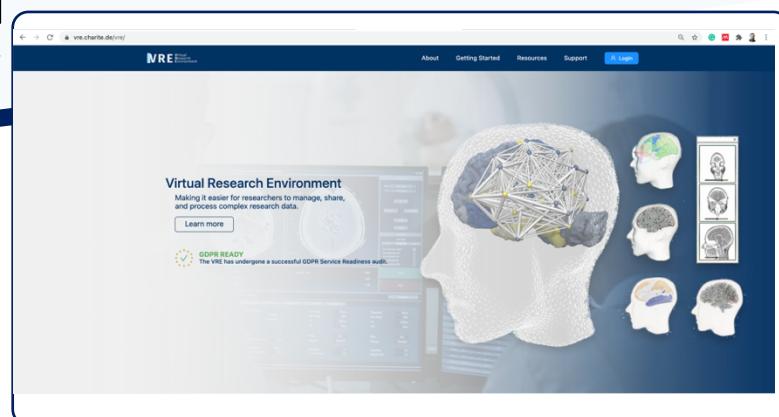




VirtualBrainCloud

Personalized Recommendations for Neurodegenerative Disease



Public deliverable report

D6.4: A functionally integrated version of VBC is evaluated for therapeutic targets (Demonstrator)

Date

December 2021

Pedro Omedas, Paul Verschure (IBEC),
Jochen Mersmann (Codebox),
Aleksandr Masny (FRAUNHOFER SCAI.BIO),
Paula Prodan, Lia Domide (Codemart),
Santiago Brandi (Eodyne), Angela Bradshaw (AE),
Fan Dong, Moyez Dharsee (Indoc/CHARITE),
Patrik Bey, Petra Ritter (CHARITE)

Authors

© VirtualBrainCloud consortium

Dissemination level

public

Website

<https://virtualbraincloud-2020.eu>



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826421



Table of content

1.	Introduction	3
2.	Partners involved	3
3.	Description of the work performed	3
4.	Results	5
4.1.	TVB-Cloud Deployment in the VRE	6
4.1.1.	TVB-Cloud Workflow	8
4.1.2.	Data discoverability	9
4.1.3.	Data analysis and visualization	10
4.2.	TVB-Cloud Software Tools	11
4.2.1.	The Virtual Brain (TVB)	11
4.2.2.	BrainX3	14
4.2.3.	SCAI.BIO View Component	20
4.3.	Use case demonstration in the VRE	21
4.3.1.	Access to VRE and data processing	21
4.3.2.	TVB Web GUI	25
4.3.3.	Jupyter Notebooks	36
4.3.4.	Data pipelines using graphical user interfaces (BrainX3)	38
4.3.5.	Blue Brain Nexus	40
4.4.	Interventions based on the Rehabilitation Gaming System (RGS)	43
4.5.	In silico research: Identifying mechanisms, augmenting diagnosis, testing interventions ..	48
4.6.	Technical coordination and integration	50
5.	Conclusion, next steps	52
6.	Glossary	53
7.	References	54



1. Introduction

2. Partners involved

Partners	Main tasks
1 CHARITE - UNIVERSITAETS MEDIZIN BERLIN (CHARITE)	Multiscale simulation models
2 UNIVERSITE D'AIX MARSEILLE (AMU)	Personalized simulation
3 FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. (FRAUNHOFER)	Semantic and knowledge module, Cloud infrastructure
8 FUNDACIO INSTITUT DE BIOENGINYERIA DE CATALUNYA (IBEC)	Neuroinformatics visualization, Immersive systems, Virtual Reality
12 CODEBOX COMPUTER-DIENSTE GMBH (CODEBOX)	Simulation engine and tools
13 CODEMART Romania (CODEMART)	Simulation engine and tools
14 EODYNE SYSTEMS S.L. ES (EODYNE)	Intervention Tools, Reports, Neurorehabilitation
15. UNIVERSITAT WIEN (UNIVIE)	Legal research and assistance
17. ALZHEIMER EUROPE (AE)	Validation of intervention tools for neurorehabilitation with target groups

3. Description of the work performed

The objective of The VirtualBrainCloud (TVB-Cloud) project is to develop and validate the VirtualBrainCloud, a dedicated cloud-based environment that leverages the potential of big data and high-performance computing (HPC) for personalized prevention and treatment of neurodegenerative diseases (NDD). TVB-Cloud is embedded in The European Open Science Cloud Initiative. It combines already existing technologies and contributes to the development of new Information and communication technology (ICT) services while ensuring the appropriate data safety and protection.

In the present report, a functionally integrated version of TVB-Cloud is evaluated for therapeutical targets. A Demonstrator (webinar) can be found on our project website and on YouTube:

<https://virtualbraincloud-2020.eu/tvb-cloud-videos.html>

<https://www.youtube.com/watch?v=qPLrkRvlgE>

The following is a description of the work performed by WP6 in each task:

T6.1 Semantic integration of data and knowledge for personalized brain simulation M3-M36
LEAD: CODEBOX, Partners involved: CODEMART, IBEC, FRAUNHOFER, CHARITE

TVBase was integrated as a component in the BrainX3 visualization software (IBEC). TVBase is a new methodology (developed by CHARITE) to map information from SCAI knowledge bases (Fraunhofer) to



parcellated brain models. This work is a continuation of the initial prototypes created by IBEC inside BrainX3 and by Codemart and CODEBOX in TVB using the Web-GUI. The TVBase component in BrainX3 is offered through a GUI component, allowing to introduce free text or composed queries. The search is performed on the SCAI knowledge bases services, returning the relevance of the results in brain areas. These results are visualized in a 3D representation of the Glasser brain atlas and allows to explore the information. The results inform the parameters for brain simulations.

T6.2 Personalized multi-scale bioinformatics visualization M01-M42

LEAD: CODEBOX, Partners involved: CODEMART, IBEC, FRAUNHOFER, CHARITE

IBEC has developed a pipeline to generate the 3D representation of brain atlas regions in MNI space from the original parcellations. We generated a representation of the Glasser (Glasser et al., 2016) and the Desikan-Killiany (Desikan et al., 2006) brain atlas and it was integrated in the BrainX3 software to map the results of the TVBase component (developed in T6.1). CODEMART and CODEBOX continued the development of features on TVB and the *tvb-rest-client*, the component that exposes the features of the TVB simulation software to BrainX3. These features are being tested and validated in a use case for interactive exploration of patient specific brain areas, connectivity and semantic knowledge. CHARITE has developed a prototype of a 3D multiscale brain viewer that can visualize computational brain models where different areas are simulated at different degrees of resolution (Figure 1).

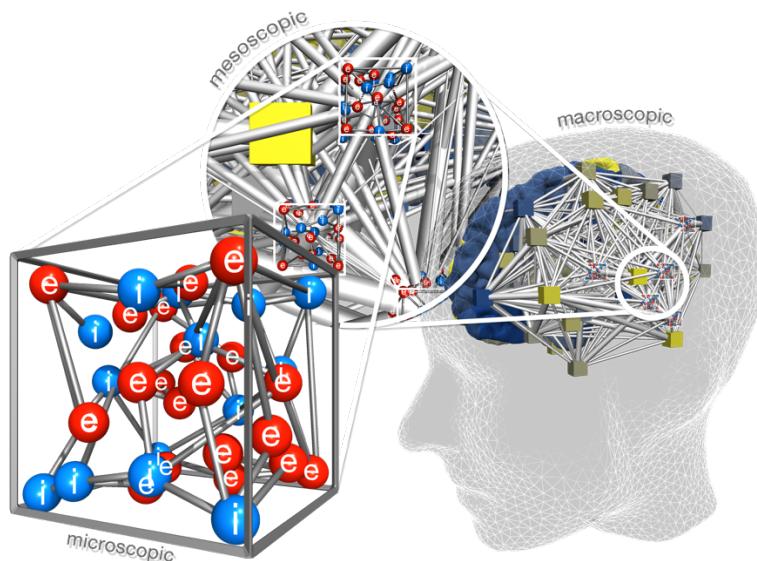


Figure 1. A prototype of a multi-scale brain model viewer has been developed by the CHARITE.

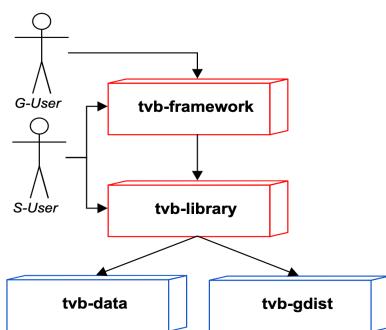
T6.3 Information fusion and model integration in a 3D scaffold that can be parameterized for personalized simulation M24-M46

LEAD: CODEMART, Partners involved: CODEBOX, IBEC, FRAUNHOFER, AMU, EODYNE

The task started in M24 of the project. CODEBOX and CODEMART worked on the integration of the TVB Ontology (TVB-O), a method developed by CHARITE, that supports the integration of mechanistic



cascades related to brain disease with personalized simulations of patient-specific brain network models. Part of the work for this task includes the adaptation of *tvb-framework* adapters and metadata on datatypes, for the integration of the new information (as identified when using BrainX3, or the new data in TVB-Cloud). Also, for making this integrative data and concept functional, we had to update and improve the *tvb-gdist* library component for computing distances on a surface, critical for surface level simulations where the main packages of tvb core are illustrated with their relations (see Figure 2).



This task also focused on the maintenance of the TVB simulation library coding standards and to make the deployment fully integrated and the implementation and integration of external datasets from the EBRAINS platforms (e.g., EBRAINS Tumour Dataset). The wizard for the simulator on the TVB Web-GUI page changed to accommodate the new workflows.

Figure 2. Main TVB core packages (also python packages on Pypi) and their relations.

T6.4 Interactive interfaces for interventions, diagnostics and prognostics M1-M46

LEAD: IBEC, Partners involved: CODEBOX, CODEMART, EODYNE

Eodyne and Alzheimer Europe (AE) have collaborated to organize an evaluation of the cognitive training tools that can be used as part of TVB-Cloud. Eodyne has developed intervention protocols on different platforms (desktop, mobile and VR) and decided with AE to use the mobile solution RGSApp (Rehabilitation Gaming System) for an initial evaluation due to the facility of deployment at home in the current pandemic situation. AE organized an evaluation focused on usability testing with the European Working Group of People with Dementia (EWGPWD). The results of the evaluations have provided feedback to Eodyne to improve the design of the cognitive protocols and tools.

T6.5: Architecture and Integration M1-M48

LEAD: IBEC, Partners involved: CODEBOX, CODEMART, EODYNE, AMU, CHARITE

This task consists of the ongoing coordination of activities across WP6 and technical interactions with other work packages. This includes the discussions, meetings, and workshops to define the software integration, data and infrastructure of the TVB-Cloud project, and existing solutions in the community. Several discussions with the members of the work package were organized, to synchronize the development and the deployment of the tools in the Virtual Research Environment (VRE), the target infrastructure for TVB-Cloud.

4. Results

The main result of this deliverable is the deployment of a TVB-Cloud integrated workflow for clinical data processing, analysis, and simulation in a secure cloud: the Virtual Research Environment (VRE). The TVB-Cloud workflow offers the following features: A web portal to access the cloud environment (<http://vre.charite.de/vre/>), secure data ingestion and pre-processing pipelines, metadata enrichment using the openMINDS standard, integration of knowledge graphs and adoption of standards for data sharing, command line workflows for anonymization and adoption of FAIR data processes, integration of the TVB simulation software, integration of a graphical user interface for data visualisation based on the BrainX3 software, integration of services for semantic searches, and integration with high performance computing (HPC) resources.



4.1. TVB-Cloud Deployment in the VRE

This deployment of the TVB-Cloud within the VRE is built on top of TVB D6.3, with several improvements and new components introduced. The objectives of this deployment are to demonstrate that 1) the VRE provides a suitable environment and adequate IT infrastructure for installing, testing and running fundamental front-end and back-end components of the TVB-Cloud; 2) datasets can flow into the VRE and be accessed by these components and by researchers; and that 3) the TVB-Cloud deployment satisfies relevant legal and ethics requirements. We describe below the process and outcomes of this effort.

Installation and testing of TVB-Cloud Components in the VRE

The TVB-Cloud back-end components are deployed on a virtual machine (VM) residing inside of the VRE Core Extension Zone and only accessible to TVB-Cloud back-end developers. This VM is configured with 16 cores, 64 GB RAM, and 500 GB hard disk capacity, and has Ubuntu (version 18.04 LTS) installed as the base operating system. The number of cores, amount of RAM, and local disk can be expanded if more capacity is needed. The TVB-Cloud back-end components were installed by the CODEMART development team. Members of this team were provided by CHARITE with a virtual private network (VPN) account, which allowed them to connect to the CHARITE network, sign in to the TVB-Cloud VM, and install, configure and test the TVB-Cloud back-end components. In this deliverable, INDOC worked with CODEMART team to integrate the identity and access management system used by TVB-Cloud back-end into VRE such that Single-Sign-On can be enabled to maintain a single source of user identity and prevent users from re-entering credentials when they switch between different VRE components. VRE users can access the TVB Web GUI and the REST server from <https://vre.charite.de/tvb>, and <https://vre.charite.de/tvb-rest>, respectively.

The IBEC team was responsible for installing and testing the TVB-Cloud front-end, BrainX3. This component was deployed on a VirtualGL enabled GPU node provided by the CHARITE HPC team. With VirtualGL, graphical rendering is redirected to the underlying GPU device and only rendered outputs are sent to a remote desktop client over the network, typically proxied by an X-server. In this deployment, Turbo Virtual Network Computing (VNC) Server is installed on the GPU node as the X-proxy. The VRE workbench VMs have Turbo VNC Viewer installed as the remote desktop client to receive the rendered frames. This setup also allows BrainX3 to communicate with the TVB-Cloud back-end components for authentication and data exchange. Following installation, the team at IBEC was able to confirm the correct functioning of BrainX3 using test datasets, viewing the remote-rendered outputs, including connectivity to the TVB-Cloud back-end components.

The Blue Brain Nexus (BBN) software, version 1.5.1, has been deployed to the VRE Kubernetes¹ cluster at CHARITE, following the documentation and Kubernetes manifests provided by Blue Brain Nexus². Dependant services, including Cassandra³, Elasticsearch⁴, Blazegraph⁵ as well as BBN Delta and Web UI were all deployed as Stateful Sets in the utility namespace without any node label selector. Ingress rules were created to allow external path-based routing under the VRE domain to the BBN services. The BBN software was then integrated with Keycloak⁶ by registering a BBN client within Keycloak and configuring BBN with the VRE realm. The three dependant services and BBN Delta all made use of persistent volume claims (PVC) for their storage requirements with these PVCs being backed by the CHARITE NFS.

¹ <https://kubernetes.io/>

² <https://bluebrainnexus.io/>

³ <https://cassandra.apache.org>

⁴ <https://www.elastic.co/elasticsearch/>

⁵ <https://blazegraph.com/>

⁶ <https://www.keycloak.org/>



The DataLad⁷ software, version 0.14.8, has been deployed to the VRE workbench VMs. For DataLad to track file contents and metadata, git⁸ client and git-annex software dependencies, version 8.202108-4-g7961c5a98, was also deployed to the VMs. GitLab Community Edition, version 11.7.4, was used as a Git repository manager for DataLad, which was also deployed as a VRE service with Git Large File Storage⁹ (LFS, version 2.3.4) enabled. Upon execution of the 'datalad push' command, file name metadata (file name associated with symbolic link) is stored within the 'master' branch of GitLab¹⁰, while the annexed file content metadata (file content locations on LFS/special remote) is stored within the "git-annex" branch. These two branches allow for pulling both file names and associated file contents from their respective locations, upon cloning of the GitLab repository and subsequent execution of the "datalad get" command. GitLab was then integrated with Keycloak, allowing users to sign in with their VRE portal credentials. Ingress rules were created to allow external path-based routing under the VRE domain to the GitLab services.

As a result of recent VRE development on improving its storage backend, it became possible now to deploy workbench VMs in the VRE Green Room (the protected zone in the VRE for processing of sensitive datasets) for authorized users to access and process Green Room data. A project member can request a dedicated Green Room workbench VM for a certain period of time. Once approved, that authorized member can have access to that workbench VM through Apache Guacamole¹¹, an open source remote desktop gateway, from the VRE portal. Using this remote access, the user can use the VRE Command Line Interface (CLI) on the VM to retrieve the Green Room data for further processing. Data retrieval is controlled by the user-based access policy configured at the storage backend. Inside the workbench VM, users can also install software packages, execute Singularity containers, and launch graphical application interfaces.

The HPC cluster with SLURM¹² workload management system used by the VRE is fully managed and maintained by Charité GB-IT. VRE users request HPC access from Charité GB-IT and, upon approval, they can log into a job submission node by IP address via Workbench tools integrated into the VRE Portal. The HPC environment provides a scratch space for users to store intermediate results generated during HPC computations. Although the scratch space is a shared network storage area, users have private scratch folders within this space that are only accessible to themselves, thus ensuring segregation of user project data. The scratch space is purged periodically in accordance with data retention policies of the GB-IT HPC facility. Users are instructed to store final outputs of their HPC jobs within their home folder, which can then be pushed back to VRE Core storage using the VRE CLI.

⁷ <https://www.datalad.org/>

⁸ <https://git-scm.com/>

⁹ <https://git-lfs.github.com/>

¹⁰ <https://gitlab.com/>

¹¹ <https://guacamole.apache.org/>

¹² <https://slurm.schedmd.com/>



4.1.1. TVB-Cloud Workflow

With all TVB components deployed inside the VRE, researchers can upload and process their data through the VRE portal, VRE workbench environment, as well as the HPC resources.

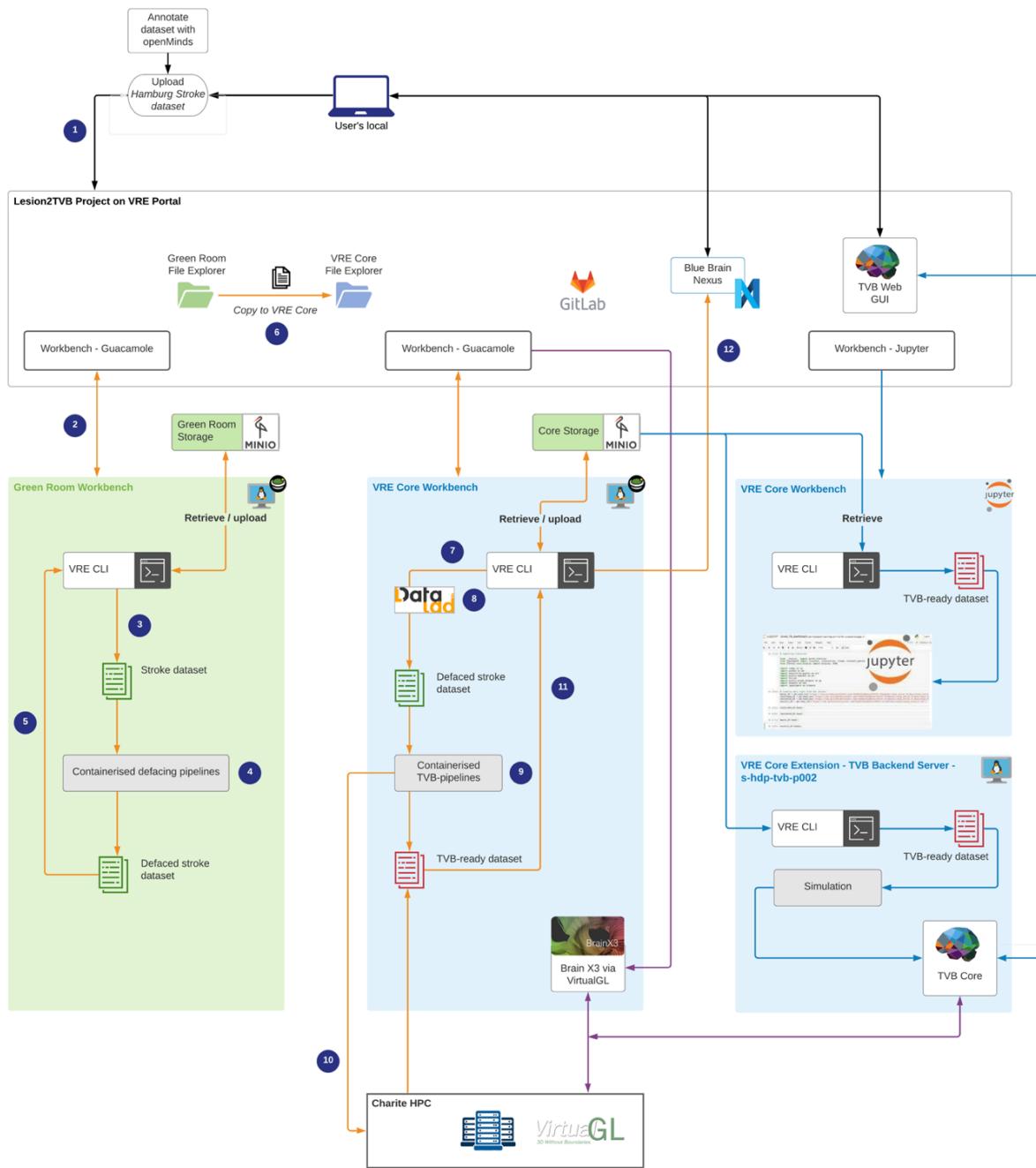


Figure 3: Deployment of the TVB-Cloud components in the VRE

Below is a detailed description of the workflow of the TVB-Cloud components (Figure 3).

- The VRE platform administrator first creates a project, Lesion2TVB, on the VRE portal, and invites TVB-Cloud team members to join the project. One of the TVB-Cloud members is



appointed as the Project Administrator, who can then invite the rest of the team.

- A Lesion2TVB project member signs on to the VRE portal and uploads a dataset together with a JSON file containing openMINDS-based annotation of the dataset (step 1 in the diagram), both of which are saved on the VRE Green Room storage backend with data encryption-at-rest enabled.
- An authorized Lesion2TVB project member enters the Green Room Workbench VM via the VRE portal and remote desktop gateway, downloads the data onto the workbench VM using VRE Command Line Interface (CLI), and runs a containerized defacing algorithm on the data. A graphical user interface is used to inspect the results to ensure that the defacing was successful. The user then uploads the defaced outputs back to the VRE Green Room storage (steps 2-5).
- Using the VRE portal, the Project Administrator approves and copies the defaced dataset to the VRE Core, making it available to all project members (step 6).
- To further prepare the data for this workflow, a project member logs into the VRE Core zone workbench VM, retrieves the defaced dataset via the VRE CLI, creates a DataLad dataset for provenance tracking, and runs the containerized TVB pre-processing pipelines directly on the workbench VM (step 7-9) or submits a job to the SLURM workload manager on the HPC cluster (step 10).
- Using the VRE CLI, the project member finally pushes the processing results, i.e., a TVB-ready dataset, back into the VRE Core storage (step 11) and loads the openMINDS annotation into BBN (step 12).
- Once the TVB-ready data is prepared and saved in the VRE Core zone of the Lesion2TVB project, it can be retrieved by authorized VRE applications via VRE CLI, for instance, by the TVB backend server or by Lesion2TVB workbench tools such as JupyterHub¹³.

4.1.2. Data discoverability

Data annotation (openMINDS¹⁴)

The open Metadata Initiative for Neuroscience Data Structures (openMINDS) represents a current standard in storing and organizing neuroscience metadata in graph databases developed within the Human Brain Project (HBP) and EBRAINS¹⁵ community, and is the base for the integration of such data structures e.g., within the EBRAINS Knowledge Graph. It uses linked JSON (JSON-LD) files to create interlinked resources (entities) of information covering the highest level of meta information for a given dataset down to the most granular level of single file instances for a given subject. The graph-based linking enables the attribution of multiple object properties to various levels of granularity including disease states, file size or versioning. We developed an automated approach creating the openMINDS schema in JSON-LD¹⁶ format for the initial dataset in the Lesion2TVB use case as part of the containerized processing of neuroimaging data.

Knowledge graph

¹³ <https://jupyter.org/hub>

¹⁴ <https://openminds.com/>

¹⁵ <https://ebrains.eu/>

¹⁶ <https://json-ld.org/>



Blue Brain Nexus (BBN)¹⁷ is an opensource software ecosystem to ingest, manage and query data through the use of a Knowledge Graph (KG). In contrast to a SQL database, where knowledge is abstracted into tables, a KG consists of entities (nodes) that are connected to each other with relations (edges). The KG makes it easy to model and visualize complex relationships between entities in a graph, compared to purely SQL-based databases, that require separate tables for each entity and complex join operations for visualization.

As a way to describe entities and relations, BBN uses JSON-LD that natively supports the linking of JSON resources. For this deliverable the metadata of the Lesion2TVB dataset was described using JSON-LD resources that followed the openMINDS data standard.

4.1.3. Data analysis and visualization

Defacing and Data processing

Within the Lesion2TVB use case we apply containerized pre-processing of raw neuroimaging data. The processing is split in two main aspects representing the processing requirements for pseudonymisation (defacing) and creation of TVB-ready connectomes. To this end we created a pseudonymisation Singularity container based on the FSL defacing pipeline (fsl_deface) (Alfaro-Almagro et al., 2018) and deploy it within the VRE Green Room workbench VM. After visual inspection of the defacing results the defaced data is pushed back into the VRE Core storage of the Lesion2TVB project (see steps 2-6 in the workflow diagram, Figure 3).

The pre-processing of the defaced multimodal neuroimaging data is performed using Singularity containers deployed within the project VRE Core VM representing step 9 in the workflow diagram.

Container	Processing Step	Reference
EBC	Enantiomorphic brain correction	Bey et al. (in prep)
HCPPipelines	Structural processing	Glasser et al. (2013)
ParcMap	Parcellation mapping in subject space	Bey et al. (in prep)
MRtrix	Tractography	Tournier et al. (2019)

In a first optional step brain lesion induced abnormalities are corrected using subject specific healthy signal for all input modalities (T1w and FLAIR image volumes). Following the enantiomorphic correction the human connectome minimal processing pipeline (Glasser et al., 2013) is performed creating fully processed volumes within native subject T1w space. Based on the extracted brain ribbon we create subject specific brain parcellations based on the HCP-MMP1 atlas (Glasser et al., 2016) using a containerized mapping pipeline. With the created brain parcellation we can apply the containerized MRtrix3 connectome pipeline (Tournier et al., 2019) to create structural connectomes for a given subject as a base for whole brain simulations using the TVB platform.

HPC integration

To handle the computationally expensive workload of the previous processing pipeline we employ the HPC environment of Charité IT. To this end we connect via ssh from inside the project instance of the VRE core zone VM to the HPC cluster. On the cluster itself we can make use of the VRE CLI tool to retrieve the corresponding data and container images. A SLURM job is created using a standard batch script to submit the processing job to the job management tool of the cluster. After job completion, the VRE CLI is again used to synchronize the results with the project VRE Core storage.

¹⁷ <https://bluebrainnexus.io/>



BrainX3

BrainX3 is the front-end visualization to visualize and explore multimodal data. It can access the data of the TVB2Lesion use case by connecting to the TVB-Server. BrainX3 is installed in a VirtualGL-enabled GPU node running CentOS Linux and accessible to end users via the VRE portal. Users can run the application on this node and interact with it directly within a web browser.

TVB simulation backend

TVB, with its web GUI and jupyter notebook console, can get access to the VRE secured data, after mounting/retrieving it through VRE CLI.

4.2. TVB-Cloud Software Tools

4.2.1. The Virtual Brain (TVB)

The Virtual Brain (TVB)¹⁸ is a full brain activity simulator, mainly operating at the large scale. It can compute simulated signal from source level (centres of the brain regions) to EEG, sEGG, MEG. It supports personalization of the operations through large scale connectivity (extracted from a patient DWI, T1 scans) as well as from annotations on pathologies coming out of literature (TVBase, TVB-O).

TVB is deployed in the VRE in a dedicated VM, and accessible as a web graphical interface. TVB also exposes a REST API, which is used by BrainX3 for visualizing the same data as in TVB Web GUI but integrated differently. VRE has its own Keycloak installation as authenticator, and TVB uses it for its users in the Web GUI.

TVB has also integration with VRE data flow, by importing data after the pre-processing pipeline has produced it, as well as exporting data for VRE storage and indexing. The import happens in the background, but it requires a manual user trigger to happen. This process can also become fully automated, in future.

Similarly, the import of EBRAINS / HBP TVB ready dataset is initiated at the click of a button, but this time using the public dataset from EBRAINS directly.

¹⁸ <https://www.thevirtualbrain.org/tvb/zwei>

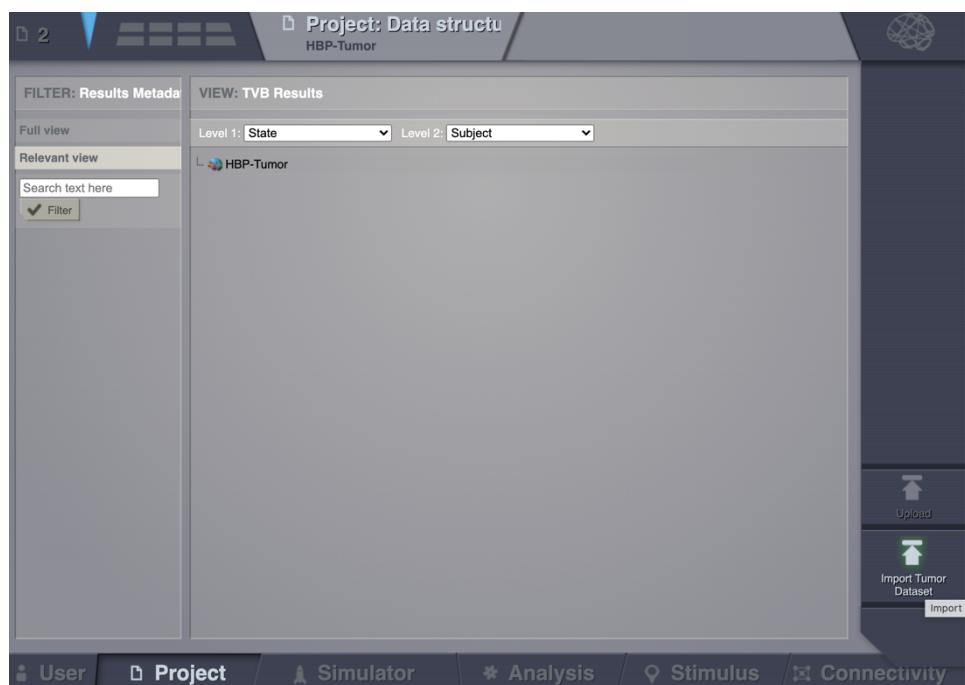


Figure 4 Screenshot from TVB-web GUI with the button to start the remote data extraction in a new TVB project (in this example in HBP Tumor dataset)

Once the import operation starts (button clicked), an info message appears in the top-right area of the screen, informing the user about this.

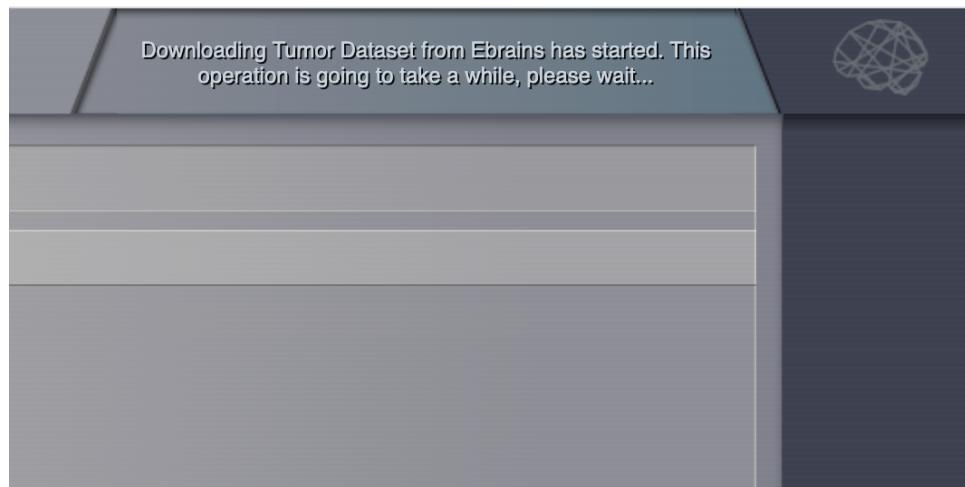


Figure 5 Top-right corner of TVB=web GUI with the info message informing the user about the operation being launched.



The actual operation now becomes visible on the Project Operations page, where we can also monitor its progress (when it is in progress it is coloured in blue, and it becomes green when it successfully finishes execution).

The figure consists of two vertically stacked screenshots of the TVB Web GUI. Both screenshots show the 'Project: Operations' page for the 'HBP-Tumor' project. The top screenshot shows an operation titled 'Import Tumor Dataset' with a status of 'Started'. The bottom screenshot shows the same operation with a status of 'Finished'. The interface includes a filter sidebar on the left and a main table with columns for Status, Control, Operation, and Simulation.

Figure 6 TVB Web GUI with the operation status

The figure shows a screenshot of the TVB-web GUI's 'Project: Data structure' page for the 'HBP-Tumor' project. The main area displays a hierarchical tree of connectivity data under the 'Raw Data' category. The tree structure is as follows:

- HBP-Tumor
 - Raw Data
 - sub-CON01
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON02
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON03
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON04
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON05
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON06
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON07
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19
 - sub-CON08
 - Connectivity [68] - 2021-10-28,16-31-19
 - CorrelationCoefficients - 2021-10-28,16-31-19

Figure 7 Screenshot from TVB-web GUI with the resulting datatypes after an import operation has finished. Different subjects have structural and functional connectivities available in TVB storage, and thus immediately to be used in BrainX3 desktop GUI also.



4.2.2. BrainX3

BrainX3¹⁹ is the visualization software of TVB-Cloud. It is a neuroinformatic tool for exploration, analysis, modelling and simulation of brain data using interactive visualizations that facilitates exploration and discovery of new insights by scientist and clinicians. It is a standalone multiplatform software that supports multiple workflows for data analysis.

BrainX3 can be used as a front-end application to the TVB simulation software. It integrates the tvb-rest-client, a component that allows to communicate with a TVB server, interchange data and execute operations on the server. This section describes the features to interact with TVB server: authentication, data exploration and visualization, and the new TVB-Knowledge base feature.

Authentication

The first step is to launch the connection to the TVB server from the BrainX3 menu and authenticate the connection. For that the user must provide the URL of the TVB server²⁰ and the software will open a browser to request the credentials of the user, using the Keycloak authentication manager integrated in the TVB-Server (Figure 8).

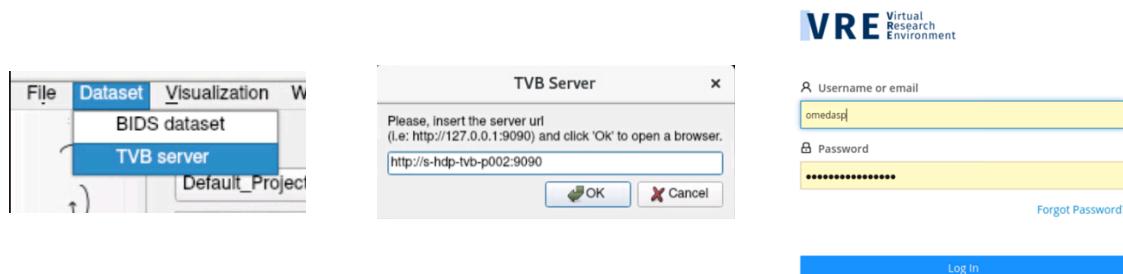


Figure 8: On the left, the option in the BrainX3 menu to launch the connection to the TVB server. On the center, the widget in BrainX3 to introduce the URL of the TVB server (tvb-rest-api) and on the right the authentication box in the browser.

Data exploration

BrainX3 allows to explore the projects stored in the TVB server and their data; this information is displayed in the dataset window. From this window the user can select the project and the data is displayed (Figure 9).

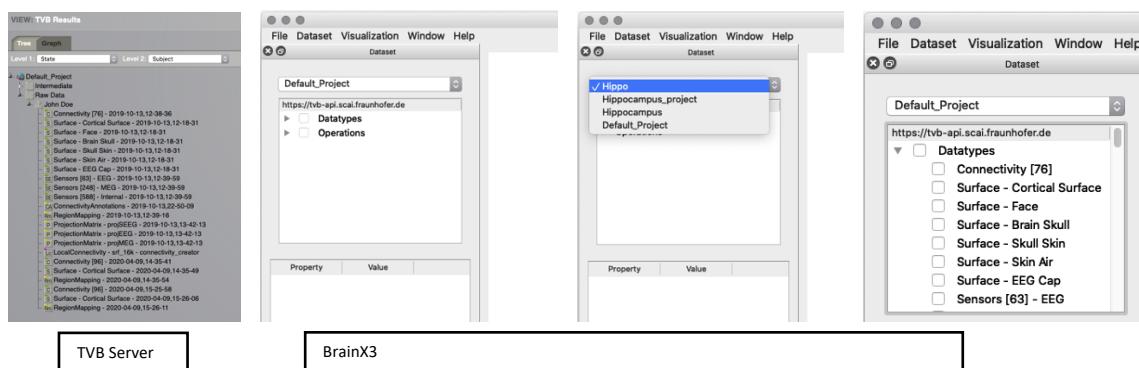


Figure 9: Dataset exploration from BrainX3. On the left it shows the project and the data as displayed in the TVB Server. On the right it shows the interface to explore the project from BrainX3.

¹⁹ <https://www.brainx3.com/>

²⁰ <https://vre.charite.de/tvb-rest>



TVB Data visualization and exploration

From the dataset widget the user can click on the listed data and visualize it. In the background, BrainX3 downloads the data to a temporary location in the VM to be used more efficiently by the software and it is displayed in the appropriate widget. If it is a mesh, volume or 3D object, it will be loaded in the 3D visualization widget and, if it is a signal, it will be loaded in the signal viewer. Once the data is loaded, the user can explore the data interactively. In the visualization window the user can use basic functionalities like zoom and rotation and can modify the properties of the data. In the current version we have integrated the following datatypes: Connectomes, Meshes, Electrodes, Timeseries and Volumes.

Connectivity data

When the user clicks on the connectivity data, it is downloaded from the TVB server to the VM where BrainX3 is running (all in the same protected network of the VRE) and it will be visualized as a 3D graph where each node corresponds to a brain area and each edge represents a measure of connectivity between the nodes. In the example below (Figure 10) we show the visualization of a connectome as a network of interconnected nodes, showing different visualization options.

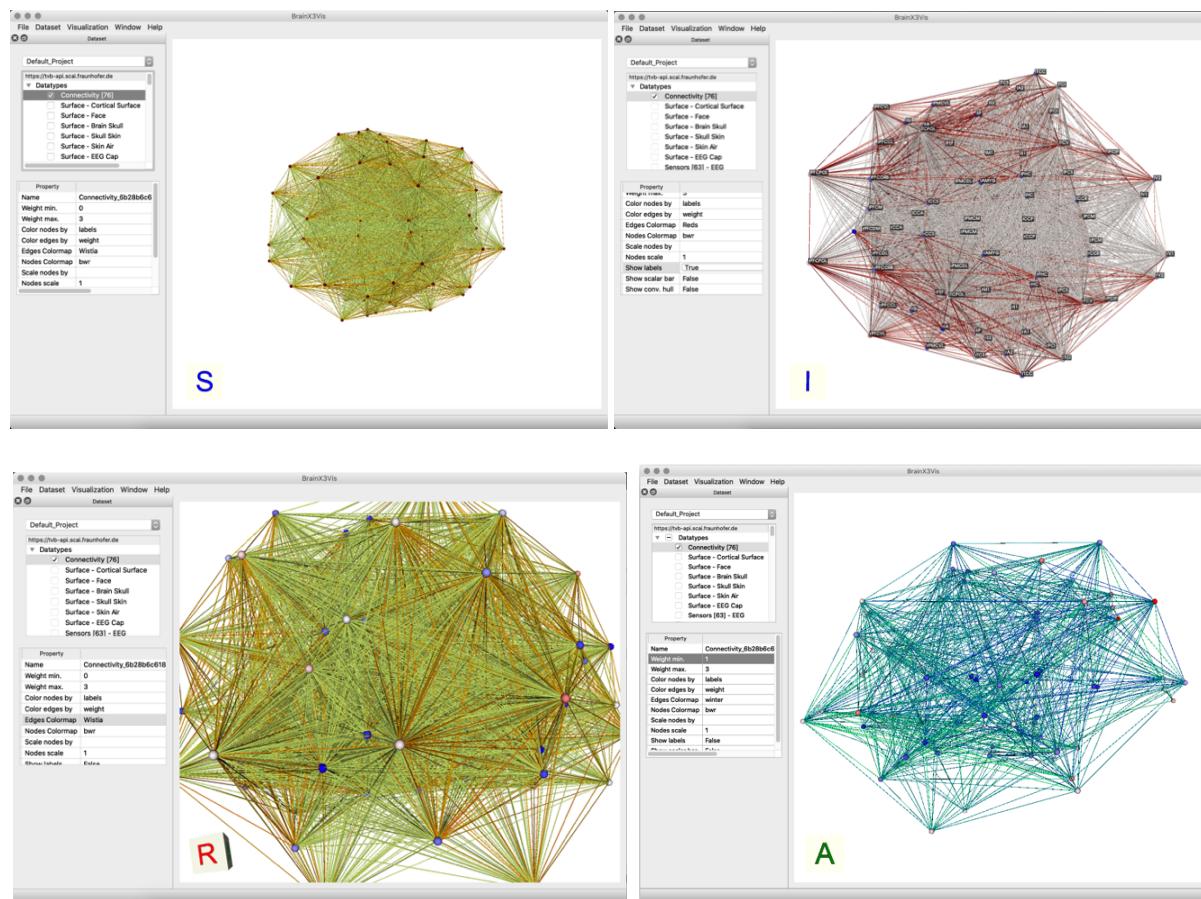


Figure 10: Connectome visualization in BrainX3 using different properties and options. In the top left is the default view, on the top right, the labels associated to the node location of the brain atlas is displayed, on the node sizes has been increased and, on the bottom right, it is using a different color map for the edge connections and is filtering the connections using the weight property of the edges.



Surface and sensors data

BrainX3 allows to explore and visualize 3D data such as surface meshes and sensors. When the user clicks on this data it is loaded in the 3D widget and the user can explore it. The data is aligned to a common 3D space, allowing to explore different modalities at the same time (Figure 11).

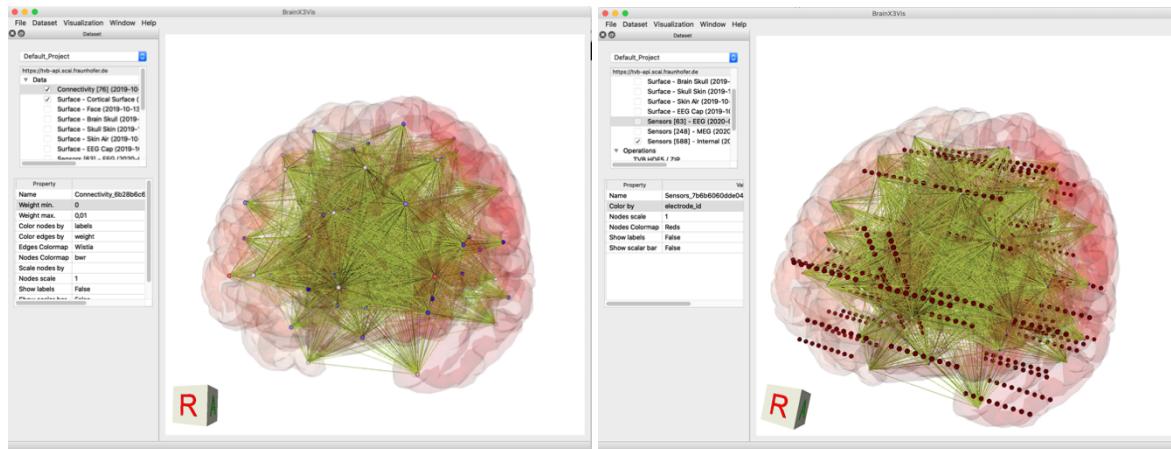


Figure 11: On the left connectivity data and the cortical surface mesh, on the right image it is added the intracranial sensors location to the visualization.

Operations

TVB offers the possibility to perform operations on the data. We integrated in BrainX3 the functionality to launch operations; this can be done by right clicking on the data object in the dataset widget. A contextual menu appears showing the available operations and the user selects the operation to be performed on the data. When the operation is selected, it is launched and executed on the TVB Server using the TVB API. On BrainX3 the user can get a list of launched operations and their status on the dataset widget (running, finish ok or finish with error is indicated with an icon) (Figure 12 left)

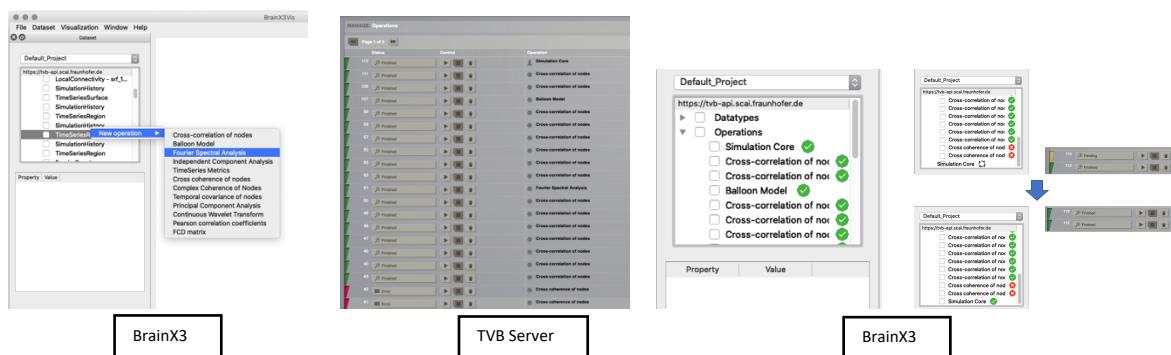


Figure 12: The steps to launch and operation on a dataset using BrainX3 and TVB. On the left the operation is launched on the selected data, the operation is launched in TVB server (second image), and the result is downloaded and the status is shown on the BrainX3 interface (third and fourth images to the right).

Simulations

The main feature of TVB is to perform brain simulations. We integrated in BrainX3 the functionality to launch simulations on the TVB Server. We have integrated in the interface a function to launch a predefined simulation on the server by right-clicking on *Operations* and clicking on *New simulation* (Figure 13Figure 13). The simulation is launched on the TVB server using the TVB API and the status is reflected in the BrainX3 interface. We plan to expand this functionality to allow the user to choose the models and the parameters of the simulations.

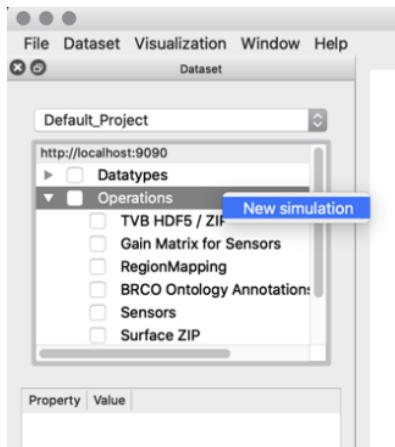


Figure 13: Interface to launch TVB simulations from BrainX3

The result of a simulation in TVB is a TimeSeriesRegion. More precisely, the signals generated by the simulation on each of the nodes of a Connectivity. Once the simulations are finished, the user can click on the result and visualize them with the reference data (connectivity matrix and surface) (Figure 14).

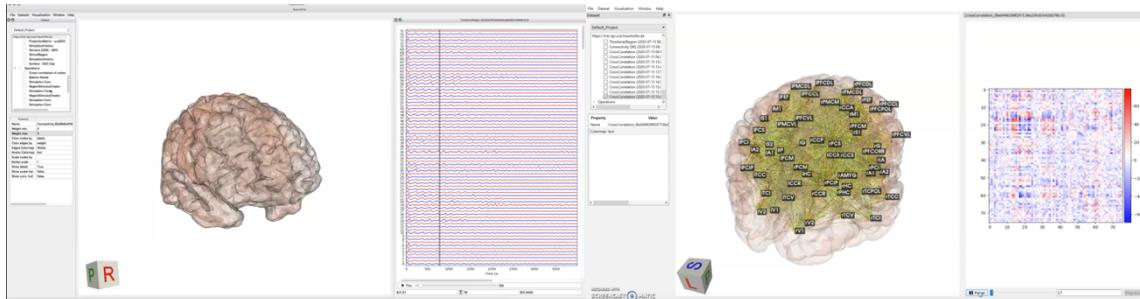


Figure 14: Visualization of brain datasets and TVB simulation results in BrainX3.

TVBase

BrainX3 integrates the functionality of the TVB Knowledge Base Adapter TVBase, a method that maps information from automated literature mining derived knowledge bases to the 3D space of brain network models – as built with TVB simulation software. Users can select any search terms or combinations to generate spatial feature maps from the knowledge base SCAIView that subsequently can inform multi-scale brain simulations. This approach allows to map in real time information from the literature-mining knowledge base of SCAIView to a 3D brain space for effortless integration of biological information in multi-scale whole brain models. This method has been implemented in Python code that can be used in a Jupyter Notebook.

The TVBase functionality has been integrated to BrainX3 to facilitate the use and expand the possibilities to explore the results in a 3D brain space. A graphical user interface has been created to introduce the search terms and show the results produced by the library, an interactive plot of the cortical surface of the Glasser atlas showing the relevance of the results in a colour scale (Figure 15)

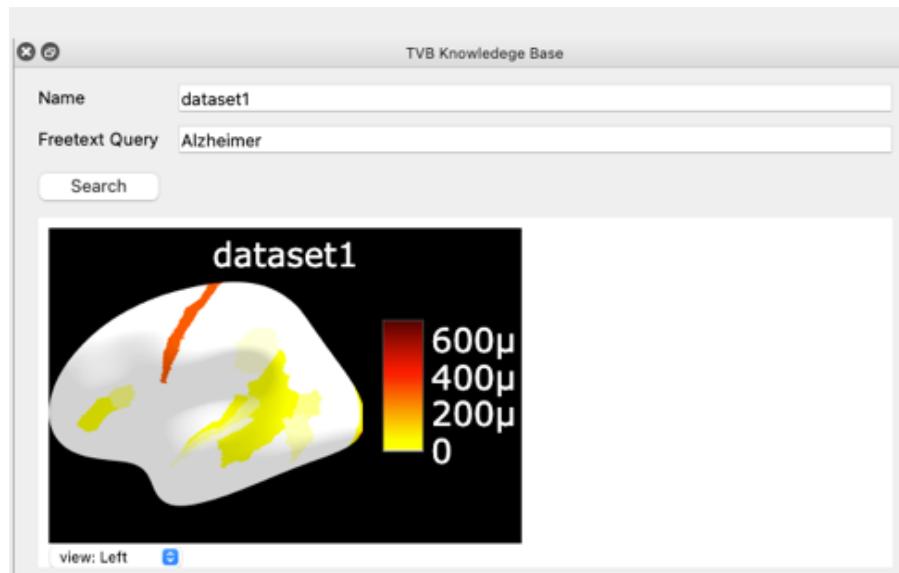


Figure 15: The interface of TVBase in BrainX3

To allow interactive exploration of results linked to semantic knowledge, a 3D representation of the Glasser Atlas in MNI space has been integrated. The results of TVBase are mapped to the 3D presentation and are displayed in the main window, allowing to align multiple modalities with the results of TVBase and explore.

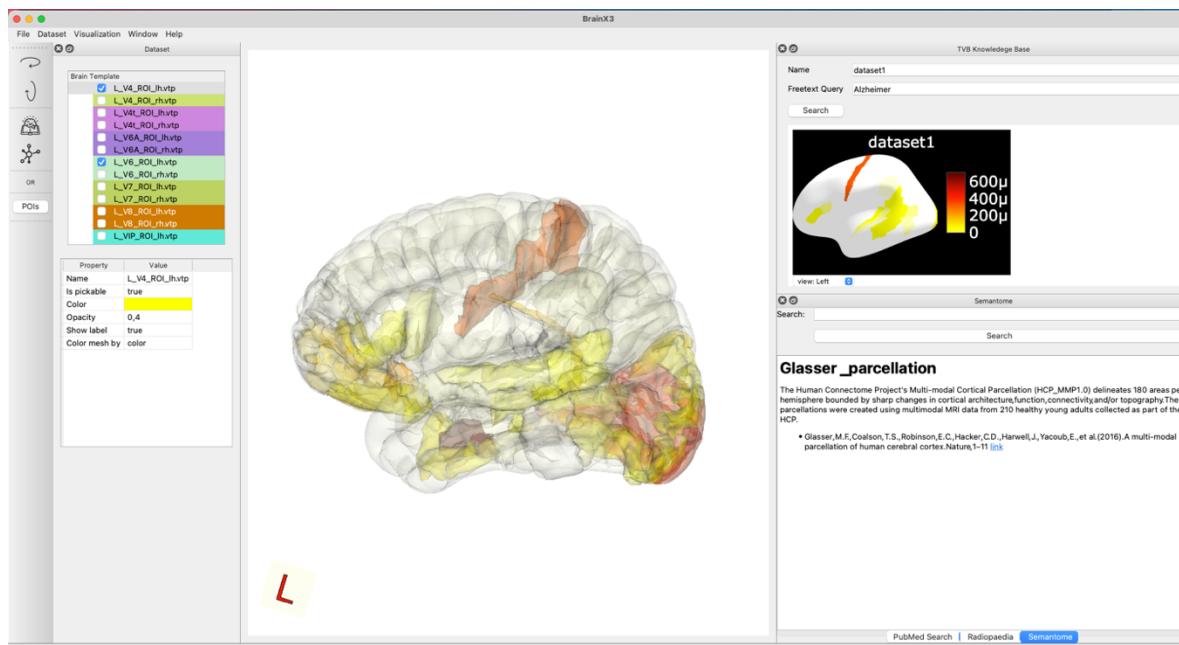


Figure 16: Results of TVBase mapped into brain areas of the Glasser atlas, showing the relevance of the result in colours

The user can start to explore and interact with the results in the anatomical representation. The areas can be selected and it will be highlighted with the name of the area. Semantic information and searches into knowledge base databases will be triggered from these selections as a next step.

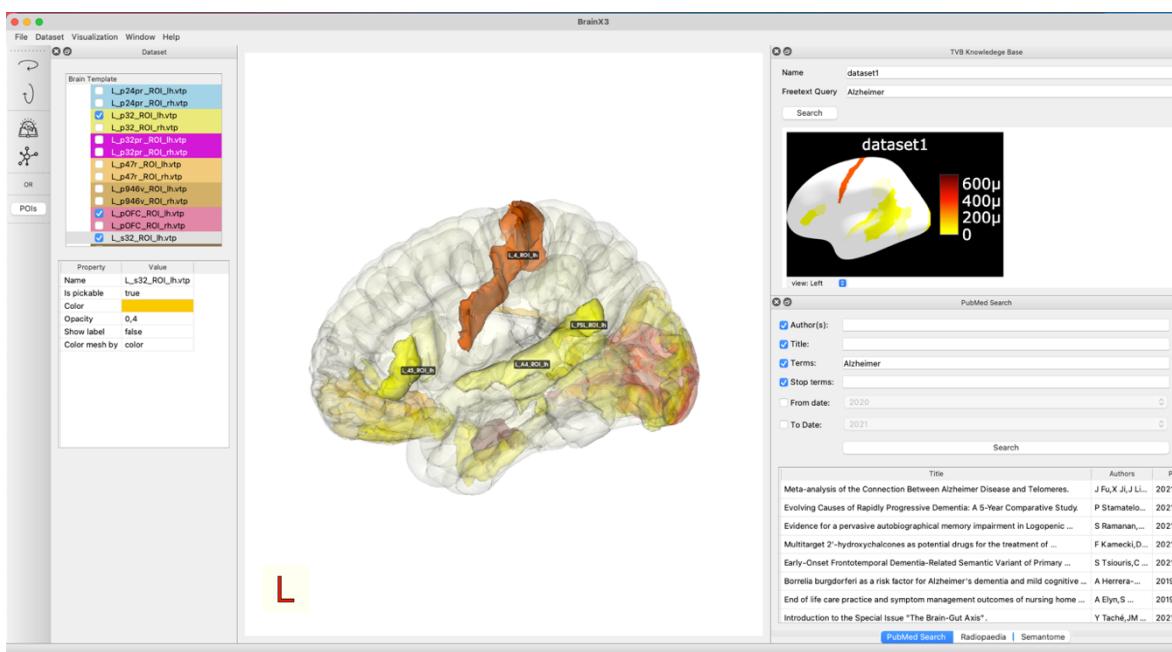


Figure 17: Highlighted areas of the Glasser atlas and the identification

Additional data can be loaded, like MRI images and tractographies. The user can also start to define Regions of Interest (ROI) and add annotations (See figure Figure 18).

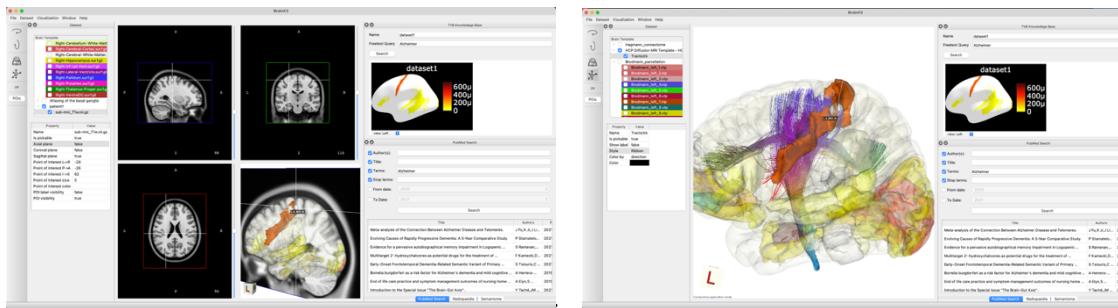


Figure 18: On the left, the results of TVB aligned with an MNI template. On the right the TVBase results are shown with an average tractography template.

Atlas integration

Since BrainX3 works with a variety of neuroimaging modalities, it is critical to have access to a variety of atlases to map information to specific brain areas. Our goal is to make BrainX3 not just beneficial for researchers, but also a useful tool for physicians. To map spatial data onto the highly parcellated atlas, we decided to use the Glasser atlas. We also merged the Desikan-Killiany atlas to address the spatially poor data in the case of Electroencephalogram or Magnetoencephalogram (EEG/MEG) data onto the average parcellated brain atlas.

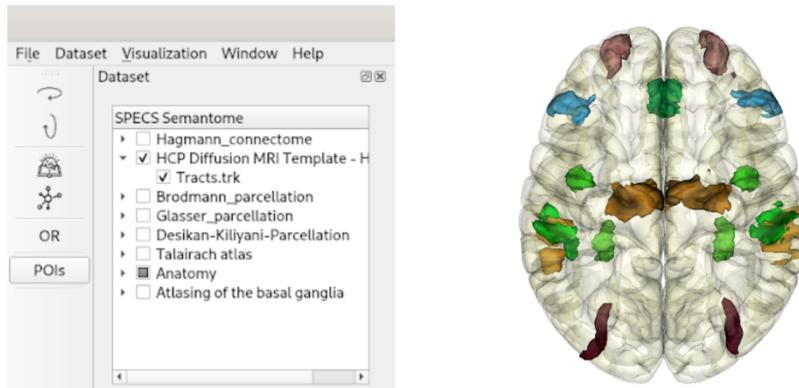


Figure 19: On the left panel, the newly added atlases are displayed. From the atlas of interest, one can visualise structures on the 3D MNI template. For example, the labels from Glasser atlas are shown in the image below

To process the MRI volume, we utilized the Python packages nibabel and nilearn. The MRI volume containing label information was first resampled to MNI space using the MNI152 template, after which the volume was partitioned on one axis to capture only one hemisphere. Using the Python VTK module, each hemisphere was then used to produce 3D brain surfaces and using BrainX3, the surfaces were displayed.

4.2.3. SCAI.BIO View Component

The RESTful APIs of the following services have been maintained and improved:

- Semantic search engine SCAIView²¹: API v7 was developed, that is based on Docker, Docker compose and Eureka (artifacts can be downloaded from the SCAI Docker registry);
- NDD knowledge Base (KB) and Biomedical Knowledge Miner (BiKMi) and explorer²²
- Semantic Aggregation Machine (SAM²³);
- Virtual Brain Lookup service (based on OLS²⁴).

All services provide an OpenAPI (Swagger REST) interface, that help to get acknowledged with the API's endpoints and the functionalities. It is also possible to use the OpenAPI Generator and generate API client libraries or server code²⁵.

As it was already mentioned in section 4.3.2. BrainX3, the semantic search engine SCAIView was used to create the TVB-Knowledge Base (TVB-KB), providing a literature-mining service. It is also planned to deploy SCAIView semantic search engine in the VRE to provide a stable interaction with TVB-KB.

²¹ <https://api.neuro.scaiview.com/swagger-ui.html>

²² <https://graphstore.scai.fraunhofer.de/studio/index.html#/>

²³ <https://sam.neuro.scaiview.com/swagger-ui.html>

²⁴ <https://ols.neuro.scaiview.com/index>

²⁵ <https://github.com/OpenAPITools/openapi-generator>



4.3. Use case demonstration in the VRE

In the previous sections we have described the VRE infrastructure, deployment of tools in the VRE, and detailed features of these tools that support the TVB-Cloud workflow. In this section we describe in detail the steps that can be performed in the VRE by a registered user.

4.3.1. Access to VRE and data processing

The first step is to access the VRE from the public URL²⁶ (Figure 44). Only registered users can get access to portal.

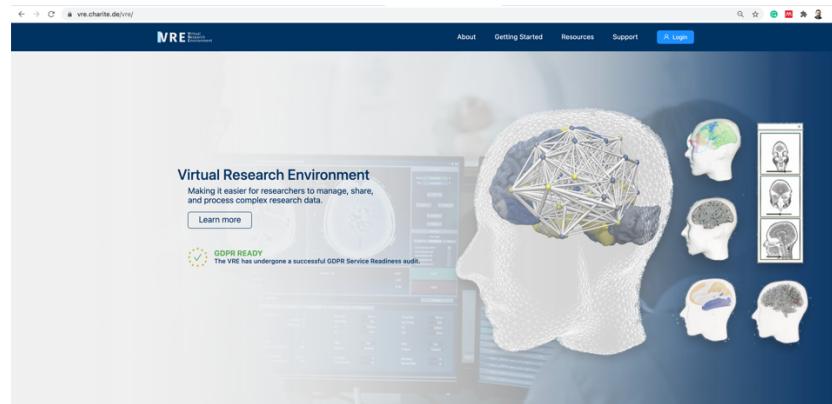


Figure 20 The VRE portal

The user is invited by a Project Administrator to access a project, in this case the project Lesion2TVB. We here describe the containerized data processing and handling as implemented for the current Lesion2TVB use case project set up inside the VRE portal (Figure 21Figure 21).

Figure 21: View of the projects in the VRE portal

A dataset from a subject (sub-0000), composed by a Nifti image and the corresponding openMINDS metadata schema are uploaded into the VRE Green Room storage.

²⁶ <https://vre.charite.de/vre/>



		Name	Added	Created	Size	Action
<input type="checkbox"/>		openMINDS	beyp	2021-11-10		
<input type="checkbox"/>		sub-0000	beyp	2021-11-10		

Figure 22: Dataset uploaded in the VRE portal

Upon upload, the data are stored in the VRE Green Room storage area of the Lesion2TVB project. After integrating the data using the VRI CLI tool (a command line client that provides access to VRE managed services and data), we run the described defacing pipeline for the given anatomical input modalities and can visually check the results using a Nifti visualization tool (MRICroGL²⁷) inside the VRE Green Room VM.

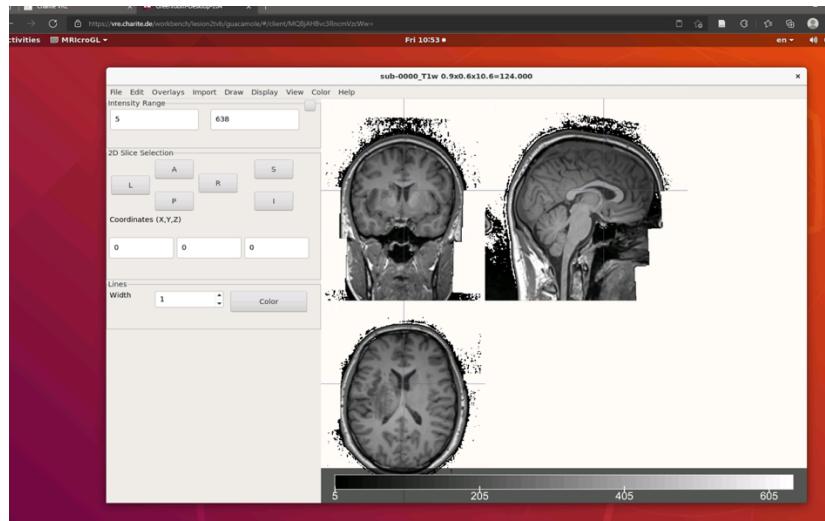


Figure 23: The visualization of the results of the defacing pipeline processed in the VRE and visualized from a browser running in a VRE workbench VM

The results are then uploaded back into the VRE Green Room storage to be approved and copied by the Project Administrator to the project's VRE Core storage.

²⁷ <https://www.nitrc.org/projects/mricrogl/>

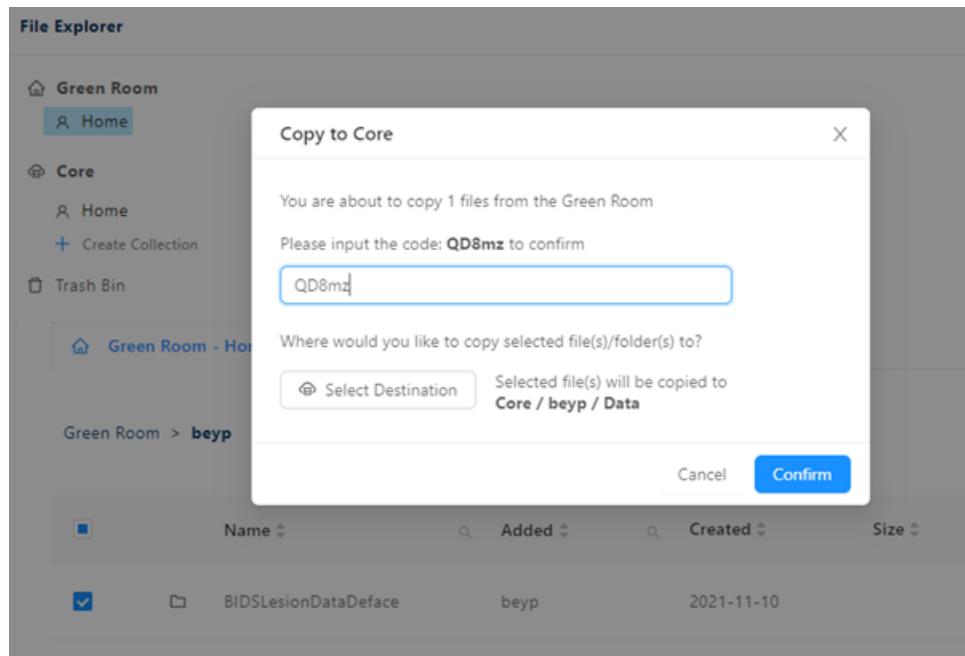


Figure 24: Process to copy the data from the Green Room to the Core zone

Once the upload the VRE Core storage is complete, we access a VRE Core VM to further process the data. First we download the data inside the VM using the VRE CLI tool.

```
beyp@s-hdp-vre-d013:~/Lesion2TVB/Data$ vrecli file sync lesion2tvb/bepy/ Data/BIDSLesionDataDeface . -z vrecore
Preparing status: READY_FOR_DOWNLOADING
start downloading...
Downloading lesion2tvb_1636546702.9238212.zip |████████████████████████████████| 100% 00:00
File has been downloaded successfully and saved to: ./lesion2tvb_1636546702.9238212.zip
beyp@s-hdp-vre-d013:~/Lesion2TVB/Data$
```

Figure 25: Data download in the Virtual machine in the VRE

In a first processing step we perform lesion correction using a developed singularity image performing enantiomorphic brain correction (EBC) for all input modalities using the following code snippet.

```
cd $HOME/Lesion2TVB/Container

singularity run \
-bind "${HOME}/Lesion2TVB/Data/BIDSLesionDataDeface":/data
-env Input="/data/sub-0000/ses-01/anat/sub-0000_T1w.nii.gz" \
-env Mask="/data/sub-0000/ses-01/anat/T1w_lesion_mask.nii.gz" \
Ebc
```

To validate the results we use the GUI enabled VRE Core VM and visually validate the performance.

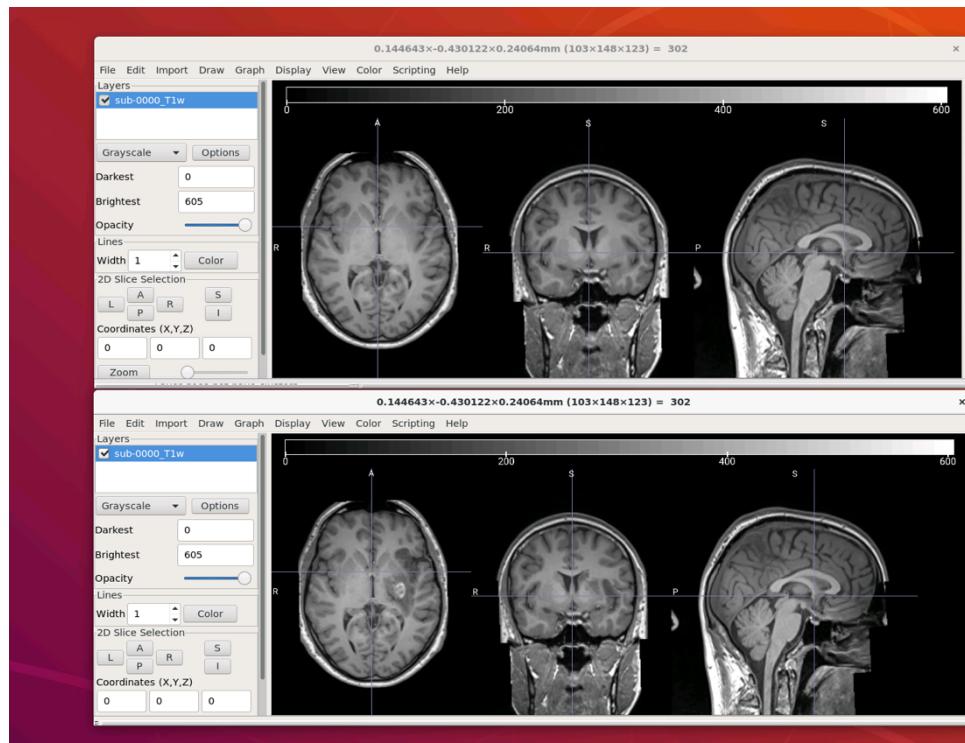


Figure 26: Example performance for ebc for a given subject before (bottom) and after correction (top).

Using the corrected data, we start the computationally expensive processing using the Human Connectome Project (HCP) minimal processing pipeline inside the HPC environment of Charité IT. To this end we connect using ssh to the cluster.

```
beyp@s-hdp-vre-d013:~$ ssh beyp@s-sc-test3
Welcome: Scientific Computing Team @Charite
This is frontend node s-sc-test3.charite.de of the Charite Compute Cluster.

You can find the User Documentation, various HOWTOs and contact information
in our Gitlab Wiki:

https://git.bihealth.org/charite-sc-public/sc-wiki
Have a lot of fun!: Scientific Computing Team @Charite
Last login: Tue Nov  9 22:42:50 2021 from 10.32.42.205
[beyp@s-sc-test3 ~]$
```

Figure 27: Connection the HPC cluster from the VRE virtual machine in the core zone.

Inside the cluster we can use VRE CLI again to retrieve the created data and submit a batch script to the SLURM job management tool of the cluster, calling the corresponding HCP singularity image for processing.



```
cd Lesion2TVB/Container
```

```
Path="home/beyp/Lesion2TVB/Data/BIDSLesionDataDefaceCorrected"
OutDir="/home/beyp/Lesion2TVB/Data/Processed"
SubID="0000"
Steps="PreFreeSurfer FreeSurfer PostFreeSurfer"

sbatch \
HCPPipelinesBatch.sh \
--path="${Path}" \
--subject="${SubID}" \
--outdir="${OutDir}" \
--stage="${Steps}"
```

In a similar fashion we can process the diffusion imaging data to create structural connectomes for the patient and create TVB-ready data that is then pushed back into project-specific VRE Core storage for subsequent retrieval by the TVB backend.

4.3.2. TVB Web GUI

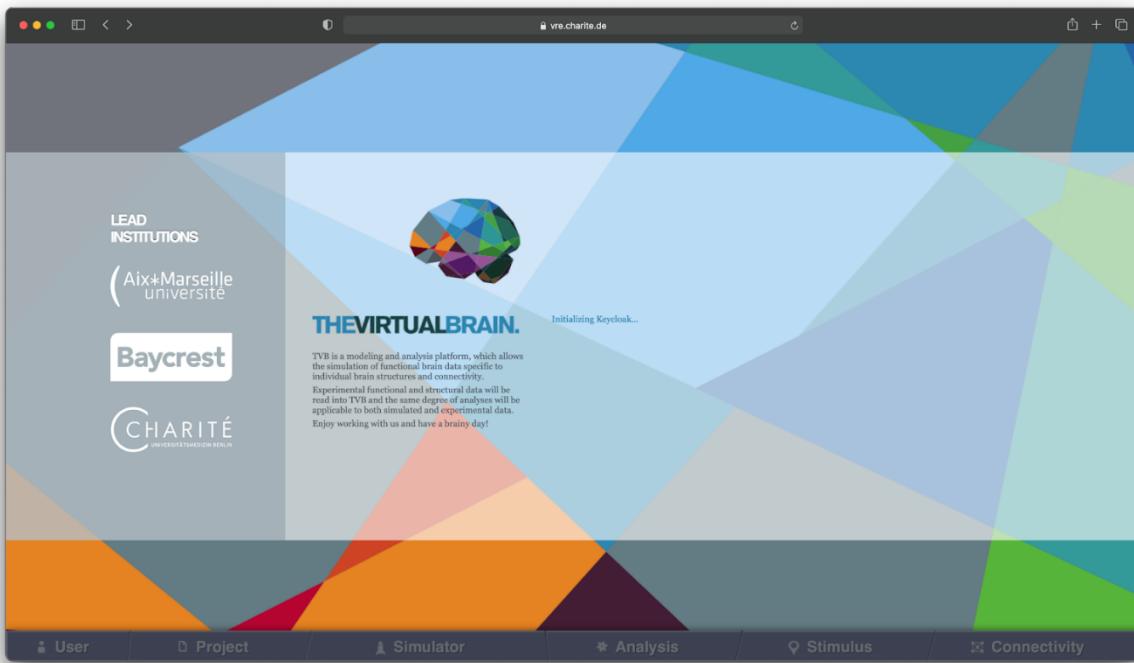
To continue the workflow, we can perform analysis and simulations with the data uploaded in the previous step using the TVB simulation software installed in the VRE. To access TVB, the user can launch a Core VM from the “Guacamole” option (the remote desktop gateway).

In the Core VM, from the Terminal, the user can download the Lesion2TVB TVB-ready data via the VRE CLI tool. Below is an example of using the VRE CLI to download a connectome from the Lesion2TVB project:

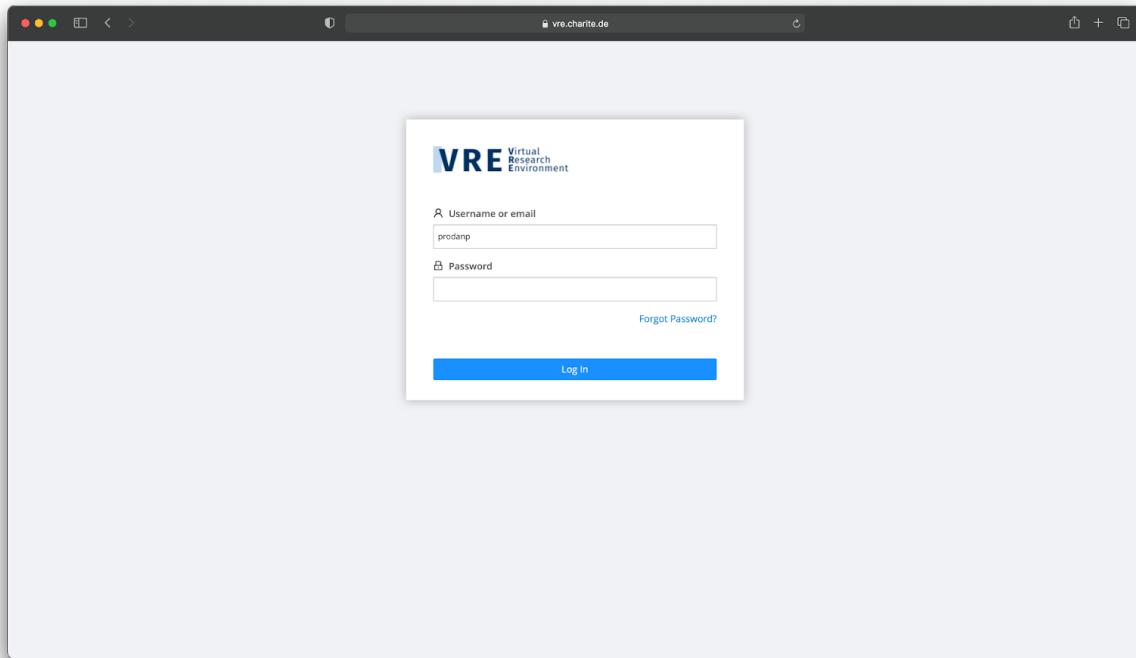


```
prodanp@s-hdp-vre-d013:~$ vrecli user login
Username: prodanp
Password:
Welcome to the VRE Command Line Tool!
prodanp@s-hdp-vre-d013:~$ vrecli file sync lesion2tvb/bepy/Data/TVB_ready/connectivity_383.zip . -z vrecore
```

In order to access the TVB Web GUI available in the VRE, the user opens the following URL in a local web browser: <https://vre.charite.de/tvb>. This launches the TVB Web GUI, first redirecting the user to the VRE login page.



The user signs on using their VRE credentials. Only users with a VRE account can log into this instance of TVB Web GUI.

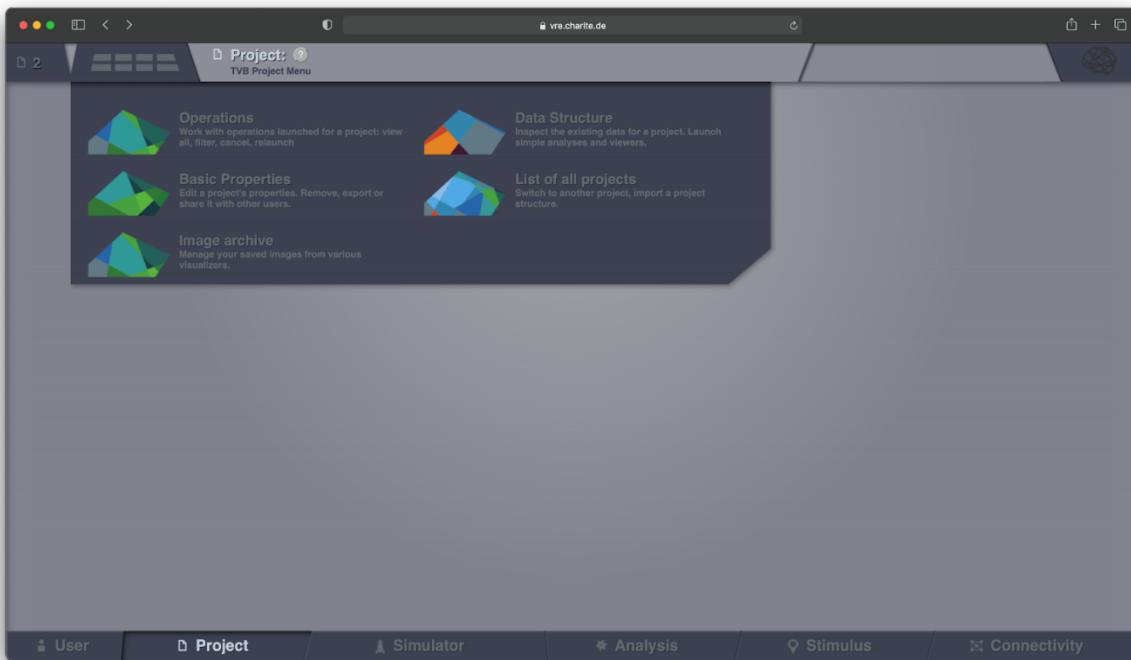


Once the Login button is clicked, the user is redirected to the TVB Web GUI landing page. Here, the user can see their user profile and some details about the current version of TVB that is installed on the VRE.

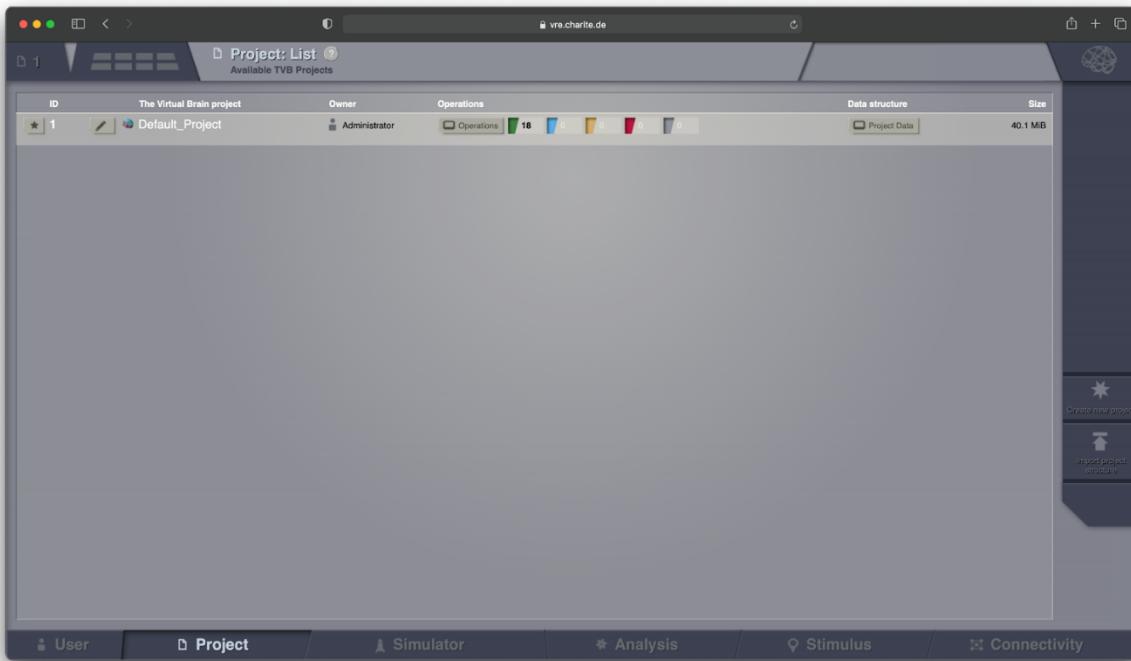
On the bottom-side of the page, the user has access to the main menu. By clicking on “Project”, a



submenu opens on the upper-side of the page, allowing the user to choose “List of projects”.



A web page then opens where the user can see a list of projects that they own, along with projects that they have been given permission to access by the project owners.



In order to create a new project, the user can click on the button highlighted in red below:



This screenshot shows the 'Project: List' page of the VirtualBrainCloud interface. At the top, there's a header bar with the title 'Project: List' and a sub-header 'Available TVB Projects'. Below this is a table with columns: ID, The Virtual Brain project, Owner, Operations, Data structure, and Size. There is one entry: ID 1, The Virtual Brain project 'Default_Project', Owner 'Administrator', Operations section showing 18 items, Data structure 'Project Data', and Size '40.1 MB'. On the right side of the main content area, there's a sidebar with a 'Create new project' button, which is highlighted with a red box. The bottom of the screen features a navigation bar with tabs: User, Project (which is selected), Simulator, Analysis, Stimulus, and Connectivity.

On the next page, the user first chooses a name and an optional description for the project. On the right-side of this page, the user can select which other VRE users can access the project. As shown in the example below, no user has access to this project, other than the current user which is the project owner.

As the owner of this project, the current user can allow other users to access this project by checking the box in front of their name and clicking the “Save changes” button.

This screenshot shows the 'Project: Properties' page. The left panel is titled 'EDIT: Project attributes' and contains fields for 'Project Name' (set to 'Lesion2TVB') and 'Project description'. The right panel is titled 'SHARE: Data with other users' and shows a list of users under 'Page 1'. The users listed are: admin, Administrator, Fan Dong, Moyez Dharsee, Pedro Omellas, Susan Evans, Shan Wang, and Zhejin Zhang. None of these users have checked the 'admin' checkbox next to their names. At the bottom right of the right panel, there is a 'Save Changes' button, which is highlighted with a red box.



After the project is created, it is displayed in the “List of projects” page:

ID	The Virtual Brain project	Owner	Operations	Data structure	Size
2	Lesion2TVB	Paula Prodan	Operations: 19, 0, 0, 0, 0, 0	Project Data	0.0 kB
1	Default_Project	Administrator	Operations: 19, 0, 0, 0, 0, 0	Project Data	56.6 MB

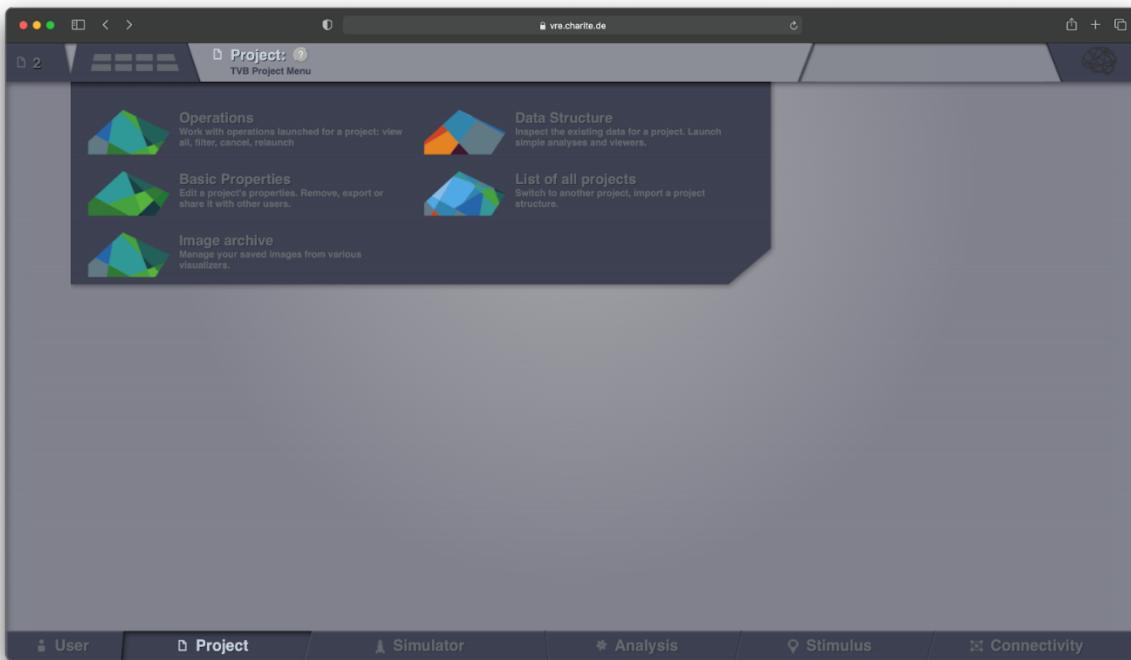
By clicking on the “Edit project” button, highlighted in red, the user can review and edit the project details:

ID	The Virtual Brain project	Owner	Operations	Data structure	Size
2	Lesion2TVB	Paula Prodan	Operations: 19, 0, 0, 0, 0, 0	Project Data	0.0 kB
1	Default_Project	Administrator	Operations: 19, 0, 0, 0, 0, 0	Project Data	56.6 MB

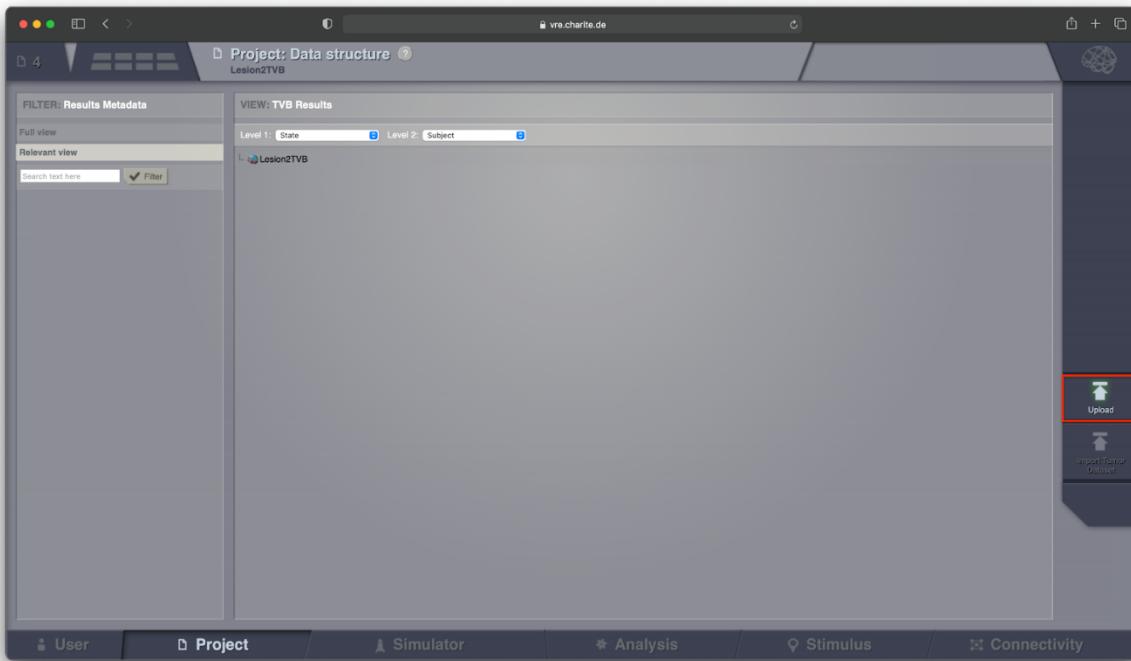
After this action, the Lesion2TVB project remains selected as the current working project in the TVB GUI. The user is brought back to this page, where the “Data structure” could be chosen as a next



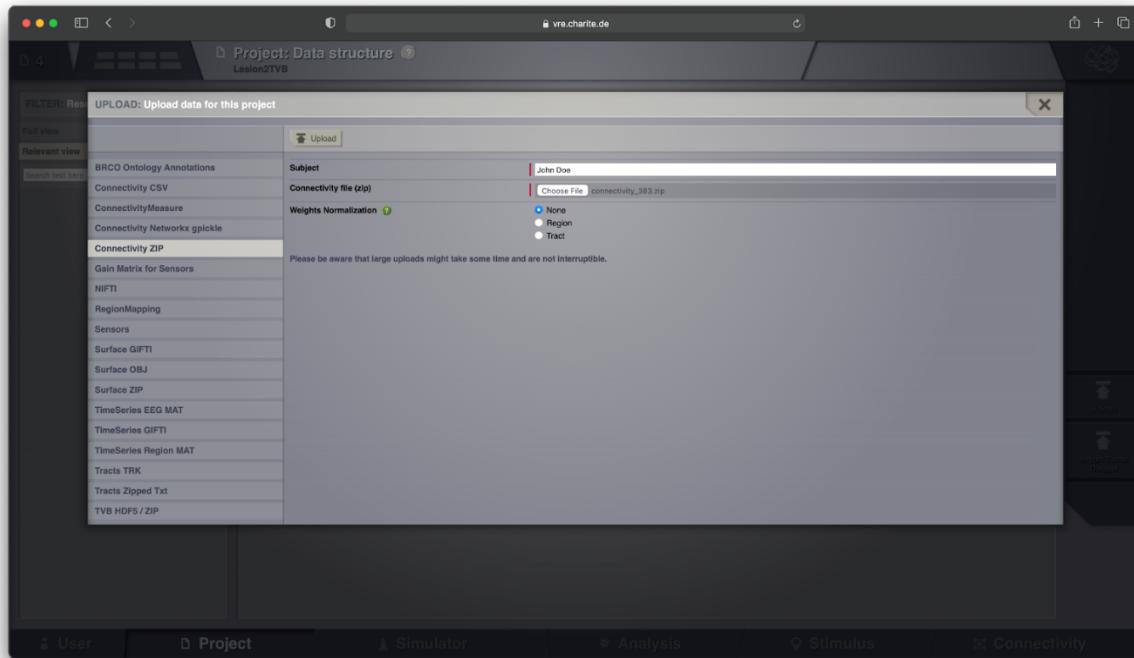
step, to inspect the data in the Lesion2TVB project.



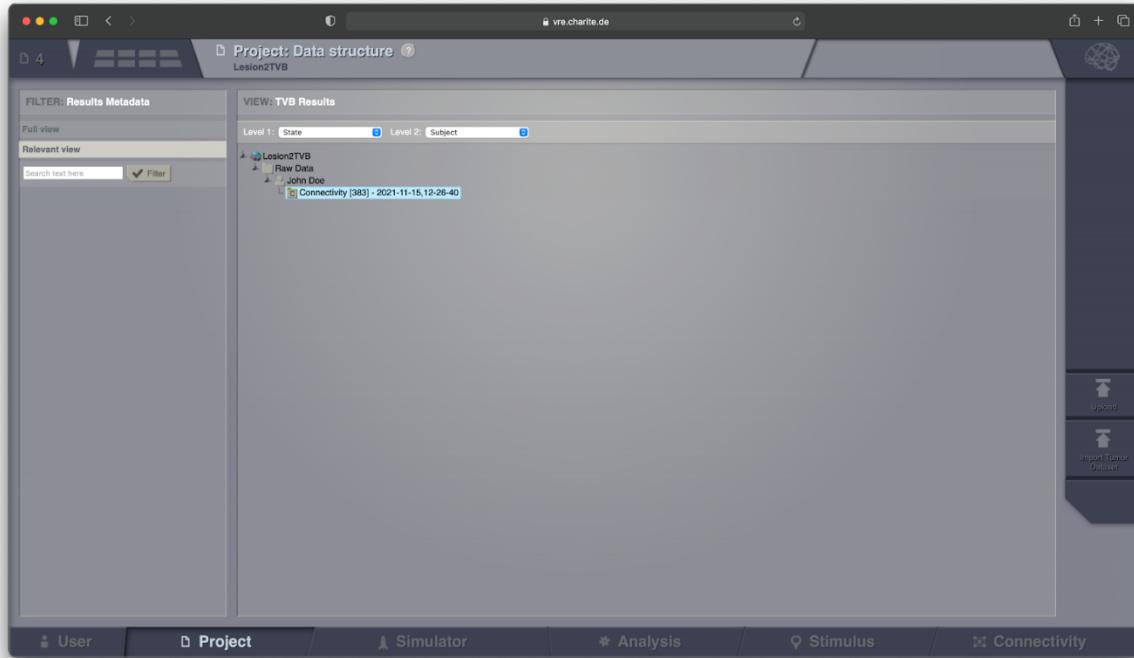
On the “Data structure” page, we can see that there is no data in the current project:



By clicking the “Upload” button, highlighted in red, an overlay pops-up and allows the user to upload data into the current Lesion2TVB project:



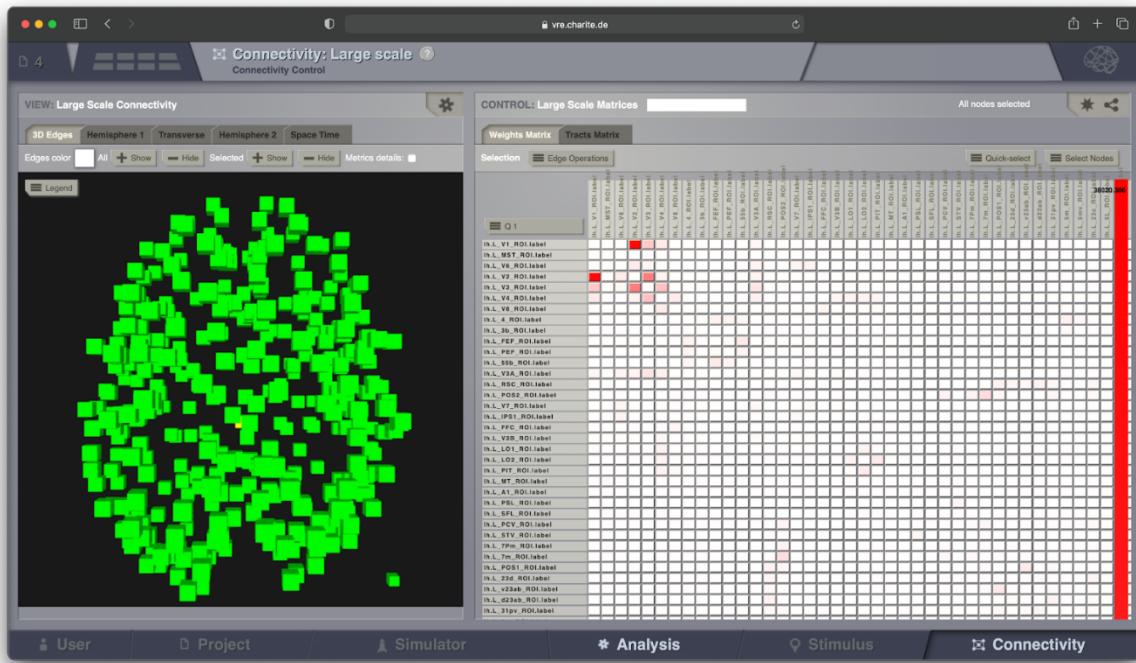
Here, the user chooses to upload the connectome that was previously downloaded from the VRE Core storage. After the upload is complete, the user is redirected to the “Data structure” page, where the Connectivity datatype is now available:



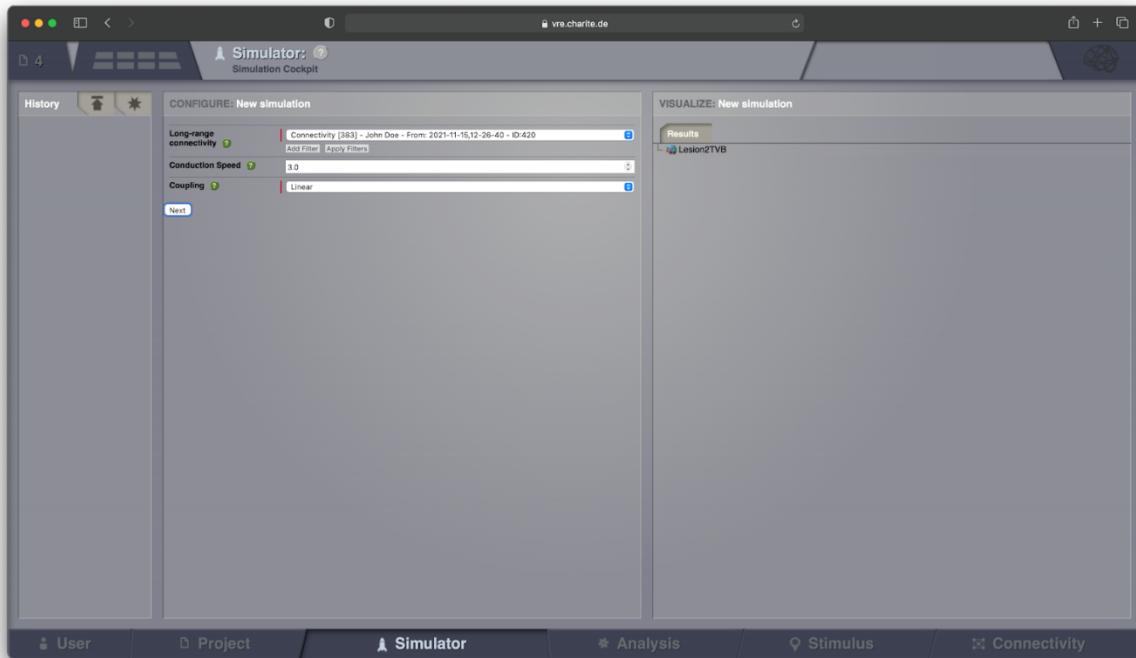
By clicking on the Connectivity, an overlay pops up and here the user can choose a visualizer for this datatype:



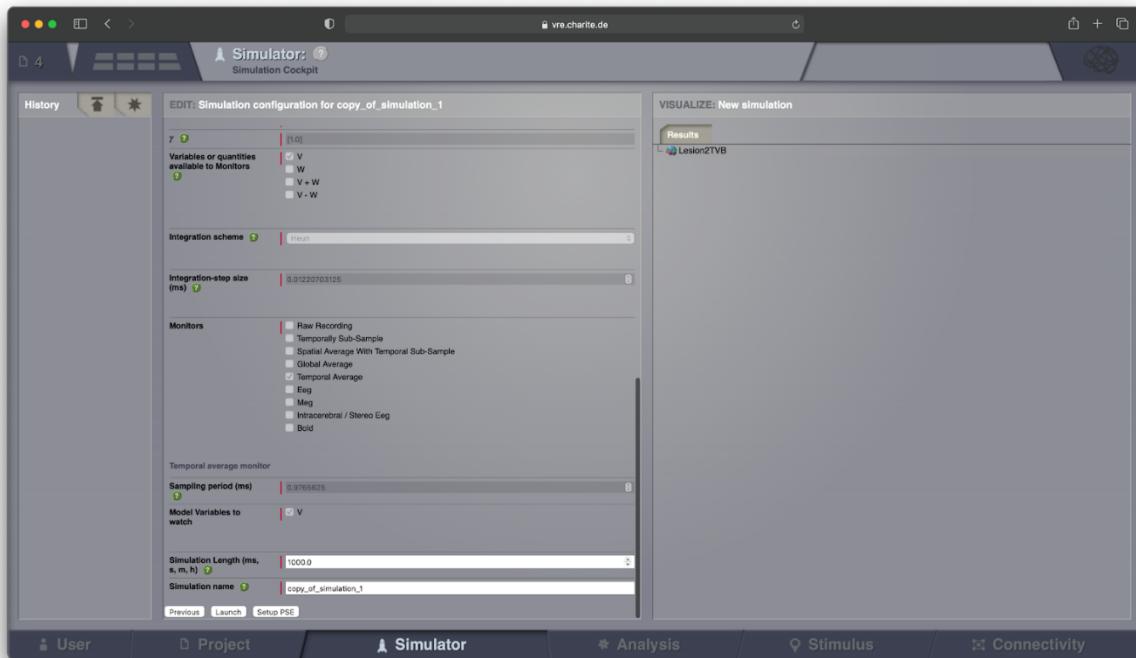
And within the visualizer, the user can inspect or manipulate the current Connectivity in different ways:



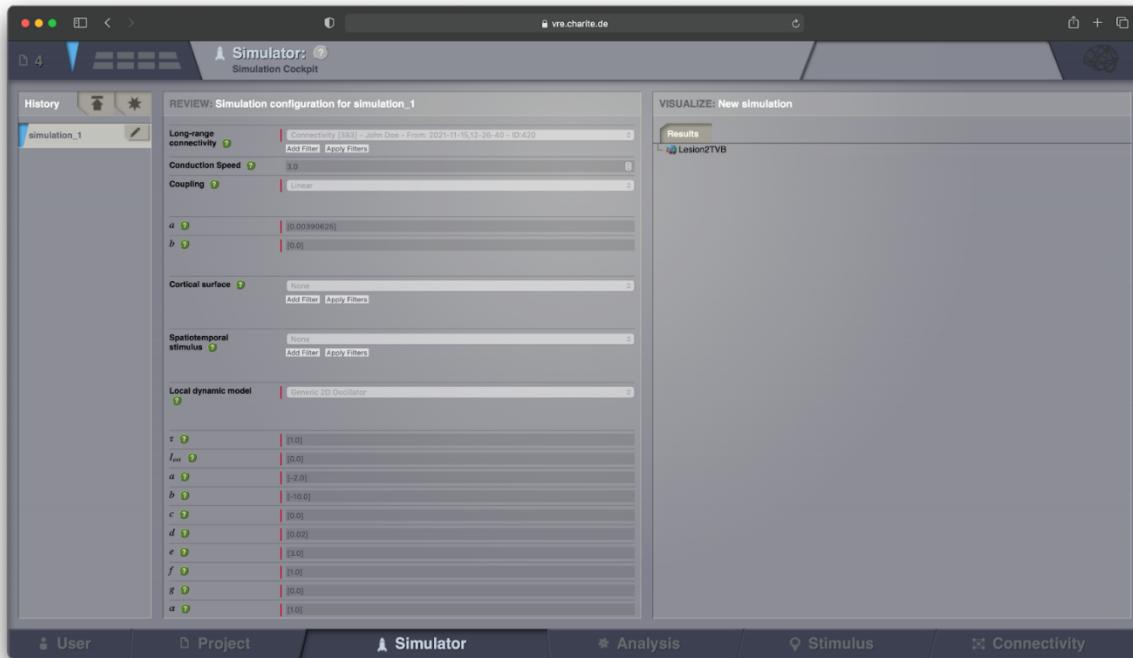
As a next step, the user can perform simulations based on this Connectivity, by choosing the "Simulator" from the bottom menu navigator:



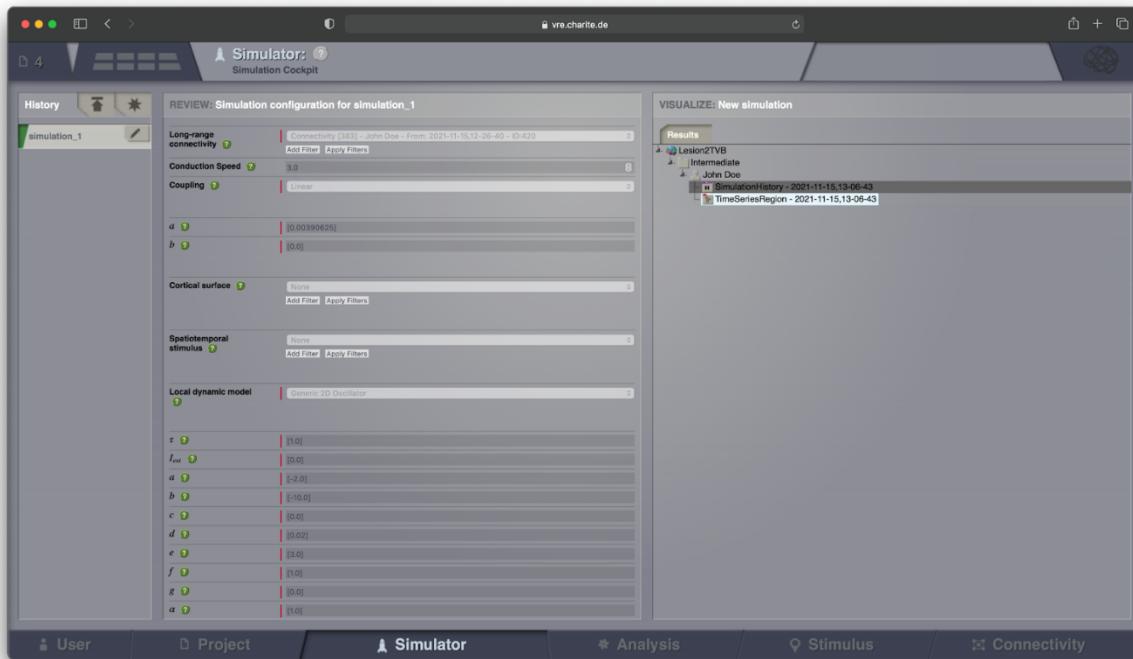
The user can configure simulation settings by moving with the “Next” buttons throughout this wizard and, as a final step, can start the simulation by clicking “Launch”. A parameter space exploration can also be set up by choosing the “Setup PSE” button.



The user can then see the progress of the simulation on the left-side, with blue indicating that the simulation has started:



Once the simulation is finished, the status indicator on the left-side turns green and, on the right-side, the results of the simulation are displayed:



Similar actions can be performed for a variety of data types that result from pipeline processing.

The user can also work with these projects in the BrainX3 interface. The integration between TVB and BrainX3 is implemented via a REST API. TVB offers a REST client SDK to be incorporated in BrainX3 for easier access to the TVB REST endpoints and Keycloak authentication.

In order to view the TVB REST server documentation, the user can open the following URL in a browser:



(<https://vre.charite.de/tvb-rest/doc>)



As a result, the page below is shown and the user can inspect the available endpoints.

The screenshot shows the 'Rest services for TVB' documentation page. It lists several API endpoints under different categories:

- api/users**: TVB-REST APIs for users management.
- api/projects**: TVB-REST APIs for projects management.
 - GET /api/projects/{project_gid}/data**: return a list of DataType instances (subclasses) associated with the current project.
 - PUT /api/projects/{project_gid}/members**: Add members to the given project.
 - GET /api/projects/{project_gid}/operations**: return a list of project's Operation entities.
- api/datatypes**: TVB-REST APIs for datatypes management.
- api/operations**: TVB-REST APIs for operations management.
 - GET /api/operations/{operation_gid}/results**: return list of DataType instances (subclasses), representing the results of that operation if it has finished and
 - GET /api/operations/{operation_gid}/status**: return status of an operation
 - POST /api/operations/{project_gid}/algorithm/{algorithm_module}/{algorithm_classname}**: generic method of launching Analyzers
- api/simulation**: TVB-REST APIs for simulation management.
 - POST /api/simulation/{project_gid}**: start a simulation using a project id and a zip archive with the simulator data serialized

4.3.3. Jupyter Notebooks

Within the VRE portal, users can also access the JupyterHub service, in order to access TVB via its command line interface.

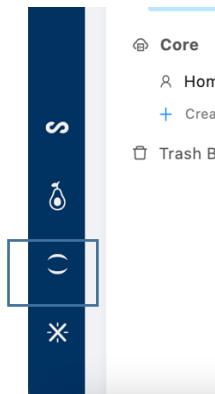


Figure 28: Icon to launch a Jupyter Notebook

It is possible for users to create their own Jupyter kernels that are persisted in the VRE for later usage. This way, the user has full control of upgrading/adding libraries in the long term. Libraries installed using the 'pip' Python package installer in a Python virtual environment (venv) can persist in JupyterHub and can be accessed in future sessions without requiring re-installation. Following the instructions provided on VRE User Manual wiki²⁸, the user can carry out all the steps needed to create a virtual environment on their JupyterHub workbench.

²⁸ https://vre.charite.de/xwiki/wiki/vrepublic/view/Main/user_guide/Analyzing%20Data/JupyterHub/



In the below image, the user has already created a “tvb-env” kernel where TVB is already installed. Thus, the next step is to create a Jupyter notebook that uses that kernel.

The screenshot shows the JupyterHub interface. At the top, there's a navigation bar with 'jupyterhub' and links for 'Logout' and 'Control Panel'. Below it, a 'Clusters' tab is selected. A sidebar on the left shows a file tree with a folder named 'tvb-env'. On the right, a dropdown menu is open under 'New' with options: 'Notebook' (selected), 'Python 3', 'tvb-env', 'Text File', 'Folder', and 'Terminal'. A tooltip says 'Create a new notebook with tvb-env'.

The user can check whether TVB works as expected, by doing a simple import:

The screenshot shows a Jupyter Notebook interface. The title bar says 'jupyterhub Untitled Last Checkpoint: a few seconds ago (unsaved changes)'. The toolbar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A status bar at the bottom right shows 'Trusted' and 'tvb-env'. In the main area, there are two code cells. The first cell, 'In [1]:', contains the code 'import tvb'. The second cell, 'In [2]:', is currently active and empty.

The VRE CLI tool can be used to bring the TVB-ready data from VRE Core storage into the current user folder of JupyterHub. Then, the user can start configuring and running simulations:

The screenshot shows a Jupyter Notebook interface titled 'jupyterhub TVB_Simulation Last Checkpoint: 2 minutes ago (unsaved changes)'. The toolbar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A status bar at the bottom right shows 'Trusted' and 'tvb-env'. The notebook contains several code cells. The first cell, 'In [14]:', imports 'tvb_simulator.lab' and 'numpy'. The second cell, 'In [12]:', defines a 'connectivity' object and loads data from files like 'centers.txt', 'weights.txt', and 'tract_lengths.txt'. The third cell, 'In [13]:', performs some calculations with arrays 'rw', 'S', 'C', and 'ds'. The fourth cell, 'In [5]:', creates a 'Simulator' object with parameters like 'model', 'coupling', 'integrator', 'monitors', and 'simulation_length'. The fifth cell, 'In [6]:', runs the simulation. The sixth cell, 'In [9]:', prints the shape of the resulting data array. The output 'Out[9]:' shows '(5000, 1, 383, 1)'.

The TVB demo videos website (<https://docs.thevirtualbrain.org/demos/Demos.html>) provides further details on how to run JupyterHub-based TVB simulations.



4.3.4. Data pipelines using graphical user interfaces (BrainX3)

The first step is to launch the Guacamole clientless remote gateway from the VRE through the icons on the bottom left. If it is the first time the user should request the access.

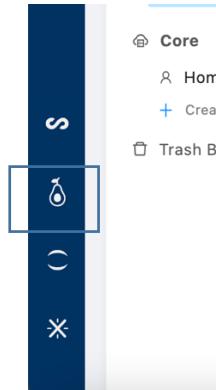


Figure 29: Icon to launch the Guacamole workbench.

In the next step, the module shows the available VMs that can be launched. For analysis using software that requires a graphical user interface, the user will launch a Core Desktop machine.

ALL CONNECTIONS

- Core-CommandLine-205
- Core-Desktop-205
- Greenroom-CommandLine-204
- Greenroom-Desktop-204
- TVB

Figure 30: View of the available Virtual Machines accessible through the Apache Guacamole clientless remote desktop gateway

The desktop of the remote machine is visualized in the browser and the user can run the BrainX3 software. The software allows to load data available in the VRE (Subject-0000 uploaded in the initial steps) and can inspect it. The user can use tools to define a region of interest (e.g., a lesion in the brain).

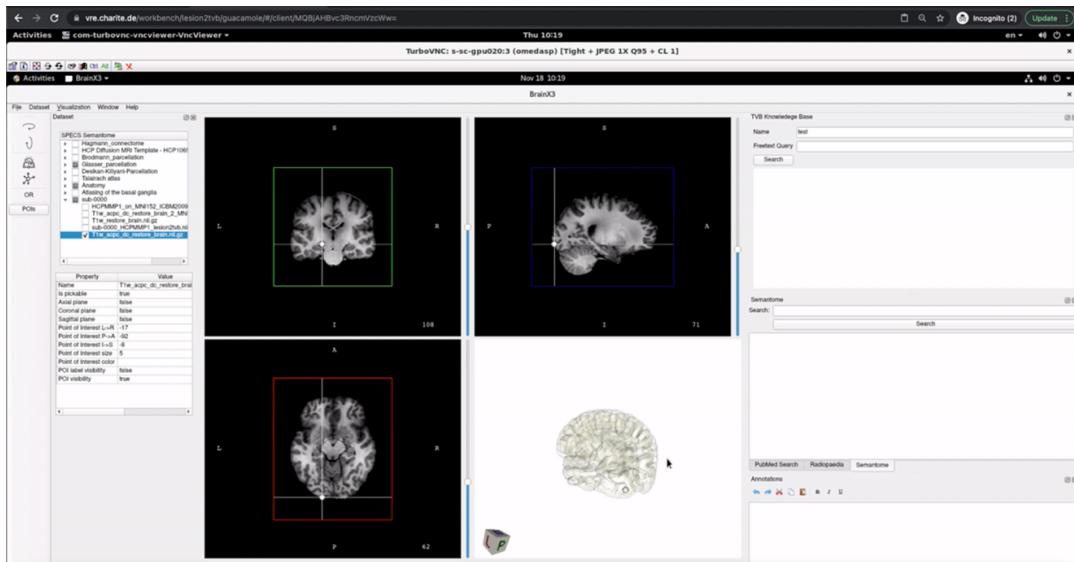


Figure 31: MRI visualization of data shared in the VRE



In the next steps, the user can explore the subject data interactively and load several generic and brain atlases. Using the TVB Knowledge Base feature integrated in the software, the user can perform a search using terms. The result is a mapping of molecular signalling information retrieved from the literature-mining knowledge base of SCAIView to a 3D brain space using the Glasser atlas, allowing integration of biological information in multi-scale whole brain models.

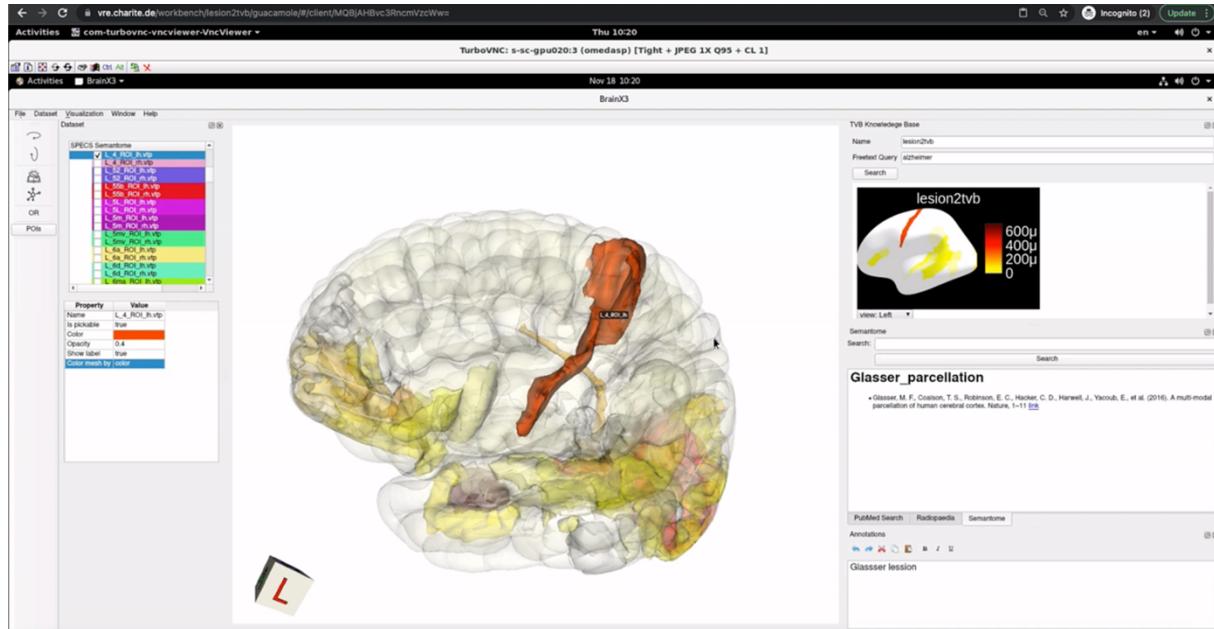


Figure 32: Visualization of the results of TVB Base

The user can connect to the TVB server from BrainX3 and launch operations and simulations that runs on the VRE VMs and HPC resources.

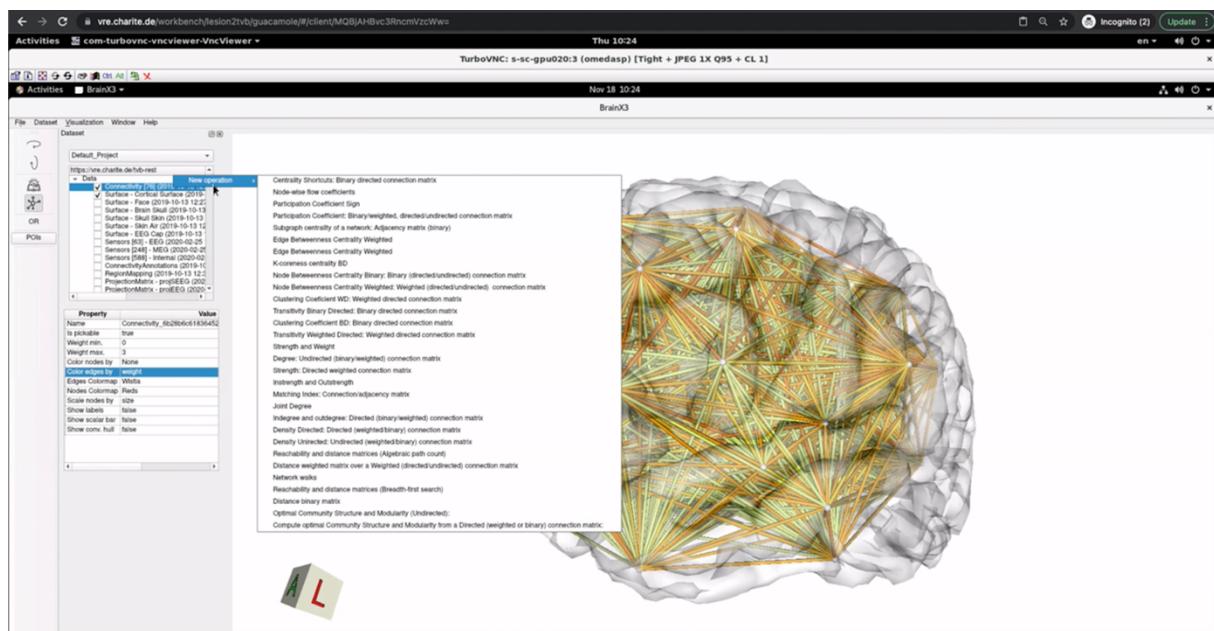


Figure 33: Interface to launch operations in TVB server



Finally, the results are retrieved and can be visualized back in BrainX3.

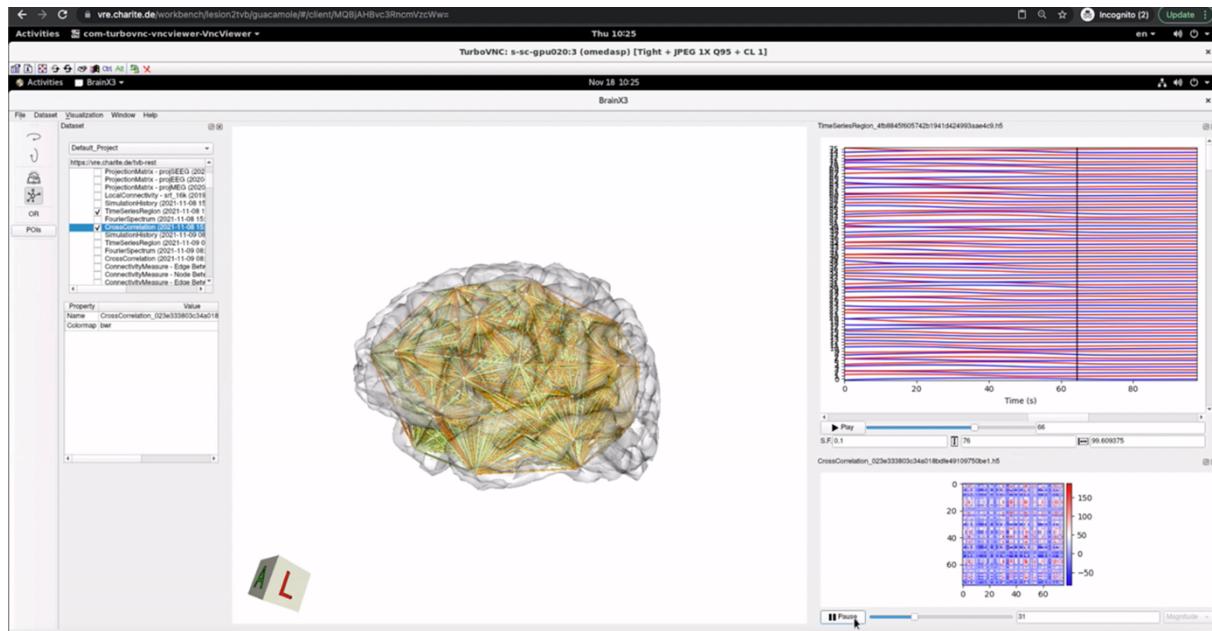


Figure 34: Visualization of analysis and simulations performed with TVB

4.3.5. Blue Brain Nexus

All information in the Blue Brain Nexus (BBN) system is represented as resources, either the information of system configuration or user data. User data is represented as sub-resources to BBN Projects, which in turn are sub-resources of BBN Organizations. BBN Organization and Project resources provide a logical grouping and isolation allowing for variation in configuration and access control policies, which also implies that BBN Organization and Project identifiers are part of a resource addressing scheme.

So, as the first step of BBN resource modelling, a dedicated BBN Organization (Charite) and Project (Lesion2TVB) were created in BBN to host all the Dataset annotations (in JSON-LD format) imported as resources. Within the VRE Project Lesion2TVB, the Dataset annotations were subsequently copied from the VRE Green Room to VRE Core storage and ingested into BBN using the VRE CLI from a workbench VM.

The best way to directly interact with the BBN KG is through the exposed REST API. The VRE CLI was developed to wrap the REST API functionalities and make the upload of many resources such as a fully modelled openMINDS dataset easier (Figure 35). Another common way to interact with the KG is BBN Fusion, a web app that was integrated with the VRE's Single Sign On authentication mechanism.

Since BBN was deployed with VRE Keycloak federation, all VRE users can log into BBN Fusion with their Charité Active Directory credentials. After login, either using the VRE CLI or the BBN Fusion web interface, authenticated users can create, edit, view and query BBN resources. All user permissions are configured through the BBN Access Control List APIs.



After the dataset annotations are imported into the BBN KG through the VRE CLI, Lesion2TVB Project members can access BBN Fusion to review the imported resources under ‘charite’ Organization and Lesion2TVB project. Resources can be visualized as a graph displaying relationships with other resources as defined by openMINDS annotations.

BBN Fusion also provides an aggregated Search UI, powered by the Elasticsearch component of BBN. To provide VRE users with access to the BBN Fusion Search UI, the Elastic Search View resource was created to define the resources (all imported dataset annotations) that can be queried in the BBN Project (charite/Lesion2TVB), as well as a SearchConfig resource which enables the Fusion Search UI and refers to the Elastic Search View resource. As a result, Lesion2TVB Project members can apply filters or perform free text searches using the Fusion Search UI on any field or term of interest. For instance, searching for the term “stroke” will return all resources containing “stroke” as part of JSON-LD attributes.

The following screenshots illustrate BBN Fusion integrated within the VRE, and capabilities to manage, query and visualize the data in the KG (Figure 36Figure 35 - Figure 40). The backend of BBN manages Elasticsearch and Blazegraph indices that can be used respectively for text-based and graph-based search queries.

```
$ vrecli kg_resource import ./stroke_schema_mockups
File imported:
stroke_schema_mockups/admin/stroke_schema_mockups_1112/dataset/dataset.json,
stroke_schema_mockups/admin/stroke_schema_mockups_1112/datasetType/e4395e8a-22d4-11ec-a97f-0242ac110002.json,
stroke_schema_mockups/admin/stroke_schema_mockups_1112/datasetVersion/datasetVersion.json.json
$
```

Figure 35: Import Dataset annotations modeled by openMINDS into BBN by VRE Command Line Tool.

The screenshot shows the BBN Fusion interface within the VRE. The top navigation bar includes 'Search or View', 'Clear', and 'Copy token'. The main content area is titled 'charite | Lesion2TVB'. On the left, there's a sidebar with 'Admin' selected, followed by 'Studios', 'Workflow', and 'Search'. The main panel shows a 'Default Query' with 11 results, filtered by type ('Dataset'). The results list includes several entries: 'e4395e8a-22d4-11ec-a97f-0242ac110002/026testing1112/testing_DatasetType', 'e4395e8a-22d4-11ec-a97f-0242ac110002/026testing1112/testing_DatasetVersion', and 'e4395e8a-22d4-11ec-a97f-0242ac110002/0191026testing1112/testing_Dataset'. Below the results, there are sections for 'testresource', 'Dataset', 'context.org', 'ElasticSearchSetup', 'BasicSearchView', and 'View'. The right sidebar contains 'Project Tools' with options for 'Create Resource', 'Search Query Editor', 'Elasticsearch Query Editor', and 'View Project's permissions'. It also features a file upload area with instructions to 'Click or drag file to this area to upload' and a dropdown for 'Single or bulk upload'.

Figure 36: BBN Fusion showing an overview of uploaded JSON-LD resources inside of Charite/Lesion2TVB (Organization/Project). The left column offers a simple search interface to filter all uploaded resources by their type or keywords. The right column offers functionalities to create new resources or Views using Elastic Search DSL or SPARQL.



The screenshot shows the BBN Fusion interface for the 'charite | Lesion2TVB' project. A specific DatasetVersion resource is selected, showing its JSON-LD representation. The JSON content includes fields like '@context', '@vocab', and various '@id' links to other resources such as 'funding', 'license', 'author', and 'custodian'. The interface also displays navigation buttons and a sidebar with a list of other resources.

Figure 37: Clicking on a resource opens a detailed description of the JSON-LD resource. This figure shows attributes of a DatasetVersion resource, that has outgoing links (relations) to many other resources using the "@id" attribute.

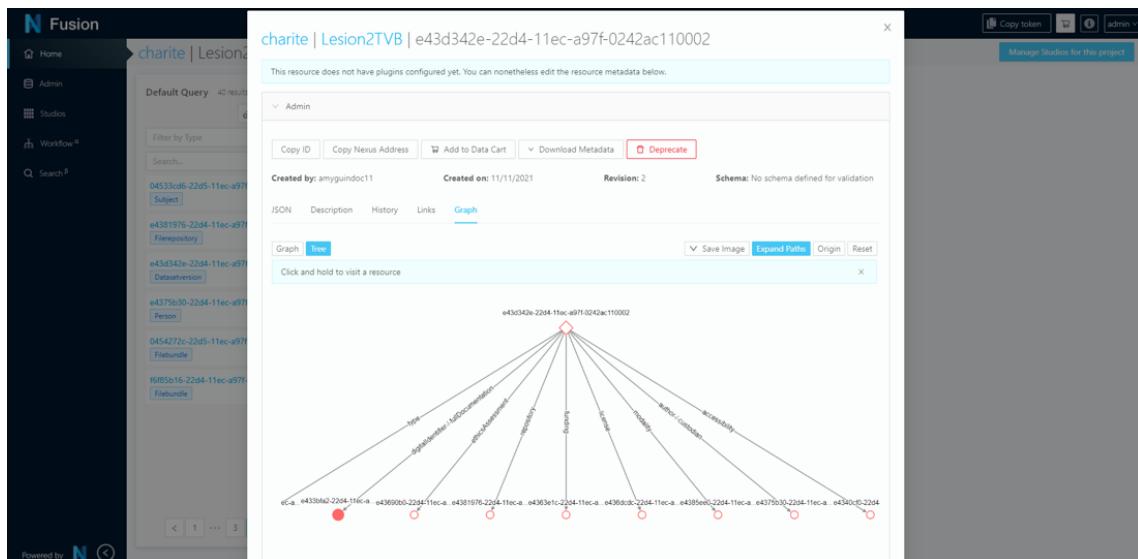


Figure 38: The Graph view in BBN Fusion shows each resource (entity) visualized with its relations (edges) to other resources. This figure shows the DatasetVersion resource that has relations such as funding, license, author, etc. and its links to the resources.



The screenshot shows the BBN Fusion search interface. On the left, there's a sidebar with a navigation menu and a 'Search' input field. Below it is a 'Filter' section with a dropdown for 'PROJECTS' (selected) and a dropdown for 'TYPES'. The main content area displays a grid of 40 resources. Each resource card includes a thumbnail, the resource ID, its type (e.g., 'Fileinstance', 'Organization'), and a brief description. At the bottom right of the grid, there are page navigation buttons for '1' and '2'.

Figure 39: BBN Fusion Search interface. Resources can be filtered by their type (left column) and by use of the free text search (top search field). In this figure no filters are applied and all available resources are displayed in the search results.

This screenshot shows the BBN Fusion search interface after performing a search for the keyword 'stroke'. The search bar at the top now contains 'stroke'. The main content area displays a grid of 3 resources. Each resource card includes a thumbnail, the resource ID, its type (e.g., 'Subjectgroupstate', 'Dataset'), and a brief description. The results are labeled as 'Showing 3 of 3 Resources'.

Figure 40: BBN Fusion showing all matching resources for a successful free text search with the keyword "stroke". Only resources that contain the string "stroke" in one of their attributes are displayed in the search results.

4.4. Interventions based on the Rehabilitation Gaming System (RGS)

The interventions module of TVB-Cloud intends to close the loop between the pipelines for personalized analysis, modeling, simulation, and prognosis of patients with Alzheimer's disease and use this information to plan clinical treatments. We are exploring this using the Rehabilitation Gaming System (RGS), a solution for personalized treatment which combines theoretical neuroscience grounded, Virtual Reality based, and AI intervention protocols. RGS delivers immersive embodied and goal-oriented training for multiple domains, i.e., motor, affective, cognitive, through virtual reality delivery and optimizes rehabilitation by personalizing the therapy through artificial intelligence algorithms. As a result, the treatment of the patients is individualized and can be planned and adjusted based on the diagnostics and performance.



Eodyne, as the developer of the RGS, started a collaboration with Alzheimer Europe (AE) to evaluate the use of the RGS solutions with the target patients of TVB-Cloud. A focus group was established to present the RGS applications, decide the ones that are more suitable for the group and plan a test with the users to collect their feedback. The focus group consisted of members of the European Working Group of People with Dementia (EWGPWD), which was launched by AE and its member organisations (national Alzheimer's associations) in 2012. The EWGPWD is entirely composed of people with dementia, supported (where required) by a family member, friend, or representative from the national association. Currently, the EWGPWD includes 14 people with dementia from 10 different countries in Europe, with different types and at different stages of dementia.

Focus Group – Consultation

The first EWGPWD consultation, in December 2020, included one participant from Eodyne doing the presentation, three coordinators from AE (two of them acting as moderators), and 14 participants with different types and at different stages of dementia. This consultation served as an introduction of Eodyne and the different possible platforms for cognitive training to the group to get a first impression of potential user acceptance. Eodyne provided a non-technical introduction of the three platforms: 1) existing PC-based platform, used in a clinic setting (RGSClinic, RGSHome); 2) smartphone/tablet apps (RGSAp); 3) Virtual Reality/VR Head-Mounted Display (RGSVR). The presentations were mostly graphical, avoiding the excessive use of text so participants would get a good idea of how applications would look and feel like. Approximately 15 minutes of discussion were dedicated to each platform. Each presentation focused on the pros and cons of the different applications in three domains of interest: cognitive training, physical activity, and social interactions.

The response was overall very positive, with most participants willing to try one or more platforms. Among the feedback and comments from the working group, we highlight the following:

- Although some participants seemed interested in the HMD platform, others felt it could raise some safety issues, such as dizziness or falls. In a previous experience with VR, one participant had fallen from the chair.
- Half of the participants had used games or apps for dementia or memory.
- Some participants expressed that the objects in VR should look as realistic as possible for them to be able to recognize them, as opposed to having a cartoon-like aspect.
- When shown examples of virtual rehabilitation protocols based on Activities of Daily Living (ADLs) developed by Eodyne for stroke patients, some participants with more advanced dementia expressed that the tasks would be too complex for them.

After the first consultation, the coordinators from Eodyne and AE proposed to test the smartphone/tablet RGSAp solution based on the feedback from EWGPWD regarding the perceived safety issues with the HMD solution, the fact that half of the group members had previous experience with cognitive training apps, and that the distribution of dedicated equipment (e.g., motion tracking cameras) was not required as the group members or their relatives already had smartphones or tablets.

RGSAp

RGSAp Rehabilitation Gaming System version is a mobile application that focuses on the training and rehabilitation of motor, affective, and cognitive areas. In addition, it allows patients with reduced mobility to train their upper extremities without gravity by sliding the phone on the surface of a table, taking the first steps to recover the flexo-extension function of the shoulders and elbows and facilitate first-class physical exercise regimes. RGSAp includes an algorithm for recognizing patient gestures from the information provided by the accelerometer and gyroscope of the smartphone. It includes virtual rehabilitation programs adapted to mobile phones that are previously prescribed.



The RGSapp is installed on the mobile phone or tablet of the patients. They can open the app and login with the credentials provided by the health professional. Inside the app, there are main sections: The Home screen, the History and the Profile that shows the goals and tasks of the day are listed as prescribed by the professional and their progression (Figure 41). The home screen shows the goals and tasks of the day listed as prescribed by the professional. As the patient completes the different tasks RGSapp will offer feedback on the progress of the daily goals. These tasks can include rehabilitation exercises, questionnaires, and measurements. All the information is collected by a backend in the cloud, accessible to the therapist to follow up remotely the performance.

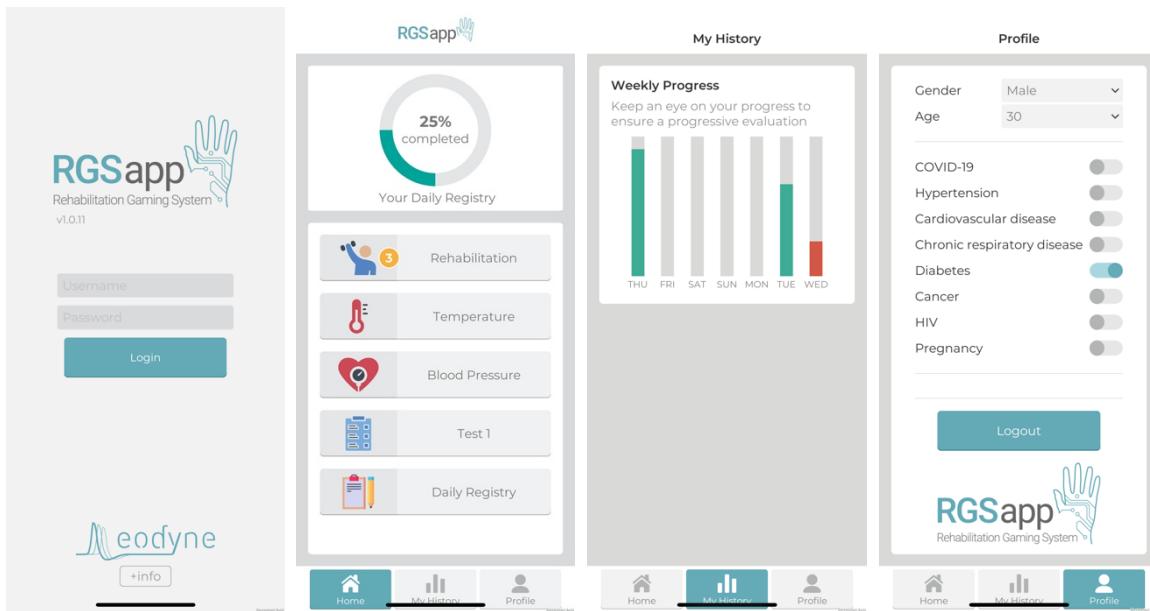


Figure 41: Sections of the RGSApp

RGSapp presents a variety of rehabilitation exercises that train a wide range of motor and cognitive functions. It provides protocols that require that the patient makes movements of different amplitudes and trains cognitive functions and motor inhibition (

Figure 42).

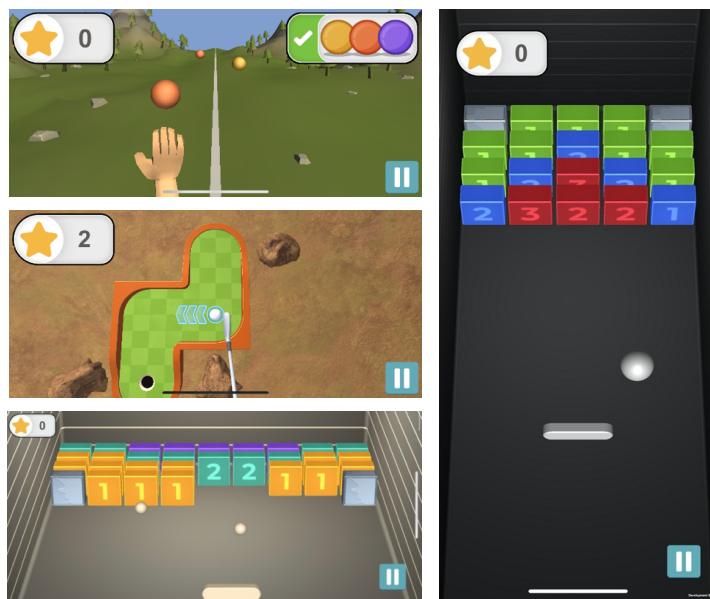


Figure 42: The RGSApp protocols: Spheroids (left-top), Minigolf (left-middle), Armanoid (left-down), and Knock a Bloc (right) protocols.

Focus Group – Testing

As preparation for testing with the EWGPWD, Eodyne and AE made a test run, where the AE team followed the testing script to assess the clarity and suitability of the instructions and tasks. A few adaptations were made after the test run, such as adapting the app to tablet format and adding a timer to the different activities. Moreover, the app was translated to Finnish, Brazilian Portuguese, and Swedish, and a series of documents were created, including a manual to download the app, a set of questions to be answered by participants daily while using the RGSapp for a week, and a list of questions for the testing day. Where necessary, AE provided additional support to the EWGPWD members via WhatsApp or zoom calls.

The test consisted of three parts. First, an initial meeting for app installation and walkthrough on April 29, 2021. This session was split in two groups according to their English level (native speakers and non-native speakers). During these sessions, Eodyne guided the installation and testing of the solution while the AE team acted as moderators. A total of nine participants and four of their supporters with different types of dementia tested the RGSapp solution. This was followed by a one week testing period, in which participants were asked to use RGSapp daily, when possible, and to fill in a daily questionnaire (see below for an example of the daily questionnaire). Finally, a feedback session was held on May 7, 2021, where participants shared their experiences with RGSapp.



TABLE 1 DAILY RECORDING

Day	Games	Did you play this game for 3 minutes today?		If NO, what was the main reason?	How easy did you find the game?	How did you feel about the game?	Overall, after playing the game today, you felt:
		Yes	No	ONLY IF "NO"			
Minigolf	X				DIFFICULT	FRUSTRATING	Mainly negative
	Spheroids	X			DIFFICULT	FRUSTRATING	Mainly negative
	Pinch	X			OK	STIMULATING	Mainly positive
	Armanoid	X			DIFFICULT	FRUSTRATING	Mainly negative
	Knock a block	X			OK	FUN	Mainly positive
	Fit in	X			OK	STIMULATING	Mainly positive

Figure 43: Daily questionnaire for EWGPWD members - exemplar

The results of the daily questionnaires are shown in the following plots:

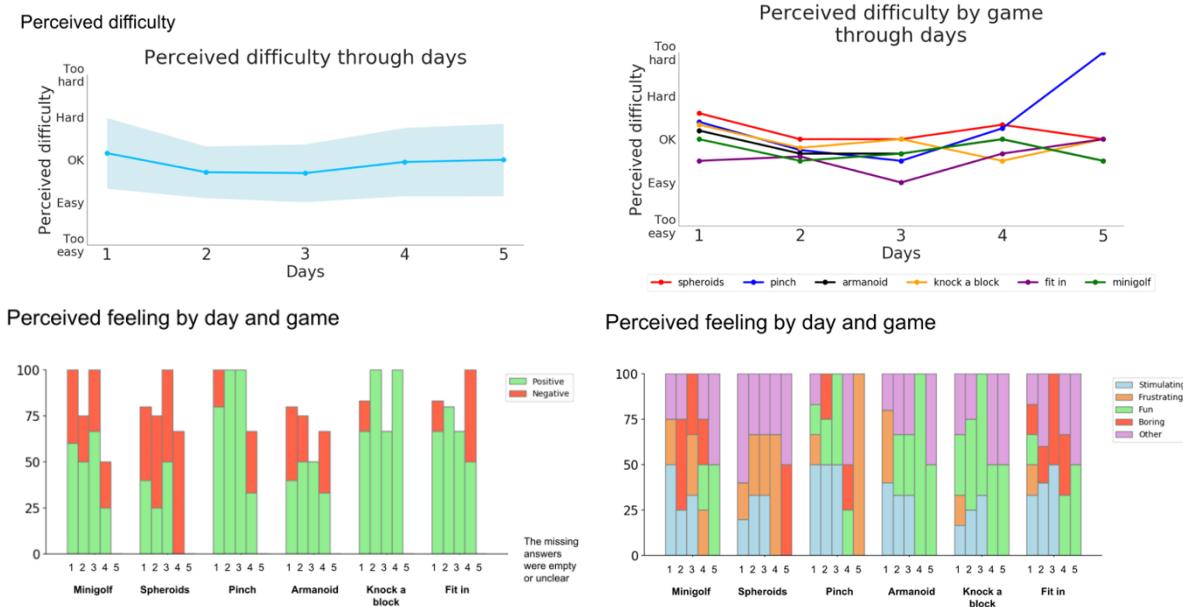


Figure 44: Results from the daily questionnaires filled in by participants using RGSapp for one week.

As it can be seen in the upper plots, the general and individual perceived difficulty of the activities seemed appropriate throughout the testing week, without being too easy or too hard with the exception the *Pinch* activity in day five. Moreover, there is a clear difference between the perceived feelings for the *Minigolf*, *Spheroids*, and *Armanoid* activities, and the rest of the activities. This might be explained by the fact that the former activities required users to move the phone to complete the activities, and the movement detection algorithm in RGSapp was found to malfunction with some of the smartphone models.

During the final feedback session, we collected unstructured feedback in the form of a focus group consultation, moderated by two AE staff members. The impressions were very heterogeneous, which might be partially explained by the diverse range of participants, who have different levels of comfort with smartphones, varying linguistic backgrounds, and different types of dementia. Among the feedback received, we highlight:



- Some participants could lose interest in activities that were very easy after a few days.
- Some participants felt that the easy activities were relaxing.
- In general, participants agreed that it is very important for the application to measure and show the progress made by the user quantitatively.
- The malfunction of the movement detection in some phones and bugs in general could make participants feel frustrated.
- For some participants, activities that were difficult to understand at first, could become motivating after learning how to use them.
- Some participants referred to other commercially available game apps with fairly high levels of gamification.

In general, the reception of RGSapp by the EWGPWD group was positive. However, a few adjustments and improvements must be made to motivate and engage users with dementia. In the first place, it seems necessary to distinguish users according to the stage and type of dementia, and to tailor the complexity of the activities accordingly. This could be achieved by personalization algorithms that adapt the training according to the user's performance. Second, gamification seemed to play a crucial role for the EWGPWD group, as activities with higher levels of gamification were perceived as more fun than others, and participants would refer to other commercially available game apps not necessarily intended for cognitive training. Third, the RGSapp accelerometer-based movement detection algorithm did not work properly for all users, which is probably due to the heterogeneous quality of the accelerometers across the different phone brands and models (which included iPhone 6/7/8/12 as well as android-based phones such as Samsung Galaxy & Note, Xiaomi POCO). Therefore, if we wish to integrate physical activity in the cognitive training, we must explore additional approaches that guarantee optimum performance across devices. Eodyne has taken on board this feedback and is redesigning the RGSApp.

4.5. In silico research: Identifying mechanisms, augmenting diagnosis, testing interventions

TVB-Cloud enables researchers for clinical in silico research (Figure 45)

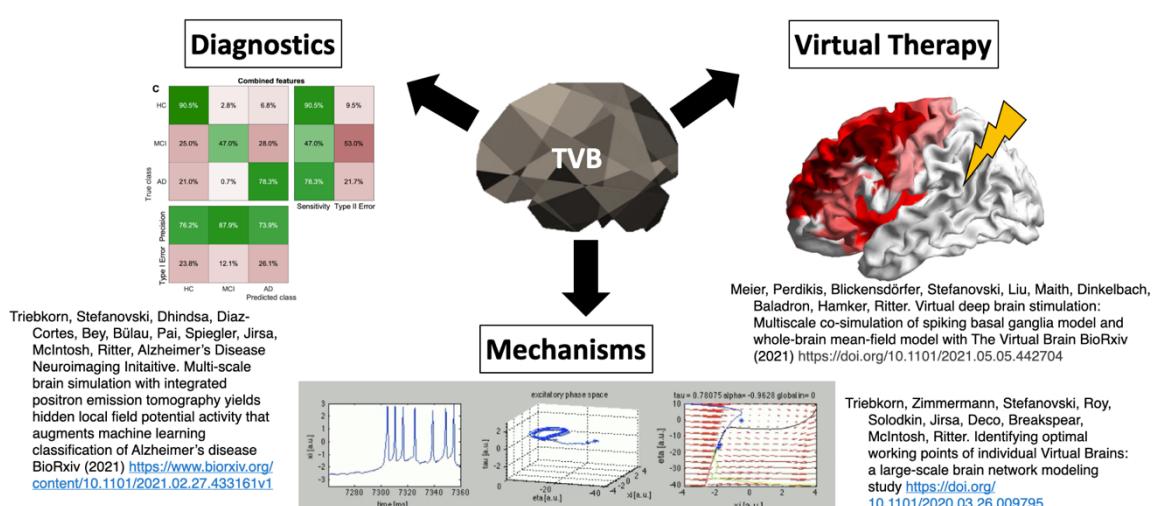


Figure 45: TVB-Cloud is an enabler for in silico research. Main clinical innovation potential lies in the domains of simulation augmented diagnostics and prediction, in silico intervention testing and inference of disease mechanisms.



In a recent study we demonstrated the potential of brian network simulations for drug testing (Stefanovski et al. 2019). Modeling the N-methyl-D-aspartate (NMDA) receptor antagonism of memantine in local population models, reveals potential functional reversibility of the observed large-scale alterations (reflected by EEG slowing) in virtual AD brains (

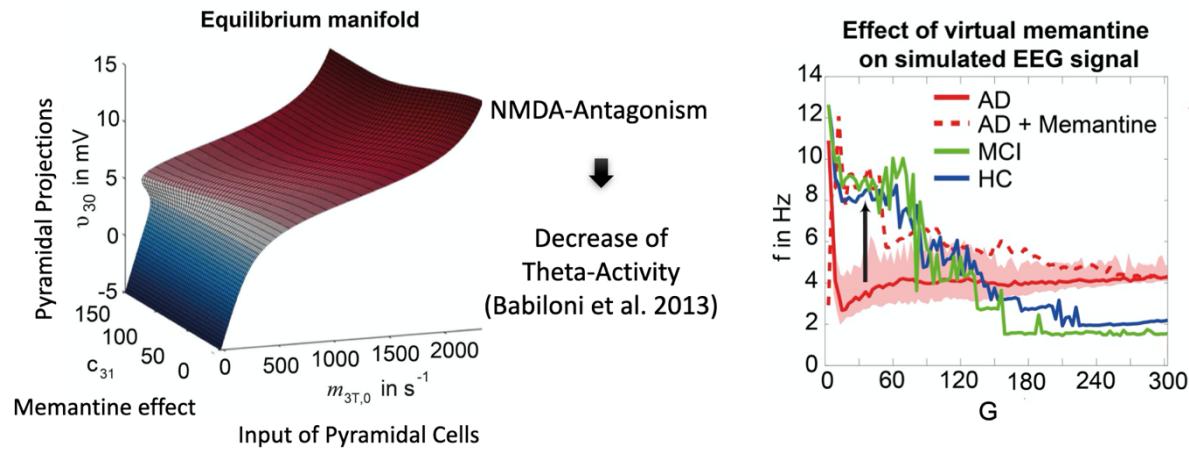


Figure 46)

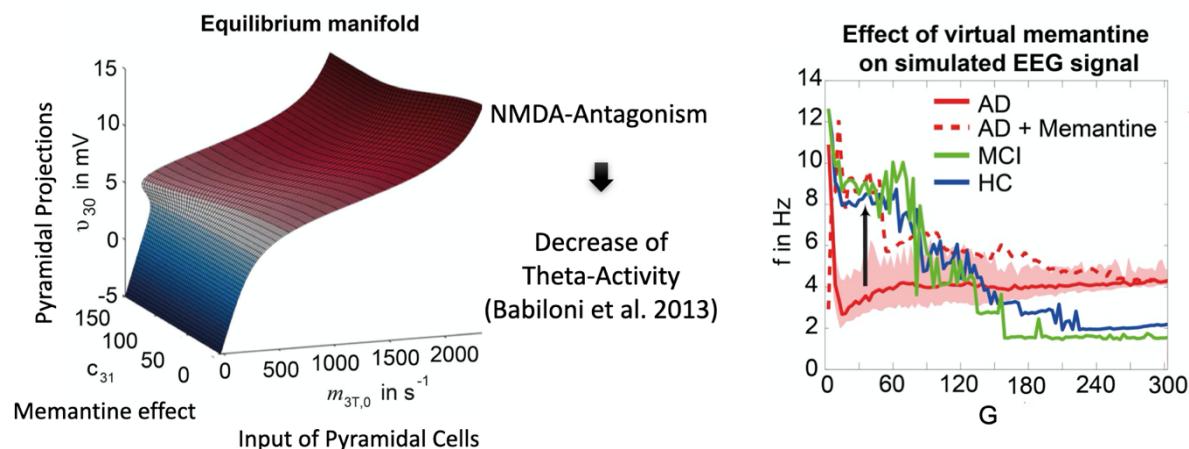


Figure 46: Virtual therapy with the NMDA antagonist memantine. Left: Demonstration of the dose dependent effect of memantine for generating a buffer that prevents from hyperexcitation and excitotoxicity with increasing excitatory drive due to disinhibition. Right: Normalization of the spectral EEG properties (reversal of EEG slowing) in AD virtual brains upon administration of virtual memantine. (Stefanovski et al. 2019)

Using our newly developed multiscale brain simulation framework (Schirner et al. 2021, Figure 1) we developed a Use Case on virtual deep brain stimulation (DBS) in Parkinson's Disease (Meier et al. 2021).

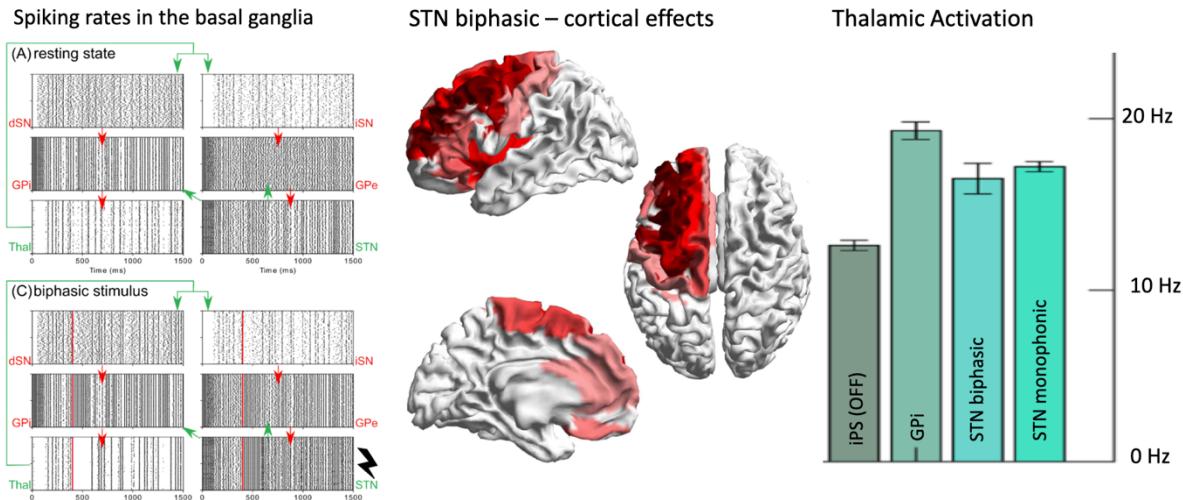


Figure 47: Demonstration of *in silico* testing of deep brain stimulation using TVB-multiscale Software (Meier et al. 2021).

The goal is to use *in silico* DBS in future for personalized planning of the optimal stimulation site and stimulation paradigm.

4.6. Technical coordination and integration

The work performed in WP6 has been a coordinated effort between the TVB-C consortium partners and not-for-profit SME partner Indoc Research who was subcontracted by Charité to take the role of overall technical coordination of the project. Since May 2020, more than 78 technical coordination meetings have taken place, bringing together TVB-C partners to discuss a variety of topics including the overall architecture of TVB-Cloud, the release management process, integration of TVB software tools, specific use cases and clinical applications, and legal and ethical topics relevant to the processing and sharing of TVB-Cloud data. Meetings are documented in a shared collaboration space, Confluence <https://conf.thevirtualbrain.org/display/TC>, with managed access to TVB-Cloud consortium members (Figure 48). Meeting agendas are determined in advance based on current priorities and general topics relevant to the project and are posted prior to each meeting in both Confluence and the TVB-C internal site accessible by TVB-C consortium members at <https://internal.virtualbraincloud-2020.eu> (Figure 48).



The screenshot shows a Confluence page titled "Technical Coordination". On the left, there's a sidebar with a navigation tree containing sections like "Meetings", "Technical Coordination", and "Meeting Schedule". The main content area displays a table of meeting schedules from 2021-12-17 to 2021-09-10, with topics ranging from "Data Governance Act, Artificial Intelligence Act and the EHDS – key elements and implications for the TVB-Cloud project" to "Part 6 of planning for Deliverable 6.4 - "A functionally integrated version of VBC is evaluated for therapeutic targets": defining updates to the BrainX3 integration in VRE". A sidebar on the right lists "Meeting Notes" with links to various meeting documents. At the bottom right is a small graphic of a brain scan.

Figure 48: The TVB-Cloud Technical Coordination web-based project management tool on Confluence, which contains documentation of all meetings and discussions, joint development processes and decisions, and web links and resources relevant to technical coordination.

The screenshot shows the TVB internal site with a "TC" space selected. The main header reads "Technical Coordination" and "Collaboration space for TVB-C Technical Coordination". Below the header, it shows "38 Posts", "101 Members", and "0 Followers". The left sidebar has a "Space menu" with options like Stream, Wiki, Files, Tasks, Gallery, Polls, and About. The main content area features a post by Sue Evans dated 14 Oct 2021, which announces a meeting on Friday 22-Oct-2021 at 1500 CEST. It includes a list of topics and a reminder about meeting frequency changes. There are "Like" and "Comment" buttons at the bottom of the post.

Figure 49: The TVB-Cloud Technical Coordination (TC) dedicated space in the TVB internal site <https://internal.virtualbraincloud-2020.eu> maintained by WP1 leaders tp21. The TC space is used to disseminate information on upcoming meetings and other announcements to all TVB-C consortium members.

We also have a Jira dedicated projects for managing these integration tasks:

- Cloud Integration tasks:

<https://req.thevirtualbrain.org/projects/VBC/issues/VBC-68?filter=allopenissues>



- Team administrative tasks:
<https://req.thevirtualbrain.org/issues/?jql=project%20%3D%20ADMN>

The successful completion of this deliverable can be attributed to frequent planning, communication and continual refinement of the roadmap and technical integration tasks. Since June 2021 six technical coordination meetings have been dedicated specifically to Deliverable 6.4 with additional in-depth meetings involving both the technical and legal/data protection teams to plan and realize the detailed logistics of the software integration and GDPR compliant processing. This effort also relied on the assistance and cooperation of the Charité GB-IT team which has been instrumental in providing the necessary hardware and system configurations critical to the integration in the Virtual Research Environment.

5. Conclusion, next steps

The TVB-Cloud is deployed within the GDPR-compliant VRE data platform, providing researchers with a seamless web interface to data and tools for multi-scale brain simulation. This integration provides the following functionality:

- Data upload and isolation: secure and reliable transfer of large, sensitive datasets to the protected storage resources of the VRE Green Room.
- Pre-processing of sensitive data: execution of containerized pipelines on sensitive datasets, e.g., for the purposes of generating defaced structural MRI images, and to inspect and validate the outputs of this processing before copying these outputs to the VRE Core zone.
- Execution of TVB pipelines: processing of raw data stored in the VRE Core by executing containerized pipelines to generate TVB-ready datasets.
- Data standardization and discoverability: annotation of datasets based on the openMINDS schema and ingestion of these metadata into the VRE Knowledge Graph, enabling researchers to query and find available datasets.
- Simulation analysis: execution of TVB simulation pipelines via Jupyter notebooks or via the TVB Web GUI.
- Exploration or simulation results: ability to visualize simulation results using the BrainX3 desktop application, running within the VRE workbench, and utilizing specialized HPC graphical hardware, and integrating brain region mapping of molecular pathways extracted from the SCAI View knowledge base and semantic search services.

This represents a significant advance in the capabilities of the TVB-Cloud, towards enabling researchers to perform complex end-to-end multi-scale analyses, from raw data through to simulation results. As the team continues to work towards a production release of TVB-Cloud in 2022, the following deployment and integration activities are planned:



- Data discoverability enhancements, including expansion of knowledge graph to ingest additional metadata based on openMINDS, BIDS standards, and TVB Ontology.
- Further support of DataLad, including the ability to generate and access DataLad datasets using the VRE Portal and the VRE Workbench.
- Incorporation of Clinical Data Viewer and terminology services.
- Deployment and integration of TVBase as a standalone brain region mapping service running within the VRE.
- Deployment and integration of SCAI View Knowledge Base and semantic search services within the VRE.

6. Glossary

API – Application programming interface

DBS -Deep brain stimulation

DWI – Diffusion Weighted Image

EEG – Electroencephalography

EWGPWD - European Working Group of People with Dementia

GB - Gigabytes

GDPR – General Data Protection Regulations

GUI – Graphical User Interface

HCP – Human Connectome Project

HPC – High performance computing

JSON-LD - JavaScript Object Notation for Linked Data

MEG - Magnetoencephalography

OpenMINDS – open Metadata Initiative for Neuroscience Data Structures

PVC - persistent volume claims

RDP – Remote desktop protocol

sEEG - Stereoelectroencephalography

TVB – The Virtual Brain

TVB-Cloud – The Virtual Brian Cloud

VM – Virtual Machines

VPN – Virtual Private Network

VRE - Virtual Research Environment



7. References

References Alfaro-Almagro, F., Jenkinson, M., Bangerter, N. K., Andersson, J. L. R., Griffanti, L., Douaud, G., Sotiropoulos, S. N., Jbabdi, S., Hernandez-Fernandez, M., Vallee, E., Vidaurre, D., Webster, M., McCarthy, P., Rorden, C., Daducci, A., Alexander, D. C., Zhang, H., Dragoni, I., Matthews, P. M., ... Smith, S. M. (2018). Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank. *NeuroImage*, 166, 400–424.

<https://doi.org/10.1016/J.NEUROIMAGE.2017.10.034>

Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., Albert, M. S., & Killiany, R. J. (2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, 31(3), 968–980. <https://doi.org/10.1016/J.NEUROIMAGE.2006.01.021>

Glasser, M. F., Coalson, T. S., Robinson, E. C., Hacker, C. D., Harwell, J., Yacoub, E., Ugurbil, K., Andersson, J., Beckmann, C. F., Jenkinson, M., Smith, S. M., & Van Essen, D. C. (2016). A multi-modal parcellation of human cerebral cortex. *Nature* 2016 536:7615, 536(7615), 171–178. <https://doi.org/10.1038/nature18933>

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., Van Essen, D. C., & Jenkinson, M. (2013). The minimal preprocessing pipelines for the Human Connectome Project. *NeuroImage*, 80, 105–124. <https://doi.org/10.1016/J.NEUROIMAGE.2013.04.127>

Meier, Perdikis, Blickensdörfer, Stefanovski, Liu, Maith, Dinkelbach, Baladron, Hamker, Ritter. Virtual deep brain stimulation: Multiscale co-simulation of spiking basal ganglia model and whole-brain mean-field model with The Virtual Brain BioRxiv (2021) <https://doi.org/10.1101/2021.05.05.442704>

Schirner, Domide, Perdikis, Triebkorn, Stefanovski, Pai, Prodan, Valean, Palmer, Langford, Blickensdörfer, van der Vlag, Diaz-Pier, Peyser, Woodman, Zehl, Fousek, Petkoski, Kusch, Hashemi, Marinazzo, Mangin, Flöel, Akintoye, Stahl, Deco, McIntosh, Hilgetag, Morgan, Schuller, Upton, McMurtrie, Dickscheid, Bjaalie, Amunts, Mersmann, Jirsa, Ritter. Brain simulation as a Service: The Virtual Brain on the European Research Platform EBRAINS (2021) ArXiv <https://arxiv.org/abs/2102.05888>



Stefanovski, Triebkorn, Spiegler, Diaz-Cortes, Solodkin, Jirsa, McIntosh, Ritter; for the Alzheimer's Disease Neuroimaging Initiative (2019). Linking molecular pathways and large-scale computational modeling to assess candidate disease mechanisms and pharmacodynamics in Alzheimer's disease. *Frontiers Computational Neuroscience*

Tournier, J. D., Smith, R., Raffelt, D., Tabbara, R., Dhollander, T., Pietsch, M., Christiaens, D., Jeurissen, B., Yeh, C. H., & Connelly, A. (2019). MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202, 116137.
<https://doi.org/10.1016/J.NEUROIMAGE.2019.116137>