, neuronal circuitry and brain function and can quickly add context to experimental findings. However, the major challenges for visual analytic workflows arise from accessing, fusing and visualizing spatial brain data, such as brain gene expression, structural and functional connectivity, and non-spatial data, like genes associated with a given brain function. A particular challenge when exploring such heterogeneous neurobiological data is the alignment of their spatial reference. Depending on the data acquisition

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12021-019-09428-9) contains supplementary material, which is available to authorized users.

✉ Florian Johann Ganglberger
ganglberger@vrvis.at

[1] VRVis Research Center, Vienna, Austria

[2] Research Institute of Molecular Pathology, Vienna, Austria

Published online: 25 June 2019

technique, it can be volumetric or region-wise, and different resources are not necessarily in the same reference space. This can lead to time consuming workflows that involve manual aggregation of the data that do not work continuously on different scales.

The entry point for many neuroscience workflows are local brain regions/areas and/or gene expression sites (sites where the gene creates products, such as proteins (Lein et al. 2007)) that are linked to a specific brain function. The functional annotations of such sites are typically the results of neuronal recording, imaging, optogenetics and behavioral neurogenetic studies (e.g. amygdala subnuclei in emotional processing (Haubensak et al. 2010; Kim et al. 2017)). The knowledge of where these local regions/areas- and/or primary expression sites are connected to, is a first step to relate them to a specific brain circuit or a particular function. This information is encoded in so called spatial networks. In these networks, nodes represent regions/areas in the brain, while edges describe their structural (Oh et al. 2014), functional (Betzel and Bassett 2017) or genetic (Richiardi and Altmann 2015) relation/connectivity. Since the size of these networks increases squarely to the number of nodes, these networks can easily grow to hundreds of gigabytes, with billions of edges.

Comparing different types of connectivity is essential for identifying neural circuits. For example, two brain regions can have a high structural connectivity (a connection via neurons) but do not necessarily express the same genes (e.g. a so called ligand-receptor binding (Young and Wang 2004)). Depending on data acquisition techniques, different types of networks are not necessarily available at similar resolution and scale (Betzel and Bassett 2017). Besides being time-consuming, up-sampling networks to higher resolutions requires more storage space, while down-sampling to a lower resolution or even region-level would waive information. When operating on different anatomical scales, i.e. different levels of anatomical parcellation, it is necessary to perform cumulative operations on these networks (e.g. calculate region-wise connectivity from voxel-wise connectivity, aggregate voxel-connectivity of brain areas) to map the networks' common brain space. In this case large parts of the network need to be loaded and aggregated. The size and complexity of these networks created a need for sophisticated data handling techniques to allow further analyses and exploration (Bassett and Sporns 2017).

Several interactive frameworks for querying connectomic data in neuroscience have been published in recent years. The Allen Brain Institutes's BrainExplorer as well as its web interface (Oh et al. 2014) can identify pre-computed incoming and outgoing connections of pre-defined locations (injection sites) and anatomical regions in mice. Meso-scale source/target sites are visualized in 3D at voxel level. For quantitative examination this data is ordered by brain region and shown in a list. Although this is an easy-to-use tool for neurobiologists, results cannot be compared directly to other connectivity data or examined with respect to user-generated data. Other tools allow to locally explore the connectomes built by neurons traced on a single EM stack (volumetric electron microscopy) like CATMAID (Saalfeld et al. 2009) and ConnectomeExplorer (Beyer et al. 2013). They are working on a local level of a single network with a fixed scale. This also applies for Sherbseondy et al. (Sherbondy et al. 2005), who used queries on volumes of interest and pre-computed pathways to explore diffusion tensor imaging data, and Tauheed et al. (Tauheed et al. 2013), who developed tree-based spatial management techniques for dense spatial neuron simulations.

The problem of efficiently querying large-scale spatial networks was originally addressed by different domains, particularly on transportation/road networks (Barthelemy 2010). Early approaches in optimizing local queries on road network data were proposed in 1997 by Shekhar and Liu (Shekhar and Liu 1997). In principle, network nodes, and respectively their edges are stored as adjacency list. The list is ordered by a space filling curve, so nodes that are spatially close are stored on the same disk page. This reduces Input/Output (I/O) costs and therefore increases query speed. The data structure was further improved by Papadias et al. (Papadias et al. 2003) and Demir and Aykanat (Demir and Aykanat 2010) with a grid based tree-like hierarchical structure partitioning the spatial domain to efficiently process range queries (perform queries in circular range around a query point) and successor retrieval operations (get all successors of a network node).

Further techniques to speed up network queries can be found in the more general domain of graph computation (Pienta et al. 2015). A common method is the use of advanced caching/paging strategies to hold often accessed parts of a graph in memory (Kyrola et al. 2012; Han et al. 2013; Roy et al. 2013; Chi et al. 2016; Leskovec and Sosič 2016). Other approaches apply memory mapping of large-scale graphs as edge-list files to handle them on the disk programmatically as if they were in the main memory (Lin et al. 2014). This allows for graph processing with billions of edges on consumer level computers and mobile devices (Lin et al. 2014; Lin et al. 2013; Chen et al. 2015). LLAMA (Macko et al. 2015) further uses compressed row storage to harness sparsity. Recent graph computing frameworks such as FlashGraph (Zheng et al. 2015) further facilitate solid-state disks in combination with minimization of I/O operations to perform out-of-memory graph analysis algorithms (Ai et al. 2017).

Despite their universal applicability, the lack of spatial optimization results in inferior performance for *Aggregation Queries*, i.e. aggregated connectivity from, to or between a set of nodes on spatial networks (see *Section Performance Evaluation*). Some of these implementations are tailored to unweighted, binary graphs (Han et al. 2013; Lin et al. 2014; Chi et al. 2016) that are unsuitable to be generalized to

perform *Aggregation Queries* on weighted, i.e. non-binary, connectivity data.

To our better knowledge, there is currently no tool, which combines those state-of-the art techniques to allow interactive exploration of multimodal, multiresolution neurobiological connectivity on a "big data" level across local-global scales. Thus, from the neuroscientist's perspective, bridging this gap is essential for significant synergies in updating, mining, communicating and sharing brain data.

We meet this demand by proposing a data structure for integration and real time querying of heterogeneous large-scale connectivity matrices at multi-scale voxel and region level by exploiting the hierarchical organization of brain parcellations in combination with spatial indexation.

Region-wise (e.g. resting state functional connectivity) or voxel resolution (structural connectivity, spatial gene expression correlation) connectivity data is aggregated hierarchically, to bridge the gap between different scales and resolutions. The hierarchies are anatomy-driven and can be flexibly generated for different ontologies and their related spatial region annotations. On the lowest level of these hierarchies, high resolution, voxel-wise connectivities with billions of edges (matrices with hundreds of gigabytes) are stored on hard disk in spatially organized indices for high-speed data access. Therefore, aggregated connectivity from, to or between brain areas can be retrieved, from voxel-level to large anatomical brain regions, in an instant.

For direct correlation of different connectivity data at voxel level, we expect the data to reside in the same spatial reference brain space, i.e. registered to the same (multi-resolution) standard brain.[1] However, the dual indexing strategy allows us also to easily integrate and correlate data available only at region level with voxel wise data within the same brain space, but in principle also across brain spaces at region level if the corresponding regions are known. Data from public resources can be easily integrated in our data structure as well as private data generated during experiments in the lab.

We demonstrate the practical significance of this tool by presenting use cases for which we used data provided by large scale brain initiatives. We reproduced recent biological findings by performing data integration and interactive queries on heterogeneous neurobiological data from mice and humans. We created a web-based, interactive local 3D segmentation on visualized data to define volumes of interest (VOI) that can be used to query user-selected connectivity data sets accessible via our data structure. The result is the cumulative voxel-wise connectivity of the selected VOI that is visualized as intensity volume in a 3D rendering. This kind of interaction allows the researcher to relate integrated resources, for example incoming/outgoing connectivity on voxel-level, directly to spatial data like gene expressions.

In general, the proposed data structure allows for handling data of different modalities delivering volumetric and/or connectivity data, which can be used for experimental hypothesis finding. The presented framework is applicable for multilevel functional predictions and extends its relevance across species. Therefore, it is suitable for virtual screening of complex networks, like those linked to psychiatric disorders, and to functionally dissect the corresponding neural correlates in mice.

Materials and Methods

Data

The data relevant for our system can be divided into three types, which in principle can stem from any species or modality:

A Hierarchical Definition of Brain Regions and their Associated Positions on a Reference Brain. This is basically a hierarchical parcellation of a given standard brain and its related ontology. A hierarchy generally starts with the whole brain divided iteratively into sub-regions, where the lowest level contains the highest resolved regions. These regions can have a dense voxel-level representation (Lein et al. 2007) or a set of coordinates representing biopsy sites (coordinates in the brain from where the gene expression data has been sampled (Hawrylycz et al. 2012)). We exemplarily use the Allen Mouse Brain Atlas (AMBA) ontology with 1288 regions on 5 levels (Lein et al. 2007) on a 132x80x114 voxel space, and Allen Human Brain Atlas (AHBA) ontology with 1840 regions also on 5 levels (Hawrylycz et al. 2012) on 3600 MNI152 coordinate space (representing biopsy sites).

Connectivity data is given as weighted adjacency matrices. Rows/columns represent the connectivity strength between brain areas on different scales (voxel or region-wise). The weights can be in any range, positive or negative. In the context of the use-cases described in this paper, we used three different types/sets:

1. Structural connectivity: In Ganglberger et al. (Ganglberger et al. 2017) we compiled a voxel-wise structural connectivity matrix that shows the projections (efferent neurons) of ~15% of the brain from *AMBA* and respectively how voxels are structurally connected in a 132x80x114 mouse brain (100 μm resolution, i.e. the side of a voxel has a length of 100-μm). Further details in Supplementary Note 1. The 67,500 × 450,000 directed connectivity matrix is stored as an uncompressed 91.5 gigabyte *CSV* (comma separated value) file. Weights are normalized to range between 0 and 1.

[1] Such a multi resolution reference brain space is e.g. available from the Allen Institute, providing different kinds of data at 100-μm and 200-μm resolution.

2. Functional connectivity: Functional connectivity, representing correlation of BOLD fMRI signal shows the functional association of brain regions for specific tasks or resting state. We used a resting state connectome for human (Van Essen et al. 2013) and experimental mouse data, which is only available region-wise (~80 regions). Weights are undirected and represent positive correlation coefficients between 0 and 1.
3. Spatial gene expression correlation networks: Correlated gene expression networks quantify tissue-tissue relationships across genes (Lein et al. 2007; Richiardi and Altmann 2015). Details on matrix creation can be found in Supplementary Note 1. The data consists of a 60000×60000 undirected connectivity matrix for mice, that shows the transcriptional similarity for a specific gene set and 3600×3600 for humans (Hawrylycz et al. 2012). The mouse data has a resolution of 200 μm (67x41x58 voxels mouse brain), and is about 12 gigabyte as uncompressed *CSV* file. The data consists of undirected weights, showing positive correlation coefficients between 0 and 1.

A *Volume of Interest (VOI)* is a spatially related set of coordinates in a reference space. These can be arbitrary selected voxels of a user or a brain region. A *VOI* defines an area in the brain, of which the user would like to know the aggregated source or target connectivity of its individual points.

Managing and Aggregating Hierarchical Connectivity Data

The data access structure we propose is tailored to take advantage of sparseness, anatomical or hierarchical parcellations, and spatial organization of the data, which, to our best knowledge, standard graph managing frameworks such as graph databases are not optimized for.

To allow interactive (real time) exploration of the brain connectivity space, the purpose of the data structure is to retrieve the aggregated source or target connectivity of specific *VOI,* such as anatomical regions or arbitrary user defined areas, on a voxel- or region-level in an instant. These *Aggregation Queries* are executed on connectivity matrices, which we define as weighted directed adjacency matrix

$$\mathrm{C} = \left(\mathrm{c_{ij}}\right)_{\mathrm{i}=1..|\mathbf{I}|,\ \mathrm{j}=1..|\mathbf{J}|} \quad , \mathrm{C} \in \mathbb{R}^{|\mathbf{I}| \times |\mathbf{J}|}$$

of a graph, where the rows $\mathbf{I}$ correspond to outgoing-, and the columns $\mathbf{J}$ to incoming edges of spatial regions, defining the spatial and/or anatomical resolution of the respective connectivity data (which can be voxel level) in the discretized standard brain space $\mathbf{B} = \{\mathbf{p}_\mathrm{x}\}_{\mathrm{x}=1..\mathrm{n}}$, $\mathbf{p}_\mathrm{x} \in \mathbb{R}^3$.

Here and in the following, the term *region* refers to a spatial related set of positions in a standard brain space such as certain anatomical brain regions, a group of voxels or a single voxel. Furthermore, we assume the standard brain space to represent the highest occurring resolution of all data to be queried.

Spatial Mapping between Connectivity Matrices and Brain Space

We define the spatial association of the rows, respectively columns of the connectivity matrix $\mathbf{C}$, to be a set of ordered disjunct sub-regions (i.e. from anatomical regions or voxel-level), so

$$\mathbf{R}^{\mathbf{ROW}} = \left\{\mathbf{R}_1^{\mathbf{ROW}}, .., \mathbf{R}_{|\mathbf{I}|}^{\mathbf{ROW}}\right\}$$
$$\mathbf{R}_\mathrm{i}^{\mathbf{ROW}} \subseteq B, \mathbf{R}_\mathrm{i}^{\mathbf{ROW}} \cap \mathbf{R}_\mathrm{k}^{\mathbf{ROW}} = \varnothing, \forall \mathrm{i} \neq \mathrm{k} \wedge \mathrm{i}, \mathrm{k} \in \mathbf{I}$$

and respectively column associations $\mathbf{R}^{\mathbf{COL}}$

$$\mathbf{R}^{\mathbf{COL}} = \left\{\mathbf{R}_1^{\mathbf{COL}}, .., \mathbf{R}_{|\mathbf{J}|}^{\mathbf{COL}}\right\}$$
$$\mathbf{R}_\mathrm{j}^{\mathbf{COL}} \subseteq B, \mathbf{R}_\mathrm{j}^{\mathbf{COL}} \cap \mathbf{R}_\mathrm{l}^{\mathbf{COL}} = \varnothing, \forall \mathrm{j} \neq \mathrm{l} \wedge \ \mathrm{j}, \mathrm{l} \in \mathbf{J}$$

Note that $\mathbf{C}$ represents voxel-wise connectivity if

$$|\mathbf{R}_\mathrm{i}| = 1, \quad \forall \mathbf{R}_\mathrm{i} \in \mathbf{R}^{\mathbf{ROW/COL}}$$

To directly associate spatial positions in brain space $\mathbf{p}_\mathrm{x} \in \mathbf{B}$ with rows and columns of $\mathbf{C}$, i.e. incoming/outgoing connections, we define the following mapping: At first, we map positions in brain space $\mathbf{p}_\mathrm{x}$ to the indices of brain regions contained in $\mathbf{R}^{\mathbf{ROW}}$ and $\mathbf{R}^{\mathbf{COL}}$

$$\psi^{\mathbf{ROW}}(\mathbf{p}_\mathrm{x}) := \begin{cases} i & if\, \mathbf{p}_\mathrm{x} \in \mathbf{R}_\mathrm{i}^{\mathbf{ROW}} \\ \varnothing & if\, \mathbf{p}_\mathrm{x} \in B \setminus \mathbf{R}^{\mathbf{ROW}} \end{cases}, \quad \varnothing if\, \mathbf{p}_\mathrm{x} \in B \setminus \mathbf{R}^{\mathbf{COL}}$$

$$\Phi\psi^{\mathbf{COL}}(\mathbf{p}_\mathrm{x}) := \begin{cases} j & if\, \mathbf{p}_\mathrm{x} \in \mathbf{R}_\mathrm{j}^{\mathbf{COL}} \\ \varnothing & if\, \mathbf{p}_\mathrm{x} \in B \setminus \mathbf{R}^{\mathbf{COL}} \end{cases}, \quad \forall \mathbf{p}_\mathrm{x} \in \mathbf{B}$$

This creates non-unique mappings of arbitrary *VOI* in the brain reference space $\mathbf{V} \subseteq \mathbf{B}$ to rows/columns (i.e. a set of positions in the brain space can point to multiple rows or column indices)

$$\psi^{\mathbf{ROW}}(\mathbf{V}) := \left\{\mathrm{i} \mid \psi^{\mathbf{ROW}}(\mathbf{p}_\mathrm{x}) = i, \quad \forall \mathbf{p}_\mathrm{x} \in \mathbf{V}\right\}$$
$$\psi^{\mathbf{COL}}(\mathbf{V}) := \left\{\mathrm{j} \mid \psi^{\mathbf{COL}}(\mathbf{p}_\mathrm{x}) = j, \quad \forall \mathbf{p}_\mathrm{x} \in \mathbf{V}\right\}$$

Please note that specific indices in the resulting set might be represented more than once, i.e. if there are m voxels in $\mathbf{V}$ laying in region $\mathbf{R}_\mathrm{i}$, then i is m times present. This allows the application of this mapping for aggregation of connectivity.

Vice versa, we map a set of indices to the union of their corresponding regions. As the voxel wise representation of regions in the standard brain space is known, this generates a representation of connectivity at highest voxel resolution independent from the resolution or underlying parcellation of the original connectivity data.

$$\psi^{\mathbf{ROW}-1}(\mathbf{U}) := \bigcup_{i \in \mathbf{U}} \mathbf{R}_i^{\mathbf{ROW}}, \forall \mathbf{U} \subseteq \mathbf{I}$$

$$\psi^{\mathbf{COL}-1}(\mathbf{W}) := \bigcup_{j \in \mathbf{W}} \mathbf{R}_j^{\mathbf{COL}}, \forall \mathbf{W} \subseteq \mathbf{J}$$

Therefore, a connection c_{ij} might represents equal connections of several points in brain. This has several advantages (see Fig. 1):

1. Compare connectivity data defined on different resolutions of the standard brain: $\psi^{\mathbf{ROW}-1}$ and $\psi^{\mathbf{COL}-1}$ define the relation of rows respectively columns to voxel at a certain resolution. Since the overlap of regions with standard brain space is known, this enables a comparison of connectivity matrices in respect to different brain parcellations and/or different resolutions. If the resolution is smaller than the reference space, this mapping would represent up-sampling (see Fig. 1a).
2. Map region wise connectivity to voxel level: Nodes of a connectivity matrix can also represent (anatomical) brain regions to store region-wise connectivity data. Using $\psi^{\mathbf{ROW}-1}$ and $\psi^{\mathbf{COL}-1}$ allow a retrieval of the data in voxel-wise brain space and therefore also allow the comparison of connectivity with respect to different brain parcellations (see Fig. 1b).
3. Build caches: This technique can also be used to store precomputed data, such as connectivity of brain regions (from voxel level data) or pyramids representations with lower resolution (like an image pyramid). Although this increases the required storage, it improves scalability (see Fig. 1c).

A Dual Data Stucture Strategy for Aggregation Queries

Aggregation Queries are defined as follows. Let $\mathbf{V} \subseteq \mathbf{B}$ be a *VOI*. The result of a target aggregation query is the cumulated outgoing connectivity for every position in space $\mathbf{B}$

$$\boldsymbol{\tau}(\mathbf{V}) = \left(\sum_{i \in \psi^{\mathbf{ROW}}(\mathbf{V})} \mathbf{c}_{i,\psi^{\mathbf{COL}}(\mathbf{p}_x)} \right)_{\mathbf{p}_x \in \mathbf{B}}$$

and the result of a source aggregation query the cumulated incoming connectivity for every row

$$\boldsymbol{\varsigma}(\mathbf{V}) = \left(\sum_{j \in \psi^{\mathbf{COL}}(\mathbf{V})} \mathbf{c}_{\psi^{\mathbf{ROW}}(\mathbf{p}_x),j} \right)_{\mathbf{p}_x \in \mathbf{B}}$$

We are proposing a dual strategy unifying two complementary data structures to efficiently realize *Aggregation Queries*. The *Connectivity Storage* handles the data access for the *Aggregation Queries*, and the *Region-Wise Connectivity* in a Graph-Database manages queries on (anatomical brain-)region level. Figure 2 gives an overview of the overall system. Incorporating a connectivity matrix into our data structure begins with a preprocessing, that harnesses spatial-organization of the data (Fig. 2 *(1)*) and uses row-compression to minimize disk-space (and therefore reading-time for queries) (Fig. 2 *(2)*) to create a *Connectivity Storage File*. Region-wise connectivity of a hierarchical anatomical brain-region parcellation is precomputed and stored in a graph-

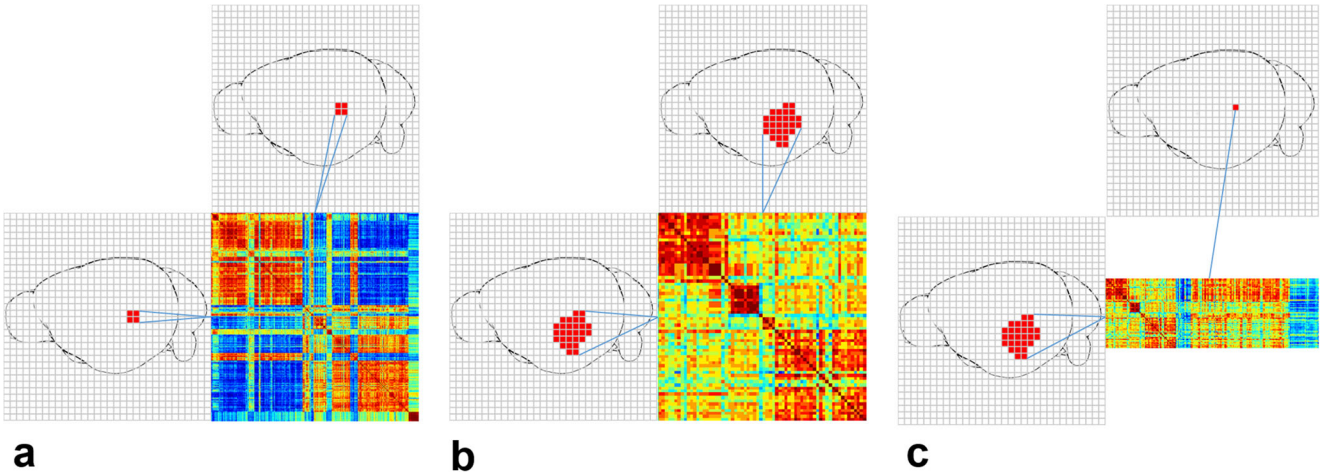

Fig. 1 **a** Connectivity matrix in a 4 times lower resolution than the reference brain space. Therefore every row/column is associated with 4 voxels. **b** Region-wise connectivity matrix. Every row/column is associated with voxels that form brain regions. **c** Region cache. Preprocessed aggregated outgoing connectivity for brain regions on voxel level

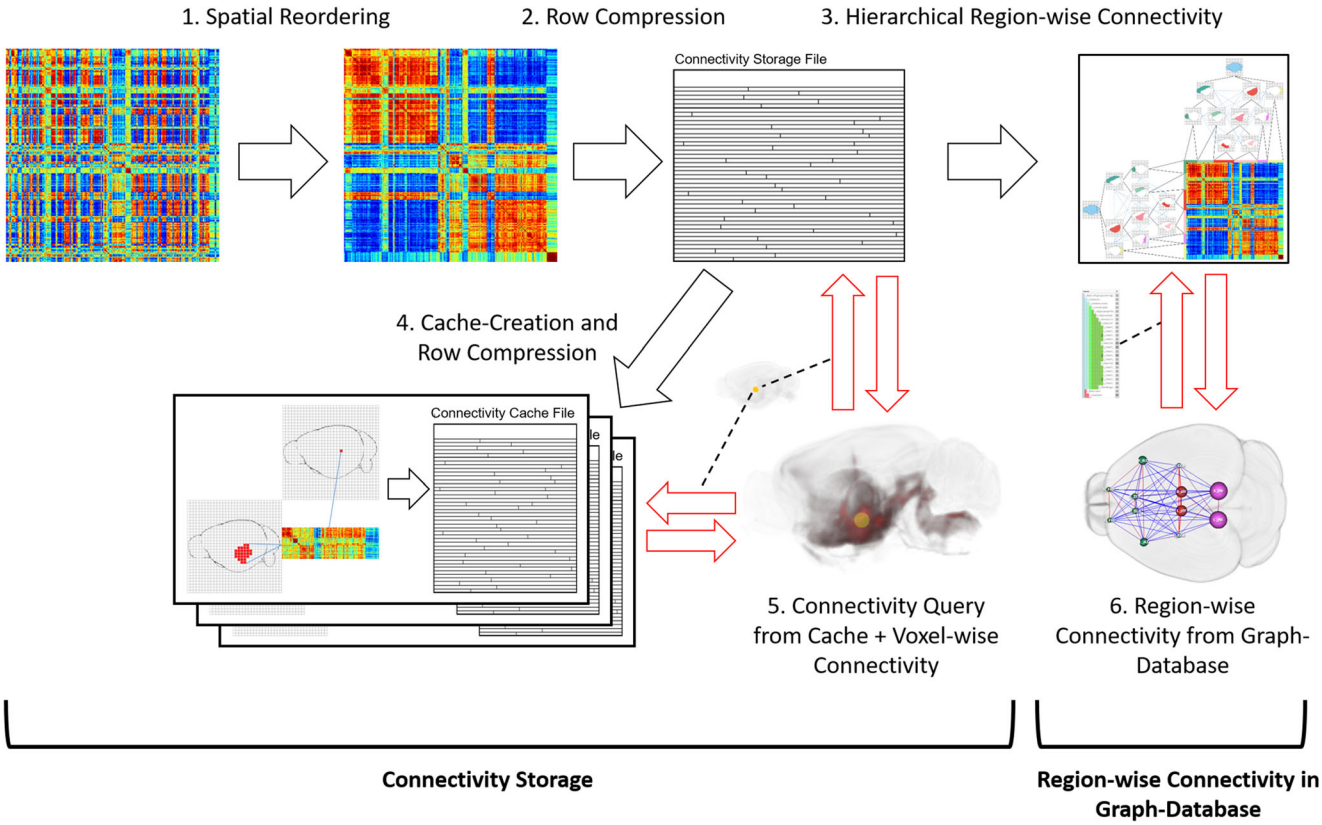

Fig. 2 Overview of Connectivity Storage and the Region-wise Connectivity in the Graph-Database. Black arrows: Preprocessing of the data. **1.** Spatial Reordering of a (voxel-wise) connectivity matrix with a space filling curve. **2.** Row-wise compression of spatially-ordered connectivity matrix. 3. Generation of hierarchical region-wise connectivity and storage in graph-database. **4.** Cache creation (preprocessed voxel-wise connectivity for predefined regions), and storage with row compression. Red arrows: **5.** Querying a *VOI* (yellow circle) on *Connectivity Cache Files*, then on *Connectivity Storage File*, resulting in aggregated connectivity (red). **6.** Querying connectivity between preselected brain regions (from a hierarchical parcellation), resulting in a region-wise connectivity graph

database Fig. 2 *(3)*). To further improve query performance, *Connectivity Cache Files* are created, that store pre-computed connectivity for faster data access (Fig. 2 *(4)*). Voxel-wise connectivity can then be queried from cache files and *Connectivity Storage Files* (Fig. 2 *(5)*), region-wise connectivity form the graph-database (Fig. 2 *(6)*). Preprocessing (Fig. 2 *(1,2,3)*) is further described in the following subsections (*Connectivity Storage, Region-wise Connectivity Database)*, and cache-creation as well as querying (*Fig. 2 (4,5,6)*) in subsection *Implementation.*

Connectivity Storage Since *Aggregation Queries* involve the reading and aggregation of whole rows or columns of connectivity matrices, we use a row-wise storage scheme. Allthough edge lists are popular for many graph management tools (Lin et al. 2014), which store connections in a < source node, target node, value> combination, they create a significant storage overhead for dense connectivity matrices.

Reducing data size allows higher query speed, since fewer data needs to be read. Therefore we apply a row-wise compression, that exploits potential sparseness of the data. First, the rows and columns of **C** are ordered by a space filling curve (Hilbert 1891) to preserve locality. The reordering causes sparse/dense areas to cluster within each row/column, since the connectivity of a region/voxel is not randomly distributed over the brain, but spatialy related. Then, a compressed row starts with the column index of the first non-zero value (*NZV*), the amount of *NZV* to follow, and the following *NZV*s. This is repeated similarly with the column index of the next *NZV* until the end of the row is reached. To identify each row in the file, an additional mapping

$$\Omega(i) := f, \quad \forall f \in \mathbf{F}, i \in \mathbf{I}$$

needs to be created, depicting the beginning of each row to their position f in the file **F**. A connection c_{ij} can be identified by going to the corresponding position of the i-th row f, reading the j-th value from the row-wise compression. Figure 3 illustrates this process.

Other compression methods would also reduce the data size, but would not allow to directly access single rows without decompression of the whole file or significant parts of the file (Barrett et al. 1994).

For every connectivity matrix, we create a separate *Connectivity Storage File,* consisting three indices as header

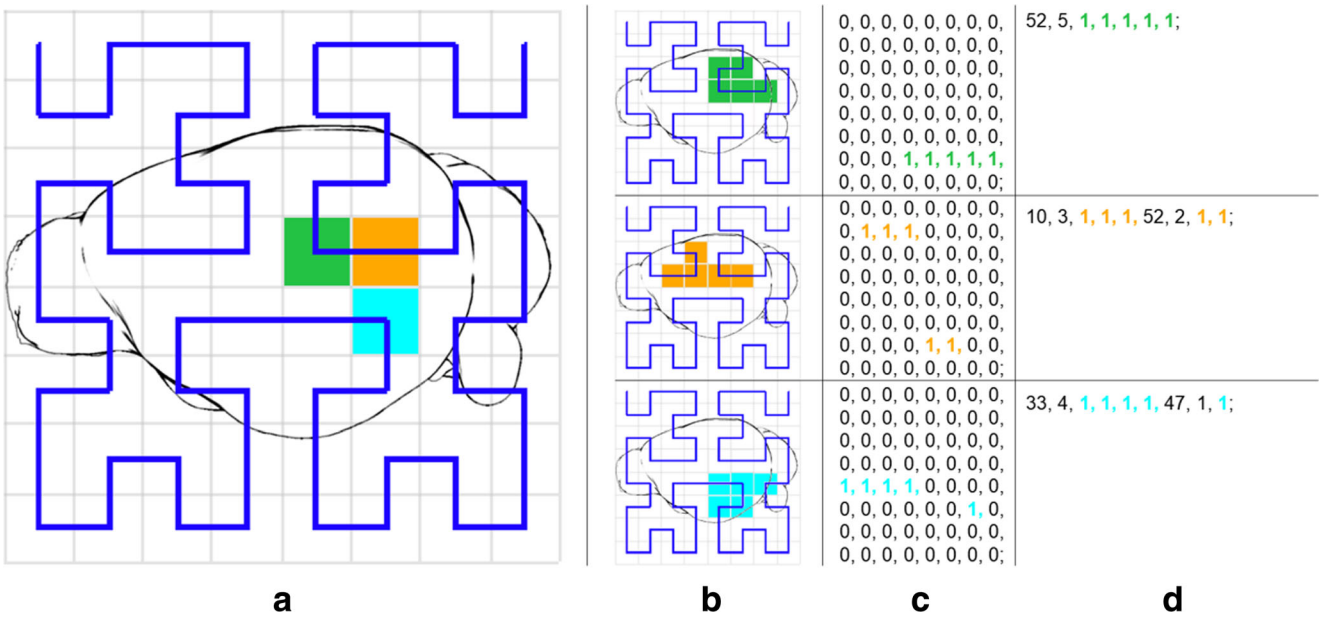

Fig. 3 **a** A mouse brain with overlayed Hilbert curve (blue), mapping the space to a one-dimensional space **b** The outgoing connectivity of the voxels corresponding to the colored voxels in *A*. These would represent 3 rows in a connectivity matrix. **c** The connectivity of B along the Hilbert curve (for simplicity in this example, connectivity is either 0 or 1). **d** Row-wise compression of *C*. The compression can be read this way: On the 52nd position, 5 *NZVs* are following (green). On the 10th position, 3, and on the 52th position 2 *NZVs* are following (orange). On the 33rd position, 4 *NZVs* and on the 47th position, there is 1 *NZV* following (cyan)

(*FILE*: $\mathbf{\Omega(i)}$, *ROW*: $\psi^{\mathbf{ROW}}(\mathbf{V})$, *COLUMN*: $\psi^{\mathbf{COL}}(\mathbf{V})$ followed by compressed rows (Fig. 4b). Even after compression, the compressed file does not necessarily fit into memory, especially when one wants to query on multiple connectivity matrices. We use memory mapping (*MMap*) to map the file into virtual address space. This allows to programmatically access rows if they were in the main memory, without overhead of system calls. Furthermore, the OS employs paging strategies, such as read-ahead paging. When performing a query for outgoing connections, the rows can be read in the order of their position in the file, and directly benefit from read-ahead paging of the operating system to reach near-sequential reading speed. This additionally

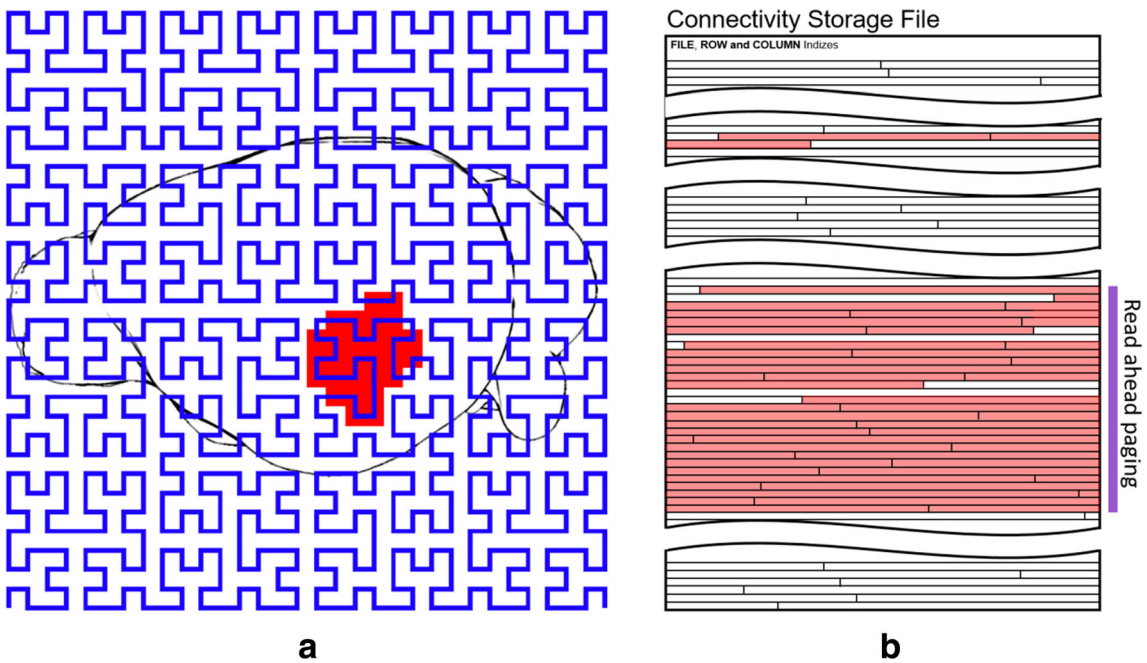

Fig. 4 **a** Brain Space overlayed with Hilbert curve (blue) and a *VOI* (red) that is queried for outgoing connectivity, **b** Connectivity matrix file (rows ordered by Hilbert curve). Red-blocks represent rows that are read in order to get outgoing connectivity of *VOI* shown in *A*. Blocks can be read sequentially. Purple rows benefit by read ahead paging

exploits the spatial organization of the data, that has been created with the ordering by space filling curve (see Fig. 4a). Multiple connectivity matrices can then be queried sequentially without loading the whole matrices into memory. Note that a connectivity matrix of a directed graph needs an additional transposed *Connectivity Storage File* to query incoming connectivities (for undirected graphs, outgoing and incoming connections are equal due to symmetry).

Region-Wise Connectivity Database On higher levels, the (anatomical brain-)region level, the aggregated connectivity of a region consists of the connectivity of its subregions (and on the lowest level voxel-wise connectivity). When looking at brain wide region-wise graphs, it is not feasible to read the entire *Connectivity Storage* and compute the connectivity hierarchically at runtime. This would be too resource consuming for real-time computation. Instead, we compute it once when the *Connectivity Storage* is created. The resulting region-wise hierarchical connectivity is stored in a graph-database. The region-wise connectivity is computed recursively bottom up: First, the lowest level regions are aggregated from the *Connectivity Storage*, then the regions above are aggregated by their levels below until the top of the hierarchy. We further compute the connectivity between the levels in a similar way. Therefore, it is not necessary to compute any region-level connectivity at runtime (Fig. 5).

Implementation As central access point for the data, we created a REST API in GO (golang). It provides calls for importing data, creating caches as wells as *Aggregation Queries*. These are executed on the *Connectivity Storage*, which was implemented in C++ for memory and performance optimization. Connections are stored in a 4-byte floating point format, which supports a range of values $\pm 1.18 \times 10^{-38}$ to $\pm 3.4 \times 10^{38}$, with single precision (about 7 decimal digits). We choose this as trade-off to storage space, since higher precision would also cause higher reading times. *ROW* and *COL* indices have a 4-byte unsigned integer format. Therefore the maximum amount of edges is limited by $4{,}294{,}967{,}295 \times 4{,}294{,}967{,}295$ ($= 1.84467 \times 10^{19}$). The *FILE* index associates rows the with 8-byte unsigned integers to file positions, limiting the file size similarly to 1.84467×10^{19} connections or 64 petabyte.

We implemented two types of *Connectivity Caches* to increase performance: A region-cache, that stores the

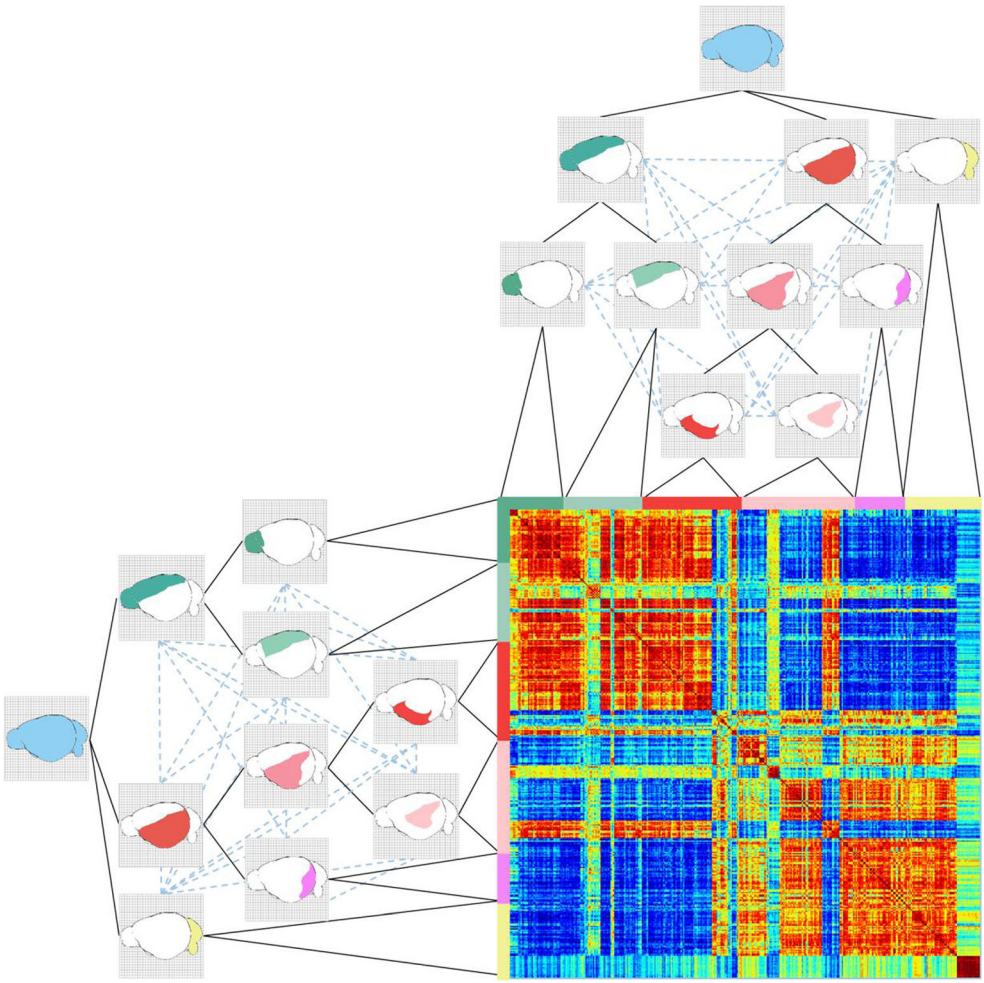

Fig. 5 Scheme of the data structure: The *Connectivity Storage* stores the connectivity on the lowest level (voxel-wise connectivity). Region-wise connectivity (dotted blue lines) is aggregated from the *Connectivity Storage* hierarchically

aggregated voxel-level connectivity of lowest-level of the hierarchical brain-region parcellation, and factor h low-resolution versions ($h \in \mathbb{N}$, $h \leq |\mathbf{I}|$) of the *Connectivity Storage*, which cumulates the connectivity of h voxels along the Hilbert curve (basically every h rows of the *Connectivity Storage* being aggregated).

When executing an *Aggregation Query* for a *VOI,* $\mathbf{V} \subseteq \mathbf{B}$, the *Connectivity Cache Files* will be accessed first to check if the *VOI* contains cached regions $\mathbf{R}^{\mathbf{ROW\ (CACHE)}}$ defined in the *ROW* index of the cache. The connectivity of a region $\mathbf{R}_c \in \mathbf{R}^{\mathbf{ROW\ (CACHE)}}$ will be added to the results from the cache, if $\mathbf{R}_c \subseteq \mathbf{V}$, i.e. all spatial positions of a region are contained within the *VOI*. Before the *Connectivity Storage* will be accessed, all *Connectivity Cache Files* will be queried until no further region in the cache can be found. Only after this, the remaining brain space positions of the VOI will be queried from the *Connectivity Storage*, hence, the total number of row-reads is minimized.

The anatomical hierarchy is represented in *OrientDB* (Garulli 2010), a graph database that can be used to store further region information, such as masks, 3D models or links to online repositories. Region-wise connectivities within those hierarchies consist of 1000–2000 regions with a maximum of 4 million edges. When querying such comparatively small graphs, the performance differences of standard graph databases to the *Connectivity Storage* is neglectable. Therefore, we store them in *OrientDB*, where it is directly linked to the brain regions.

To access the API, we created a web-component that allows visual queries that are based on selections of *VOI* directly in 2D slice views, visualized simultaneously in a 3D volume rendering. Via a spherical brush tool, a user-defined area can be marked. Figure 6a shows for example a gene-expression volume, where the spherical area is drawn on voxel with high gene-expression. After selection, *Aggregation Queries* can be used to link connectivity data with volume data. The selected area (Fig. 6b), is used as input for an *Aggregation Query* on the API. The API retrieves the connectivity from the *Connectivity Storage* to all voxels that are either *targets* or *sources* of the selection, and the web component will instantly render the connectivity as volume. This represents the cumulative connectivity to (target) or from (source) the selected area (Fig. 6c). Furthermore, the connectivity can be quantified in *Connectivity Profiles,* which shows the cumulated connectivity of the *VOI* to preselected (brain) regions (Fig. 6d).

Results

To assess the efficiency and effectiveness of the data structure in context of its practical application, we performed a quantitative and qualitative evaluation on real world data that was introduced above in *Section Data*. We quantified the effect of the data structure's parameters (row-compression, spatial ordering, caches) on query performance and compared these results with two state-of-the-art graph engines (*Section Performance Evaluation)*. We further performed two Case-Studies that we designed with domain experts in order to demonstrate the relevance of the data structure for neuroscientific research *(Section Case Study 1 and 2*).

Performance Evaluation

To verify the data structure's applicability for real-time *Aggregation Queries*, we created test queries on three voxel-level connectivities which were introduced in *Section Data*. We used one directed structural connectivity matrix *SC,* resulting in two *Connectivity Storage Files* for targets and source queries, and two undirected spatial gene expression correlation networks *CS1* and *CS2* (which are further used in Case Study 1 and 2), creating one *Connectivity Storage File* each (because they are undirected).

Creating the two *Connectivity Storage Files* for SC (91GB CSV file) took 32 min in total (19 for the first, 13 min for the transposed) while CS1 (12 GB CSV file) and CS2 (13 GB CSV file) took about 3 min each on an SSD with our REST API. Therefore, the file creation takes approximately 21 s/GB for directed, and 13 s/GB for undirected matrices. Generating the *Region-Wise Connectivity Database* lasted less than 10 min for each *Connectivity Storage* which depends on the I/O performance of the *OrientDB.*

In cooperation with domain experts, we defined 10 queries in our web-component (user-queries), which are shown in Supplementary Note 2. The *VOI* of these queries range from 0.2% to 10% of the mouse brain space. In addition, we selected 10 distinct anatomical brain regions to act as *VOI* (region-queries) with sizes ranging from 0.2% to 4% (see Supplementary Note 3). To evaluate queries on a bigger scale, we further created 100 random queries by using randomly placed spheres with random radii as *VOI*s. The sizes of these range from 0.2% to 5%, because it was not possible to place larger spheres within the mouse brain space.

We used these queries to assess the effects of individual components of the *Connectivity Storage*, such as row-compression, the spatial-ordering of rows/columns and *Connectivity Caches.* To demonstrate the data structure's relevance for performing *Aggregation* Queries, we compared the results to the state-of-the art tools FlashGraph (Zheng et al. 2015) and GraphChi (Kyrola et al. 2012). We did not evaluate the performance of the *Region-wise Connectivity Database* in the *OrientDB* specifically, since retrieving a connection between two regions only involves accessing a single database

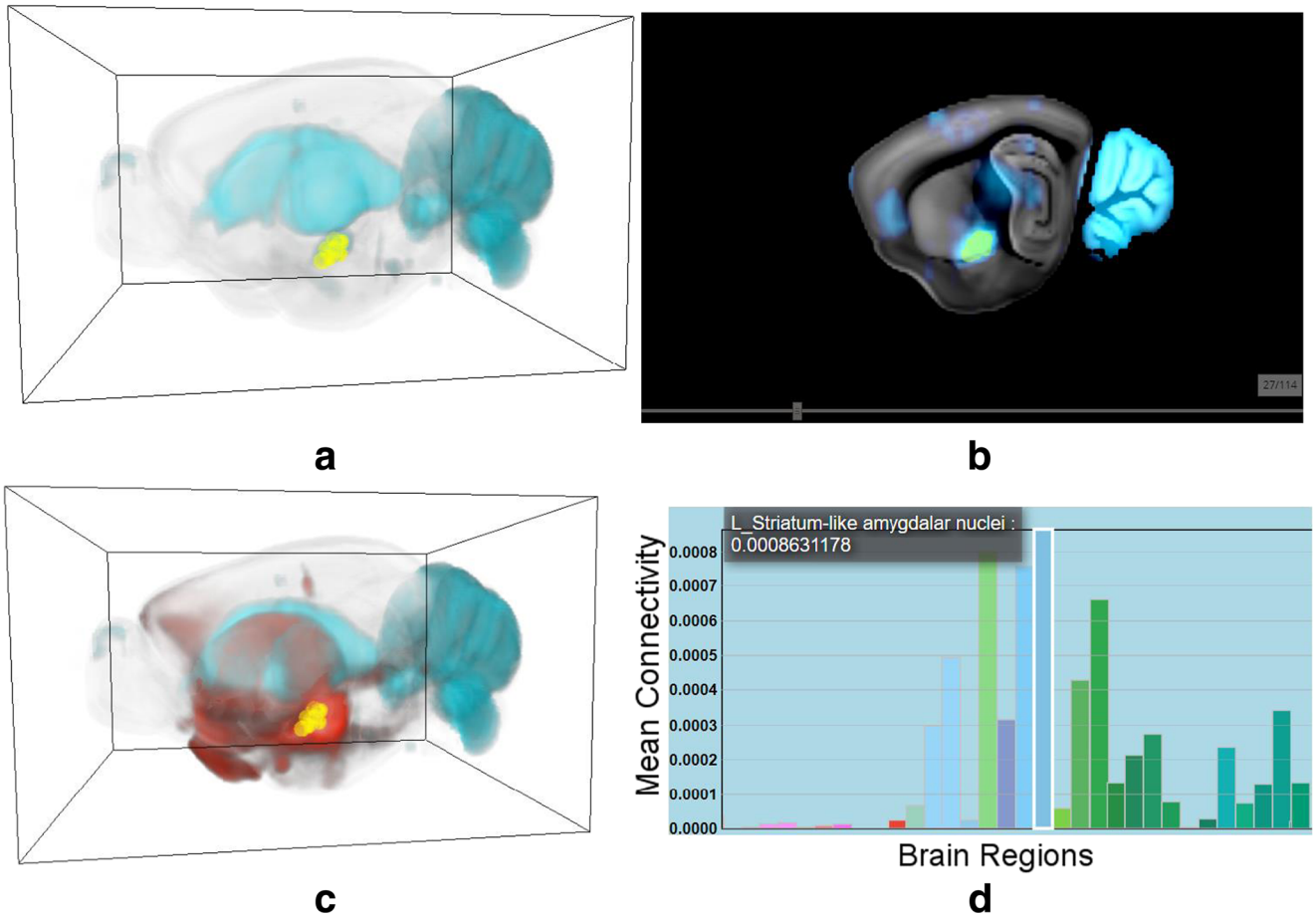

Fig. 6 **a** gene expression (cyan) with brush selection (yellow) of *VOI* in 3D **b** Selection has been performed on 2D slice views, **c** Accumulated target connectivity (red) of *VOI* in 3D **d** *Connectivity Profile* of the target query, showing the mean connectivity to each brain region

entry (<10 ms), in comparison to aggregating mega- to gigabytes of data from the *Connectivity Storage*.

Performance has been evaluated on an Ubuntu 16.10 64-bit machine with Intel Core i7–4470 CPU, 32 GB RAM and a 1 Terabyte SSD with a sequential read-speed of 520 MB/s. Test result on an HDD with 120 MB/s sequential read-speed can be found in Supplementary Note 4.

Effect of Compressed Row-Storage on Data Size For 3 connectivity matrices (*SC*, *CS1* and *CS2*), we created 4 *Connectivity Storage Files* (2 for *SC* and 1 for each *CS1 and CS2*). Figure 7a shows that *Connectivity Storage Files* with compression reduces the initial file size of *SC* by half, even if one is using random ordering of rows/columns. Spatial ordering by a Hilbert-curve further improves file size by reducing it by half. The effect is smaller for *CS1* and *CS2*, since they are not as sparse as *SC* (i.e. they contain not as many zeroes).

Effect of Spatial-Ordering on Query Speed We executed the user-, random-, and region-queries on *SC*, *CS1* and *CS2* for their sources and target connectivity. Figure 7C shows the mean query time and their standard error bars on the connectivity matrices for different query types. Not, that the spatial ordering along a Hilbert curve greatly reduces query-time compared to random-ordering, especially for the bigger *SC* matrix (from up to 20 s to <2 s). This is due to read-ahead-paging, which benefits from sequential reading. Note that the mean query time for different query types depends on the size of their *VOI*. Hence, region-queries are faster than user- or random-queries simply because they involve reading fewer data (detailed query sizes see Fig. 7b).

Effect of *Connectivity Caches* on Query Speed As described in *Section Implementation*, we created a region *Connectivity Cache* of the lowest level of the hierarchical brain-region parcellation, and factor h low-resolution *Connectivity Caches,* where every h rows of the *Connectivity Storage* are aggregated, for h =10 and h =100. Figure 8 shows the mean query time and its standard error for different cache combinations. One can see that for high resolution *Connectivity Matrices* such as *SC*, h-factor caches can save up to half of the query time, while region queries especially benefit from the region-caches. For lower resolutions (*CS1* and *CS2*), this effect of h-factor caches is not as strong. The

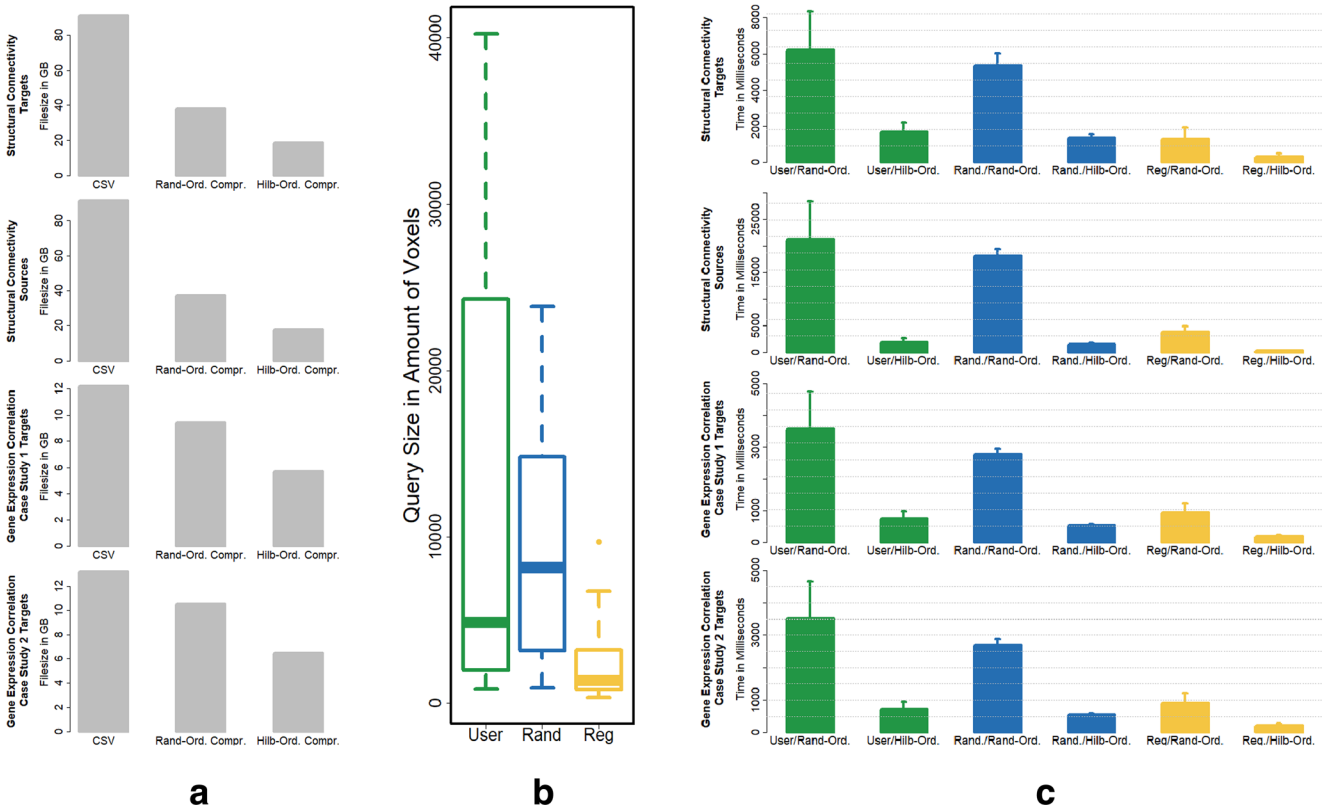

Fig. 7 **a** Effect of compressed row storage on the data size of different connectivity matrices. Bars indicate the size of the original CSV, the *Connectivity Storage* file without compression, random-ordering with compression and Hilbert-ordering with compression. **b** Boxplot of the *VOI* size as amount of voxels (i.e. the query size of 10 user-defined *VOI* queries (green), 100 random *VOI* queries (blue) and 10 region *VOI* queries (yellow) **c** Effect of spatial-ordering on query-speed on different connectivity matrices. Bars show the mean query-time with standard error of 10 user-defined *VOI* queries (green), 100 random *VOI* queries (blue) and 10 region *VOI* queries (yellow), for Hilbert-Ordering and Random-Ordering

reason is that connectivity retrieved from *Connectivity Caches* leaves "holes" in the *VOI* of the query, hence, the remaining rows that need to be read from the *Connectivity Storage File* are fragmented. This reduces the overall read speed, for it relies on read-ahead paging (the effect of sequentially reading spatially close rows has been shown in Fig. 7c). Figure 9 depicts this in further detail: In the left column, one can observe that the query time depends on the query size, and that query time benefits increasingly from *Connectivity Caches* for larger query sizes (i.e. more data to read leads to higher chances of read-ahead paging). The right column shows read-speed on *Connectivity Storage Files* vs query size and therefore the effect of reads from the *Connectivity Caches* and the resulting fragmentation. The lower read speed after cache reads is a direct cause of the higher fragmentation rate (i.e. many reads from 10-factor caches lead to more "holes" in the *VOI* than a few reads from 100-factor caches).

Comparison to State-of-the-Art Tools We compared our method to the state-of-the-art graph engines FlashGraph (Zheng et al. 2015) and GraphChi (Kyrola et al. 2012). Both tools are capable of computing graph algorithms (page-rank, breath-first-search etc.) on graphs with billions of edges on consumer level machines (i.e. without hundreds of gigabytes RAM). They achieve this by utilizing data access mechanisms that are able to load data from hard-drive on demand, instead of holding the whole graph in memory. GraphChi's approach is splitting the data into small parts (so called shards), and loading them on demand, while FlashGraph uses optimized I/O requests for SSDs. Therefore, these methods benefit from graph queries that do not involve whole graphs respectively, do not need to load entire connectivity matrices, such as *Aggregation Queries*. To compare their performance to the *Connectivity Storage*, we have implemented *Aggregation Queries* for both (see Supplementary Note 5 for details) and created edge-lists (in their common input data format <source node, target node, value>) of our connectivity matrices. Further, we have ordered the node indices spatially (according to a Hilbert curve) to test them under equal conditions. Figure 10 shows that even with Hilbert ordering, FlashGraph and GraphChi do not perform as fast as our method. While on smaller graphs (CS1 and CS2), the *Connectivity Storage* is still faster

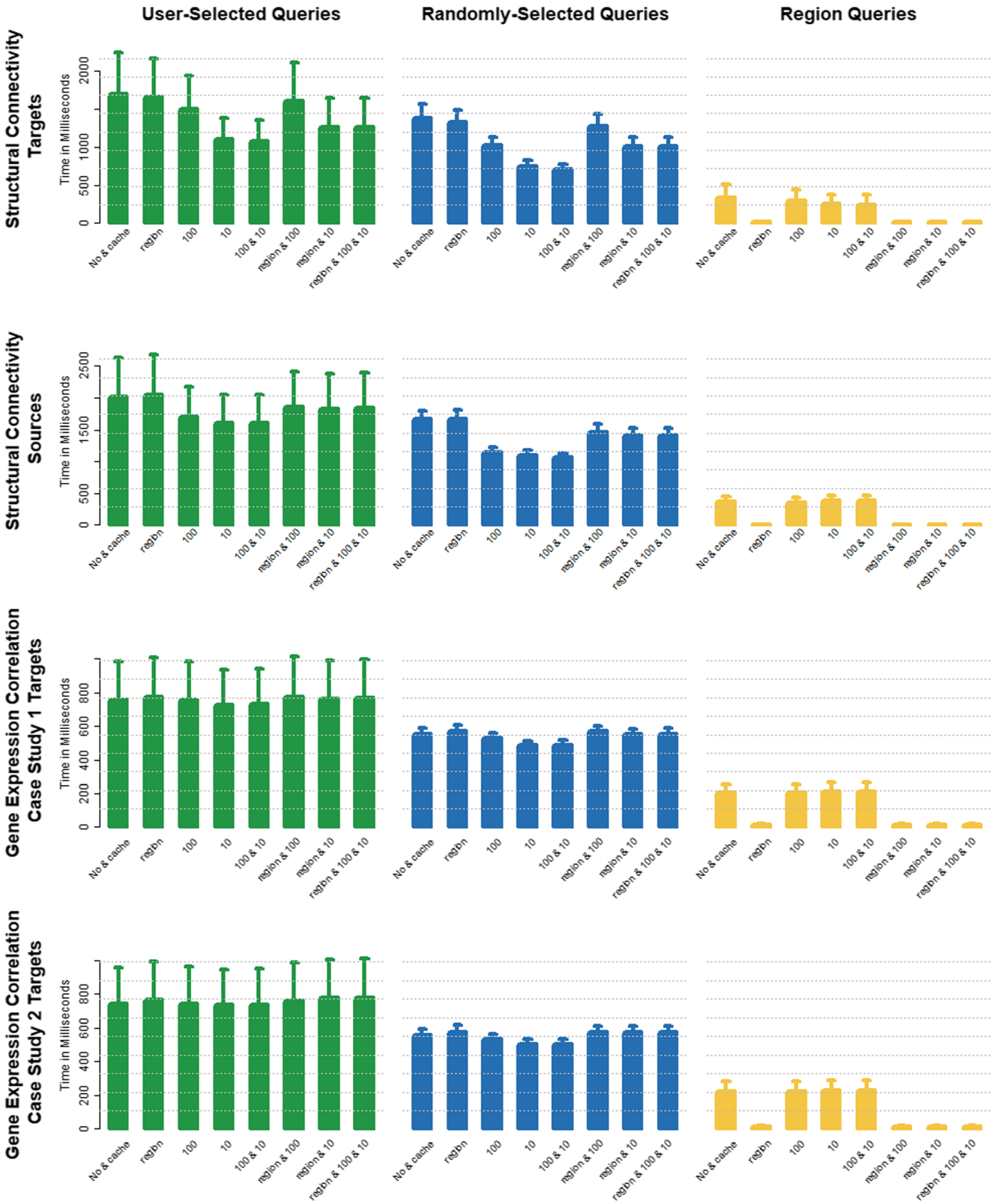

Fig. 8 Effect of *Connectivity Cache* on query-speed on different connectivity matrices. Bars show the mean query-time with standard error of 10 user-defined VOI queries (green), 100 random *VOI* queries (blue) and 10 region *VOI* queries (yellow), for different types of caches and their combination

than FlashGraph by a factor of 2–3, this effect is even stronger for larger matrices (SC1) with a factor of 6. Overall, our method performs more than 5 times faster than FlashGraph, and 160 times faster than GraphChi.

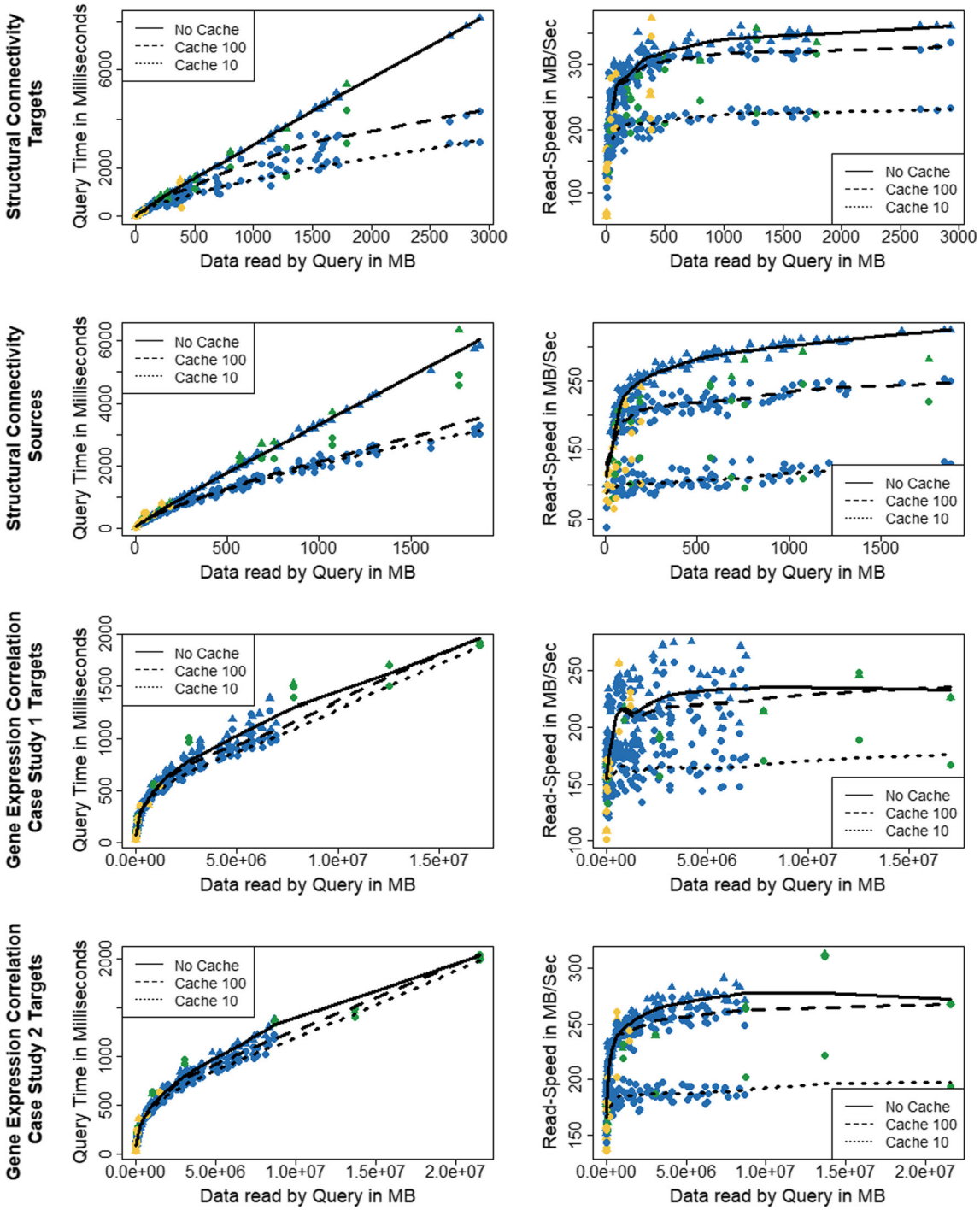

Fig. 9 Relation of query-time and read-speed on query size for different *Connectivity Matrices, Connectivity Caches* and query types. The left column depicts the query time vs query size for queries executed with cache (●) and without (▲), while the color depicts the query type (green = user query, blue = random query and yellow = region query). The right column depicts read speed on the *Connectivity Storage Files* vs query size, similarly encoded. LOWESS regression lines are added to see the overall trend for different cache sizes

One has to note, that these tools were developed for performing various graph analysis methods, thus, they are probably not optimized for *Aggregation Queries*. Especially GraphChi is more suited for analyzing whole graphs, while *Aggregation Queries* only require loading of subgraphs.

Example Video for Real-Time Performance For further demonstration, Supplementary Video 1shows a target query

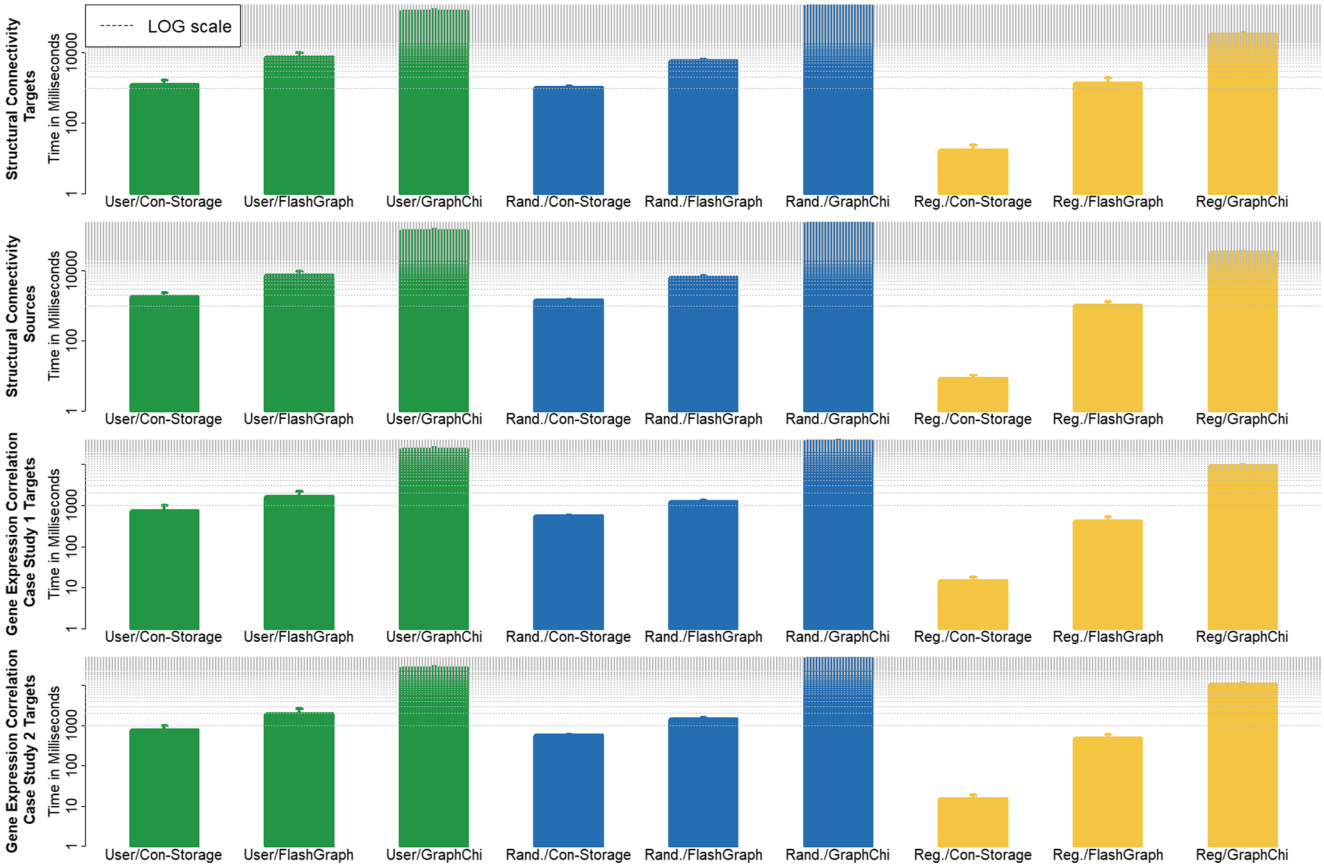

Fig. 10 Comparison of query speed with state-of-the-art tools. Bars show the mean query-time with standard error of 10 user-defined VOI queries (green), 100 random *VOI* queries (blue) and 10 region *VOI* queries (yellow), for the Connectivity Storage, FlashGraph and GraphChi. The bars are log scaled, indicated by equidistant grey dotted lines (distance between two lines represent 1 s)

on the structural connectivity matrix (similar to Fig. 6*)* performed in real-time.

Case Study 1: Exploring Different Types of Connectivity Emerging from a Brain Area of Interest

This case study has been chosen for its particular application in circuit dissection. Recent advances in circuit neuroscience (e.g. neuro- and behavioral genetics, optogenetics, imaging) identified gene sets underlying specific behavioral function. Hence, we mapped such function-related network context on a genetically well dissected microcircuitry (Radke 2009). To illustrate this case, we focused on the central amygdala (CEA), an amygdala subnucleus and hotspot expressing several functionally related genes, whose role in fear behavior is a heavily researched topic in the neuroscience community.

The connectivity data used for this case study consists of directed structural connectivity (*Data Set 1)* and undirected spatial gene expression correlation (*Data Set 3*). Hence, this case demonstrates the exploration of connectivities of different type and different resolution.

The entry point for our experts is a subset of these genes consisting of Prkcd (EntrezID: 18753), Sst (20604), Crh (12918), Dyn (18610) and Penk (18619) that have been known to regulate fear responses (Haubensak et al. 2010). We examined the gene expression density of these genes in 3D and 2D slice views for areas of high co-expression (where multiple genes are expressed). An image overlap of Prkdc, Crh and Dyn revealed an enclosed area (Fig. 11a*, red arrow)* that is selected by using a brushing tool allowing the user to interactively mark *VOI*s on 2D slice views of the brain space. We further overlaid the outlines of CEA so the selected area is this brain region indeed (Fig. 11b*, red arrow).*

After a target query on the structural connectivity (*Data Set 1)* matrix (Fig. 11c*)*, which is performed in less than a second, particularly strong connected areas are visualized and identified by *Connectivity Profiles*. It highlights, that among other regions, the bed nucleus of the stria terminalis (BNST) has a strong connection to the central amygdala (CEA) (Fig. 11, red arrow). It is important to note, that this confirms known structural anatomy from literature (Radke 2009). Interestingly, the BNST is functionally related to CEA. While CEA causes brief

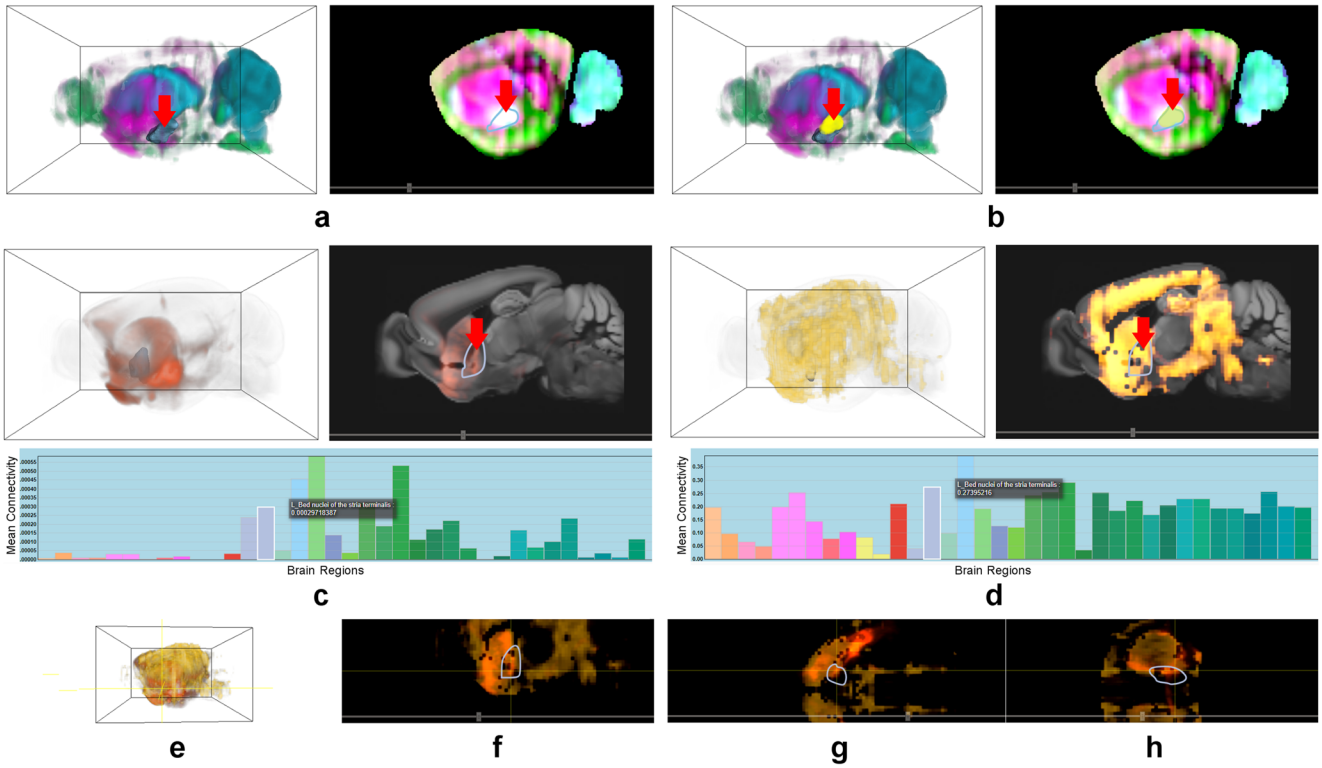

Fig. 11 ***A:*** Overlap of Prkcd (cyan), Crh (green) and Dyn (purple) by aggregating the image intensity (i.e. strong overlap is white in the 2D slice view). Outlines of CEA in blue (red arrow). ***B:*** Selecting a VOI on the image overlap (yellow), ***C:*** Structural connectivity of the VOI. Outlines of BNST in dark blue (red arrow) and its connectivity profile (bars reflect mean connectivity to (Allen Brain Atlas) brain regions, with corresponding colors). ***D***: Gene-coexpression correlation of the VOI, analogue to *C*. ***E***: Overlap of *C* and *D* in 3D. ***F***: Overlap of *C* and *D* in a 2D slice (XY). ***G***: Overlap of *C* and *D* in a 2D slice (XZ). ***H***: Overlap of *C* and *D* in a 2D slice (YZ)

phasic fear responses, BNST shows more long-lasting tonic anxiety-like states. Thus, this approach recaptures a functional CEA-BNST circuit module for fear.

To further verify the query's result quantitatively, we compared the outgoing connectivity to known region-level structural connectivity of CEA. Therefore we used the normalized projection strength of 469 sites (positions in the brain) to 590 brain regions provided by Oh et al. (Oh et al. 2014) (Fig. 3/ Supplementary Table 2), i.e. the cumulated outgoing strength of projection neurons. Out of these 469 sites, we choose the five that lie within CEA since there is a high overlap (Fig. 11b, *red arrow*) between the query's VOI and CEA. Figure 12 depicts the rank correlation of the query result to the five sites chosen by us, as well as the mean connectivity thereof. A correlation of 0.817 demonstrates the validity of the query. When using a VOI congruent to CEA, the correlation increases to 0.92.

This is repeated with the spatial gene expression correlation network (*Data Set 3)* of Prkcd, Sst, Crh, Dyn and Penk, a connectivity matrix representing the voxel-wise correlation of the gene set used for this case study. BNST has again one of the strongest connections (Fig. 11d). Figures 11 e, f, g and h visualize the overlap of both connectivities from different perspectives demonstrating a dominant structural and genetic linkage of CEA and BNST.

Case Study 2: Comparing Networks of Different Modalities and Species

Comparative visualization of human and animal models might be of particular interest for biomedical research and translational psychiatry. To investigate comparative functional networks across species the experts next assess this workflow by exploring functional connectivity and gene co-expression correlation from gene sets related to psychiatric traits, here exemplary autism in human (Li et al. 2017).

For this case study we used voxel-level undirected spatial gene expression correlation (*Data Set 3*), and region-level functional connectivity (*Data Set 2*). The data is retrieved from the *Region-Wise Connectivity Database*, which highlights the usability of our data structure on different levels of hierarchical brain parcellations. An example how the user navigates these hierarchies can be seen in Supplementary Video 2.

To explore and compare global gene expression correlation networks and functional MRI networks across species, it is necessary to find corresponding anatomical brain regions. To our better knowledge, no comprehensive mapping of brain regions between mouse and human exists. Nevertheless, finding similarities in networks can be identified by comparing them iteratively on different anatomical levels in parallel

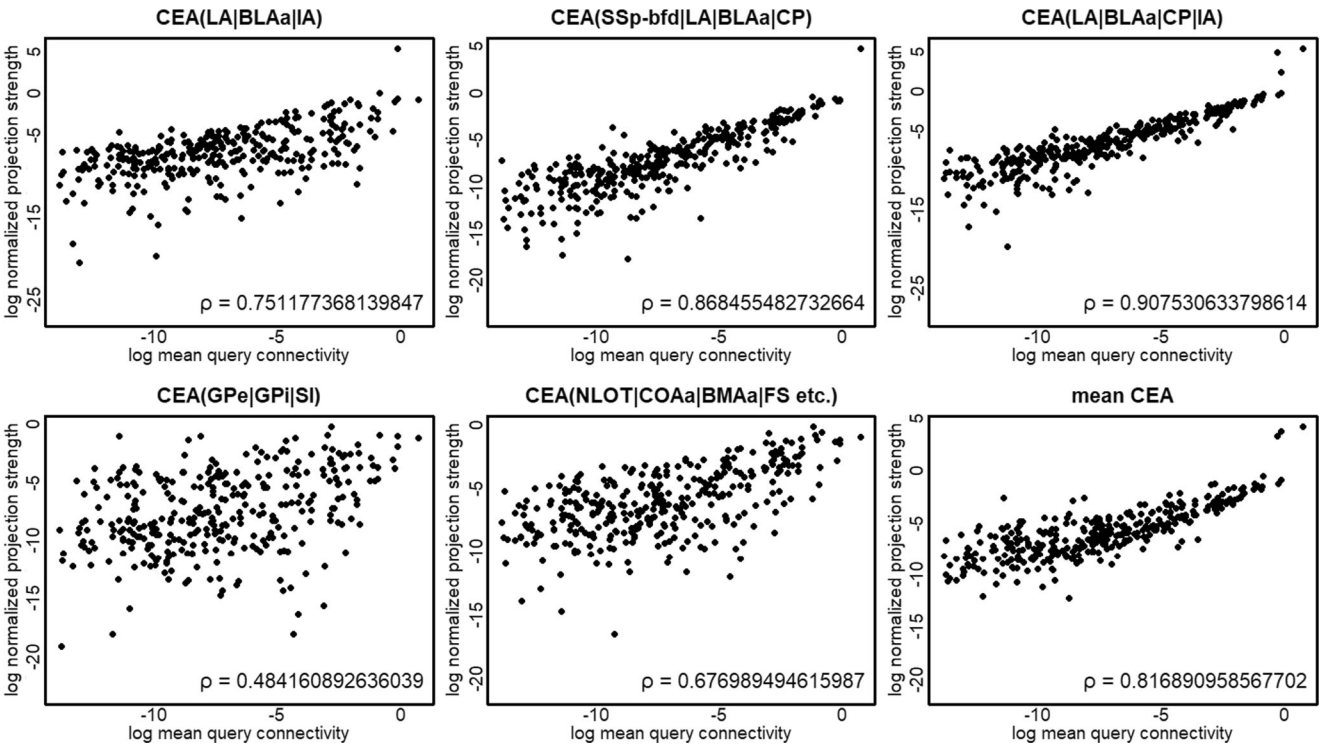

Fig. 12 Correlation between mean outgoing connectivity of the query's VOI (Fig. 11 B) and the normalized projection strength of sites within CEA according to Oh et al. Fig. 3/Supplementary Table 2 (Oh et al. 2014).

between the two species (Fig. 13 a, b, left side, colors are picked from the *AMBA* and *AHBA* and do not correspond to each other). When a user navigates the hierarchical brain parcellations (for mouse and human separately), the data structure returns the connectivity of the selected brain regions in real-time after each interaction (Supplementary Video 2). Consequently, domain experts can iteratively adapt their choice of brain regions not only by their knowledge of individual inter-species region correspondence, but also based on the network similarity.

We found high coupling mostly in cortex (agranular insular and temporal association areas) and primary sensory areas (olfactory, gustatory and somatosensory areas) to the amygdala (central and medial).

Closer inspection of a subnetwork related to social behavior in autism, consisting of higher association cortex, namely insula cortex (IC), frontal pole (FP), hypothalamus (HY) and midbrain (MB), as well as the CEA, revealed gene co-expression correlation within the autism gene set was strongest within cortical regions (FP,IC) and weaker between cortex and subcortical structures (CEA,HY,MB). Figure 13a shows the overlap (product) of functional connectivity and gene expression correlation of this subnetwork for mice, Fig. 13b is analogue for humans.

Discussion

We have shown that our spatial connectivity data structure outperforms state-of-the-art graph engines when querying connectivity of local brain areas. To achieve additional real-time access of outgoing/incoming connections without holding the whole connectivity matrices in the memory, a combination of data compression, spatial locality, memory mapping and hierarchical anatomical annotations is used.

Aggregation of outgoing/incoming connections of a brain area requires the reading of all edges of the involved network nodes. Therefore, row-wise data compression is used based on the specificity of the task: It reduces the total amount of data that needs to be read from the hard drive for whole rows, while it is neglectable that it is not optimized for reading single connections. As shown in the *Evaluation* section, spatial organization and sparsity of the data increases the compression factor by 2, compared to random ordering. Sparsity is often given in neurobiological connectomic data (Sporns 2016), and can be further improved by extended preprocessing of the data (Xu et al. 2015) which we suggest for future projects. One has to note that row-wise compression improves only the reading speed of rows, and therefore outgoing connections. This is not an issue for undirected connectivity graphs, since those are symmetrical (outgoing connections are equal to incoming connections), but requires a separate transposed *Connectivity*

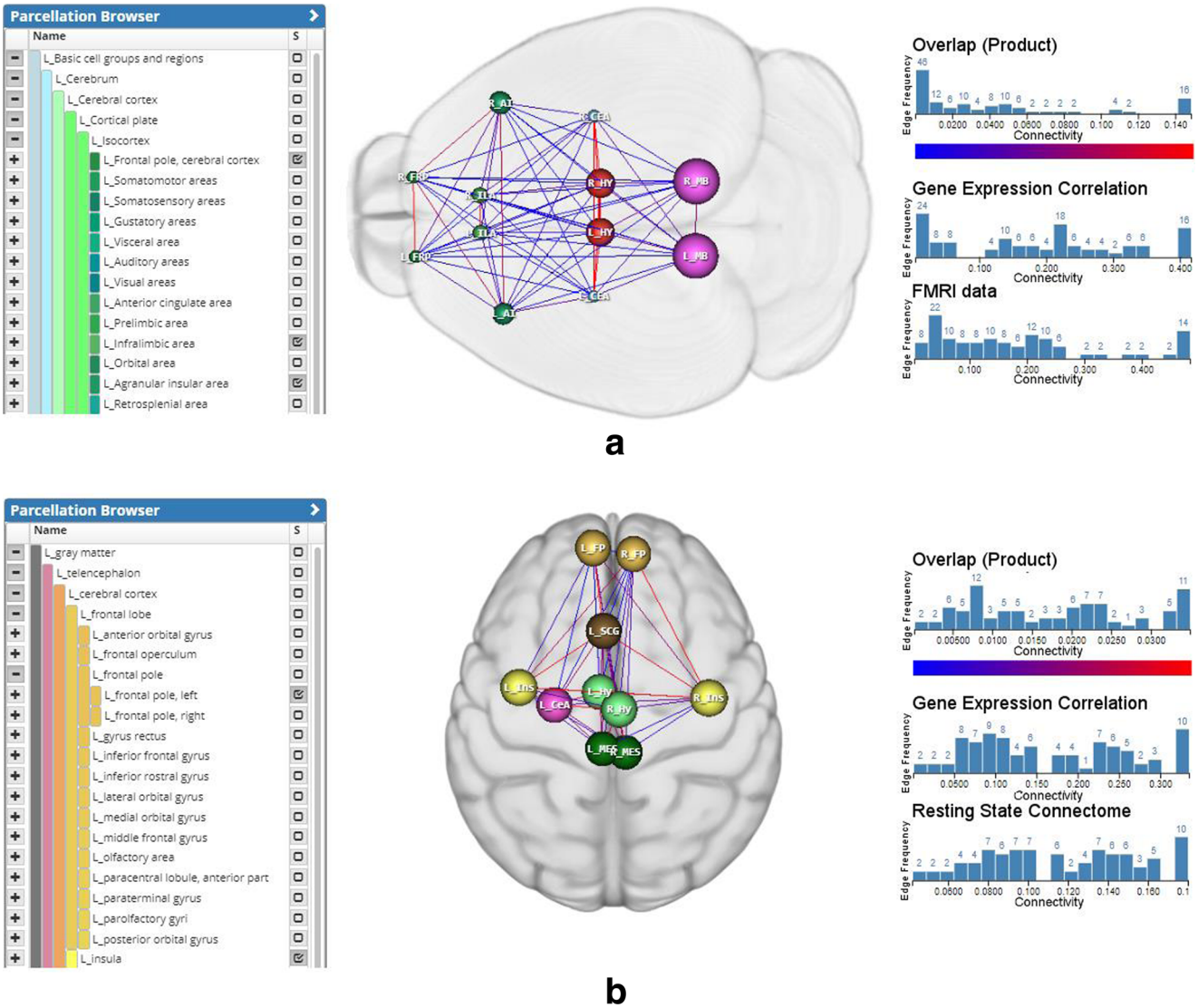

Fig. 13 **a** 2D mouse brain regions (green: cortical regions, red: HY, pink: MB and blue: CEA) and with overlap (product) of functional connectivity and gene expression correlation (blue: weak, red: strong) **b** 2D human brain regions (orange, brown and yellow: cortical regions, green: HY, dark-green: MB and pink: CEA), also with overlap (product of functional connectivity and gene expression correlation

Storage for retrieving incoming connections. While this does not influence query speed, twice the disk space is needed.

A Hilbert curve is used to generate spatial locality of rows, i.e. rows, whose nodes are spatially close in the brain space are also close in the *Connectivity Storage* file. In combination with memory mapping, read-ahead paging, this greatly increases read speed, as shown in section *Evaluation*. Further, it is not necessary to hold a *Connectivity Storage* file in the memory. Therefore, one can access large matrices, with billions of edges, and execute *Aggregation Queries* on multiple matrices sequentially without loading them into memory.

New data can be imported with a REST API which creates *Connectivity Storages* and *Region-Wise Connectivity Databases* in less than an hour for connectivity matrices <100GB, scaling linearly with file size. This enables users to integrate their own, as well as data form large-scale brain initiatives on a consumer-level machine efficiently.

Depending on data acquisition techniques, neurobiological data is available on diverse scales (Betzel and Bassett 2017). To operate on different region wise levels, we used hierarchical anatomical annotations (Lein et al. 2007) and aggregated connectivity from bottom (voxel) to top (large brain regions). Since those annotations consists of only 1288 brain regions, the additional stored connections are neglectable. To bridge the gap between region and voxel levels, we created a row and column indices. These allow retrieving voxel-wise data for brain regions and mapping lower resolution data to a common reference space enabling the comparison of connectivity of different resolution. One has to note that this only represents an upsampling of the data. Since this is done in run-time, a continuous experience in visual analytics workflows is

possible, for data does not need to be preprocessed. Furthermore, this technique can be used to create region-wise caches (voxel-wise outgoing/incoming connectivity of brain regions), or pyramids representations with lower resolution (voxel-wise outgoing/incoming connectivity of lower-resolution super voxels). Although these create additional storage overhead, we show in *Evaluation* that scalability is greatly improved by doing so, hence, future projects could work with even larger matrices in tera- or petabyte range.

Conclusion

In this paper, we present a novel data structure to explore heterogeneous neurobiological connectivity data of different types, modalities and scale for interactive visual analytics workflows. It enables domain experts to combine data from large-scale brain initiatives with user-generated data, by utilizing the hierarchical and spatial organization of the data. Connectivity data at different resolutions, such as mesoscale structural connectivity and region-wise functional connectivity can be queried on different levels on a common hierarchical reference space. On the lowest level, voxel-wise brain networks with billions of edges can be accessed/queried in real-time without having them loaded into working memory. It outperforms state-of-the-art graph engines in receiving connectivity of local brain areas, which allows continuous interactive exploration workflows on consumer level machines and/or via web. We demonstrate this with the implementation of a web-component for visual queries, based on *VOI* selections in 2D slice views. Results are visualized in a 3D volume rendering together with brain anatomy. Case studies conducted with domain experts showed that we could reproduce findings of neural circuits research which are currently extensively investigated experimentally. An inter-species comparison of multimodal brain networks linked to autism showed even more versatile applications, and potential use in studying psychiatric conditions.

For the future, we are aiming to extend this prototype to create a holistic framework for interactive exploration of neurobiological data. This should not only allow to access the data, but also include importing, preprocessing as well as computing network statistics in the web.

Information Sharing Statement

The implementation of the data structure is available upon individual request to the corresponding author, Florian Ganglberger or Katja Bühler.

Acknowledgments We want to thank Florian Schulze, Nicolas Swoboda, Markus Töpfer and Emre Tosun for creating and working on parts of the web-component. This work is the result of a joint VRVis/IMP project supported by Grant 852936 of the Austrian FFG Funding Agency. VRVis is funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG. Wulf Haubensak was supported by a grant from the European Community's Seventh Framework Programme (FP/2007-2013) / ERC grant agreement no. 311701, the Research Institute of Molecular Pathology (IMP), Boehringer Ingelheim and the Austrian Research Promotion Agency (FFG).

References

Ai, Zhiyuan, Mingxing Zhang, Yongwei Wu, Xuehai Qian, Kang Chen, and Weimin Zheng. 2017. "Squeezing out all the value of loaded data: An out-of-Core graph processing system with reduced disk I/O." In 2017 USENIX annual technical conference (USENIX ATC 17*)*, 125–37. Santa Clara, CA: {USENIX} Association. https://www.usenix.org/conference/atc17/technical-sessions/presentation/ai. Accessed 12 June 2018. Accessed 12 June 2018.

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., & der Vorst, H. (1994). *Templates for the solution of linear systems: Building blocks for iterative methods*. SIAM.

Barthelemy, M. (2010). Spatial networks. *Physics Reports.*

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

Betzel, R. F., & Bassett, D. S. (2017). Multi-scale brain networks. *NeuroImage.* https://doi.org/10.1016/j.neuroimage.2016.11.006.

Beyer, J., Al-Awami, A., Kasthuri, N., Lichtman, J. W., Pfister, H., & Hadwiger, M. (2013). ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE Transactions on Visualization and Computer Graphics.* https://doi.org/10.1109/TVCG.2013.142.

Chen, Yiqi, Zhiyuan Lin, Robert Pienta, Minsuk Kahng, and Duen Horng Chau. (2015). Towards Scalable Graph Computation on Mobile Devices. In *Proceedings* - 2014 *IEEE International Conference on Big Data, IEEE Big Data 2014*. https://doi.org/10.1109/BigData.2014.7004353.

Chi, Yuze, Guohao Dai, Yu Wang, Guangyu Sun, Guoliang Li, and Huazhong Yang. (2016). "NXgraph: An Efficient Graph Processing System on a Single Machine". In 2016 *IEEE 32nd International Conference on Data Engineering, ICDE* 2016. https://doi.org/10.1109/ICDE.2016.7498258.

Demir, E., & Aykanat, C. (2010). Efficient successor retrieval operations for aggregate query processing on clustered road networks. *Information Sciences.* https://doi.org/10.1016/j.ins.2010.03.015.

Essen, D. C., Van, S. M., Smith, D. M., Barch, T. E. J., Behrens, E. Y., Kamil Ugurbil, W. U.-M. H. C. P., & Consortium, and others. (2013). The WU-Minn Human Connectome Project: An Overview. *Neuroimage, 80*(Elsevier), 62–79.

Ganglberger, F., Kaczanowska, J., Penninger, J. M., Hess, A., Bühler, K., & Haubensak, W. (2017). Predicting functional neuroanatomical maps from fusing brain networks with genetic information. *NeuroImage.* https://doi.org/10.1016/j.neuroimage.2017.08.070.

Garulli, L. (2010). OrientDB. *Orient Technologies LTD, 2010.*

Han, Wook-Shin, Sangyeon Lee, Kyungyeol Park, Jeong-Hoon Lee, Min-Soo, Kim, Jinha Kim, and Hwanjo Yu. (2013). TurboGraph: A fast parallel graph engine handling billion-scale graphs in a single PC. Proceedings of the 19th ACM SIGKDD international

conference on knowledge discovery and data mining. https://doi.org/10.1145/2487575.2487581.

Haubensak, W., Kunwar, P. S., Cai, H., Ciocchi, S., Wall, N. R., Ponnusamy, R., Biag, J., et al. (2010). "Genetic dissection of an amygdala microcircuit that gates conditioned fear". *Nature* 468 (7321). *Nature Publishing Group*, 270–276.

Hawrylycz, M. J., Lein, E. S., Guillozet-Bongaarts, A. L., Shen, E. H., Ng, L., Miller, J. A., van de Lagemaat, L. N., et al. (2012). An anatomically comprehensive atlas of the adult human brain transcriptome. *Nature, 489*(7416). Nature Publishing Group), 391–399. https://doi.org/10.1038/nature11405.

Hilbert, D. (1891). Ueber Die Stetige Abbildung Einer Line Auf Ein Flächenstück. *Mathematische Annalen, 38*(3). Springer), 459–460.

Kim, J., Zhang, X., Muralidhar, S., LeBlanc, S. A., & Tonegawa, S. (2017). Basolateral to central amygdala neural circuits for appetitive behaviors. *Neuron, 93*(6). Elsevier), 1464–1479.

Kyrola, Aapo, Guy Blelloch, and Carlos Guestrin. (2012). GraphChi: Large-scale graph computation on just a PC disk-based graph computation. Proceedings of the 10th USENIX conference on operating systems design and implementation. https://doi.org/10.1109/HPCA.2015.7056066.

Lein, E. S., Hawrylycz, M. J., Ao, N., Ayres, M., Bensinger, A., Bernard, A., Boe, A. F., et al. (2007). Genome-wide atlas of gene expression in the adult mouse brain. *Nature, 445*(7124), 168–176. https://doi.org/10.1038/nature05453.

Leskovec, J., & Sosič, R. (2016). SNAP: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology.* https://doi.org/10.1145/2898361.

Li, K., Guo, L., Faraco, C., Zhu, D., Chen, H., Yuan, Y., Jinglei, L. V., et al. (2012). Visual analytics of brain networks. *NeuroImage.* https://doi.org/10.1016/j.neuroimage.2012.02.075.

Li, D., Karnath, H. O., & Xu, X. (2017). Candidate biomarkers in children with autism Spectrum disorder: A review of MRI studies. *Neurosci Bull, 33*, 219–237. https://doi.org/10.1007/s12264-017-0118-1.

Lin, Zhiyuan, Duen Horng Polo Chau, and U. Kang. 2013. Leveraging Memory Mapping for Fast and Scalable Graph Computation on a PC. In *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*. https://doi.org/10.1109/BigData.2013.6691739.

Lin, Zhiyuan, Minsuk Kahng, Kaeser Md Sabrin, Duen Horng Polo Chau, Ho Lee, and U. Kang. (2014). "MMap: Fast Billion-Scale Graph Computation on a PC via Memory Mapping". In *Proceedings* - 2014 *IEEE International Conference on Big Data, IEEE Big Data* 2014. https://doi.org/10.1109/BigData.2014.7004226.

Macko, Peter, Virendra J. Marathe, Daniel W. Margo, and Margo I. Seltzer. (2015). LLAMA: Efficient graph analytics using large multiversioned arrays. In *Proceedings - International Conference on Data Engineering* https://doi.org/10.1109/ICDE.2015.7113298.

Markram, H., Meier, K., Lippert, T., Grillner, S., Frackowiak, R., Dehaene, S., Knoll, A., et al. (2011). Introducing the human brain project. *In Procedia Computer Science*. https://doi.org/10.1016/j.procs.2011.12.015.

Oh, S. W., Harris, J. A., Ng, L., Winslow, B., Cain, N., Mihalas, S., Wang, Q., et al. (2014). A mesoscale connectome of the mouse brain. *Nature, 508*(7495), 207–214. https://doi.org/10.1038/nature13186.

Papadias, Dimitris, J Zhang, Nikos Mamoulis, and Y Tao. (2003). Query processing in spatial network databases. Proceedings of the 29th international conference on very large data bases. https://doi.org/10.1016/B978-012722442-8/50076-8.

Pienta, Robert, James Abello, Minsuk Kahng, and Duen Horng Chau. (2015). Scalable Graph Exploration and Visualization: Sensemaking Challenges and Opportunities. In 2015 *International Conference on Big Data and Smart Computing, BIGCOMP* 2015. https://doi.org/10.1109/35021BIGCOMP.2015.7072812.

Poo, M. m., Du, J. l., Ip, N. Y., Xiong, Z. Q., Xu, B., & Tan, T. (2016). China brain project: Basic neuroscience, brain diseases, and brain-inspired computing. *Neuron.* https://doi.org/10.1016/j.neuron.2016.10.050.

Radke, A. K. (2009). The role of the bed nucleus of the Stria terminalis in learning to fear. *J Neurosci, 29*(49). Soc Neuroscience), 15351–15352.

Richiardi, J., & Altmann, A. (2015). Correlated gene expression supports synchronous activity in brain networks. *Science, 348*(6240), 11–14.

Roy, Amitabha, Ivo Mihailovic, and Willy Zwaenepoel. (2013). X-stream: Edge-centric graph processing using streaming partitions. Proceedings of the ACM symposium on operating systems principles. https://doi.org/10.1145/2517349.2522740.

Saalfeld, S., Cardona, A., Hartenstein, V., & Tomančák, P. (2009). CATMAID: Collaborative annotation toolkit for massive amounts of image data. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btp266.

Shekhar, S., & Liu, D. R. (1997). CCAM: A connectivity-clustered access method for networks and network computations. *IEEE Transactions on Knowledge and Data Engineering.* https://doi.org/10.1109/69.567054.

Sherbondy, A., Akers, D., Mackenzie, R., Dougherty, R., & Wandell, B. (2005). Exploring connectivity of the Brain's white matter with dynamic queries. *In IEEE Transactions on Visualization and Computer Graphics*. https://doi.org/10.1109/TVCG.2005.59.

Sporns, O. (2016). Connectome networks: From cells to systems. *In Research and Perspectives in Neurosciences*. https://doi.org/10.1007/978-3-319-27777-6_8.

Tauheed, F., Nobari, S., Biveinis, L., Heinis, T., & Ailamaki, A. (2013). Computational neuroscience breakthroughs through innovative data management. *In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-40683-6_2.

Xu, T., Yang, Z., Jiang, L., Xing, X. X., & Zuo, X. N. (2015). A connectome computation system for discovery science of brain. *Science Bulletin, 60*, 86–95. https://doi.org/10.1007/s11434-014-0698-3.

Young, L. J., & Wang, Z. (2004). The neurobiology of pair bonding. *Nat Neurosci, 7*(10), 1048–1054. https://doi.org/10.1038/nn1327.

Zheng, Da, Disa Mhembere, Randal Burns, Joshua Vogelstein, Carey E Priebe, and Alexander S Szalay. (2015). "FlashGraph: Processing billion-node graphs on an Array of commodity SSDs". In *Proceedings of the 13th USENIX Conference on File and Storage Technologies*, 45–58. FAST'15. Berkeley, CA, USA: USENIX Association. http://dl.acm.org/citation.cfm?id=2750482.2750486. Accessed 12 June 2018.