

Grant Agreement:	604102	Project Title:	Human Brain Project
Document Title:	Neuroinformatics Platform v1		
Document Filename:	SP5 D5.8.4 Resubmission_FINAL.docx		
Deliverable Number:	D5.8.4		
Deliverable Type:	Prototype		
Work Package(s):	WP5.1, WP5.2, WP5.3, WP5.4, WP5.5, WP5.6, WP5.7, WP5.8		
Dissemination Level:	PU		
Planned Delivery Date:	M30 / 31 Mar 2016		
Actual Delivery Date:	M30 / 08 Apr 2016		
Resubmission Date	30 Sep 2016		
Authors:	Sean HILL, EPFL (P1); Sten GRILLNER, KI (P31)		
Compiling Editors:	Sean HILL, EPFL (P1); Sten GRILLNER, KI (P31)		
Contributors:	EPFL (P1): CNRS (P7): Andrew DAVISON JÜLICH (P17): Michael DENKER, Sonja GRÜN, Alper YEGENOGLU SKU (P49): Paul TIESINGA UPM (P59): José Maria PEÑA, Pedro LARRAÑAGA (P68): Jan BJAALIE ; Pascal FUA (EPFL) (P1).		
Coordinator Review:	EPFL (P1): Jeff MULLER, Martin TELEFONT UHEI (P45): Sabine SCHNEIDER, Martina SCHMALHOLZ		
Editorial Review:	EPFL (P1): Guy WILLIS, Lauren ORWIN, Colin MCKINNON		
Abstract:	<p>Neuroinformatics Platform v1 - preliminary release for internal consortium use (prototype)</p> <p>This Deliverable sets out the specifications for the HBP's Neuroinformatics Platform (NIP), which serves as a central brain data resource for the Project's other five Platforms. The principle elements of the NIP defined here are:</p> <ul style="list-style-type: none"> <li>• Tools for Brain Atlases (including the Brain Atlas Builder).</li> <li>• Tools for Structural Data Analysis</li> <li>• Tools for Functional Data Analysis</li> <li>• Predictive Neuroinformatics</li> <li>• Brain Atlases (Mouse and Human)</li> </ul> <p>The document also describes the overall architecture of the NIP, and provides a roadmap for the assembly of the atlases.</p>		
Keywords:	NIP, brain, atlases, data, platform, specification		

## Table of Contents

1. Purpose of this Document .....	7
2. Documentation (D5.8.5) .....	7
3. NIP: Overall Architecture (T5.6.1) .....	7
3.2 Architecture .....	8
3.3 Web components .....	12
3.4 Service components (REST APIs) .....	12
3.5 How to Access the Neuroinformatics Platform .....	13
3.6 Platform User Instructions .....	13
3.7 Platform Testing and Quality Strategy (T5.6.1) .....	14
<b>!Unexpected End of Formula</b>	
4.1 Platform User Adoption Strategy (T5.7.2) .....	16
4.2 Help and User Feedback .....	17
4.3 User Feedback Received Month 18 - Month 30 .....	17
4.4 Services provided to other Platforms .....	18
5. Tools for Brain Atlases (WP5.1) .....	18
5.1 Tools for Brain Atlases: Data Management Tools .....	18
5.1.1 Federating neuroscience data repositories .....	18
5.1.2 Registering data sets .....	19
5.1.3 Registering and annotating data through a web interface (T5.1.4) .....	20
5.1.4 Data curation workflow .....	20
5.1.5 Data type definition registration .....	21
5.1.6 Data provenance format .....	22
5.2 Text mining services (T5.1.2) .....	22
5.3 Unified search service .....	23
5.4 Tools for Brain Atlases: Functional Requirements .....	25
5.4.1 HBP-PROV: Minimal metadata specifications .....	25
6. Data standards and data model (T5.1.4) .....	26
6.1 Standard data formats .....	26
6.2 Specialised Data Access .....	27
6.2.3 Atlas .....	27
6.2.4 Electrophysiology .....	28
6.2.5 Morphology .....	29
6.2.6 Receptor Density .....	29
6.2.7 Transcriptomics .....	30
6.2.8 Kinetic model .....	30
7. DataSpace - Data federation of repositories (T5.1.1) .....	30
7.1 Archival data repositories .....	30
7.2 Active data repositories .....	32
7.3 KnowledgeGraph - Database built on HBP-PROV data model and ontologies (T5.1.3) .....	33
8. KnowledgeSpace - Semantic wiki for community ontology maintenance, data curation and living review articles (T5.1.5, T5.1.3) .....	33
8.1 Knowledge Space .....	33
8.2 LabSpace - Semantic wiki for organising and curating summaries of individual laboratory data sets .....	38
8.2.1 Automated data monitoring pipeline .....	39

8.2.2 Data access API .....	39
8.2.3 Data analysis and data mining module.....	39
8.2.4 Tools for Brain Atlases: Software .....	39
8.2.5 Tools for Brain Atlases: Physical Architecture.....	40
8.2.6 Integrating existing atlas.....	40
8.2.7 Mapping new image data to existing atlas .....	40
8.2.8 Data processing modules.....	41
8.2.9 Data ingestion to voxel brain.....	44
8.2.10 Tools for Brain Atlases: Prerequisites .....	45
8.3 Brain Atlas Builder (T5.1.6) .....	45
8.3.1 Brain Atlas Builder: Overall Goals .....	46
8.3.2 Components.....	47
<b>9. Tools for Structural Data Analysis (WP5.2 and WP5.4) .....</b>	<b>47</b>
9.1 Structural Tools: EMDigest (T5.2.1).....	47
9.2 Structural Tools: EspINA (T5.2.2) .....	47
9.2.1 EspINA.....	47
9.1.2 EM images segmentation (T5.2.2) .....	50
<b>10. Predictive Neuroinformatics (T5.4.1).....</b>	<b>50</b>
10.1 Description.....	50
10.2 Overall goal and embedding in HBP.....	51
10.2.1 Current strategy .....	51
10.2.2 Alternate approaches from outside the HBP consortium. ....	52
10.2.3 Each of the steps in more detail. ....	54
10.2.4 Use cases. ....	56
10.3 Structural Tools: 3DSynapsesSA (T5.4.2).....	62
10.3.1 SynapseGenerator .....	62
10.3.2 3DSynapsesSA.....	63
10.4 Structural Tools: 3DSomaMS (T5.4.2).....	64
10.4.1 3DSomaMS.....	64
<b>11. Tools for Functional Analysis (WP5.3, T5.3.1, T5.3.2).....</b>	<b>65</b>
11.1 Elephant .....	65
<b>12. Brain Atlases (WP5.5).....</b>	<b>68</b>
12.1 General comment.....	68
12.2 The Mouse Brain Atlas (T5.5.1) .....	69
12.2.1 AlignII .....	69
12.2.2 QuickNII .....	70
12.2.3 MeshGen.....	71
12.2.4 MeshView.....	72
12.2.5 CutNII .....	74
12.2.6 Scalable Brain Atlas .....	75
12.3 The Human Brain Atlas (T5.5.2) .....	76
12.3.1 JuBrain workflow .....	76
12.3.2 JuBrain Cytoviewer .....	77
12.3.3 Papaya.....	78
<b>Annex B: Software and Services Included in this Platform Release.....</b>	<b>80</b>
Product/Software Package/Service name:.....	80
3DSomaMS .....	80
Description .....	81
Functionality .....	81
DATA Repository service (ZENODO).....	82
3DSynapsesSA .....	84

Elephant .....	86
EspINA 88	
Papaya 92	
JuBrain Cytoviewer .....	94
Morphology Viewer .....	96
AlignII 98	
QuickNII100	
MeshGen .....	102
MeshView .....	104
CutNII 106	
Scalable Brain Atlas .....	108
SynapseGenerator .....	110
Knowledge Space .....	112
<b>Annex C: Summary - Platform Use Case Status.....</b>	<b>114</b>
Summary of use cases .....	146
New Use Cases Platform Use Case Status.....	155
Summary of use cases .....	155
Response times and numbers of users .....	159
<b>Annex D: Summary - Service IT Resource Planning .....</b>	<b>160</b>
Interfaces to data providers and other service .....	163
Response times & number of users .....	163
Overall list of tools and services offered by platform.....	163
EspINA .....	164
<b>Annex E: Summary - Service Technology Readiness Levels (TRLs) Metrics .....</b>	<b>165</b>
<b>Annex F: Backlog (Remaining bugs and new features to be added) .....</b>	<b>168</b>
Product/Software Package/Service name.....	168
Current limitations, problems and bug list.....	168
Usage levels of data storage and compute resources .....	168
Main technologies used.....	168
Remaining Bugs .....	168
Features.....	169
<b>Annex G: IPR Status, Ownership and Innovation Potential .....</b>	<b>171</b>
<b>Appendix I: Report on the Data Hosting Pilot .....</b>	<b>173</b>
<b>Appendix II: Use Cases and Use Case IDs .....</b>	<b>174</b>
<b>Appendix III: NIP Listing of Available Tools .....</b>	<b>176</b>

## List of Figures and Tables

Figure 1: Overview of the Neuroinformatics architecture.....	8
Figure 2: Overview of the Neuroinformatics Platform web applications, REST-based services and APIs, data indexes and annotations, and distributed data repositories. ....	9
Figure 3: Overview of the functions provided by the Neuroinformatics Platform Architecture....	10
Figure 4: BBP NIP HBP infrastructure .....	10
Figure 5: 3-tier infrastructure of the NIP .....	11
Figure 6: Existing components of the tiers architecture of the NIP .....	11
Figure 7: NIP component representation.....	12
Figure 8: Web components of the NIP .....	12
Figure 9: Service components of the NIP.....	13
Figure 10: Sample of monitoring activity .....	14
Figure 11: Log visualization of the NIP API.....	15
Figure 12: NIP development and deployment model.....	16
Figure 13: Data curation workflow .....	21
Figure 14: HBP-PROV formats .....	22
Figure 15: Simple example of provenance structure containing one dataset .....	22
Figure 16: Landing page view of the Search API .....	23
Figure 17: Sample of Search results.....	24
Figure 18: Metadata visible on Search API results .....	24
Figure 19: Contributors affiliation visibility on Search API.....	25
Figure 20: Restriction access to dataset on Search API .....	25
Figure 21: KnowledgeGraph, based on HBP-PROV metadata specification and data model .....	25
Figure 22: Classes of services provided by the NIP .....	27
Figure 23: KnowledgeSpace collaboration developed with INCF .....	37
Figure 24: Schema of integrating existing atlas.....	40
Figure 25: Pipeline of integration of new image data .....	41
Figure 26: Data storage structure .....	42
Figure 27: An example of hierarchical structure of the hippocampus layers in json format.....	45
Figure 28: Schema of multiple brain region smoothing .....	45
Figure 29: Brain Atlas Builder .....	47
Figure 30: Data from a recent saturated reconstruction based on electron microscopy data.....	52
Figure 31: Graphical overview of the different steps in the reverse geometric approach.....	54
Figure 32: Morphologies are accessed, evaluated and selected with the morphology viewer.....	56
Figure 33: The morphologies are transformed into Gaussian mixture models appropriate for the SynapseGenerator Pipeline. ....	57

Figure 34: The online documentation provides an overview of the different steps in the pipeline. ....	57
Figure 35: The Y matrix, with pre or post- synaptic densities, is generated. A subset of the postsynaptic densities are shown. ....	58
Figure 36: The X matrix, with sampled pre and postsynaptic elements, is generated. ....	58
Figure 37: The dendritic and axonal morphologies are regenerated using the minimum spanning tree algorithm. A few example dendritic morphologies are shown. ....	59
Figure 38: Synapse placement on example morphologies. Synapses are indicated by filled circles. ....	59
Figure 39: Validation of the constructed projection. ....	60
Figure 40: Sample of Elephant NIP use case .....	68
Figure 41: The Scalable Brain Atlas .....	75
Figure 42: The Scalable Brain Atlas .....	109
Table 1: Use cases and success scenarios (1) .....	117
Table 2: Use cases and success scenarios (2) .....	121
Table 3: Use cases and success scenarios (3) .....	148
Table 4: New use cases Platform use case status .....	155
Table 5: Use cases and success scenarios (4) .....	156
Table 6: Service IT resource planning .....	160
Table 7: KPI status .....	162
Table 8: Milestones .....	164
Table 9: Service technology readiness levels metrics .....	165
Table 10: Remaining bugs .....	169
Table 11: Features .....	169
Table 12: IPR status, ownership and innovation potential .....	171
Table 13: SP5NIP-FR Use Cases and Use Case IDs .....	174
Table 14: SP5DAT-FR Use Cases and Use Case IDs .....	175
Table 15: SP5BAS-FR Use Cases and Use Case IDs .....	175
Table 16: NIP listing of available tools .....	176

## 1. Purpose of this Document

This document provides an overview of the Neuroinformatics Platform (NIP) v1 release and related information. We have reformatted D5.8.4 using the structure D5.8.1 “specification document”.

All original content of D5.8.4 is present in this document. Task numbers have been added. All additional information provided (from the PPR) is highlighted.

## 2. Documentation (D5.8.5)

Our documentation is available both on the HBP Collaboratory within the NIP (<https://collab.humanbrainproject.eu/#/collab/47/nav/6641>) as well as on our NIP website (<https://nip.humanbrainproject.eu/documentation/>). Each [tool](#) is described, and use cases and samples of integration with the NIP are present. In the NIP website documentation you will find a [user manual](#), a [data integration manual](#) and a [technical manual](#).

## 3. NIP: Overall Architecture (T5.6.1)

The NIP and the Brain Atlases allow neuroscientists to collaboratively curate, analyse, share, and publish heterogeneous neuroscience data.

The NIP is an entirely new platform design and implementation, specifically started within the HBP and developed by a new team recruited starting with the project launch at the beginning of the Ramp-Up Phase.



## 3.2 Architecture

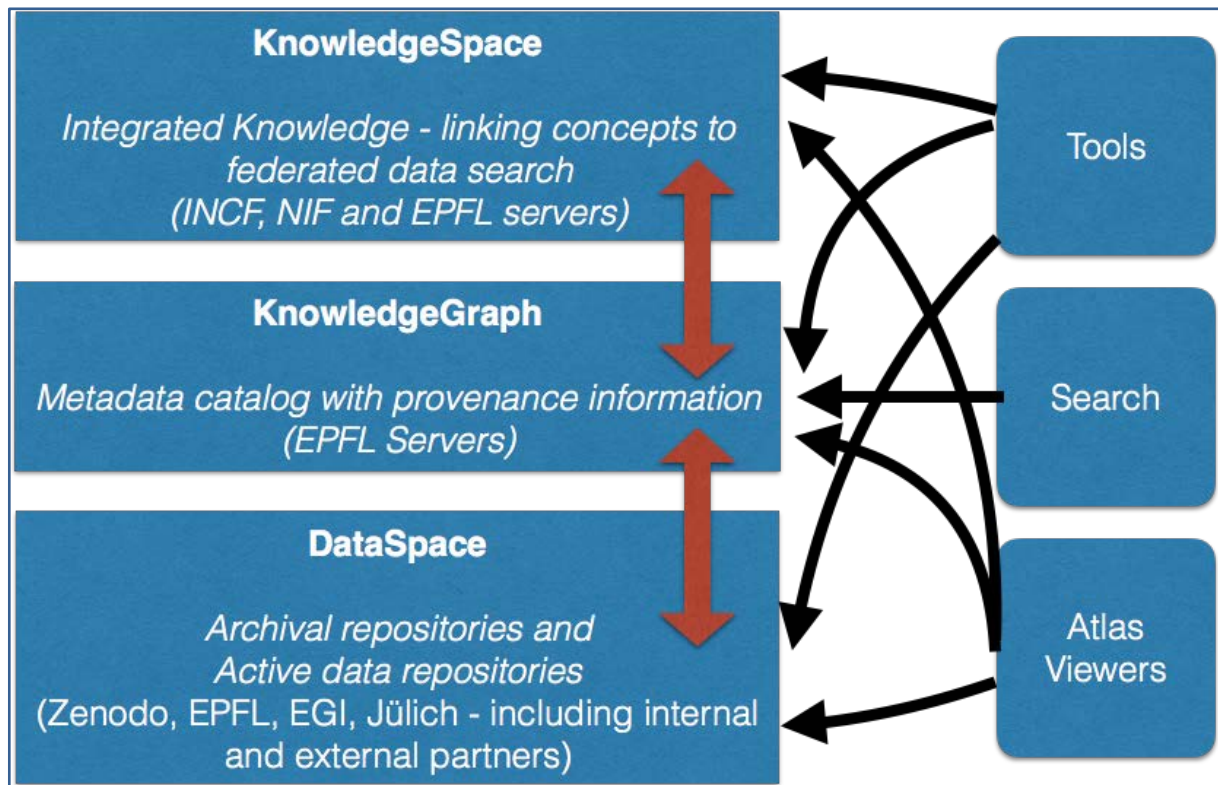


Figure 1: Overview of the Neuroinformatics architecture

The DataSpace is a federation of distributed data sources including archival and active data repositories. The KnowledgeGraph is a metadata catalog built on a provenance data model. The KnowledgeSpace provides the link between the ontologies and neuroscience concepts used to annotate the data and the diverse datasets from around the world that have been annotated with that concept or a synonym. Tools, Search and Atlas Viewers are all applications that use APIs to access the services provided.

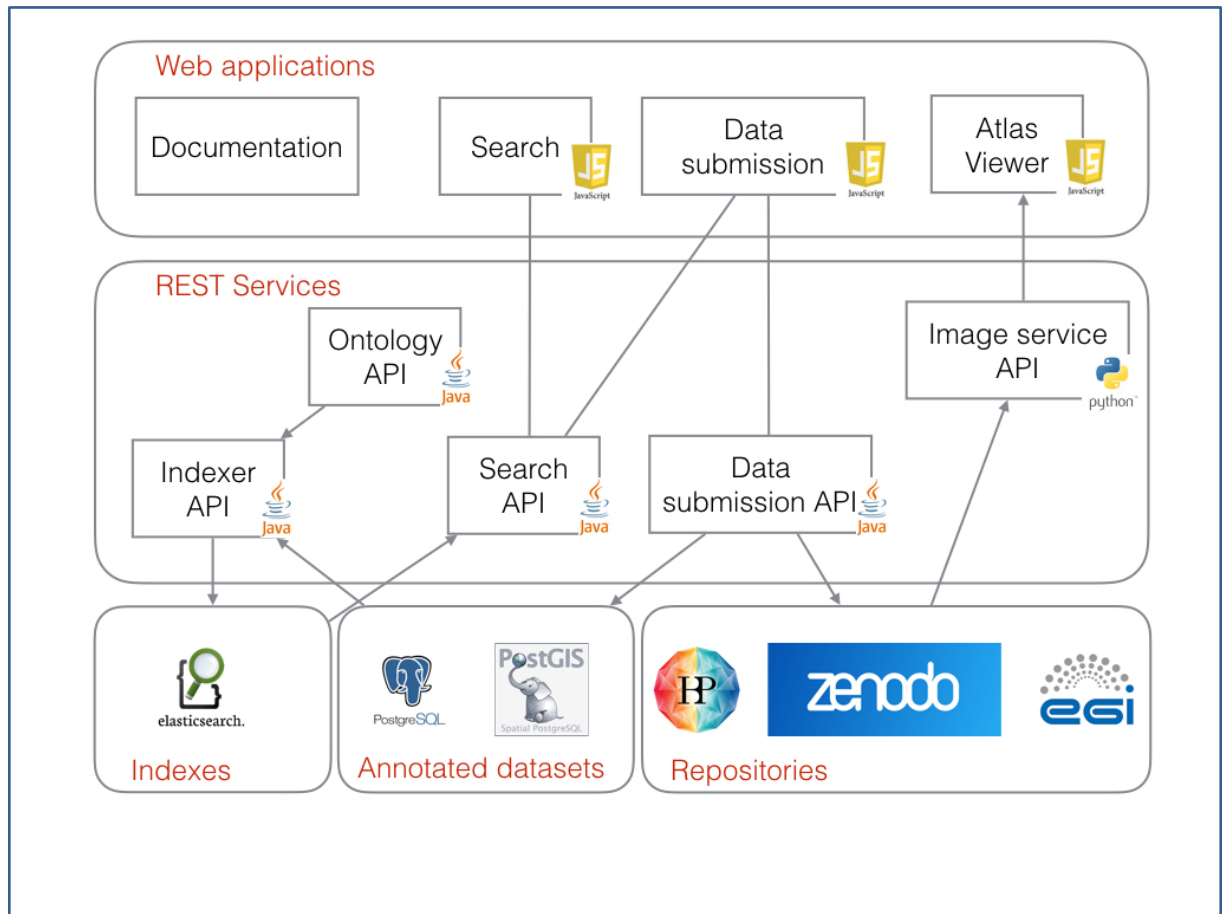
The NIP comprises a set of web services, APIs, standards, and user interfaces that enable registration, search, visualization and analysis of distributed neuroscience data. The Platform is designed to enable a modeller/user to identify data that can potentially be integrated and searchable for neuroscientists around the world (either within the HBP or not).

The Platform consists of identified neuroscience data repositories, community data portals, and data federation mechanisms (known as Dataspace), a provenance graph to which all data is registered for discovery, search, access and tracking (KnowledgeGraph), web services and viewers for multi-resolution brain atlas viewing and navigation (Atlas Viewers), and a community-oriented encyclopaedia that provides standard vocabularies, taxonomies and ontologies linked to data, models and literature from federated data sources around the world (KnowledgeSpace).

Brain atlases consist of a template brain dataset which provides the spatial coordinates, brain parcellations which define brain region/structure volume boundaries, and an ontology which provides a structured naming system for the brain parcellations. Datasets can be integrated with these atlases both semantically (by being tagged with a brain region name), spatially (by assigning specific x,y,z coordinates), and aligned to the specific brain template by specifying a transformation (linear or nonlinear) to map the dataset to the brain atlas coordinate system.

The architecture is designed to enable distributed integration of heterogeneous data multi-modal brain atlases to integrate data that may be physically located on data servers around the world.





**Figure 2: Overview of the Neuroinformatics Platform web applications, REST-based services and APIs, data indexes and annotations, and distributed data repositories.**

The NIP is a tool to facilitate data integration, but the specific data integration goals, features, requirements and specification depend on the scientific questions and objectives.

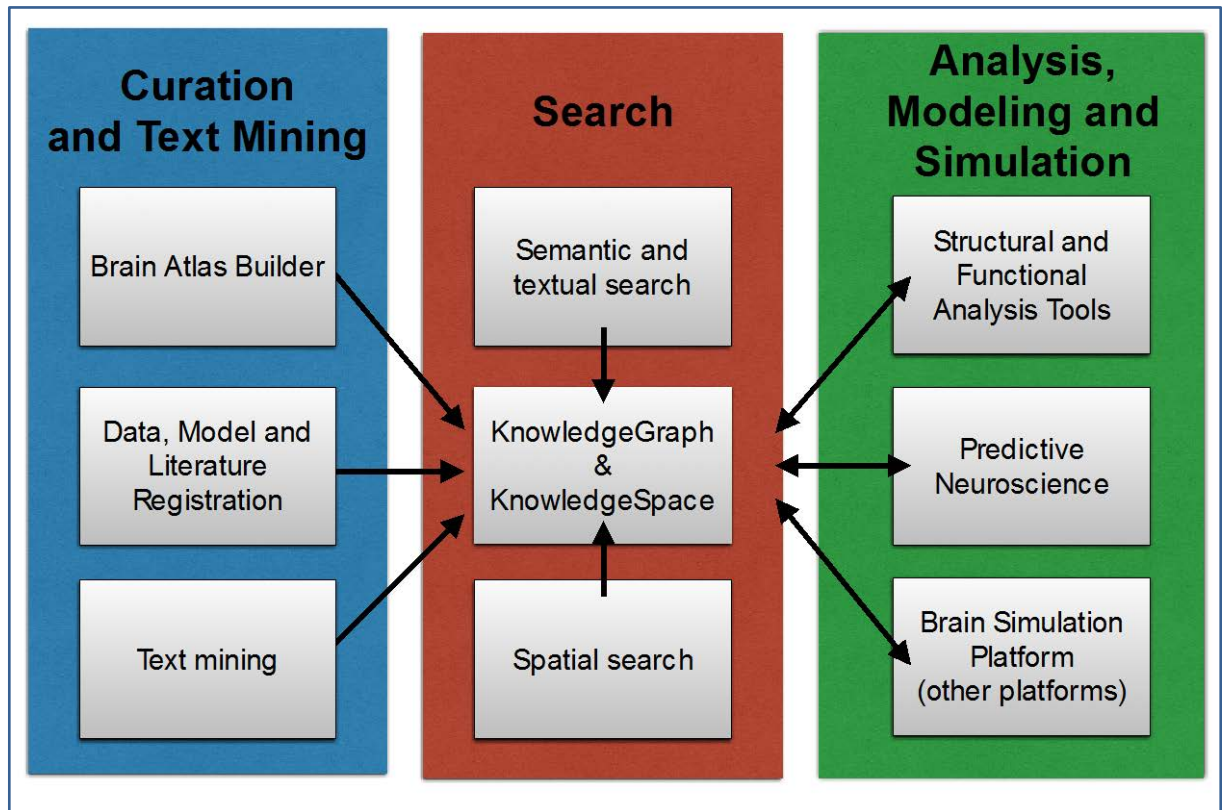


Figure 3: Overview of the functions provided by the Neuroinformatics Platform Architecture.

Data are curated and registered into the KnowledgeGraph and KnowledgeSpace or LabSpace via the Brain Atlas Builder, manual data, model and literature registration workflows, and automatic text mining. Analysis, modelling and simulation tools use an array of textual, semantic and spatial search services to find data to work with. These same tools register their results in the KnowledgeGraph and KnowledgeSpace, as appropriate.

#### *Technical details of the NIP*

The NIP components are available under the [nip.humanbrainproject.eu](http://nip.humanbrainproject.eu) domain. It is hosted in the Blue Brain Project infrastructure as depicted below:

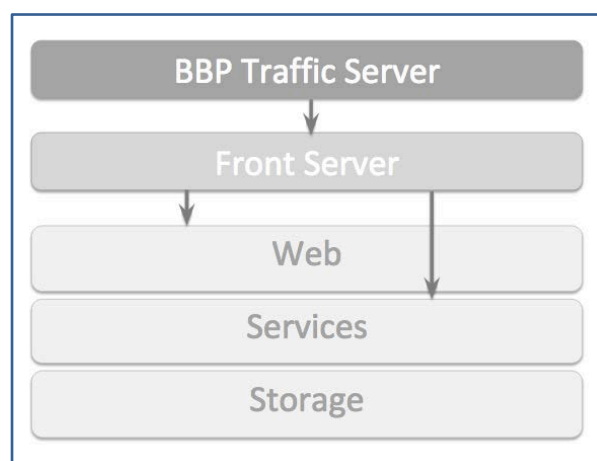


Figure 4: BBP NIP HBP infrastructure

The Blue Brain Project manages the top-level [humanbrainproject.eu](http://humanbrainproject.eu) domain. All traffic lands on the Traffic Server, which handles the first level of routing, caching and SSL handling across all the subdomains. This server runs the Apache HTTP server.

All traffic to the NIP components will be redirected to the [nip.humanbrainproject.eu](https://nip.humanbrainproject.eu) domain, which is managed by our **Front Server**. It acts as a single access point, handling routing to individual components (such as <https://nip.humanbrainproject.eu/atlas>), load balancing, and access control in some cases such as the Image Service. This server runs the NGINX HTTP server with Lua scripting extensions.

The NIP itself is built as a 3-tier architecture as depicted below:

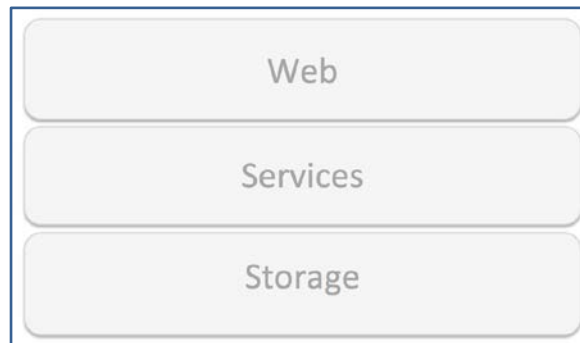


Figure 5: 3-tier infrastructure of the NIP

Before getting into the details of which components exist in each of these tiers, it is worth noting that there are two main responsibilities for the Services layer knowledgeGraph and Specialised Data Access that will be described below.

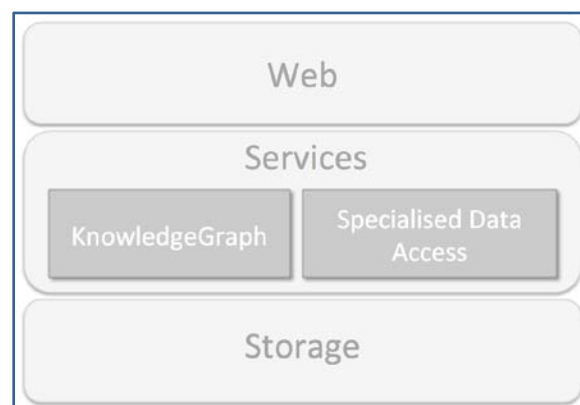
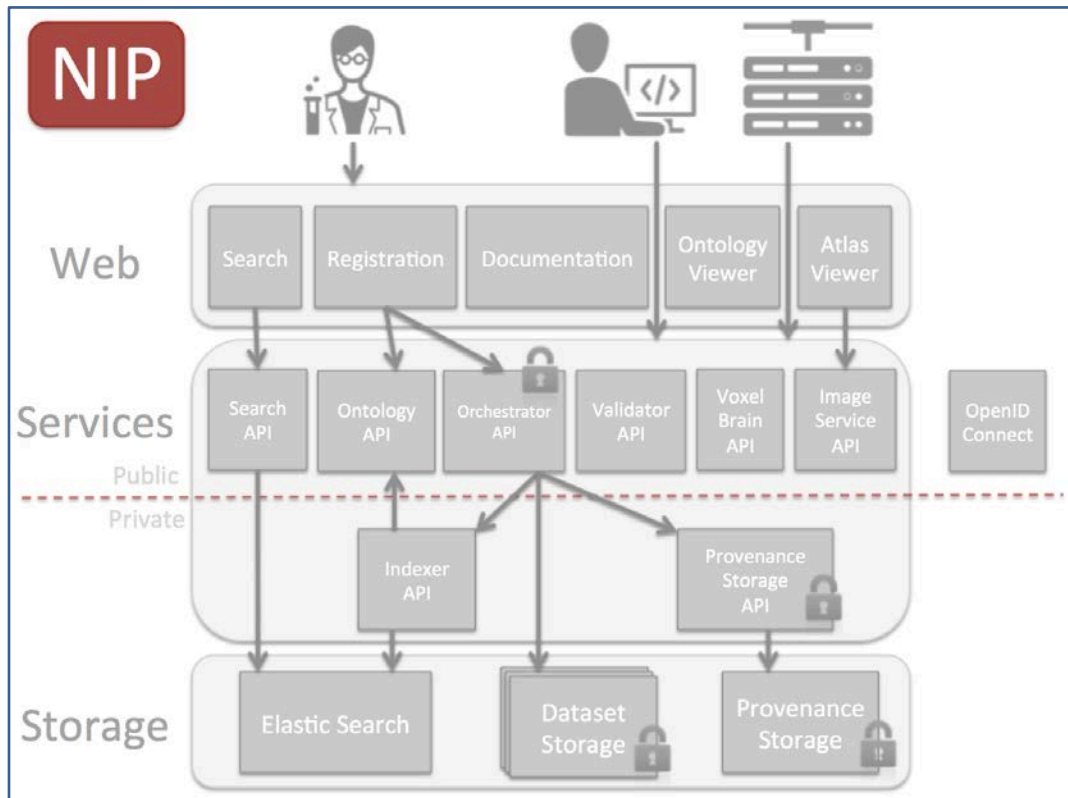


Figure 6: Existing components of the tiers architecture of the NIP

Thus, the NIP has developed several tailored services to address specific needs that were expressed by the platform users.

In the diagram below, we represent the components that belongs to each of these tiers:



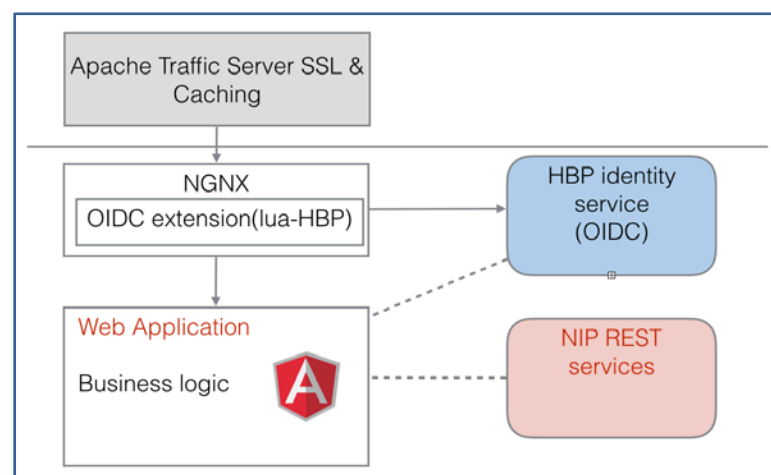
**Figure 7: NIP component representation**

Please note that the components above the red line are visible/accessible on the internet while those under are only accessible from inside the project internal network.

*Architectural templates*

### 3.3 Web components

The diagram below depicts the standard architecture for all web applications of the NIP:



**Figure 8: Web components of the NIP**

### 3.4 Service components (REST APIs)

The diagram below depicts the standard architecture for the APIs of the NIP:

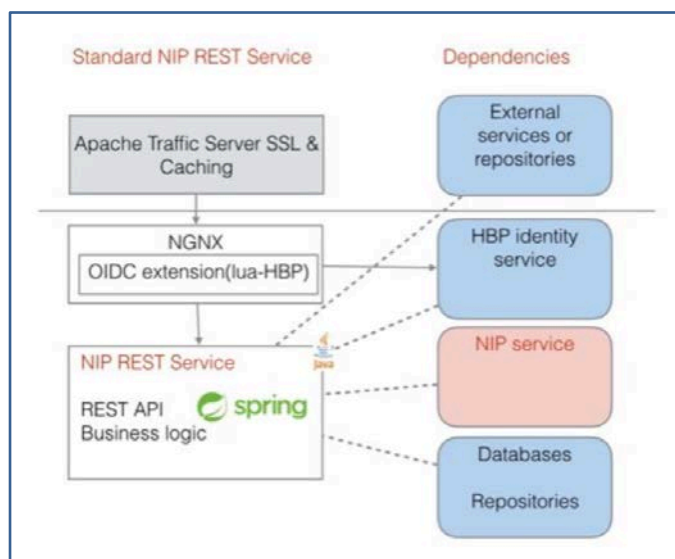


Figure 9: Service components of the NIP

If you are interested in reviewing the architecture of all our components, please refer to the technical manual within our documentation.

### 3.5 How to Access the Neuroinformatics Platform

The Neuroinformatics Platform is one of six ITC Platforms that comprise the HBP Scientific Research Infrastructure. All these Platforms can be accessed via the HBP Collaboratory web interface:

Direct link to the NIP on the Collaboratory ([collab.humanbrainproject.eu](https://collab.humanbrainproject.eu/)):

- <https://collab.humanbrainproject.eu/#/collab/47/nav/6641>
- The NIP is accessible through the Collaboratory or REST web service APIs.
- The process for user authentication will be the same as the standard HBP authentication process.
- Users can apply to register their data and will be able to search for data according to specific criteria. Data integration or linkage to other repositories offer DOIs as well as repository record ID and File Name for each dataset. This information is collected by the NIP and will be used by our Search API, for example. These datasets are not mutable, solely metadata and access rights can be changed by the contributor. The DOI is a mechanism to track citations to the data - data consumers should use the DOI as a citation in any related paper. In addition, the number of views and downloads of the datasets can be tracked.

### 3.6 Platform User Instructions

The NIP provides user support in the form of online documentation on how to use the NIP, and examples of how to perform data queries. The NIP documentation can be found at:

<https://collab.humanbrainproject.eu/#/collab/47/nav/5245>

The Platform Documentation constitutes a separate Deliverable (D5.8.5 - Neuroinformatics Platform v1 — Documentation (<https://nip.humanbrainproject.eu/documentation/>), which will include direct links to [user manual](#), a [data integration manual](#) and a [technical manual](#).

## 3.7 Platform Testing and Quality Strategy (T5.6.1)

Applications and services are automatically tested by unit and integration tests. The strategy is to have at least 90% unit test coverage for all back-end services and 80% for web application. The Platform uses continuous build tools to automatically build and deploy new versions of the code.

Additional manual tests are performed before releasing the Platform in production. Users have the possibility to report bugs, to request for additional features using our [data.nip@humanbrainproject.eu](mailto:data.nip@humanbrainproject.eu) e-mail address. This can be found in our documentation under the 'get support' section or in several places in the documentation. Specific user workflows can be found within our documentation. It includes access to the API, requirements needed in order to become a contributor, ToS, as well as Atlas contribution, restricted access or not to HBP data, architecture of the Platform, and software tools that are supported by respective software teams and guide that can be found in the HBP Collaboratory, NIP, tools section under each specific tool.

Software testing and quality aspects are covered in the respective documentations of single tools and are the responsibility of the software teams.

## 4. Platform monitoring and maintenance (T5.6.1)

In detail, all NIP web applications and service are continuously monitored and alerts are raised should one of these health checks fail. All developers of the NIP team are notified via email of such failure and they take responsibility to correct applicative faults or escalate the issue to the BBP core services if the need arises. A dashboard is visible where the developers work as an extra mean of raising awareness of failing service and diminish the time it takes to restore a failed service. The screen capture below shows this dashboard.

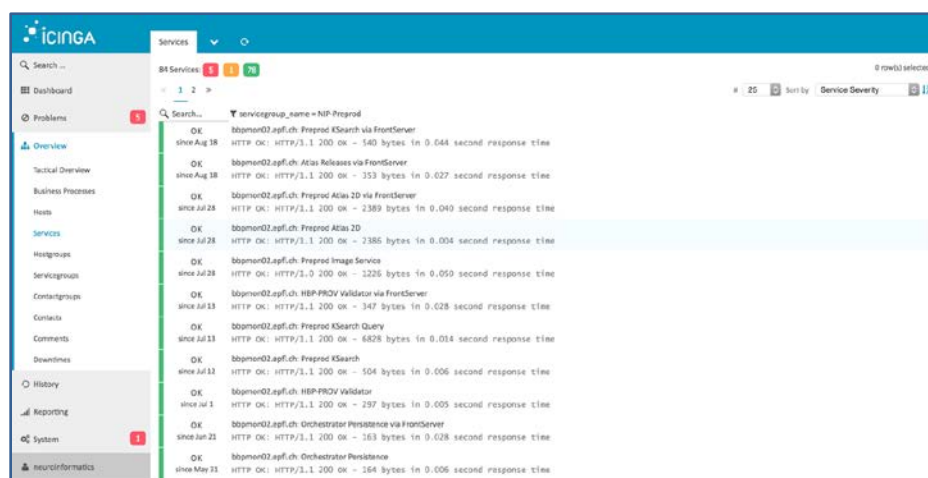


Figure 10: Sample of monitoring activity

Furthermore, all API services have their application logs collected and centralised using [FileBeat](#) (log shipping), [LogStash](#) (log processing) and [Kibana](#) (log inspection and dashboard facilities). The figure below shows a basic Kibana dashboard, highlighting that some applications have reported ERROR logs in both dev and staging environment. Clicking on the red section of the pie chart (bottom right panel) allows the developer to see the specific logs (left side panel), even if they potentially came from different virtual machines.



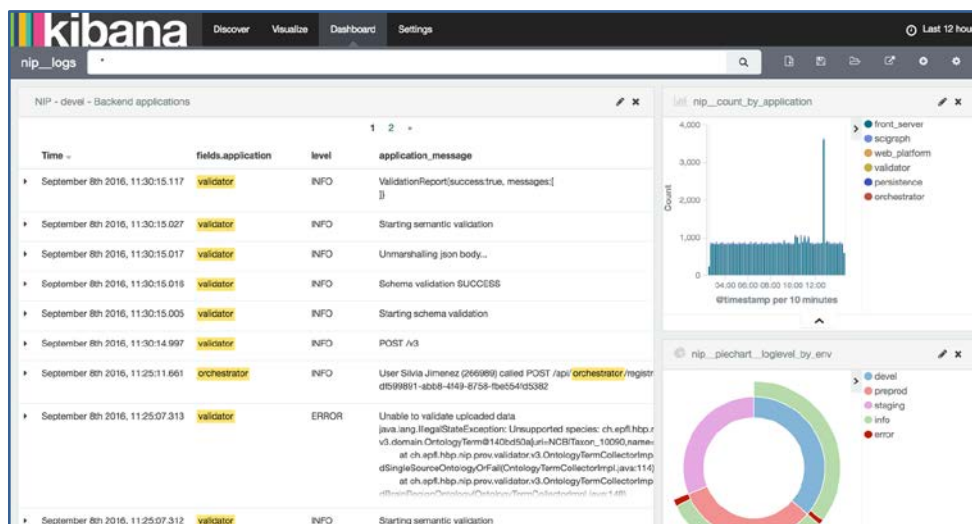


Figure 11: Log visualization of the NIP API

More details information can be find in our NIP website technical manual documentation (<https://nip.humanbrainproject.eu/documentation/technical-manual.html>)

#### *Software development and service deployment (T5.6.1)*

The functionalities of the NIP are exposed via REST services. In addition, web client applications provide interactions with the content and functionalities of the Platform. The NIP REST services, web applications and databases are hosted on BBP infrastructure. All Platform modules are developed and deployed following BBP development best practices.

All services and UI components are developed using the following best practices:

- Unit testing and assessment of test coverages
- Utilisation of code review
- Automated Integration testing suites (automated post-deployment system testing)
- Continuous integration
- Ticket management in Jira for project feature planning
- Coding standards

The Platform is developed following Agile methodology ([SCRUM](#)), with new features and functionalities developed iteratively and released continuously. Thus, repeatable deployment software is a crucial component to deliver continuous improved software to users.

Best practices from Agile software development have been adopted and implemented with a pragmatic DevOps model.

- Deployment lifecycle in development, staging and production environments for all services and web application,
- Continuous integration, i.e. unit testing, integration testing, automated software builds and package releases,
- VM configuration managed in source control and associated with a specific VM,
- Continuous automated deployment of the latest version of the software in dev,
- Automated release of the Platform in staging environment; the Platform is released in production after running integration and manual tests,
- Internet Gateway with caching proxy server.



The NIP follows the BBP Standard development and deployment model, as described in the HBP System Engineering documentation and represented below:

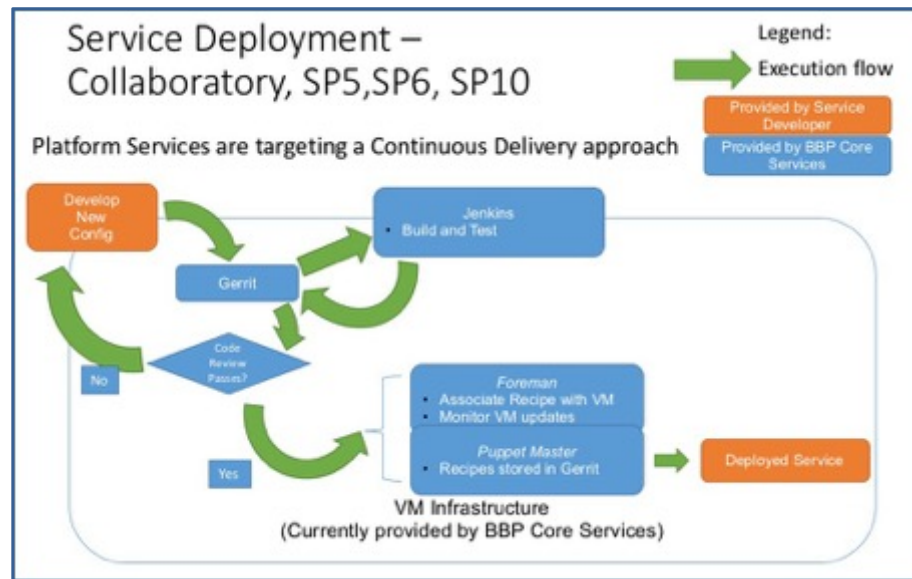


Figure 12: NIP development and deployment model

This process makes extensive use of the following open source tools:

- **Git** - a leading distributed Version Control system [\[link\]](#)
- **Jenkins** - a leading Java-based Continuous Integration system [\[link\]](#)
- **Gerrit** - a leading Git and Java-based source code change review system [\[link\]](#)
- **Puppet** - a leading *devops* system configuration system [\[link\]](#)
- **Foreman** - associates Puppet recipes with particular hosts [\[link\]](#)

More details information can be find in our NIP website technical manual documentation (<https://nip.humanbrainproject.eu/documentation/technical-manual.html#software-development-and-service-deployment>)

## 4.1 Platform User Adoption Strategy (T5.7.2)

The NIP seeks to engage users by supporting specific use cases in data search, queries of data for model building, atlas viewing and navigation and data integration. The core driving use case is to enable the query of data to support specific modelling use cases. Two key initial modelling use cases are building single cell models and building brain regions.

The single cell modelling use case will form a key part of our initial user adoption strategy involving a joint educational workshop with the Allen Institute for Brain Science. This workshop will be held as a satellite meeting with the European FENS neuroscience meeting in July. The same material will form part of a course held in conjunction with the US Society for Neuroscience meeting in October, while an extended workshop will be given as part of the HBP Education program at the end of November, 2016.

The specific use cases will be using the KnowledgeSpace to support computational modelling of neurons and circuits. In addition, the data-driven pipelines for modelling single neurons and brain regions using specialised APIs from the NIP will be central to these workshops. Other aspects to be demonstrated are data search and the variety of brain atlases available on the NIP.

We will publicise the NIP at the FENS neuroscience forum (Copenhagen, 2016), SfN (San Diego, 2016) and Neuroinformatics (Reading, 2016) to attract new users and identify priority community use cases. Community outreach will also extend to other relevant outreach events as applicable.

Moving forward we aim to integrate the NIP with new data sources and additional informatics infrastructures to support additional user stories and use cases. In addition, we are participating in additional calls under the E-INFRA H20202 program. This will help us integrate the NIP with additional existing European neuroimaging infrastructures and communities with thousands of active users.

The NIP will be represented at the Global Brain Workshop held at Johns Hopkins University 7-8 April, 2016 to plan international cooperation activities. In addition, the NIP will be presented during the Young Researchers events in Budapest, 12 April, 2016, where we will demonstrate some use cases. We will be part of the Keystone conference called the State of Brain (22-26 May, 2016) that will bring together leading representatives from HBP, the US Brain initiative, and the Chinese and Japanese Brain initiatives, the aim being to explore possibilities to interact and cooperate.

## 4.2 Help and User Feedback

To obtain help in using the platform, please start by checking the online user documentation here:

<https://collab.humanbrainproject.eu/#/collab/47/nav/4752>

On the documentation website you will find a user manual, a data integration manual and a technical manual. Moreover a forum exist for more interactive answers :  
<https://forum.humanbrainproject.eu/t/neuroinformatics-platform/165>

If you need personal assistance, want to provide feedback or contribute to the on-going development of the platform, please contact: [data.nip@humanbrainproject.eu](mailto:data.nip@humanbrainproject.eu)

## 4.3 User Feedback Received Month 18 - Month 30

We provide samples of the feedback from users and help we provide in order to integrate datasets and needs from other SPs (SP2 and SP6 respectively).

### SP2:

Integration of whole brain datasets produced in the context of SP2 and their visualisation on the NIP was done in close collaboration with SP2 scientists. In particular, their feedback on the first version of the Atlas Viewer led to the improvement of the navigation of whole brain datasets, and to a number of feature requests that is guiding the development of the application.

### SP6:

Modellers are primary users for the search API and applications. Data integration has to be deep enough for modellers to find the data they need for model building and navigation.

The search API is used in the modelling pipelines developed by SP6. Single cell morphologies are retrieved based on criteria matching the modellers needs, while feedback on electrophysiological data integration led to a review of the metadata and an alternative way of grouping electrophysiological traces and additional metadata to be provided to modellers.

The voxel brain API which exposes information at the voxel level, such as cell density and cell distribution, was co-designed with the modellers to fulfil the region/whole brain model building. Short iterations were done and deployed enabling rapid feedback and improvement of the API and data format.

For further enquiries, customers will be able to access the Platform from March 30th. We will therefore from this date be able to receive feedback from users via the contact information e-mail ([data.nip@humanbrainproject.eu](mailto:data.nip@humanbrainproject.eu)). Between Month18 and 30, APIs to access data were refined in collaboration with modellers to ensure that the right metadata were captured to enable specific queries.

In a first step of monitoring NIP component, using google doc analytics we are able to follow the development of the NIP, gather first statistical information and KPIs. We are in the process of standardising this approach within the HBP Collaboratory and the other Platforms in order to provide key information on NIP efficiency and relevance for the community.

## 4.4 Services provided to other Platforms

- Data federation
- Data registration
- Data annotation
- Data curation
- Data standard definitions
- Data mining services
- Ontology lookup
- Ontology editing
- Ontology curation
- Atlas navigation
- Atlas data anchoring
- Atlas data search
- Atlas curation
- Data analysis services
- Predictive neuroscience services
- Semantic wiki services
- Manual curation workflows
- Wiki documentation of data, model and literature samples

## 5. Tools for Brain Atlases (WP5.1)

### 5.1 Tools for Brain Atlases: Data Management Tools

#### *5.1.1 Federating neuroscience data repositories*

The NIP allows external contributors to register data that they host themselves (as opposed to being copied to and hosted in the NIP), as long as the data is accessible through a standardised interface. This is what we call **Active Data Repositories**. For instance, a BBIC Image Stack dataset could be served from a third-party Image Service instance, as long as its REST API is identical to the NIP Image Service.

The KnowledgeGraph enables such federation scenarios by not being tied to the sole NIP storage components. It can refer to external services (Active Data Repositories) as ways to access a dataset. A given dataset may also be accessible through multiple services hosted in multiple places.

For instance, we collaborated with the European Grid Infrastructure (EGI-see Annex I) on deploying instances of the Image Service on multiple datacenters across Europe. All public Atlas datasets were therefore accessible through any of the EGI datacenters, in addition to being accessible through the NIP Image Service instance.

### **5.1.2 Registering data sets**

Data integration and curation are major challenges in neuroscience. Whether or not a dataset is of sufficient quality is inextricably linked to the scientific question that will be asked using the data. Similarly, integration of neuroscience datasets may be achievable to differing degrees depending on the requirements of the scientific question and precision required.

Data curation requires examining the data, ensuring it is in an appropriate format (NIFTI or TIFF for volumetric data, a readable electronic format), evaluating the suitability of the data for a specific scientific purpose and annotating the data with an appropriate ontology (e.g. brain regions, cell types, etc.) to ensure that the data can be clearly identified. In many cases, curation also involves extending the ontology with new categories. Depending on the specific scientific use case, additional metadata may be required (e.g. for cellular modelling the resting potential, input resistance, junction potential, etc. may be required from intracellular electrophysiological data)

Depending on the techniques used to acquire the data, the precision may be insufficient for one use, yet perfectly adequate for another. When defining a curation task, it is essential to clearly define the scientific requirements - what specific ontologies are required for the annotation, what are the quality checks that need to be performed, how is the data intended to be queried? Defining the curation requirements is also essential to defining the requirements of the ontology being used to annotate the data.

In order to assess data quality the following information is required to be specified:

- 1) What are the models to be developed and their scientific objectives?
- 2) What data are needed to constrain these models?
- 3) What features of the data are required to constrain the parameters and to validate the models?
- 4) What is the feasibility of constraining these parameters from existing literature and data? How feasible is it to validate the models based on available literature and data?
- 5) What quality level (e.g. signal-to-noise ratio, stain completeness, input resistance, etc.) is required of the data to extract the required features?
- 6) Do feature extractors (analysis tools) exist or do they need to be developed? What are the data quality requirements of the feature extractors?
- 7) Do the data need to be combined or linked with other key data to be valuable? (For example, do the electrophysiological recordings need to come from exactly the same cell as the morphological data?) Is this feasible?
- 8) What are the curation requirements to make the data searchable and accessible by the model building pipelines?

The data curation process requires having the data to begin the process. Most data from the Ramp-Up Phase is only now being delivered. Once the data is received, the data curation can begin. The minimal metadata to register data is defined by HBP-CORE. Curating a histological

brain image (e.g. images of a brain labelled for anterograde connectivity projections) requires first registering the HBP-CORE metadata (e.g. specimen, contributor, protocols, brain region, data type, and the data location URI). Subsequent curation (e.g. registering to a brain atlas) requires manually identifying the slicing plane and angle in a template atlas (e.g. the Waxholm Rat Atlas) and manually aligning the image of the histological data and scaling it to fit with the template brain. This is a time-consuming process. The resulting metadata (e.g. the affine transformation necessary to map the data to the template brain) produced by this process is then stored in the KnowledgeGraph and used by the Atlas Viewers to overlay the atlas annotations with the histological data. If extracted features are required, the data must be evaluated for signal-to-noise ratio and ensure that the feature extractor (e.g. cell count algorithm) can successfully operate on the data. This will require iteration and validation with control datasets to ensure the ultimate quality.

### ***5.1.3 Registering and annotating data through a web interface (T5.1.4)***

The registration app is a web application allowing an authenticated user to submit experimental data (datasets and their corresponding metadata) to the NIP. The main metadata to provide for any data submission are defined in Minimal Information for Neuroscience Datasets ([MINDS](#)).

#### **General dataset metadata**

- Dataset title: information describing the dataset such as type of data, brain region, cell type, version. A good title is important in order to search and find a dataset using [Ksearch](#). Some examples of dataset titles: 'P40 mouse hippocampus oligodendrocyte single-cell transcriptomics', 'Subcellular signalling kinetic model of the striatal medium spiny neuron (MSNs)'.
- Dataset description: more detailed information about the data submitted, including type of data, brain region, cell type, if the data was annotated with a brain template or any other important information. This description can be used to find the dataset as well.
- Publications: PubMed and/or DOI identifiers of the publication that describe the generation of the dataset.
- Production date: the date by which the production of the dataset has been finished.
- License: the license of the dataset.
- Data categories: the broad categories (modalities) of the data, such as electrophysiological data or simulation data. It is possible to pick more than one category.

#### **Metadata about the sample**

It is possible to fill the following metadata regarding the sample: sex and age of the specimen used for the experiment, list of brain regions, comments.

#### **Contributors**

Information about agents that contribute to data generation can be provided along with their roles.

#### **Methods and protocol used to generate the data**

The app allows to describe dataset generation protocols and to list the measurement and analytical methods used.

### ***5.1.4 Data curation workflow***

Data contributors can submit their data in one of the three ways described below:

- 1) The Registration Application

- 2) Via the manual curation pipeline
- 3) Using the programmatic ingestion pipeline

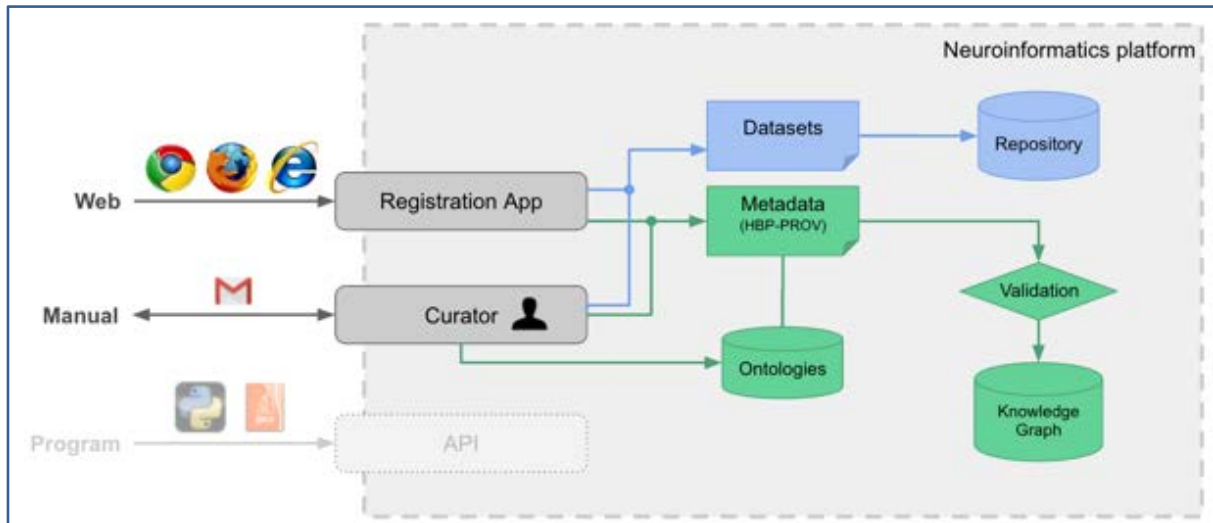


Figure 13: Data curation workflow

#### 1) Registration Application

If you have a few datasets, you can use the “[Registration Application](#)” to submit your data.

#### 2) Manual curation pipeline

If your data is very complex and it requires manual curation, please contact [data.nip@humanbrainproject.eu](mailto:data.nip@humanbrainproject.eu) to submit your data. We will contact you in order to better suit your needs for the integration of your datasets.

#### 3) Programmatic pipeline

The programmatic integration using an API will be possible by the end of 2016.

### 5.1.5 Data type definition registration

The [KnowledgeGraph](#) uses [HBP-PROV](#) to capture provenance of neuroscience datasets. It is an exchange data format intended to represent how data was created, which organism was used, how it was processed - including which contributor participated, which software was used - and what dataset/files resulted from it.

The structure contains causality information, time is regarded as crucial information in the provenance data.

It allows to answer questions such as:

- When was the data created?
- What are the characteristics of the animal this sample was derived from?
- How was the sampling organized, what protocols have been used? How does the workflow looks like?
- What brain region the image content belongs to? Which coordinates on an atlas does it have?
- How to read the files in this dataset?
- Which files have the same content and differ only in format?
- What analysis and transformations were applied to the data? What software has been used for that?



- Which organisations and people were involved in the creation of the dataset?

## 5.1.6 Data provenance format

Provenance is defined as a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data. In particular, the provenance of information is crucial in deciding whether information is to be trusted, how it should be integrated with other diverse information sources, and how to give credit to its originators when reusing it (source [W3C](#)).

The following figure depicts the current schema for a json that describes the registration activity, for a detailed information please go to section [HBP PROV format](#):

schema for a the json that describes the registration activity.		
registration	object	Registration activity, regrouping all activities described in file.
resources	array object	List of the data sources used in the registration. Those data sources are typically files or databases that represent the results of the observations.
activities	array[1..] object	List of registered activities.
agents	object	list of all agents associated with the activities described in the registration.
specimen	array[1..] object	An individual animal, plant, or single-celled life form.
samples	array[1..] object	Biological material.
datasets	array[1..] object	List of generated observations to be registered.
models	array[1..] object	In silico models, used to perform simulations and in silico experimentation.
classifications	array[1..] object	A list of classification terms derived from a manual or automated analysis.

Figure 14: HBP-PROV formats

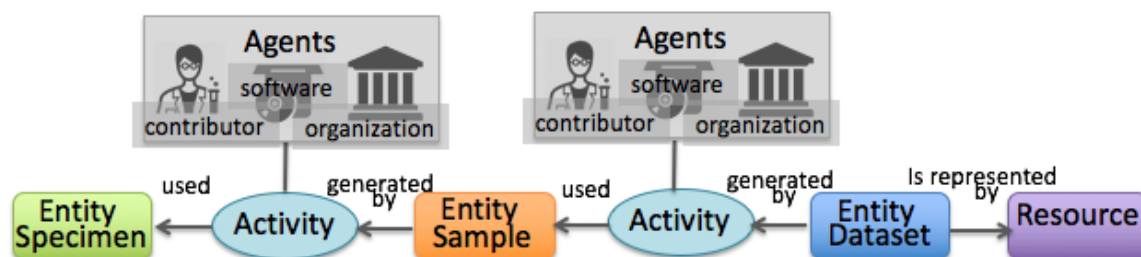


Figure 15: Simple example of provenance structure containing one dataset

### Data storage

The datasets submitted to the NIP are stored in several places:

- Web storage - storage accessible for web download on a dedicated web page indicated by the data provider
- Archival storage (long term) - at one of these storage places:
  - [Zenodo](#) (HBP community)
  - HBP document service
  - BBP GPFS

More details information can be find in our NIP website curation manual documentation <https://nip.humanbrainproject.eu/documentation/data-integration-manual.html>

## 5.2 Text mining services (T5.1.2)



The collaboration with OpenAIRE has focused on integrating the OpenAIRE Zenodo service with the NIP. Use cases for text mining of neuroscience literature for model parameters are provided to the OpenMINTED consortium to develop suitable text mining services. The intention has been to integrate the OpenMINTED infrastructure for text mining in the NIP. Workshops on vocabulary services, volumetric data services and overall data fabric for neuroscience data and knowledge have been held in collaboration with RDA. Discussions with EUDAT have examined how the B2SHARE service could be integrated as additional archival data services and how EUDAT may engage in active data repository services. The documentation about the collaboration with EGI is contained in an extensive report provided with the deliverables (SP5 D5.8.4 Resubmission Appendix I FINAL.pdf).

## 5.3 Unified search service

The Search client is an information retrieval system design to help neuroscientists, students, and users to find and download data registered in the NIP through free text and semantic tags queries.

The Search client is a front end application to the KSearch REST API and directly uses features it provides, albeit in a simplified and non-exhaustive way.

### Quick filters

The landing page of the Search client offers **2 levels of quick filters** to narrow down your search to the relevant subset of data: by species/order and then brain region. If you select a species with few datasets registered, you will not see the 2nd level of quick filters (brain region).

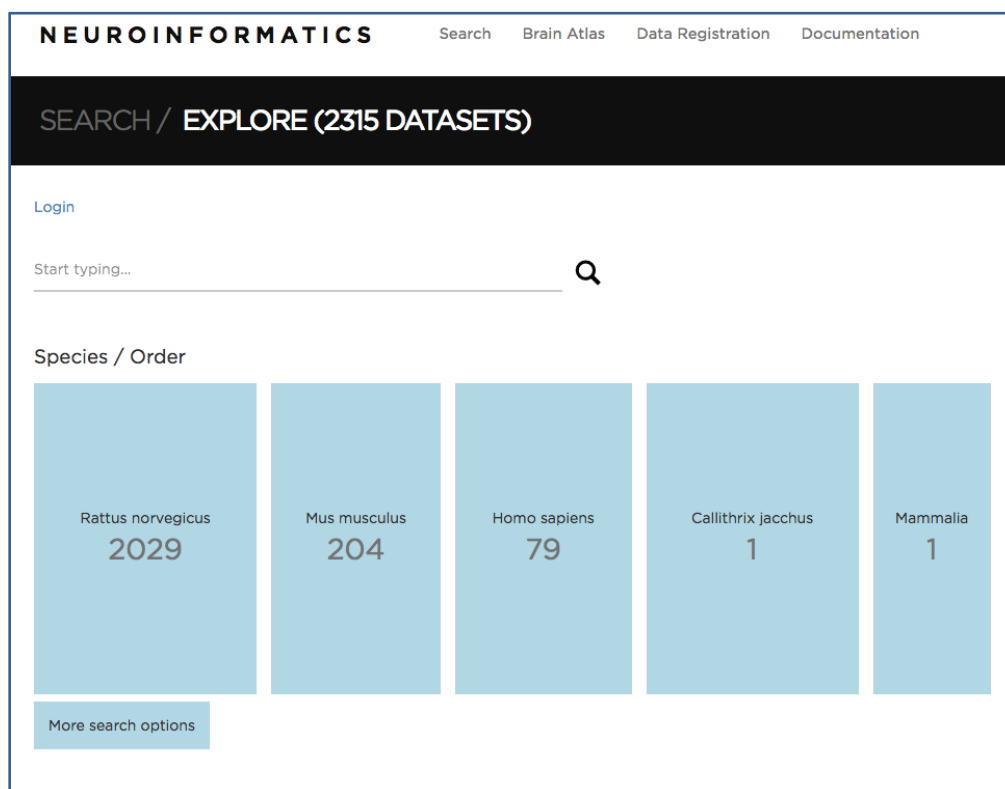


Figure 16: Landing page view of the Search API

Therefore, you are able to select for 'Rattus Norvegicus'. If you want to **bypass the quick filters**, you can click on the "More search options" button to show all datasets. You can also enter a free form text query to directly land to the search results for the query.

## Free form text query

At any time, you can **type a free form text query** in the input field at the top of the page, and press the Enter key or click on the looking glass icon to apply the query. Please note that all existing filters will be kept.

## Paginated list of results

The main screen consists in a list of paginated search results (10 results per page) with a list of filters on the left.

Bipolar cell	9	
<b>Brain regions</b>		
Hippocampus	9	<b>Single cell morphology reconstruction [P14 Wistar rat hippocampus CA1 pyramidal cell morphology]</b> <a href="#">Maurizio Pezzoli, Henry Markram, Olivier Hagens</a> Cell types: Hippocampus CA1 pyramidal cell
Somatosensory cortex hin...	181	
Somatosensory cortex hin...	356	
Somatosensory cortex hin...	247	
Somatosensory cortex hin...	320	
Somatosensory cortex hin...	143	
Field CA1, stratum radiatum	15	
Field CA1, stratum pyrami...	31	
Field CA1, stratum oriens	24	
Field CA1, stratum lacunos...	1	<b>Single cell morphology reconstruction [P16 Wistar rat hippocampus CA1 pyramidal cell morphology]</b> <a href="#">Maurizio Pezzoli, Henry Markram, Olivier Hagens</a> Cell types: Hippocampus CA1 pyramidal cell
Field CA1, pyramidal layer	12	
Field CA1	26	
<b>Protocols</b>		
Slice Staining for Morphol...	1249	
Slice Extraction for Electr...	1832	
Reconstruction of hippoc...	23	
Reconstruction of a rat hi...	18	<b>Single cell morphology reconstruction [P14 Wistar rat hippocampus CA1 pyramidal cell morphology]</b> <a href="#">Henry Markram, Olivier Hagens</a> Cell types: Hippocampus CA1 pyramidal cell
Reconstruction of a mous...	12	
Morphological Reconstruct...	1249	
Molecular Characterisatio...	454	
Labeling of Cells for Morp...	1703	
Generation of the distribu...	1	
Generation of pyramidal c...	1	
Generation of oligodendr...	1	
Generation of mural cell sl...	1	
Generation of mixed neur...	1	<b>Single cell morphology reconstruction [P14 Wistar rat hippocampus CA1 pyramidal cell morphology]</b> <a href="#">Henry Markram, Olivier Hagens</a> Cell types: Hippocampus CA1 pyramidal cell
Generation of microglia sl...	1	
Generation of interneuron...	1	
Generation of endothelial ...	1	
Generation of astrocyte sl...	1	
Generation of action pote...	3	
Generation of action pote...	3	
Generation of action pote...	3	
Generation of action pote...	10	
Generation of action pote...	7	

Figure 17: Sample of Search results

## Adding / removing filters

Filters are ordered by category. You can **add a filter** by clicking on it in the left sidebar. It will refresh the list of results and list of available filters. The active filters and categories are moved to the top of the sidebar. By clicking on the cross icon next to one of them, you can **disable the filter**.

To see detailed info about a given result, click on it. It will display more metadata along with a download link if relevant.

33	
33	<b>Single cell morphology reconstruction [VD110524_IDB.asc]</b> <a href="#">Henry Markram, Yun Wang, Meystre Julie, Jean-Vincent LeBe</a> Species / Ord... Rattus norvegicus Brain Location: Somatosensory cortex hindlimb layer V Cell types: Untufted pyramidal cell Protocols: Morphological Reconstruction of a Single Cell Slice Staining for Morphological Cell Reconstruction Labeling of Cells for Morphological Reconstruction Slice Extraction for Electrophysiology Experiments Download dataset
33	
33	
33	
7	
12	
33	<b>Single cell morphology reconstruction [FLUO15_LEFT.asc]</b> <a href="#">Henry Markram, Jean-Vincent LeBe, Ying Shi, Meystre Julie</a>
33	

Figure 18: Metadata visible on Search API results

You can put the cursor over a contributor's name to see the contributor's affiliations.

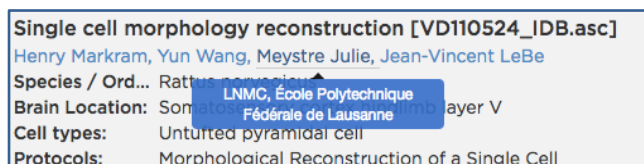


Figure 19: Contributors affiliation visibility on Search API

## Download dataset

Dataset download is restricted to HBP members. If you're not logged in yet, you will be prompted to do so. You can login at any time through the link at the top left of the page.

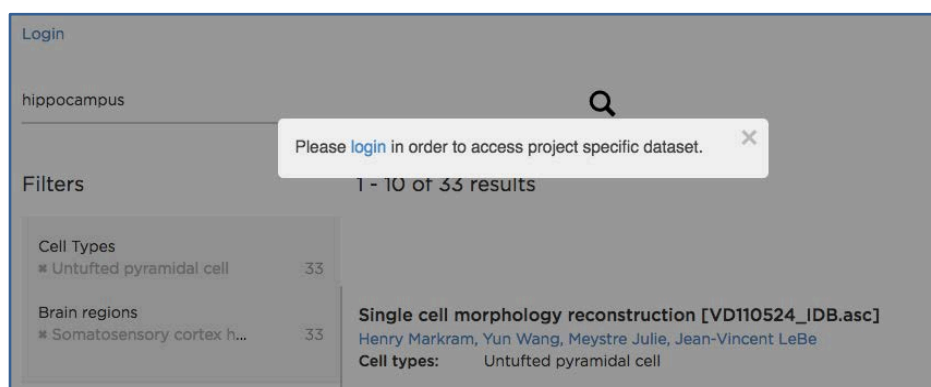


Figure 20: Restriction access to dataset on Search API

More details information can be find in our NIP website [user manual](https://collab.humanbrainproject.eu/#/collab/47/nav/5245) documentation (<https://collab.humanbrainproject.eu/#/collab/47/nav/5245>).

## 5.4 Tools for Brain Atlases: Functional Requirements

### 5.4.1 HBP-PROV: Minimal metadata specifications

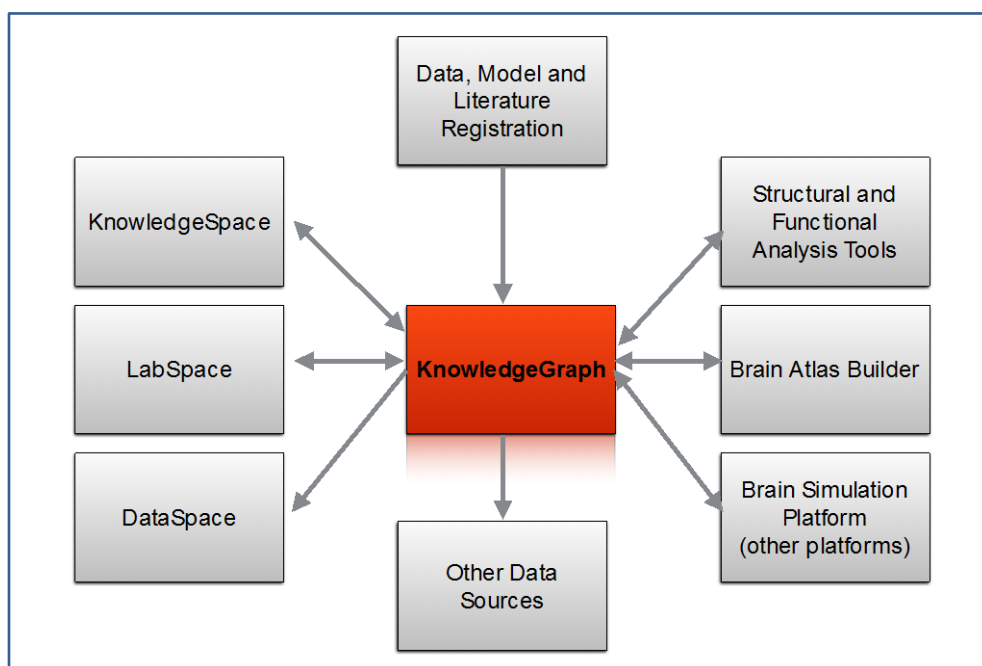


Figure 21: KnowledgeGraph, based on HBP-PROV metadata specification and data model

This serves as the common data store and provenance tracker for all types of data, models, and literature produced and consumed by other services, tools and platforms in HBP.

## 6. Data standards and data model (T5.1.4)

There are many areas of standardisation applicable in the NIP. These include standard APIs, data models, metadata, ontologies, data formats, etc. Establishing a broad community accepted standard is an extensive process requiring real use cases, technical specification, and tool support. Wherever possible, existing standards are adopted.

The NIP APIs are all built on established W3C REST-based standards. These use standard formats such as JSON and XML as serialisation formats. The core data model for the NIP, HBP-CORE is built on W3C PROV-O standards. The metadata are built on international standards from EBI, UBERON, OBO, NIF, etc.

Vocabulary service APIs are another area where SP5 is participating in standards activities. Stemming from another RDA workshop initiated by SP5 but merged with an existing RDA effort, discussions have been held on standardising APIs for vocabulary services. Currently, SP5 uses SciGraph, an open source vocabulary service also employed by the Neuroscience Information Framework.

SP5 is joining a consortium of existing neuroimaging infrastructures in Europe and around the world to integrate infrastructures and define standards for infrastructure interoperability, brain atlasing standards, APIs, data formats, repository standards, etc.

This work will integrate the Neuroimaging Data Models standard developed in the INCF Datasharing Taskforce (also based on W3C PROV) in order to enable search and meta-analysis of large imaging datasets.

In addition to the [Generic Dataset Metadata](#), we will add more information in this section about the data categories we currently support in the Platform and what our recommendations are with respect to metadata, data format and level of maturity.

### 6.1 Standard data formats

The NIP maintains documentation for recommended standard data formats for specific types of data, including volumetric image data, electrophysiological data, morphological reconstructions, etc. Many derived datasets are provided in Excel or CSV formatted text files. As part of the curation process the best data format for the specific use and access pattern is identified and the data converted to the appropriate format.

One standard data format employed by the NIP is BBIC (a multiresolution volumetric image container) used by the Volumetric Imaging Services and active data repositories. This is an open source specification made available via github. A Research Data Alliance (RDA) workshop was also organised by SP5 to examine establishing standards for accessing volumetric datasets. This led to ongoing evaluation of several existing APIs and imaging service technologies. Discussions have also been held with Google, which is providing its own API. SP5 is also working with the Array Database working group in the RDA to evaluate Array Databases, their performance and APIs.

SP5 also participates in the BigNeuron initiative (<http://bigneuron.org>) which is focused on sharing large-scale image data of stained neurons, using machine vision to extract neuron morphologies, sharing the neuron reconstructions and analysis of the resulting data. In the context of this group there is ongoing work to define a standard for neuron reconstruction formats. Currently SWC, Neurolucida and several HDF5 based formats are being evaluated.

The Neurodata without Borders (<https://crcns.org/NWB>) initiative brought together scientists from the Allen Institute, leading labs around the world, members of the INCF data sharing task

force and members of the HBP. The result is an HDF5-based electrophysiology data format that will be adopted. We are currently establishing the pipelines to convert Igor binary wave files to NWB.

In general, the data format is selected according to the specific requirements of the data itself, but more importantly the scientific use case and intended access patterns add important constraints to the data format selection.

The purpose of the KnowledgeGraph component is to give access to the provenance and to all data integrated in the NIP. It provides the means for scientists to browse data using their favorite web browser while allowing developers and software components to access the same data programmatically. In addition, for people working on the integration of data into the Platform, it gives tools to describe the provenance (HBP-PROV) as well as workflows to integrate both provenance and raw data into the Platform.

## 6.2 Specialised Data Access

While the KnowledgeGraph is a general purpose engine to record the provenance of scientific processes and their outcome, the NIP hosts very specialised data types (e.g. large volumetric datasets) that deserve specialised data access to render them useful to the Neuroscience community. Thus, the NIP has developed several tailored services to address specific needs that were expressed by the platform users.

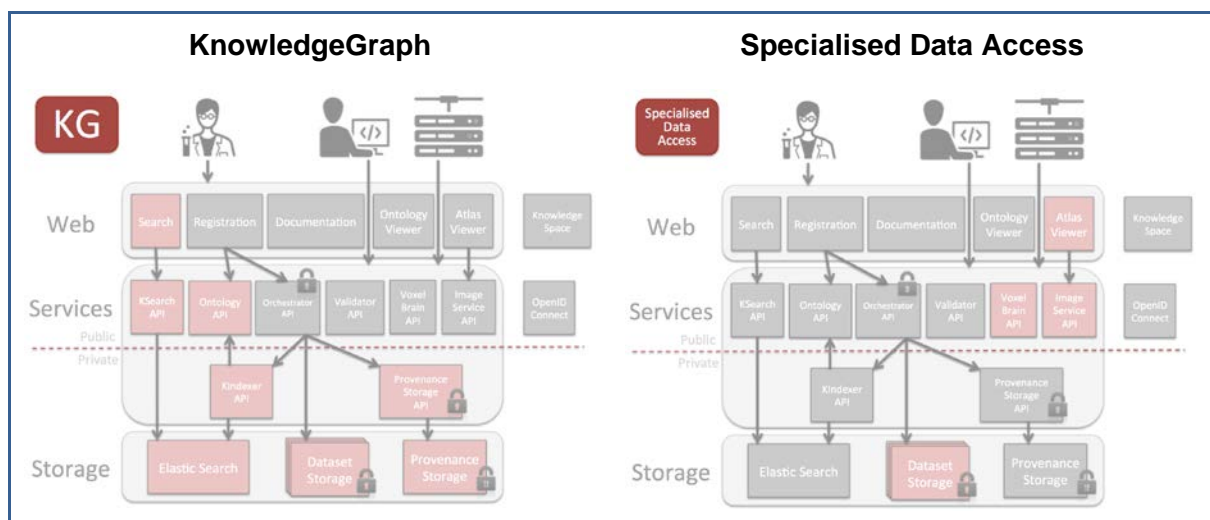


Figure 22: Classes of services provided by the NIP

### 6.2.3 Atlas

#### Specialised Metadata

In addition to the MINDS the following information is needed for the subsequent types of data:

#### Resolution and directions

The resolution is the measure of the sharpness of an image or of the fineness with which a device can produce or record such an image, usually expressed as the total number or density of pixels in the image. For brain it can be expressed in microns per pixel to millimeters per pixel.

The following values are recorded for resolution:

- Anterior\_posterior
- Superior\_inferior

- Left\_right resolution

The above resolution values represent the anisotropic image spacing of the raw data in a 3D volumetric context.

- Coronal
- Axial
- Sagittal resolution

These values represent the isotropic image spacing of the processed image data in each 2D plane.

We use the [attributes ontology](#) to add these values.

### Reference Atlas

If a specific reference atlas was used as a reference space for the aligning of the images, this should be captured. We currently have an ontology for [reference atlases](#).

Ex. Waxholm Space Rat Brain Atlas v.2.0.

### Data Formats

The current available formats are:

#### NIFTI

NIFTI-1 is a new Analyze-style data format, proposed by the NIFTI DFWG as a short-term measure to facilitate inter-operation of functional MRI data analysis software packages.

NIFTI-2 is a 64-bit update to the NIFTI format.

#### TIFF

Large volumetric data could be represented by the Tiff format in one orientation such as sagittal. Tagged Image File Format, abbreviated as TIFF or TIF, is a computer file format for storing raster graphics images.

#### NRRD

Nrrd ("nearly raw raster data") is a library and file format for the representation and processing of n-dimensional raster data. It is intended to support scientific visualisation and image processing applications.

## 6.2.4 Electrophysiology

### Specialised Metadata

In addition to the MINDS the following information is needed for the subsequent types of data:

#### Cell type

Electrical type of the cell being measured according to the experimenter, if possible using existing ontology terms.

#### Stimuli

- time step : acquisition time interval (inverse of sampling rate)
- time unit : second, millisecond.
- stimulus name : e.g. IDRest, Long Square, IRhyperpol, Pulse, Ramp, Spontaneous.
- stimulus description : stimulus start time, stimulus end time, duration
- data units : millivolts (mV), volts (V), picoampere (pA), ampere (A).



We use the [attributes ontology](#) to add these values.

## Data formats

The current data formats are:

**.dat**

This is a neurolucida file, a standard data format defined and maintained by MBF Bioscience.

**.abf**

A file format from Axon Instruments which supports both continuous data recording and episodic (computer controlled stimulus) recording.

**.ibw**

Igor binary waves file format generated by the Igor Pro software, it contains a single voltage or a current recording.

As a future development we will be transforming all the electrophysiology data into [NWB format](#).

### 6.2.5 Morphology

#### Specialised Metadata

In addition to the MINDS the following information is needed for the subsequent types of data:

#### Morphology cell type

Description of the cell morphology type according to cell ontology. e.g. Layer II/III pyramidal cell.

#### Data formats

**.asc**

File type provided by Neurolucida (ASCII file) which can contain single or multiple cell reconstructions.

**.swc**

File type provided by the AIB and Neuromorpho which could contain single or multiple cell reconstructions.

### 6.2.6 Receptor Density

#### Specialised Metadata

In addition to the MINDS the following information is needed for the subsequent types of data:

#### Receptor type

We use MeSH (Medical Subject Headings) National Library of Medicine's controlled vocabulary thesaurus to describe the receptor whose density has been measured. These terms have a hierarchical structure that permits searching at various levels of specificity.

The Receptor types associated to a dataset are annotated as attributes using the [attributes ontology](#).

#### Data formats

**TIFF**

Tagged Image File Format, abbreviated TIFF or TIF, is a computer file format for storing raster graphics images.



### 6.2.7 Transcriptomics

#### Specialised Metadata

In addition to the MINDS the following information is needed for the subsequent types of data:

#### Public accession numbers or identifiers

We recommend the use of public accession numbers or identifiers as well their version. e.g. [CAA29860.1](#)

#### Data formats

.cef

Cell Expression Format files are human-readable, tab-delimited text files that can be easily parsed and generated from any scripting language, designed to simplify the exchange and manipulation of very large-scale transcriptomics data, particularly from single-cell RNA-seq.

.raw

Free text format.

### 6.2.8 Kinetic model

#### Data formats

sbml+xml

The Systems Biology Markup Language ([SBML](#)) is a machine-readable exchange format for computational models of biological processes. Its strength is in representing phenomena at the scale of biochemical reactions, but it is not limited to this.

More detailed information can be find in our NIP website curation manual documentation (<https://collab.humanbrainproject.eu/#/collab/47/nav/5245>).

## 7. DataSpace - Data federation of repositories (T5.1.1)

Although data accessible from any URI can be registered, the NIP uses two key types of data repository deployments: archival and active. Archival repositories serve for the deposition and archival of all type of data in large, stable long term archives. Active data repositories usually serve a specific type of data (e.g. volumetric data or time series data) and combine computational resources with storage to enable data type specific services.

### 7.1 Archival data repositories

Zenodo ([www.zenodo.org](http://www.zenodo.org)) is the primary *archival data repository* used by the NIP. It has been developed as part of the OpenAirePlus project (<https://www.openaire.eu>) and is operated by the data services of CERN using the extensive data storage infrastructure of CERN to provide long-term storage, with replication and tape backup of large quantities of data. The storage in Zenodo is provided to HBP for free and the organisation has agreed that up to 1PB of data can be stored by the HBP at no cost. This is sufficient for depositing virtually all deliverables (including documents, presentations, etc) in the HBP and accepting community contributions. In addition, Zenodo is integrated with the European Commission reporting system and has a dedicated HBP community area.

#### DATA Repository service (ZENODO)

Our repository service is based on Based on Zenodo. It builds and operates a simple and innovative service that enables researchers, scientists, EU projects and institutions to share, preserve and showcase multidisciplinary research results (data and publications).

## Metadata

Category	Service
Maintainers	Zenodo is developed by CERN under the EU FP7 project OpenAIREplus (grant agreement no. 283595).
Homepage	<a href="https://zenodo.org/">https://zenodo.org/</a>
Documentation	<a href="https://zenodo.org/features">https://zenodo.org/features</a>
Support	<a href="https://zenodo.org/contact">https://zenodo.org/contact</a>
Source Code	<a href="https://zenodo.org/dev">https://zenodo.org/dev</a>
License	Missing License
Current Version	REST API — includes support for uploading your research outputs. OAI-PMH API — allows you to harvest all or parts of Zenodo via the OAI-PMH protocol.
All Versions	The REST API is versioned. We strive not to make backward incompatible changes to the API, but if we do, we release a new version. Changes to the API are documented on this page, and advance notification is given on our Twitter account.

## Description

Zenodo enables researchers, scientists, EU projects and institutions to:

- easily share the “long tail” of small research results in a wide variety of formats including text, spreadsheets, audio, video, and images across all fields of science.
- display their research results and receive credit by making the research results citable and integrating them into existing reporting lines to funding agencies such as the European Commission.
- easily access and reuse shared research results.

Deliverables:

- An open digital repository for everyone and everything not served by a dedicated service; the so called “long tail” of research results.
- A modern look and feel in line with current trends in state-of-the-art online services.
- Integration with OpenAIRE infrastructure and assured inclusion in OpenAIRE corpus.
- Easy upload and semi-automatic metadata completion by communication with existing online services such as DropBox for upload, Mendeley/ORCID/CrossRef/OpenAIRE for upload and pre-filling metadata.

- Easy access to research results via an innovative viewing option, open APIs, integration with existing online services, and the preservation of community independent data formats.
- A safe and trusted service by combining community based curation with short- and long-term archival and digital preservation strategies in accordance with best practices.
- Persistent identifiers, Digital Object Identifiers (DOIs), for sharing research results.
- Service hosting according to industry best practices in CERN's professional data centres.

An easy way to link research results with other results and products, funding sources, institutions, and licenses.

## 7.2 Active data repositories

Active data repositories combine computing services with storage resources. By providing computational resources with high speed access to large data stores, active data repositories enable interactive visualisation, feature extraction and analysis to be performed on the data without requiring the data to be moved between multiple locations. Moving data remains a relatively slow and costly endeavour and should be minimised. Rather, the approach here is to bring the compute to the data, rather than moving the data to the compute. Key datasets that require interactive visualisation (e.g. volumetric brain scans, large EEG datasets, electron microscopy data) will be replicated to active data repositories to be made available to users of the NIP.

The operation of active data repositories requires data centres with the appropriate resources and willingness to operate the resources with public (or authenticated) web-based access for multiple simultaneous users.

Starting in April 2015, SP5 began working closely with the European Grid Infrastructure (EGI) to establish a scalable approach to deploying active data repositories. Active data repositories can be operated anywhere in the world and are the basis of ongoing discussions (in the RDA) for establishing international standards on serving large-scale volumetric data. Support for additional types of data (e.g. time series) will be developed in the future. SP5 and EGI have developed use cases, test scripts, success criteria, sample data and services (packaged as a Docker container). The wiki page describing the history and use cases are available here at the following link: <https://wiki.egi.eu/wiki/HBP#Reference>.

The current status is that at least two active data repositories have been successfully deployed, tested and load-balanced for up to 20 simultaneous users on two member infrastructures of EGI (DataOne and dCache). EGI is now developing a service level agreement with SP5 to provide a scalable approach to deploying active data repositories throughout Europe. A distributed approach to deploying data repositories with a diversity of infrastructures is valuable as it ensures redundancy, performance and engagement with multiple interested parties.

A report on the work done in collaboration with EGI is attached as Appendix I.

The template brain atlases will be archived (if necessary) on Zenodo. They will be deployed on active data repositories operated by EGI, EPFL or SP7 according to size and performance requirements. Strategic data repositories will be established as required in different European member states. Some repositories may be data-type specific active data repositories, for example to serve electron microscopy data, while others may be more general archival repositories for long-term storage of all member state data. Specifically, we have identified that specialised data sets such as the Human Brain Polarized Light Imaging datasets, which have been imaged at very high resolution (~1µm voxel size), will be housed and analysed by the Jülich Supercomputing Center, while other active active and archival data repositories will be established at CINECA, for example, to accommodate the data produced at LENS by Francesco

PAVONE. Other data repositories will be established according to the data and analysis requirements using SP7 and EGI resources.

Currently all community contributed data and most HBP data are earmarked to be deposited in the Zenodo repository in the HBP community repository (the only exception is large datasets such as the human PLI dataset that will be held in specialised repositories and backed up from there). Data is determined to be either open to the public or private to the HBP consortium. Large datasets that require active data repository services will be replicated to appropriate repositories during the curation process (described below).

The active data repository strategy is a key element of establishing a globally distributed federation of data repositories, ultimately supporting diverse data types, visualisation, analysis, machine vision, machine learning and other data mining approaches.

### **7.3 KnowledgeGraph - Database built on HBP-PROV data model and ontologies (T5.1.3)**

The KnowledgeGraph lies at the heart of the NIP. It is a database in which the key metadata for all data artefacts (biological or from simulations) can be registered. It is built around a data model called HBP-CORE, which itself is based on the W3C provenance standard PROV-O. Building this graph around a provenance standard provides the foundation to track all data operations, the related data sets (e.g. a subset of this whole brain dataset was analysed and resulted in these statistics), the specific tools and algorithms, and attributions to all contributing individuals or organisations.

A web-based data registration application is used to register the key metadata for each dataset (and upload them to the Zenodo repository). HBP-CORE captures the minimal metadata necessary to usefully search and integrate datasets: Specimen details (species, age, etc); Contributor (PI, Lab, individual); Method protocols (sample prep, labelling technique, recording method, etc.); Data type (category of data and binary format); and URI (a hyperlink to the data access point).

The metadata is validated upon entry against the Ontology service and the data URL is confirmed during data curation.

The KnowledgeGraph is indexed for search using Elasticsearch which is a standard, high-performance and highly-scalable open source search engine. Elasticsearch is currently used by many very large search applications and can be extended easily by adding additional compute nodes. The system is designed so that the graph can be readily migrated and ingested into new database and graph search technologies as they emerge.

The KnowledgeGraph enables search across all the key dimensions provided by the standardised metadata in the HBP-CORE data model. A graphical search interface as well as an API is provided to enable searching.

## **8. KnowledgeSpace - Semantic wiki for community ontology maintenance, data curation and living review articles (T5.1.5, T5.1.3)**

### **8.1 Knowledge Space**

KnowledgeSpace (KS), previously Brainpedia (5.1.5, and 5.1.3), is a community-based encyclopaedia that provides living review articles, literature, neuroscience ontologies and the

links between neuroscience concepts and a diversity of federated data sources. KS builds on a vocabulary service, populated with an integrated set of neuroscience ontologies with initial content coming from the Neuroscience Lexicon (NeuroLex), and the Brain Architecture Management System (BAMS). It links to an expanding set of data sources through the Neuroscience Information Framework (NIF) federated search infrastructure. Integrated data resources include HBP KnowledgeGraph, Neuromorpho.org, Allen cell types morphologies, Allen cell types electrophysiology, Neuroelectro.org, Cell Image Library, NIF Integrated Connectivity, ChannelPedia.net, Ion Channel Genealogy, Blue Brain Neocortical Microcircuit, ModelDB, OpenSourceBrain, and GenSat. This list will be extended in an ongoing manner.

## Metadata

Category	Resource
Maintainers	The KnowledgeSpace consortium ( <a href="http://www.knowledge-space.org">www.knowledge-space.org</a> )
Contributors	Sean HILL, Maryann MARTONE, Tom GILLESPIE, Mathew ABRAMS, Linda LANYON, Jeff GRETHE, Willy WONG, Catherine ZWAHLEN, Renaud RICHARDET
Homepage	<a href="https://www.knowledge-space.org">https://www.knowledge-space.org</a>
Documentation	<a href="https://www.knowledge-space.org/NeuroKS/index.php/Documentation">https://www.knowledge-space.org/NeuroKS/index.php/Documentation</a> <a href="https://nlp.humanbrainproject.eu/documentation/user-manual.html#knowledgespace">https://nlp.humanbrainproject.eu/documentation/user-manual.html#knowledgespace</a>
Source Code	<a href="https://github.com/NeuroscienceKnowledgeSpace/KnowledgeSpace">https://github.com/NeuroscienceKnowledgeSpace/KnowledgeSpace</a>
License	CC-by-4.0
Current Version	First public release

All Versions

## Description

KnowledgeSpace (KS) is a community-based encyclopaedia for neuroscience, which links a common vocabulary for neuroscience concepts with data, models and literature available from sources around the world. KS will allow neuroscience community contributions and will support data curation, annotation, and text mining. KS is the result of recommendations from a community workshop held by the INCF Program on Ontologies of Neural Structures in 2012 and include the report attached below to community workshop. It is an open project and welcomes participation and contributions from members of the global research community.

[2012 INCF Workshop Report](#)

In the discovery and advancement of scientific knowledge, the iterative cycle of hypothesis and experiment is well established. In modern science, this cycle and the testing of the results by disparate laboratories often results in large amounts of diverse data providing evidence to support or contradict the definition of concepts developed in the hypothesis. This data is often very diverse, distributed and challenging to discover and integrate. This very fragmentation poses a challenge for assessing the state of knowledge in modern neuroscience. How much evidence is there to support the definition of a concept, such as a neuron type, a brain region or a connection pathway? The advancement of the field relies on an accurate assessment of the concrete evidence available to support the definition of a concept.

With the amount of data referenced in the literature in the public domain increasing, we rely on automated agents such as search engines to find relevant information. The availability of these search engines technically means that the fragmentation of knowledge should be decreasing. However, despite improved algorithms for natural language processing and text mining, it is still technically difficult to find data relevant to a specific concept.

Establishing a common vocabulary for neuroscience (organized as taxonomies and ontologies) is essential to even the earliest stages of evaluating the extent to which a concept is supported by evidence. In our data-driven world, there are those that might claim that this common vocabulary is not necessary, but humans communicate through concepts and they provide a very compact and understandable way for us to apply human knowledge to data.

Historically, there has been resistance in the biomedical community, particularly the neuroscience community, to use common vocabularies for expressing our concepts, e.g., brain regions, cell types. Researchers generally perform their research within a cultural context that prefers a specific vocabulary. Neuroscience also is still very much arguing about fundamental concepts such as what constitutes a brain region or cell type and how many there are.

A misconception in the neuroscience community is that we have to agree on a set of common terms and their definition in order to move forward. We argue, however, that the need for common vocabularies is not so that we can arrive at consensus definitions, but precisely so we expose our concepts to the full power of big data. In addition, tracking and integrating all equivalent concepts as synonyms is an essential part of data integration.

The use of common, machine-processable vocabulary allows us to integrate datasets linked with common concepts, but perhaps of most importance, it allows us to compare *assertions* about data to the data themselves. In other words, by making data linked to a concept findable, we can evaluate the degree to which the data supports the definition of a single, well developed concept. Modern methods in machine learning and statistical analysis provide techniques for evaluating multidimensional datasets and the extent to which they form single classes - and the features or properties that best define those classes.

The KS was developed to address these aspects of organising and disseminating neuroscience data and knowledge. The aim of the KS is to provide a unique, global interface between current scientific concepts and the data and analyses that support or weaken their definition. The KnowledgeSpace provides a link between neuroscience concepts and a federated search of neuroscience data, models and literature.

The KS has been developed in the context of an international collaboration to develop standards for neuroscience vocabularies and federated search of diverse neuroscience repositories. This collaboration includes the International Neuroinformatics Coordinating Facility ([www.incf.org](http://www.incf.org)) and the Neuroscience Information Framework (<http://neuinfo.org>). The participation of these two key organisations is key to engaging the broader neuroscience community in establishing consensus standards for ontologies and data curation.

The current KnowledgeSpace integrated data sources are:

- Neurolex



- Allen Brain Institute cell-types
- Neuroelectro.org
- Cell Image Library
- NIF Integrated Connectivity
- Channelpedia.net
- Ion Channel Genealogy
- Neuromorpho.org
- ModelDB
- OpenSourceBrain
- GenSat
- Pubmed
- and more to come.

The KnowledgeSpace is built upon many existing internationally standard ontologies including UBERON, NIF, OBO, etc. HBP ontologies are also integrated.

All HBP ontologies are publicly accessible in github:  
<https://github.com/NeuroscienceKnowledgeSpace/methodsOntology/tree/master/ttl>.

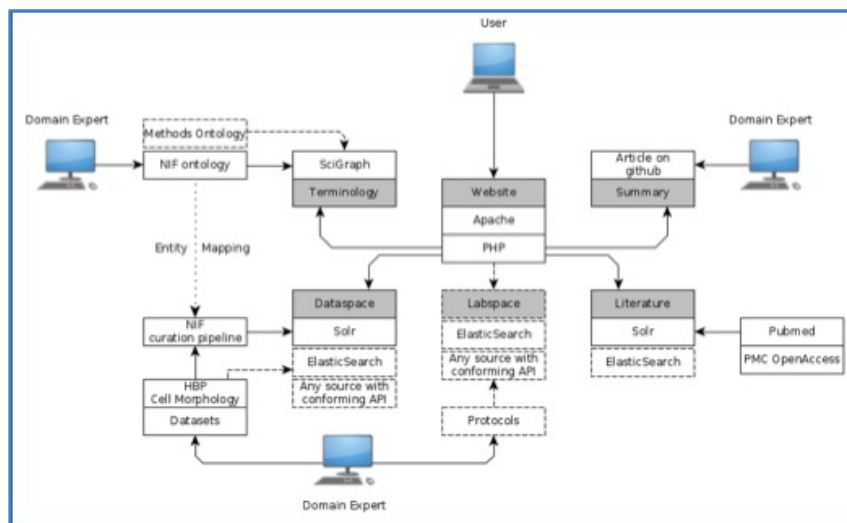
This currently includes ontologies for:

- File types
- Methods
- Data properties
- Roles
- Scales
- Sex
- Storage
- Taxonomy
- Allen Brain regions
- Waxholm brain regions
- Human brain regions
- Data Modality
- Data Type
- Cell Types
- Activities
- Molecular complexes

The KnowledgeSpace is developed in the context of an international collaboration to develop global standards for neuroscience vocabularies and federated searching of diverse neuroscience repositories. This collaboration includes the [International Neuroinformatics Coordinating Facility](#)



(INCF) and the [Neuroscience Information Framework](#). The KnowledgeSpace development is coordinated by the INCF at the Karolinska Institutet (KI).



**Figure 23: KnowledgeSpace collaboration developed with INCF**

The website is developed in PHP. All ontologies are loaded in a Neo4j database using [SciGraph](#). Domain experts contribute to the knowledge about concepts by contributing articles to a [GitHub repository](#). Link to data tagged by NIF ontology concepts are curated and added to the Data space component for indexing. All indexes are built with the [Apache Solr](#) framework. The ontologies available in KnowledgeSpace are published in [GitHub](#).

### Project management

KI has been and continues to be the central coordinator and project manager of all KS development and groups of people active in the project, including the following:

- KI has been central in bringing together and coordinating the efforts of the Neuroscience Information Framework (based at UCSD, USA) and Blue Brain Project / Human Brain Project (EPFL, Switzerland).
- KI led the work to draft the KS governance model and coordinate community inputs (using the INCF network)
- KI organises and documents the KS governing board meetings
- KI sits on the governing board in a *de facto* chair role that has responsibility for setting the agenda for meetings, and oversight of the project, ensuring appropriate use of resources and expenditure of funds including the European funding for HBP. The KI project manager for KS reports on project progress to the KS governing board.
- The KI project manager organises, facilitates, and documents the weekly meetings of the KS working group (a consortium of developers, ontologists, and neuroscientists committed to the development of KS). The working group is open to the neuroscience community at large.
- KI coordinated the gathering of user requirements and selection of use cases that led to the development of the KnowledgeSpace alpha prototype.
- KI recruited test users, as well as coordinated the design, implementation, and evaluation of community testing.
- KI coordinated the development of a neuroscience methods ontology (v.1) by facilitating the efforts of the Neuroscience Information Framework (NIF) and the Blue Brain Project. The methods ontology is available at: [github.com/OpenKnowledgeSpace/methodsOntology](https://github.com/OpenKnowledgeSpace/methodsOntology)

- A KI semantic technologist worked on *ontologies* for the KS prototype.
- KI has led outreach activities to get additional data sources into KS. KI has gained a commitment from the Neurodata without Borders Neurophysiology Initiative (funded by Kavli Foundation) to connect its data sources to KS.

*Deliverables: neuroscience methods ontology (v.1), user feedback to guide next round of development, new data sources to include in the core KS, KS working group, and KS governance model (v.1).*

### **Development**

KI has expanded its initial JAVA-based proof-of-concept prototype to a PHP-based, alpha prototype. The alpha prototype consists of the core KS with its initial data sources as well as the software required to render “living” pages about core neuroscience concepts. KS pages currently integrate information about neuroscience concepts terminologies (i.e. NeuroLex) with dynamic information from the data sources in KS. KI has completed concept and entity mappings for the source data contained in the core KS. The alpha prototype is currently available to the public ([Knowledge-Space.org](http://Knowledge-Space.org)).

### **Documentation**

KI, in association with its partners at NIF, has maintained a github repository ([github.com/OpenKnowledgeSpace/KnowledgeSpace](https://github.com/OpenKnowledgeSpace/KnowledgeSpace)) where code, milestones, governance model, and use case information are stored in an effort to maximise transparency and encourage collaboration. In addition to the github repository, KI also records weekly meeting minutes and posts them to the KS website, as well as the KS working group space on Trellis<sup>1</sup>.

*Deliverables: KS github repository, KS working group space on Trellis, and KS code, milestones, and governance model (v.1).*

### **Future Plans**

The beta prototype of KnowledgeSpace will be released in November 2016 at the Society for Neuroscience meeting in San Diego.

### **Current KS groups**

#### **KS Governing Board**

KI collaborates with the Neuroscience Information Framework and Human Brain Project to deliver the initial prototype of KnowledgeSpace. Current members: Linda LANYON (KI INCF Secretariat) Maryann MARTONE (UCSD) and Sean HILL (EPFL), *KS working group members*: Mathew ABRAMS (KI) *Project Manager*, Tom GILLESPIE (KI), Jeff GRETHE (University of California, San Diego), Willy WONG (UCSD), Silvia JIMENEZ (EPFL), Samuel KERRIEN (EPFL), Francois SY (EPFL).

## **8.2 LabSpace - Semantic wiki for organising and curating summaries of individual laboratory data sets**

Ksearch is the search component of the NIP. It is a REST API allowing [searching](#) of [curated](#) datasets using different filters that are mainly taken from [MINDS](#).

### **Features**

---

<sup>1</sup> Trellis is a new, digital communication and collaboration platform from the American Association for the Advancement of Science (AAAS). The KS groups are within the portfolio of the INCF on Trellis.

Ksearch allows searching of datasets shared in NIP. There are five main features:

- Full text search:
  - free text search on all dataset fields: all datasets fields will be searched,
  - field free text search: specific dataset fields can be targeted when search for a term,
- Boolean search:
  - Boolean operator (AND, OR,...) can be specified between the keywords of a text
- Filtering: a set of filters are available to narrow down search results
- Query expansion through term synonyms and specialisation: for example a search for “Rodentia” will be able to retrieve datasets annotated with “Rattus Norvegicus” because “Rattus Norvegicus” is a rodent
- Document aggregation: Ksearch results can be grouped by data modality, brain regions by other metadata available in [MINDS](#). This feature enables document exploration and faceted navigation.

Details of how the previous Ksearch features are implemented in term of endpoints are described [here](#).

### **8.2.1 Automated data monitoring pipeline**

Similar to Section 4.

### **8.2.2 Data access API**

Although the general search APIs are available to all types of users, specialised APIs may be needed for specific use cases.

To support SP6 modeling use cases, several APIs have been developed to support specific modelling applications. These APIs enable queries for specific morphological and electrophysiological data. The morphological queries allow searching for morphologies identified with a specific morphological type ontology (MType, which specified 55 morphological classes of neurons). The electrophysiological queries support retrieving electrophysiological traces according to an EType (11 electrical types of neurons), stimulus type and sweep number (multiple stimuli sequences). These APIs are used by SP6 to support the Single Cell Modelling Pipeline.

An additional API has been developed to support brain region modelling: the VoxelAPI. This allows for querying and accessing data and properties according to specific voxel locations with a volumetric brain atlas. This is used by SP6 for the Brain Region Building Pipeline.

### **8.2.3 Data analysis and data mining module**

For text mining, SP5 is participating in an international consortium of text mining infrastructures (OpenMINTED) and working closely to ensure the text mining capabilities of this infrastructure serve the needs of the NIP in HBP. Specifically, the neuroscience use cases are mining the literature for neuron types and properties, synapse types and properties and brain connectivity statements.

### **8.2.4 Tools for Brain Atlases: Software**

The [Multiresolution Atlas Viewer](#) allows you to easily visualise brain imaging datasets coming from several species, mapped to several brain atlases spaces. This interactive context allows you to specifically target a region of the brain in the 3 axes (sagittal, coronal and axial) rapidly. The

Multiresolution Atlas Viewer sits upon the KSearch and Image Service REST APIs to discover and display the brain imaging datasets currently registered in the NIP.

## 8.2.5 Tools for Brain Atlases: Physical Architecture

Neuroscience requires accepted maps, terminology, coordinate systems, and reference spaces in order to perform accurate and effective analysis and communication within the field and to allied disciplines. A brain atlas allows capture of such physical characteristics and could potentially overlay an infinite set of features in a fashion analogous to a geographic atlas such as google map. These features might include cytoarchitecture, chemoarchitecture, connectivity, behavior functions, metabolic rates or pathological information, etc.

### Requirement

A brain atlas in HBP is envisioned to be **visualisable**, e.g. for viewing brain parcellation; and **quantitative**, e.g. retrieve a specific set of deterministic or probabilistic responses from one certain region. Through a user-defined query, these responses can be structural identification, receptor density, clinical information, bibliography, or related datasets.

## 8.2.6 Integrating existing atlas

To integrate an existing atlas submitted by the data provider, several processing steps are required as shown in the schema below. First, it needs to be stored on the GPFS server following certain [rules](#). Secondly, it will be preprocessed to remove the imaging artifacts and resampled to isotropic. [Ontology](#) of the atlas will then be parsed and converted into Turtle format to integrate into Ontology service. Finally, it will be ingested into the [Voxel Brain](#) and converted to BBIC format to be visualised in the [Atlas viewer](#).

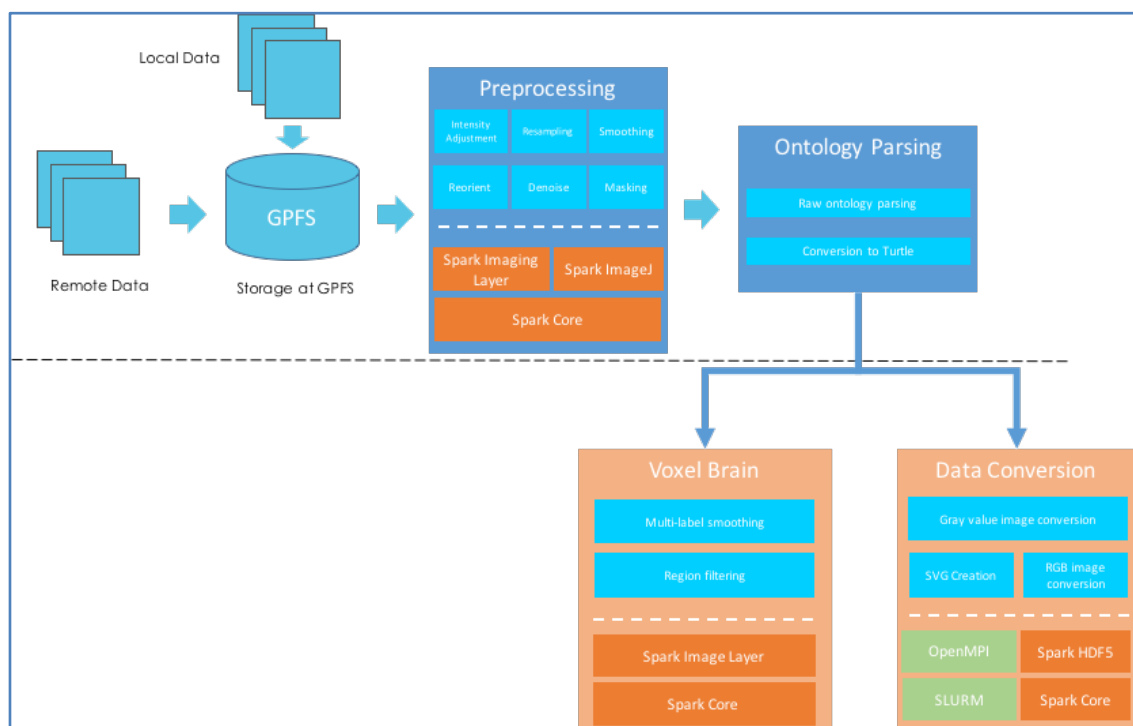


Figure 24: Schema of integrating existing atlas

## 8.2.7 Mapping new image data to existing atlas

To integrate a new image data submitted by the data provider, several processing steps are required as shown in the schema below. First, it needs to be stored on the GPFS server following

certain [rules](#). Secondly, it will be preprocessed to remove the imaging artifacts and resampled to isotropic. Spatial registration is then applied to transform the subject image into the atlas space. Finally, it will be ingested into the [Voxel Brain](#) and converted to BBIC format to be visualised in the [Atlas viewer](#).

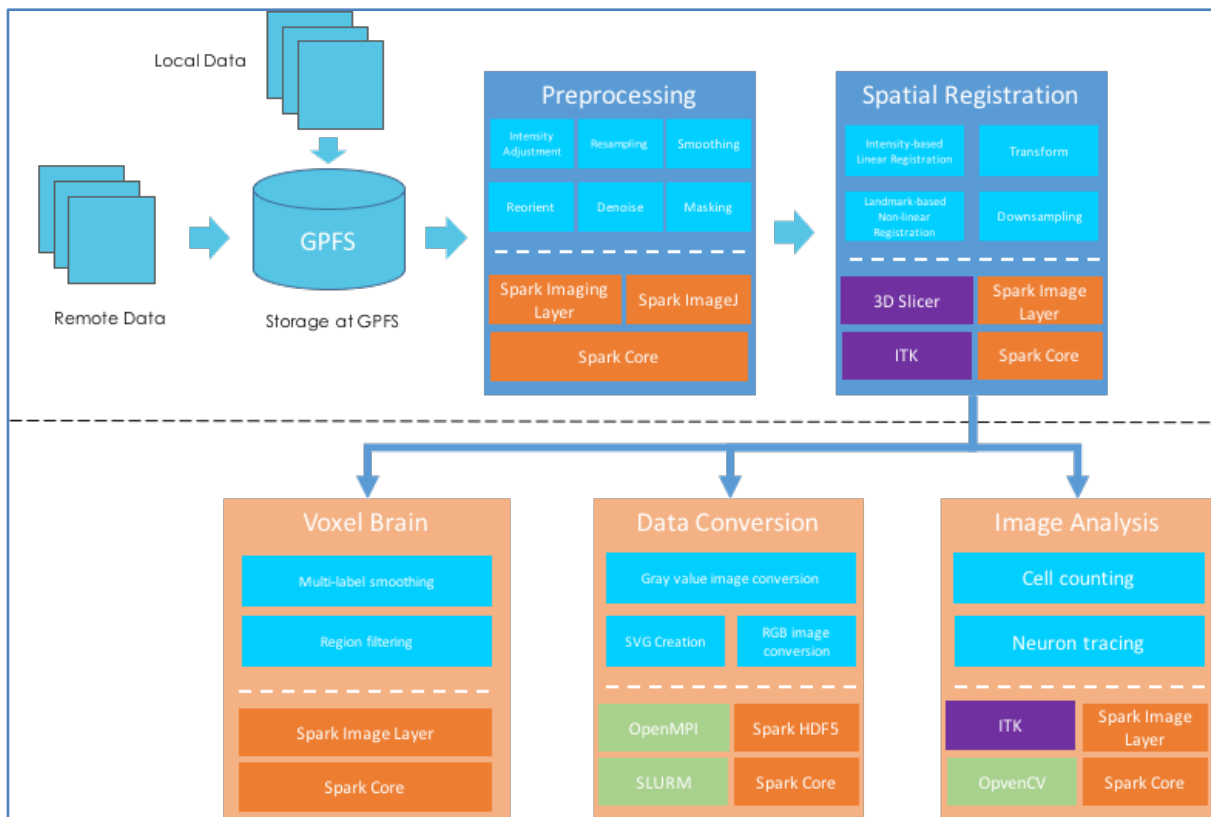


Figure 25: Pipeline of integration of new image data

### 8.2.8 Data processing modules

The above pipelines of integrating image data use the processing modules that are individually developed in-house. The tools are written in Python and C++. The Python tools aim at simple image operations which require less computation and memory usage. Algorithms that need high-performance computing power are developed in C++ which can be run in both standalone desktop and cluster.

#### Data storage

All the image data is stored at proj39 on GPFS. The data are classified and stored into two directories: **templates** and **stacks**, representing atlases and subject images. Both directories are organised per species. In stacks, data is further categorised into the atlas space it maps to. The subdirectory then contains:

- raw: raw data fetched from the data provider
- preprocessing: preprocessed data
- bbic: data converted into BBIC format for visualisation in the atlas viewer

A typical data storage structure is shown as follow:

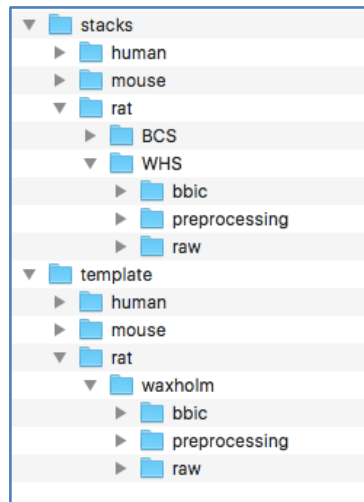


Figure 26: Data storage structure

### *Data preprocessing*

Raw image data received from the data contributor usually does not conform with the requirement for further processing, analysis and visualisation, therefore data preprocessing needs to be applied.

### **Intensity correction**

Many microscopic imaging modalities suffer from the problem of intensity inhomogeneity due to uneven illumination or camera nonlinearity, known as shading artifacts. A typical example of this is the unwanted seam when stitching images to obtain a whole slide image. Elimination of shading plays an essential role for subsequent image processing such as image registration, cell counting and visualisation.

### **Isotropic Resampling**

The atlas viewer treats the image data as isotropic at each orientation. For anisotropic data, especially microscopic images which have much larger inter-slice spacing than in-plane spacing, the image plane whose orientation differs from the original scan would be stretched or squeezed in the viewer.

### **Thresholding**

Due to potential errors in image acquisition, the raw image file occasionally would have extreme hyper or hypo intensity values. In the histogram, these intensities are far from the correct intensity distribution, and therefore can be easily removed from a simple threshold algorithm. The thresholding operation is used to change or identify pixel values based on specifying one or more values (called the threshold value).

User can perform the thresholding using Threshold.py, which is implemented based on the SimpleITK library. This filter is used to transform an image into a binary image by changing the pixel values according to the rule.

### **Intensity rescaling**

Due to various imaging acquisition methods, the images usually have different intensity distributions, resulting in pixel value types ranging from unsigned 8-bit to signed 32-bit. However, the web-based Atlas viewer only handles standard 8-bit images; therefore, all the intensity values need to be rescaled to 0-255.

### **Ontology parsing**



Atlases usually comprise terminologies of brain regions or parcels. These terminologies are defined as [ontology](#), which needs to be submitted to a [Github](#) repository.

An ontology parse tool is developed to convert the raw ontology file in various formats (e.g. xml, csv, etc.) to a Turtle file (<https://www.w3.org/TeamSubmission/turtle/>). Essentially, several fields are extracted including urls of prefix and base.

## Image registration

Image registration is the process of determining the spatial transform that maps points from one image to homologous points on an object in the second image. In the context of brain atlasing, one can register the subject image to an atlas space to allow further analysis of high-dimensional voxel representations in an identical space.

Two sequential registration steps are used: first, a linear registration to globally map the subject data to the atlas space; secondly, a deformable registration that warps the fine structure of the brain to the atlas.

3D Slicer (<https://www.slicer.org/>), which contains a rich extension of registration tools and libraries, is used to annotate anatomical landmarks and perform linear registration. Non-linear warping is performed by a standalone TPS transform tool.

## Linear registration

Affine registration is used to capture the global transformation between the subject image and the atlas image. It represents a rigid 3D transformation followed by a perspective projection.

## Deformable registration

Deformable registration is a process consisting of establishing functional or spatial correspondences between two images. The term deformable (as opposed to linear or global) is often used to denote the fact that the observed signals are associated through a non-linear dense transformation, or a spatially varying deformation model.

## Data conversion for atlas viewer

The size of processed data can be terabytes, which therefore cannot be loaded into the memory for visualisation in the Atlas Viewer. To achieve this, the processed data is converted to an in-house data format called BBIC, which is a HDF5 file containing multi-resolution of image tiles. A more detailed description of the format can be found [here](#).

## Image conversion

Image volume is converted into a hierarchical multi-resolution fashion and stored in a BBIC format. The depth of the hierarchy is computed as  $\log(N)$  where  $N$  is the number of pixels in one dimension. At each depth, the image is downsampled into resolution of  $N/\log(N)$  and divided into image tiles of given size. These tiles are then stored in a HDF5 file together with the meta information of the image.

## Annotation conversion

Each brain imaging dataset comprises of sectional images and brain region annotations. Depending on the dataset, brain regions could be already in SVG format. However, some datasets have accompanied annotations as label for each voxel or pixel. In this case, we need to convert annotations for each section of specific orientation into SVG layer in order to display in Multiresolution Atlas Viewer.

This conversion means tracing boundary of each brain regions on raster image and save in vector format such as SVG. For this operation, we run python script `SvgCreate.py` and `SvgCreateFunctions.py` locally. These scripts were developed in-house.

There are two use cases.

- 1) Annotation volume could be in nrrd or NIfTI format.
- 2) Annotation for each image section of specific orientation is in PNG, TIFF or JPEG. We will have three folders for all orientations.

### 8.2.9 Data ingestion to voxel brain

An integrated atlas or an image that mapped into an atlas space could be ingested into the [Voxel Brain](#). This will allow the Voxel Brain API to provide a high dimensional representation of the brain regions based on user queries. The data ingestion includes a process of generating the brain regions according to the input ontology hierarchy and potentially apply smoothing on the data.

#### Brain region generation

The raw atlas usually stays in a different format, such as text file, image stack or volumetric data. To ingest into the voxel brain, the raw image is first converted into nrrd format (<http://teem.sourceforge.net/nrrd/format.html>), which is the default data format in the Voxel Brain. Then, a brain region hierarchy based on the input ontology is generated. Finally, each brain region defined in this hierarchy is extracted from the input image data and stored as an individual data named with the brain region id.

```
{
  "id": 0,
  "name": "CA",
  "children": [
    {
      "id": 100,
      "name": "CA1",
      "acronym": "",
      "children": [
        {
          "id": 103,
          "name": "CA1a",
          "acronym": "",
          "children": [
            { "id": 1, "name": "SLM", "children": [], "acronym": "" },
            { "id": 8, "name": "SR", "children": [], "acronym": "" },
            { "id": 15, "name": "SP", "children": [], "acronym": "" },
            { "id": 22, "name": "SO", "children": [], "acronym": "" }
          ]
        },
        {
          "id": 104,
          "name": "CA1b",
          "acronym": "",
          "children": [
            { "id": 2, "name": "SLM", "children": [], "acronym": "" },
            { "id": 9, "name": "SR", "children": [], "acronym": "" },
            { "id": 16, "name": "SP", "children": [], "acronym": "" },
            { "id": 23, "name": "SO", "children": [], "acronym": "" }
          ]
        }
      ]
    }
  ]
}
```

Figure 27: An example of hierarchical structure of the hippocampus layers in json format.

## Brain region smoothing

Many atlases are annotated on a 2D slice basis. While smooth boundaries of the brain regions can be guaranteed in the annotation plane, unsmoothed borders often occur in the other image orientations since it is difficult for the annotator to coordinate the annotation between slices from a 2D view. This unsmoothed border results in discontinuity or spiky shape of the brain region, leading to sub-optimal outcome of the further analysis and simulations.

Instead of smoothing all the labels simultaneously, which is a mathematically and computationally challenging problem, we first smooth the label individually using the morphological operator and median filter. Then, we merge the smoothed labels into one space. Nearest neighbor strategy is employed to regularise the conflicting voxels as well as the gap voxels in the space.

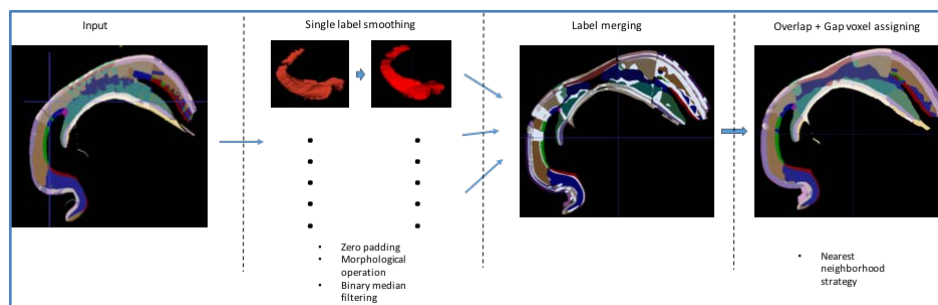


Figure 28: Schema of multiple brain region smoothing

### 8.2.10 Tools for Brain Atlases: Prerequisites

Light integration vs. deep integration reflects both the requirements presented by a scientific question and the effort necessary to integrate datasets into specific brain atlases.

Light integration is achieved by registering data using HBP-CORE metadata to the KnowledgeGraph. When this is complete, the dataset is searchable according to species, age, contributor, protocols, brain location, data types, file format, etc. If the brain location metadata is provided in a standard atlas coordinate or semantic reference system (Allen Mouse, Waxholm Rat or Allen Human) then the data can be effectively considered fully integrated in the NIP and atlases. This integration still requires a curation effort to ensure correct metadata and extension of the ontologies as appropriate to the dataset.

Deep integration only applies to some datasets that require precise integration in order to answer specific scientific questions. An example of this is co-registering multimodal whole brain datasets including fMRI, MRI, Nissl stained, polarised light imaging and receptor maps in order to answer specific questions about brain connectivity and function. This work requires careful planning of the scientific question, the hypothesis, the required measurements, the precision necessary, the landmarks to align the data across modalities, etc. Deep integration simply cannot be performed or would be meaningless if it is performed in an *ad hoc* fashion without careful planning and quality control in the original acquisition of the data. Also, many scientific questions do not require deep integration - often it is sufficient to know more general properties of cell densities, electrophysiological properties, connectivity measurements, etc. in order to build a model.

## 8.3 Brain Atlas Builder (T5.1.6)

## 8.3.1 Brain Atlas Builder: Overall Goals

Brain atlases consist of a template brain dataset which provides the spatial coordinates, brain parcellations which define brain region/structure volume boundaries, and an ontology which provides a structured naming system for the brain parcellations. Datasets can be integrated with these atlases both semantically (by being tagged with a brain region name), spatially (by assigning specific x,y,z coordinates), and aligned to the specific brain template by specifying a transformation (linear or nonlinear) to map the dataset to the brain atlas coordinate system.

Anchoring data to brain atlases is supported through the brain atlas building tools provided by AlignII and QuickNII. These tools provide the user with the ability to align image datasets to standard brain atlas templates and generate the metadata necessary to transform the data (through affine transformations) to the template space. During the Ramp-Up Phase the tools have only focused on linear transformations but additional registration approaches, including nonlinear mesh warping, will be evaluated for adoption during SGA1.

Current brain atlases in the Neuroinformatics Collaboratory include:

### Multiresolution atlases:

- Allen Mouse Common Coordinate Framework (Mus musculus C57BL/6, p56)
- Allen Mouse v2 (Mus musculus C57BL/6, p56)
- Waxholm Rat (Rattus norvegicus, Sprague-Dawley, p80)
- Golgi-Cox labeled Rat (Rattus norvegicus, Sprague-Dawley, p70)
- R602 Rat Projections (Rattus norvegicus, Sprague-Dawley, p80)
- BigBrain Human (Homo sapiens, age 65)
- JuBrain Colin27 (Homo sapiens, adult)
- Infant Human (Homo sapiens, 1 year)

### Scalable brain atlases:

- Waxholm Mouse (Mus musculus C57BL/6, p56)
- Allen Mouse (Mus musculus C57BL/6, p56)
- Waxholm Rat (Rattus norvegicus, Sprague-Dawley, p80)
- MRH Atlas of Developing Rat (Rattus norvegicus, P80)
- In vivo MRI Template Rat (Rattus norvegicus)
- DTI Rat Atlas (Rattus norvegicus, P72)
- Population-averaged DTI Rat (Rattus norvegicus, Sprague-Dawley)
- Calabrese-Paxinos Macaque (Macaca mulatta)
- Paxinos Macaque (Macaca mulatta)
- NeuroMaps Macaque (Macaca mulatta)
- Markov Macaque (Macaca fascicularis)
- Gray Short-tailed Opossum (Monodelphis domestica)
- Paxinos-Watson Marmoset Atlas (Callithrix jacchus)
- JuBrain Human Cytoarchitectonic (Homo sapiens, adult)
- Harvard Oxford Human (Homo sapiens, adult)

- Neuromorphometrics, Inc Human (Homo sapiens adult)
- SRI24/LBPA40 Multichannel Human (Homo sapiens, adult)
- Brodmann 05 Conte69 Human (Homo sapiens, adult)
- BigBrain Human 400µm (Homo sapiens, age 65)

### 8.3.2 Components

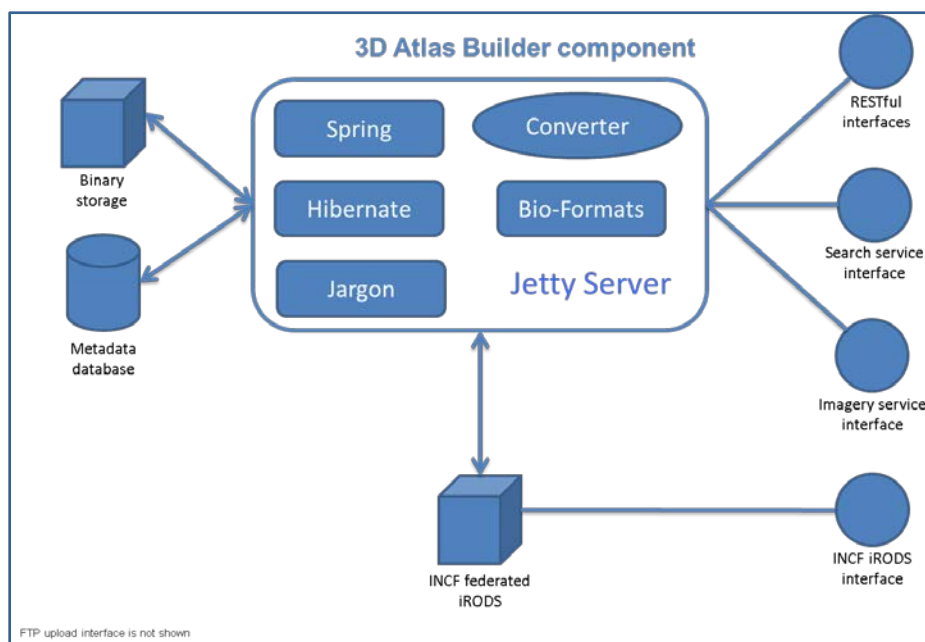


Figure 29: Brain Atlas Builder

## 9. Tools for Structural Data Analysis (WP5.2 and WP5.4)

### 9.1 Structural Tools: EMDigest (T5.2.1)

Discussed below.

### 9.2 Structural Tools: EspINA (T5.2.2)

#### 9.2.1 EspINA

EspINA is a user-friendly tool which performs segmentation and analysis of elements in a reconstructed 3D volume of the cerebral cortex, and greatly facilitates and accelerates these processes. It allows visualization and segmentation of large image stacks datasets, both from electron microscopy (e.g. FIB/SEM) and light microscopy.

#### Metadata

Category

Application

Tags

image-segmentation

	microscopy analysis
Partners	Universidad Politécnica de Madrid École Polytechnique Fédérale de Lausanne
Maintainers	<a href="#">Felix de las POZAS</a>
Contributors	Jose María (Chema) PEÑA Felix DE LAS POZAS Jorge PEÑA Pol MONSO Carlos BECKER Juan MORALES Pablo Marquez NEILA, Antonio LATORRE and Pascal FUA
Homepage	<a href="http://cajalbbp.cesvima.upm.es/espina/">http://cajalbbp.cesvima.upm.es/espina/</a>
Documentation	<a href="https://www.youtube.com/channel/UCN3kLTMxaJXkEJrvsJbx-ww">https://www.youtube.com/channel/UCN3kLTMxaJXkEJrvsJbx-ww</a>
Support	<a href="https://plus.google.com/115332585861057402895/posts">https://plus.google.com/115332585861057402895/posts</a>
Source Code	<a href="https://launchpad.net/~cbbp/+archive/ubuntu/main">https://launchpad.net/~cbbp/+archive/ubuntu/main</a>
Source Code	<a href="https://launchpad.net/~cbbp/+archive/ubuntu/main">https://launchpad.net/~cbbp/+archive/ubuntu/main</a>
License	GPL v3
Current Version	2.1.2
All Versions	1 and 2.1.2

## Description

### *Explore*

**Natural 3D navigation over the orthogonal planes.** Fluent navigation through the different sections of the stack using three planar views. Simultaneous different channel visualisation can



be performed in a customised staining and intensity/contrast properties. Navigation bookmarks helps to easily find stack spots.

## *Segment*

**Brain structures can be individually segmented to later analysis.** Identify as many elements as you want using our classification or expand it as needed. Reconstruct them using our manual or semi-automated tools.

To this, end we have introduced a new method for synapse detection that leverages available 2D or 3D context information to provide high-quality segmentations. For mitochondria, we used a similar approach combined with structured prediction to enforce common geometric constraints, such as connectivity, and relationships, such as being surrounded by a membrane.

In parallel, we have been working on Active Learning techniques to reduce the amount of effort required to train the Statistical Machine Learning algorithms they rely on. We have presented a novel approach to Active Learning that is geared towards segmenting 3D image volumes and also applicable to ordinary 2D images. By design, it takes into account geometric constraints to which regions should obey and makes the annotation process more user-friendly.

## *Edit*

**Extensive toolkit to improve segmentations.** If the result of the segmentation is not completely satisfactory, we provide a set of tools to enhance their morphology automatically or manually.

## *Visualise*

**Different 2D and 3D representations.** Geometrical features of segmented elements are relevant to their function. Different 3D visualisations provide detailed views for every segmented element.

## *Analyse*

**Obtain multiple measures from segmented structures.** The final step of the working session is the extraction of measurements for each object segmented. A list of morphometric features and parameters can be selected to be easily exported to standard .csv or .xls spreadsheet format.

## **Functionality**

- Image stack navigation.
- Multi-channel visualisation.
- Metadata information: Acquisition protocol, specimen, region, provenance information, etc.
- Manual, assisted and automatic segmentation of structures.
- Editing and refinement tools.
- Visualisation in 2D and 3D representations.
- Analysis and reporting of segmented objects.
- Plugin-based extensibility.

## **Technical aspects**

### *Development*

EspINA is implemented in C++ as a standalone client-based tool. It required a series of external libraries:

- Qt 4.8
- Boost
- VTK (Visualization Toolkit)
- ITK (Insight Toolkit)

We developed an EspINA plug-in that integrates our methods for synapse and mitochondria segmentation. This plug-in was improved following the requests and recommendations of the UPM neuroscientists. We also upgraded it for two subsequent versions of EspINA that had a new, backward-incompatible API.

### 9.1.2 EM images segmentation (T5.2.2)

Development of new tools for automatic and semi-automatic segmentation of synapses and mitochondria in EM images.

#### Research activities undertaken

We have introduced a new method for synapse detection that leverages available 2D or 3D context information to provide high-quality segmentations. For mitochondria, we used a similar approach combined with structured prediction to enforce common geometric constraints (e.g., connectivity) and relationships (e.g., being surrounded by membrane) regarding mitochondria.

In parallel, we have been working on Active Learning techniques to reduce the amount of effort required to train the Statistical Machine Learning algorithms they rely on. We have presented a novel approach to Active Learning that is geared towards segmenting 3D image volumes and also applicable to ordinary 2D images. By design, it takes into account geometric constraints to which regions should obey and makes the annotation process convenient.

**Results of the collaboration (e.g. publication, new code, patent, etc). If there were no results in the ramp up, when will they be expected?**

We developed a plug-in for the EspINA platform integrating our methods for synapse and mitochondria segmentation. This plug-in was improved following the requests and recommendations of the UPM neuroscientists. We also upgraded it for two subsequent versions of EspINA that had a new, backward-incompatible API.

We gave keynote presentations at microscopy workshops associated to the CVPR and MICCAI conferences, which are top conferences in the fields of Computer Vision and Medical Imaging, respectively. In addition, one of our PhD students won the First HBP Science Writing Competition and gave a presentation at the 2015 HBP Summit.

We started a close collaboration with Prof. Fred HAMPRECHT from U. Heidelberg and worked with his group on porting our synapse and mitochondria plug-ins into Ilastik. The collaboration is currently ongoing to integrate more tools into this platform.

## 10. Predictive Neuroinformatics (T5.4.1)

### 10.1 Description

The overall goal of RUP task 5.4.1 was to provide a *first draft algorithm for the brain addressing system*, which was also the deliverable. T5.4.1 was therefore primarily comprised of a study of an algorithm to build long-range projections at cellular resolution that is an alternative to the currently used algorithm within the Blue Brain Project/Human Brain project (described in a 2015 publication by Markram et al in Cell). A more detailed description of this algorithm, and the reasons for it being more powerful, can be found in a technical report. The

overall philosophy of the approach was outlined in a Current Opinion in Neurobiology publication [Tiesinga et al., 2015]. The T5.4.1 team also provided a Matlab implementation to demonstrate the algorithm, which was included in the NIP. Two additional tools were developed by the team. The first, the morphology viewer, aimed to facilitate running the Matlab implementation of the algorithm by using a python script on the NIP to import morphologies from internal sources (NIP) and external sources (Neuromorpho.org). Because this functionality was also more broadly useful within the HBP consortium, we decided to develop it as a standalone tool. The second tool was the incorporation of the Scalable Brain Atlas (SBA, developed by INCF) into the NIP and updating of the atlases provided therein. This was not strictly part of the DoW for T5.4.1, but was performed in order to provide a pilot tool for the analysis to be conducted for T5.2.6 within SGA1, as the SBA is able to provide visualisations of, for instance, the relation between the mouse meso-connectome and the gene expression atlas provided by the Allen Institute for Brain Science. As the SBA makes these functionalities also available for use in the rodent (5.2) and human (5.3) atlasing work packages in SGA1, we decided to make this contribution explicit in the PPR. A description of the SBA functionalities was recently published in the journal Neuroinformatics [Bakker et al., 2015].

Tiesinga, P., Bakker, R., Hill, S. & Bjaalie, J.G. (2015). Feeding the human brain model. *Current Opinion in Neurobiology*, 32: 107-114.

Bakker, R., Tiesinga, P. & Kotter, R. (2015). The Scalable Brain Atlas: Instant Web-Based Access to Public Brain Atlases and Related Content. *Neuroinformatics*, 13: 353-366

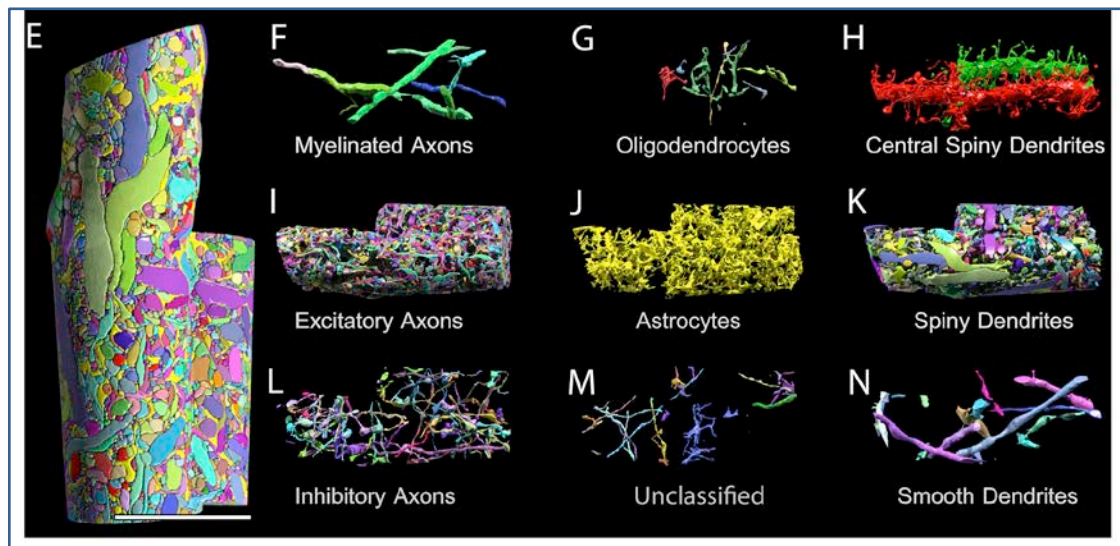
## 10.2 Overall goal and embedding in HBP.

The HBP aims to build the infrastructure necessary to simulate the entire human brain. To achieve this goal the connectivity at a cell-to-cell resolution needs to be constructed in way that is consistent with the morphology of the constituent neurons. Hence, the location of the synapses on the dendrite and axon needs to be specified in terms of the branch structure as well. There are multiple strategies for achieving this goal. One is to grow neurons by modelling developmental processes. Given the computational resources and the available information to constrain the simulation, this strategy does not seem feasible at the moment (Zubler & Douglas, 2009; van Ooyen, 2011). The strategy currently employed in the circuit builder tool (developed in SP6) of the consortium is to clone the morphology of 55 classes of reconstructed neurons and place the somas at the appropriate locations and with the appropriate densities (Markram *et al.*, 2015). In the next step, close appositions between axons and dendrites are determined as potential sites for synapses. This strategy leads to an excess number of synapses ( $6 \times 10^8$ ) that needs to be pruned by an order of magnitude in order to obtain the correct number of synapses ( $4 \times 10^7$ ) as well as the correct multiplicity of connections (i.e. the number of synapses per connection).

### 10.2.1 Current strategy

The currently used strategy is feasible as demonstrated by the validation in a recent publication (Markram *et al.*, 2015; Reimann *et al.*, 2015) and predicts a number of experimentally observed features. The aggregate statistics obtained from the appositions in the reconstructed morphologies match connectivity statistics sampled experimentally (Hill *et al.*, 2012). Nevertheless, the morphologies used for the reconstruction are obtained from different brains and therefore do not belong together. Furthermore, when placed in the simulation domain they may physically not fit together, because they have overlap with each other. The exquisite filling of space by the different elements of neural tissue is demonstrated in a recent saturated reconstruction based on electron microscopy data (Kasthuri *et al.*, 2015). It is difficult to pack the diversely shaped neural elements in box such that as little as possible of unused space

remains, when the elements do not belong to the same tissue. In addition, new investigations may uncover additional selection criteria that govern the matching of pre and postsynaptic partners into synapses that deviate from the version of Peters' principle that underlies the current strategy (Mishchenko *et al.*, 2010; Druckmann *et al.*, 2014).



**Figure 30: Data from a recent saturated reconstruction based on electron microscopy data.**

The figure shows (E) dense packing of (F-N) different neural elements that is difficult to reproduce using morphologies obtained from different brains without adapting these. Figure taken from (Kasthuri *et al.*, 2015).

Ultimately we aim to reconstruct networks that are capable of performing a particular computation relevant to the brain area in which the cells reside. The morphology of cells in different cortical areas, for instance sensory areas, are not that different, yet they give rise to circuits that perform different transformations of their afferent input and whose recurrent connectivity subserves different computations on the outcome of these transformations. It is therefore likely that the structure of the cell-to-cell connectivity is different than that of a network whose connectivity is based on random selection of a subset of close appositions.

The goal of the Task T5.4.1 was therefore to develop alternative strategies to construct the connectivity, with a special focus on the long-range projections, in a way that makes it easy to incorporate current and future experimental data from sources different than morphological reconstructions. We report on the implementation of our *first draft algorithm* and evaluation of this particular strategy that we have termed the reverse geometric approach. In addition, we look ahead to upcoming SGAs in order to formulate new strategies based on the first draft algorithm.

### 10.2.2 Alternate approaches from outside the HBP consortium.

Three other approaches are relevant to contrast to the reverse geometric approach. Oberlaender and co-workers divide the neural tissue domain into large voxels, with sides of 50  $\mu\text{m}$  and use cloned reconstructed morphologies to determine the number of pre or postsynaptic elements of each cell in each voxel and then pair its pre- and post-synaptic elements randomly (Oberlaender *et al.*, 2012; Egger *et al.*, 2014; Narayanan *et al.*, 2015). The disadvantage is that the synapse location so determined has a spatial uncertainty of 50  $\mu\text{m}$  relative to the putative location of the pre- and post-synaptic processes that gave rise to the synapse. The impact of this imprecision has not been fully evaluated, but the aggregate connectivity statistics as well as population dynamics of the model have been validated by experimental data. Van Pelt and coworkers developed simple yet realistic generative models for neurites (Koene *et al.*, 2009; van Pelt & van Ooyen, 2013; McAssey *et al.*, 2014; van Ooyen *et al.*, 2014; van Pelt *et al.*, 2014). These models were

used to generate a large number of axonal and dendritic trees representative of cultures, which were placed in the same volume to determine potential synapses, from the placement of which various connectivity statistics were determined. Luczak has proposed growing dendritic trees based on a diffusion-limited aggregation process (Luczak, 2006), which yields shapes that compare well to experimental morphologies.

*Overview of the reverse geometric approach.*

The **first** step is to represent a selection of reconstructed morphologies by a simple but flexible density model. This is done for both the axonal and dendritic trees. (Please refer to Figure 31 for a graphical representation of this step and the following steps in the procedure).

The **second** step is to place neurons with randomly sampled soma-center positions at the appropriate density in the neural tissue and represent each by a cloned version of the fitted density model.

The **third** step is to represent the neural tissue by small voxels (less than  $1 \mu\text{m}^3$ ) and sample for each cell the synaptic elements from the density. None of the voxels should be filled by more than one presynaptic or more than one postsynaptic candidate.

The **fourth** step is to pair pre\_ and post-synaptic elements in the same voxel and declare a synapse. From this, the connectivity matrix follows.

The **fifth** step is to take the synaptic elements for each neuron and construct a weighted minimum spanning tree to generate the morphology that is consistent with the synapse placement as well as the currently unpaired elements belonging to that neuron. The synaptic placement precedes the determination of the morphology rather than vice versa as for the other approaches reviewed in the preceding paragraphs, hence we refer to this procedure as the reverse geometric approach.



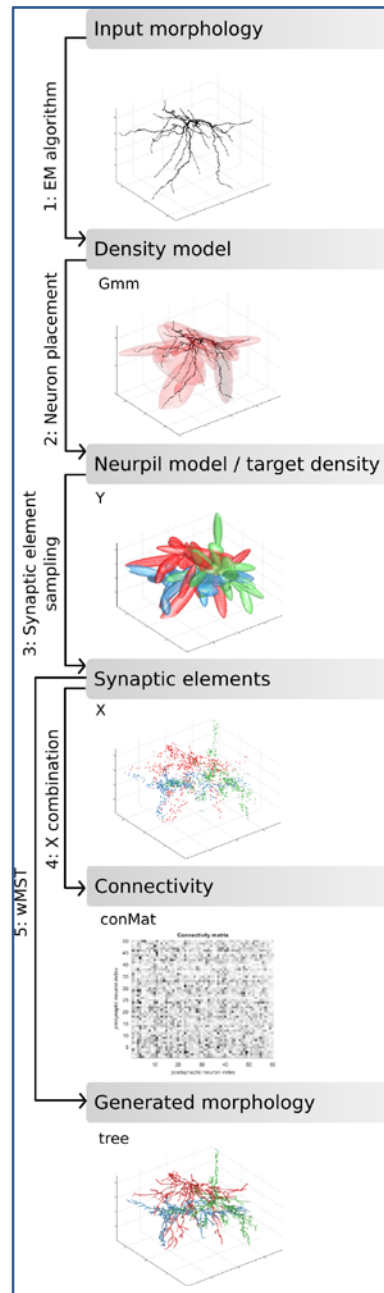


Figure 31: Graphical overview of the different steps in the reverse geometric approach.

(A) The input morphology is used to define (B) a synaptic element density model. (C) Somas are placed within the neural tissue and the density model is cloned with some variability and translated to each soma location. (D) Synaptic elements are sampled according to one of the possible procedures (see text). (E) When pre and postsynaptic elements are paired, a connectivity matrix can be determined and (F) the underlying morphology can be reconstituted.

### 10.2.3 Each of the steps in more detail.

**Step 1. Density models for neurites.** We take an experimentally reconstructed morphology and construct a density, from which we sample an adequate number of points. These points are then represented by a Gaussian mixture model using an expectation-maximisation procedure. We choose on the order of twenty Gaussians and allow the covariance matrix to vary freely, subject only to a minimum value for the diagonal elements of the variance matrix, which represents a regularisation to avoid Gaussian collapsing to a single point. The fits are not perfect and need to be improved. For this we will be exploring two strategies. First, to develop a new primitive that



is more appropriate for the elongated stick-like features of the neurites. Second, to develop an algorithm that uses spheres and cones as primitives, not to develop a representation for a specific neuron, but a filled density template that fits a population of neurons of the same type.

**Step 2. *Soma placement and cloning.*** We distribute the somas in the neural domain according to the densities determined experimentally. A density model is assigned to each soma, and cloned with the remaining free parameters drawn from the appropriate distribution. Currently, this is only a rotation across the long axis. For axons coming from neurons outside the directly modelled neural domain, we place seed points within the neural domain from which an axonal branch emerges that is restricted to the domain, rather than place the soma far away and model the initial part of the axon prior to entering the grey matter. A key future improvement to this step would involve a better model for variability of the neural density models, expressed in a small parameter set, based on experimental data stored in the Neuroinformatics platform as well as the neuromorpho online database (Parekh & Ascoli, 2013).

**Step 3. *Sampling of synaptic elements.*** For each type of synaptic element, pre- or post-synaptic, we independently sampled elements for each neuron, such that no voxel was occupied by more than one element of the same type. When sampled independently according to a Bernoulli process with probability  $p_{ij}\Delta V$ , where  $p_{ij}$  is the probability density in voxel  $i$  for a synaptic element of neuron  $j$  and  $\Delta V$  is the volume of the voxel, it is likely that the voxel is occupied by elements from more than one neuron. We have explored and evaluated multiple approaches to prevent this overlap. One can just remove the additional elements; however, as this introduces correlations between the element placement of different neurons, it consequently also introduces a bias in the density. By this we mean that the expected fraction of times an element is filled is different from  $p_{ij}\Delta V$  to an extent that depends on the aggregate element density of all the other neurons. More sophisticated methods also introduce bias, sometimes severe. We have implemented and evaluated various methods to sample in order to reduce bias. For instance, we have tried the Bernoulli scheme with different overlap corrections schemes, implemented an Ising model with an interaction term that reduces overlap, developed a LASSO-based approach to produce a sparse density of synaptic elements, and implemented a procedure wherein one element was placed according to probability  $\sum_j p_{ij}\Delta V$  and assigned to neuron  $j$  according to probability  $p_{ij} / \sum_j p_{ij}$ . This procedure is referred to as the synapse-selection rule (SS). The LASSO based approach was computationally too expensive and unstable for large system sizes. Only the SS rule was both computationally efficient and had near to the lowest bias. The bias arose when  $\sum_j p_{ij}\Delta V > 1$  that is when the expected number of elements in a voxel exceeded unity, which led to missing synaptic elements. In order to still obtain the appropriate number of synaptic elements per neuron, we redistributed the excess density. We used a simple iterative rule in which the excess probability was redistributed to the voxels where the sum was less than one. When this yielded other voxels where the expected number of elements exceeded one, we applied the procedure again. Usually this converged within a few iterations. This rule can be further improved to reduce the bias.

**Step 4. *Pairing of synaptic elements.*** The current procedure is simple - whenever a pre- and post-synaptic element are present in the same voxel, a synapse is placed. The procedure can be adapted to incorporate more biological detail. For instance, the synapse can be placed according to statistical rules that enforce a specific multiplicity distribution. In addition, one could assign a hidden variable to each neuron, and make the probability for a synapse dependent on the combination of hidden variables. This could generate connectivities that mimic feature-specific coupling. To achieve this specificity, more elements should be placed than desired, as some would be eliminated based on these additional rules.

**Step 5. *Regenerating the morphologies.*** Given the placement of synaptic elements for each neuron and/or the assigned synapses, the morphology can be regenerated by constructing the weighted minimum spanning tree as described in Cuntz *et al.*, 2010. When the procedure is internally consistent, the morphology of the cells that were used in the density fit should re-

emerge when reconstructed based on the sampled elements. In the current implementation we have found this not to be the case. We aim to resolve this issue by improving the primitives for the density model and reducing the bias in the sampling procedure as discussed in the preceding text.

#### 10.2.4 Use cases.

The tools described in this report are still in the pilot phase and are implemented in Matlab. The underlying algorithm is a first draft and the implementation needs to be improved to scale up better, as well as yield consistent morphologies as mentioned in the preceding text. We describe here a typical use case in which a researcher wants to build a long-range projection and compare the resulting statistics to experimental data. The other use case mentioned in the NIP specification document is comprised of evaluating the goodness of a projection obtained by other means, this involves running the latter half of the first use case with other data, and will not be explicitly illustrated here. We focus on reconstructing the thalamocortical projection from the ventral posteromedial nucleus in the thalamus to layer 4 of the barrel cortex, as most data is available for this case.

The first step is to identify the experimental data necessary for the prediction (note that the numbering of the different steps mentioned does not map one to one on the procedure outlined in the preceding text as we want to illustrate additional steps necessary for the use case). For this tool this would be the axonal termination of the projection neurons, for which we have a single reconstruction provided by Francisco CLASCA (SP1), as well as the dendritic morphologies of the target neurons, for which data is available on neuromorpho.org as well as stored in the NIP. In the initial pilot run of this use case we have chosen to use the latter source of data. In order to select and store the data in an appropriate format, we call up the Morphology Viewer from the NIP. This service is used in conjunction with a Python notebook to inspect morphologies, select them and produce a zip file with the properly processed and selected morphologies.

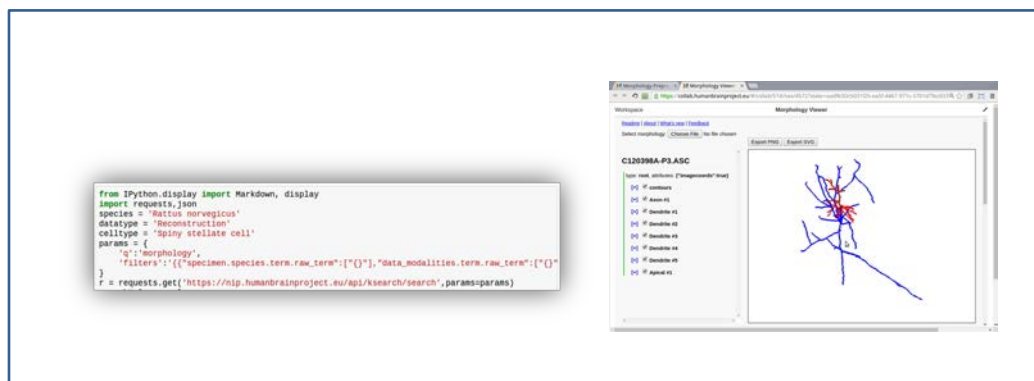


Figure 32: Morphologies are accessed, evaluated and selected with the morphology viewer.

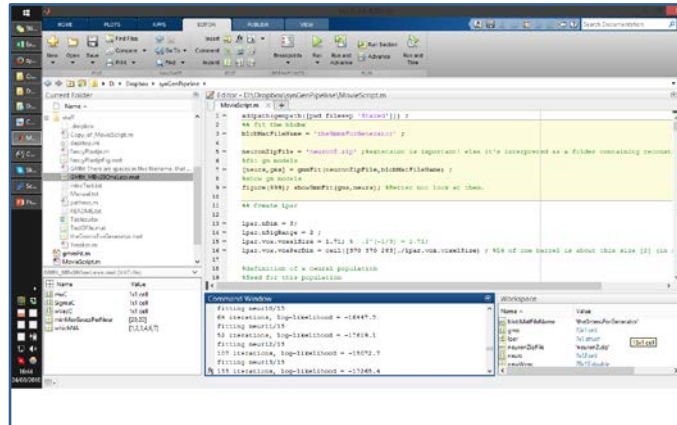


Figure 33: The morphologies are transformed into Gaussian mixture models appropriate for the SynapseGenerator Pipeline.

The second step is to process the selected morphologies into Gaussian mixture models. This is achieved by calling the matlab function `gmmFit`, which performs sanity checks on the morphology files, resamples the morphology and applies an expectation maximisation procedure. The function outputs a model specification that can be inspected by `showGmmFit`. The user can control the parameters for resampling procedure as well as the number of Gaussian blobs used in the fit and the regularisation of their shape based on the outcome of the visual inspection afforded by `showGmmFit` as well as the goodness of fit statistics.

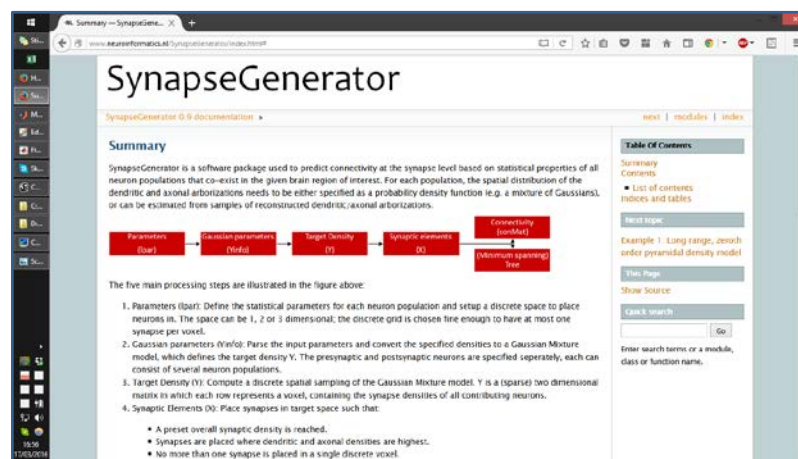
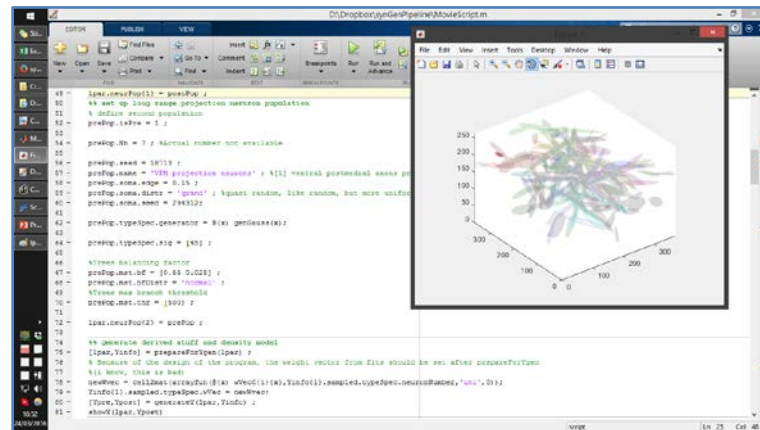


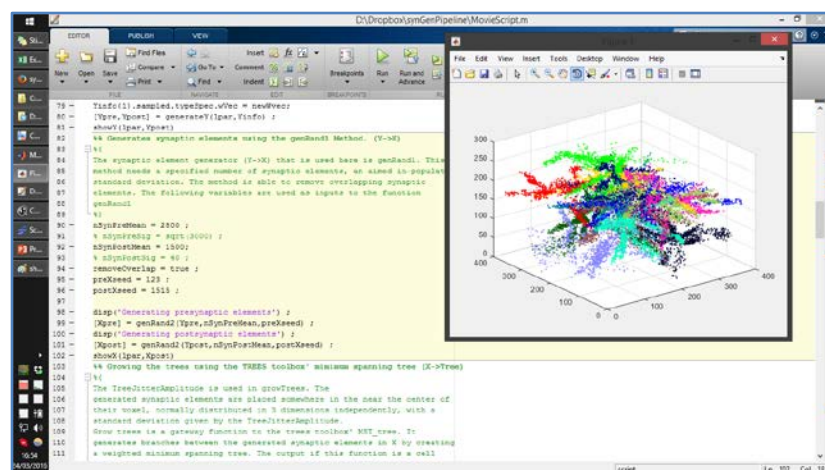
Figure 34: The online documentation provides an overview of the different steps in the pipeline.

Next the SynapseGenerator pipeline is run. There is a documentation page that can be consulted to appreciate the different steps of the pipeline (Figure 34). Furthermore, there is a template matlab script provided, in which all the necessary parameters are already specified. The third step is therefore to specify/adapt the `Ipar` struct, which specifies the number of cells, what shape they have and where to place them.



**Figure 35: The Y matrix, with pre or post- synaptic densities, is generated. A subset of the postsynaptic densities are shown.**

In the fourth step the density of synaptic elements is generated in the form of the Y matrix. The Y matrix is comprised of columns containing (linearised) voxel densities for each neuron, and is generated separately for the pre and postsynaptic population (Figure 35). Each voxel contains densities for a number of different neurons.



**Figure 36: The X matrix, with sampled pre and postsynaptic elements, is generated.**

In the fifth step the generated densities in the Y matrix are sampled to generate the synaptic elements, which is stored in the X matrix. This happens separately and independently for both the post and presynaptic population. This leads to a distribution of filled voxels, illustrated as dots colored by the identity of the neuron to which it belongs (Figure 36).

In the sixth step, the elements belonging to a given neuron are used to regenerate the neuron's morphology. For this we use the minimum spanning tree algorithm provided by the TREES toolbox (Cuntz *et al.*, 2010) supplied with the appropriate value of the parameter *bf* that weighs the relevance of the wire length relative to the path length to the soma. A key issue is that these morphologies should be consistent with the morphologies that were used as input for the Gaussian mixture fit. Currently, this is not the case.



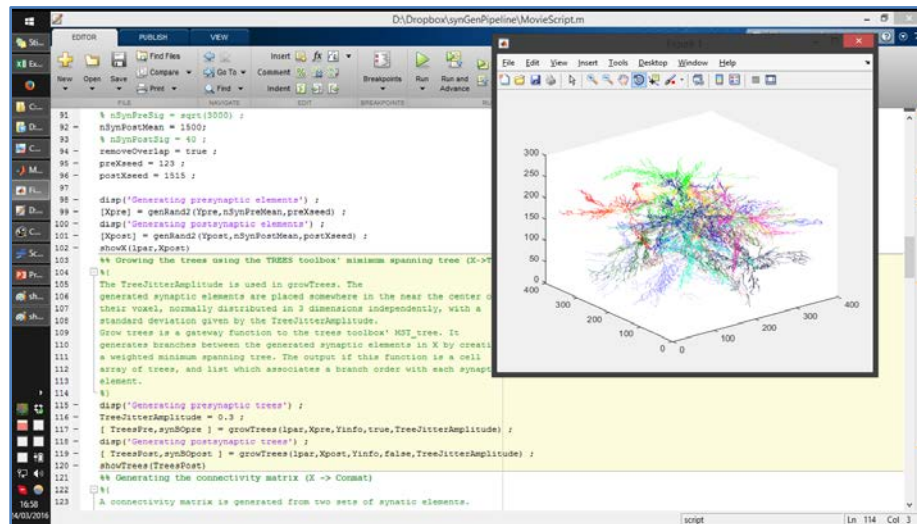


Figure 37: The dendritic and axonal morphologies are regenerated using the minimum spanning tree algorithm. A few example dendritic morphologies are shown.

In the seventh step, the synapses are constructed. Here this procedure is illustrated for the simplest rule: a synapse is called when a voxel contains both a pre- and post-synaptic element. At this stage, specific rules about the relative probability of making a synapse depending on the identity of the pre- and post-synaptic population, the number of already made synapses can be incorporated.

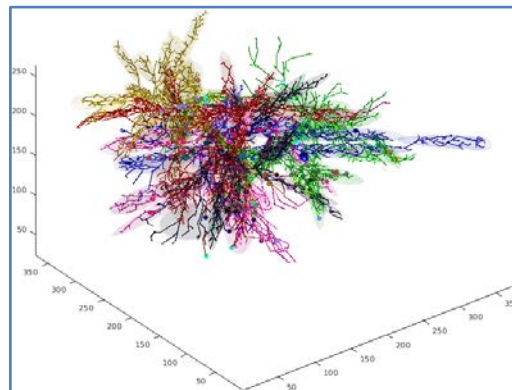
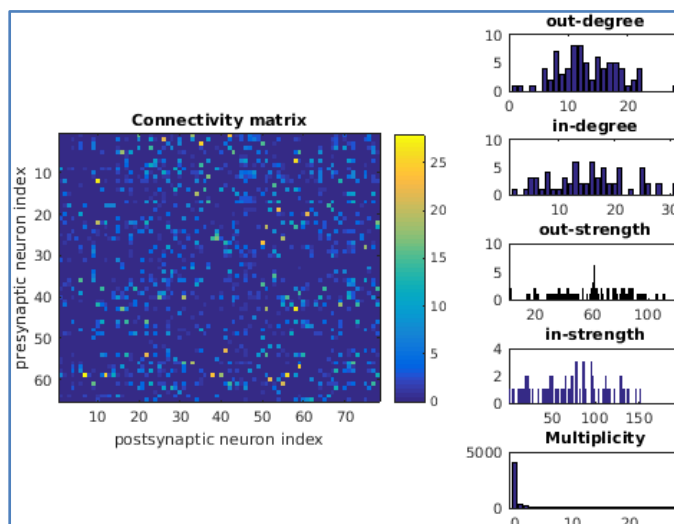


Figure 38: Synapse placement on example morphologies. Synapses are indicated by filled circles.

In the eighth step, the projection is statistically evaluated in order to compare to experimental data for validation. The validation is in terms of the statistical properties of the connections, such as the in and out degree of a neuron, the multiplicity of connections and the motif statistics. In addition, the statistics of the location of synapses: whether they are distributed randomly across the dendritic length or whether there is dependence on branch order (not shown).



**Figure 39: Validation of the constructed projection.**

This report summarises a number of results, which are more fully documented in other documents provided as attachments.

M. Bakker, Connecting the Dots, Master thesis (2016). Describes the implementation and evaluation of the reverse geometric approach in more detail.

L. Cazemier, Novel methods for assessing neuronal connectivity: a review (2015).

Tiesinga, P., Bakker, R., Hill, S. & Bjaalie, J.G. (2015). Feeding the human brain model. *Current Opinion in Neurobiology*, 32: 107-114. A publication describing the overall strategy for this work package.

The SynapseGenerator tool, its documentation and tutorials can be found online in the Neuroinformatics platform and at <http://www.neuroinformatics.nl/SynapseGenerator/index.html>



## List of references

- Cuntz, H., Forstner, F., Borst, A. & Hausser, M. (2010). One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS Computational Biology*, 6.
- Druckmann, S., Feng, L., Lee, B., Yook, C., Zhao, T., Magee, J.C. & Kim, J. (2014). Structured synaptic connectivity between hippocampal regions. *Neuron*, 81: 629-640.
- Egger, R., Dercksen, V.J., Udvary, D., Hege, H.C. & Oberlaender, M. (2014). Generation of dense statistical connectomes from sparse morphological data. *Frontiers in Neuroanatomy*, 8: 129.
- Hill, S.L., Wang, Y., Riachi, I., Schurmann, F. & Markram, H. (2012). Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits. *Proceedings of the National Academy of Sciences of the United States of America*, 109: E2885-2894.
- Kasthuri, N., Hayworth, K.J., Berger, D.R., Schalek, R.L., Conchello, J.A., Knowles-Barley, S., Lee, D., Vazquez-Reina, A., Kaynig, V., Jones, T.R., Roberts, M., Morgan, J.L., Tapia, J.C., Seung, H.S., Roncal, W.G., Vogelstein, J.T., Burns, R., Sussman, D.L., Priebe, C.E., Pfister, H. & Lichtman, J.W. (2015). Saturated Reconstruction of a Volume of Neocortex. *Cell*, 162: 648-661.
- Koene, R.A., Tijms, B., van Hees, P., Postma, F., de Ridder, A., Ramakers, G.J., van Pelt, J. & van Ooyen, A. (2009). NETMORPH: a framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinformatics*, 7: 195-210.
- Luczak, A. (2006). Spatial embedding of neuronal trees modeled by diffusive growth. *Journal of Neuroscience Methods*, 157: 132-141.
- Markram, H., Muller, E., Ramaswamy, S., Reimann, M.W., Abdellah, M., Sanchez, C.A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., Kahou, G.A., Berger, T.K., Bilgili, A., Buncic, N., Chalimourda, A., Chindemi, G., Courcol, J.D., Delalondre, F., Delattre, V., Druckmann, S., Dumusc, R., Dynes, J., Eilemann, S., Gal, E., Gevaert, M.E., Ghobril, J.P., Gidon, A., Graham, J.W., Gupta, A., Haenel, V., Hay, E., Heinis, T., Hernando, J.B., Hines, M., Kanari, L., Keller, D., Kenyon, J., Khazen, G., Kim, Y., King, J.G., Kisvarday, Z., Kumbhar, P., Lasserre, S., Le Be, J.V., Magalhaes, B.R., Merchan-Perez, A., Meystre, J., Morrice, B.R., Muller, J., Munoz-Céspedes, A., Muralidhar, S., Muthurasa, K., Nachbaur, D., Newton, T.H., Nolte, M., Ovcharenko, A., Palacios, J., Pastor, L., Perin, R., Ranjan, R., Riachi, I., Rodriguez, J.R., Riquelme, J.L., Rossert, C., Sfyraakis, K., Shi, Y., Shillcock, J.C., Silberberg, G., Silva, R., Tauheed, F., Telefont, M., Toledo-Rodriguez, M., Trankler, T., Van Geit, W., Diaz, J.V., Walker, R., Wang, Y., Zaninetta, S.M., DeFelipe, J., Hill, S.L., Segev, I. & Schurmann, F. (2015). Reconstruction and Simulation of Neocortical Microcircuitry. *Cell*, 163: 456-492.
- McAssey, M.P., Bijma, F., Tarigan, B., van Pelt, J., van Ooyen, A. & de Gunst, M. (2014). A morpho-density approach to estimating neural connectivity. *PloS One*, 9: e86526.
- Mishchenko, Y., Hu, T., Spacek, J., Mendenhall, J., Harris, K.M. & Chklovskii, D.B. (2010). Ultrastructural analysis of hippocampal neuropil from the connectomics perspective. *Neuron*, 67: 1009-1020.
- Narayanan, R.T., Egger, R., Johnson, A.S., Mansvelder, H.D., Sakmann, B., de Kock, C.P. & Oberlaender, M. (2015). Beyond Columnar Organization: Cell Type- and Target Layer-Specific Principles of Horizontal Axon Projection Patterns in Rat Vibrissa Cortex. *Cerebral Cortex*, 25: 4450-4468.
- Oberlaender, M., de Kock, C.P., Bruno, R.M., Ramirez, A., Meyer, H.S., Dercksen, V.J., Helmstaedter, M. & Sakmann, B. (2012). Cell type-specific three-dimensional structure of thalamocortical circuits in a column of rat vibrissa cortex. *Cerebral Cortex*, 22: 2375-2391.

- Parekh, R. & Ascoli, G.A. (2013). Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron*, 77: 1017-1038.
- Reimann, M.W., King, J.G., Muller, E.B., Ramaswamy, S. & Markram, H. (2015). An algorithm to predict the connectome of neural microcircuits. *Frontiers in Computational Neuroscience*, 9: 120.
- van Ooyen, A. (2011). Using theoretical models to analyse neural development. *Nature Reviews Neuroscience*, 1: 311
- van Ooyen, A., Carnell, A., de Ridder, S., Tarigan, B., Mansvelder, H.D., Bijma, F., de Gunst, M. & van Pelt, J. (2014). Independently outgrowing neurons and geometry-based synapse formation produce networks with realistic synaptic connectivity. *PLoS One*, 9: e85858.
- van Pelt, J. & van Ooyen, A. (2013). Estimating neuronal connectivity from axonal and dendritic density fields. *Frontiers in Computational Neuroscience*, 7: 160.
- van Pelt, J., van Ooyen, A. & Uylings, H.B. (2014). Axonal and dendritic density field estimation from incomplete single-slice neuronal reconstructions. *Frontiers in Neuroanatomy*, 8: 54.
- Zubler, F. & Douglas, R. (2009). A framework for modeling the growth and development of neurons and networks. *Frontiers in Computational Neuroscience*, 3: 25.

## 10.3 Structural Tools: 3DSynapsesSA (T5.4.2)

### 10.3.1 SynapseGenerator

SynapseGenerator is a pipeline used to predict connectivity at the synapse level.

#### Metadata

Category	Library
Maintainers	<a href="#">Rembrandt BAKKER</a>
Contributors	<a href="#">Max BAKKER</a> <a href="#">Rembrandt BAKKER</a> <a href="#">Paul TIESINGA</a>
Homepage	<a href="http://www.neuroinformatics.nl/SynapseGenerator/index.html">http://www.neuroinformatics.nl/SynapseGenerator/index.html</a>
Documentation	<a href="http://www.neuroinformatics.nl/SynapseGenerator/index.html">http://www.neuroinformatics.nl/SynapseGenerator/index.html</a>
Source Code	<a href="https://collab.humanbrainproject.eu/#/collab/47/nav/2784?state=uuid%3D79042414-6074-4daf-a4c0-6b9ead269cfe">https://collab.humanbrainproject.eu/#/collab/47/nav/2784?state=uuid%3D79042414-6074-4daf-a4c0-6b9ead269cfe</a>
License	GPLv3

Current Version	0.9+4
-----------------	-------

All Versions	0.9, 0.9+1, 0.9+2, 0.9+3
--------------	--------------------------

## Description

SynapseGenerator is a pipeline used to predict connectivity at the synapse level based on statistical properties of all neuron populations that co-exist in the given brain region of interest. For each population, the spatial distribution of the dendritic and axonal arborisations needs to be either specified as a probability density function (e.g. a mixture of Gaussians), or can be estimated from samples of reconstructed dendritic/axonal arborisations.

The pipeline returns the microconnectivity between individual neurons, thereby specifying the branch order at which they connect.

SynapseGenerator is written in Matlab, source files are available [for download](#).

### 10.3.2 3DSynapsesSA

SynapsesSA is a tool designed to process and analyse patterns in the three-dimensional spatial distribution of cortical synapses.

## Metadata

Category	Library
----------	---------

Partners	Universidad Politécnica de Madrid
----------	-----------------------------------

Maintainers	<a href="#">Luis Rodríguez LUJÁN</a>
-------------	--------------------------------------

Contributors	<a href="#">Pedro LARRAÑAGA</a> <a href="#">Concha BIELZA</a> <a href="#">Laura Antón SÁNCHEZ</a> <a href="#">Luis Rodríguez LUJÁN</a>
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------

Homepage	<a href="http://vps136.cesvima.upm.es:3838/hbp/synapsesSA/">http://vps136.cesvima.upm.es:3838/hbp/synapsesSA/</a>
----------	-------------------------------------------------------------------------------------------------------------------

Category	Service
----------	---------

Maintainers	<a href="#">Rembrandt BAKKER</a>
-------------	----------------------------------

Contributors	<a href="#">Rembrandt BAKKER</a> , <a href="#">Paul TIESINGA</a>
Homepage	<a href="https://neuroinformatics.nl/HBP/morphology-viewer/">https://neuroinformatics.nl/HBP/morphology-viewer/</a>
Documentation	<a href="https://neuroinformatics.nl/HBP/morphology-viewer/">https://neuroinformatics.nl/HBP/morphology-viewer/</a>
Support	hbp@scalablebrainatlas.org
Source Code	<a href="https://github.com/rbakker/HBP-morphology-viewer">https://github.com/rbakker/HBP-morphology-viewer</a>
License	GPLv3
Current Version	1
All Versions	1

## Description

The morphology viewer is a web-based service that interactively displays a 3D visualisation of neuron morphologies, and allows parts of the neuron to be switched on/off to allow a detailed inspection of the morphology. Supported formats are SWC and Neurolucida (asc, dat, nrx or xml).

There are two user scenarios. The first is to use the service as a standalone viewer, whereby the user visits the website, loads a morphology file from his own computer, and views the neuron. The second is to use the service from within the NIP, whereby search queries that return neuron morphologies link directly to the service.

Example: [direct link](#).

## 10.4 Structural Tools: 3DSomaMS (T5.4.2)

### 10.4.1 3DSomaMS

3DSomaMS is an R package for soma reparation, segmentation and characterisation.

## Metadata

Category	Library
Maintainers	<a href="#">Luis RODRIGUEZ</a>
Contributors	Pedro LARRAÑAGA Concha BIELZA

Sergio Luengo SÁNCHEZ  
Luis Rodríguez LUJÁN

Homepage <http://vps136.cesvima.upm.es:3838/hbp/somaMS/>

Documentation <http://vps136.cesvima.upm.es/Docs/somaMS/>

Support [luis.rodriquezl@upm.es](mailto:luis.rodriquezl@upm.es)

License GPL 2 or later

Current Version 1

All Versions 1

## Description

The 3DSomaMS component is designed to characterise quantitatively a 3D soma, according to morphological features taken from its image reconstruction, to build a mathematical model that also will allow simulation of artificial somas.

## Functionality

Selection of soma 3D representation generate table with quantitative characterisation reconstruction of the soma.

## 11. Tools for Functional Analysis (WP5.3, T5.3.1, T5.3.2)

### 11.1 Elephant

Electrophysiology Analysis Toolkit

## Metadata

Category Library

Maintainers [Michael DENKER](#)  
[Alper YEGENOGLU](#)  
[Andrew DAVISON](#)

Contributors The Elephant Community

Homepage	<a href="http://elephant.readthedocs.org">http://elephant.readthedocs.org</a>
Documentation	<a href="http://elephant.readthedocs.org/en/latest/index.html">http://elephant.readthedocs.org/en/latest/index.html</a>
Support	<a href="https://github.com/NeuralEnsemble/elephant/issues">https://github.com/NeuralEnsemble/elephant/issues</a>
Source Code	<a href="https://github.com/NeuralEnsemble/elephant/">https://github.com/NeuralEnsemble/elephant/</a>
License	BSD
Current Version	<p><i>Release of analysis tools: Elephant 0.3.0</i></p> <p><i>Anticipated: March 8, 2016 Integration on Collab: March 20, 2016</i></p> <p><i>(incl. spike sorting SpikeSort (0.12) or SpyKING CIRCUS (latest))</i></p>
All Versions	0.2.0, 0.1.0

## Description

Elephant (Electrophysiology Analysis Toolkit) is an open-source library for the analysis of electrophysiological data in the Python programming language. The focus of Elephant is on generic analysis functions for spike train data and time series recordings from electrodes, such as local field potentials (LFP) or intracellular voltages. In addition to providing a common platform for analysis code from different laboratories, the Elephant project aims to provide a consistent and homogeneous analysis framework built on a modular foundation.

The Python software is available for use on the Collaboratory using the Task system or as Jupyter notebook.

## Use case sample of NIP Elephant integration

The aim of WP5.3 was to provide tools for the analysis of functional datasets, in particular spike and local field potential data, acquired by both experiments and theory. Due to the large amount of data analysis methods to be implemented, it was decided to split work into two Tasks covering different focus areas expected to be of relevance for the modeling efforts of HBP: (a) analysis of the concerted action of large neuronal populations (T5.3.1), (b) bridging scales of observations including the analysis of population signals such as local field potentials (T5.3.1), (c) tools for analysing the dynamics and statistics of single cell activity (T5.3.2) and (d) implementation of spike sorting capabilities, a common preprocessing step for electrophysiological data (T5.3.2). Since both LFP and spike train data are typically recorded together, and the underlying data model (Neo python library) aims at a joint description of these two data modalities, it was decided to bundle both types of functionalities into a single library package (the Elephant package), with the exception of spike sorting, which was implemented in the Collaboratory by building on existing efforts due to the complexity of this method.

The current version of Elephant (<http://neuralensemble.org/elephant/>) contains an initial set of functionality covering aspects of both Work Packages, for example: state-of-the-art methods to detect higher-order correlations in spike data as mentioned by the reviewers' comment (relates to point (a) above), spectral methods to analyse LFP data and LFP-spike relationships such as



spike-field coherence methods (relates to point (b) above), spike train statistics and spike train metrics such as spike train similarity measures (relates to point (c) above) and inclusion of the spike-sort package (relates to point (d) above). In practice, the teams of Tasks 5.3.1 and 5.3.2 worked closely together on achieving these goals, and a clear division between the Tasks was not always possible or wanted.

Elephant is available for public use on the HBP Collaboratory.

We agree with the reviewers that with the initial set of tools of the Elephant library in place, it is imperative to construct workflows that embed Elephant and related tools into a Collaboratory-enabled workflow. To prepare for this, during the first part of the RUP WP5.3 already began to port data analysis tasks to the HBP collaborative infrastructure. In the second part of the RUP, WP5.3 constructed the NEST-SpiNNaker-Elephant workflow (<https://collab.humanbrainproject.eu/#/collab/507/nav/6326>) together with SP6, SP7, and SP9, presented during the opening day (<https://www.youtube.com/watch?v=bvesBlwvH2M>) and the review. This workflow served as a first demonstration of how the Collaboratory can be used to simulate data using HPC (SP7) and neuromorphic hardware (SP9), and subsequently analyse the data output using the Elephant package (SP5) to validate the results of different models and/or simulation engines. Due to the success of this demonstrator, the work is now continued in SGA1 as part of T9.1.5.

Also building on these efforts of the RUP, in WP4.5 of SGA1 we are in the process of extending this workflow to include a comparison of simulated data to experimental datasets. For this analysis workflow, it will become vital to establish the connection of the Collaboratory to the NIP to get data access. We envision the following action points for SGA1:

#### **Annotating a dataset with metadata**

Supplying the full metadata information for a complex dataset featuring multi-modal, high-dimensional activity data in addition to behavior. We have recently performed the detailed metadata acquisition of such datasets and described the procedures and results. [Zehl, L., Jaillet, F., Stoewer, A., Grewe, J., Sobolev, A., Wachtler, T., Brochier, T.G., Riehle, A., Denker, M., and Grün, S. (2016). Handling Metadata in a Neurophysiology Laboratory. *Frontiers in Neuroinformatics* 10, 26.].

#### **Publishing the dataset**

The annotated and fully described dataset must be made publicly available on a scientific database store. We are currently in the process of providing the data described above to the community and are preparing to publish the data with a complete description of them in the new Journal 'Scientific Data' (<http://www.nature.com/sdata/>).

#### **Integration of the dataset into NIP**

This step involves three aspects: First, making sure the data itself is accessible through the NIP, and second, selecting a set of relevant minimal metadata from the complex metadata collection to expose it through the NIP portal search. In a third step, the terminology of such exposed metadata must be matched with existing efforts to build up an ontology for electrophysiological data (e.g., G-Node <http://www.g-node.org/projects/odml/terminologies>, NWB data format (Teeters, J.L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., Asari, H., Peron, S., Li, N., Peyrache, A., et al. (2015). Neurodata Without Borders: Creating a Common Data Format for Neurophysiology. *Neuron* 88, 629-634), Neurolex [<http://neurolex.org/>]).

#### **Inclusion of the data into the analysis workflow**

Through the NIP interface, the analysis workflow on the Collaboratory will query for the data and then access the corresponding datasets. Data can be loaded using the Neo Python library (already available on the HBP Collaboratory) using either the Task system or the Jupyter Notebook functionality of the Collaboratory. In addition, the odML library will be made available through the Collaboratory such that the full metadata (contained within the retrieved dataset) can be

accessed and used to enrich the Neo data objects. At this stage, the data can be treated in a similar fashion to that described in the NEST-Spinnaker-Elephant workflow.

## Using NIP to analyse electrophysiological data using Elephant

Jake wants to build a model of a cortical column of the visual cortex V1. Although he has tuned his model neurons and network parameters of his simulation to display firing rates that match findings in the literature, he realises that he needs to compare the output of his model of connected neurons to recordings of V1 activity in order to fully validate the network model. In a first step, Jake launches the NIP portal in his web browser and searches for datasets that could potentially match his needs for suitable experimental ground truth data. Jake uses a variety of search terms to locate possible candidate datasets, such as for the brain area (query: areas is V1), the minimum number of parallel recordings he is interested in (query: number of channels larger than 50), or tasks that are potentially suited for the 'dark screen' type of stimulus Jake presents to the model (query: Task is of type that contains a blank screen). Among the search results, Jake selects a dataset that is closest to his requirements and model.

Subsequently, in the Collab Jake uses to simulate his network model, he launches a new Jupyter notebook, and downloads the dataset he has identified from the NIP to the Collab storage. Using the Neo and odML libraries, he is able to load the data and metadata information into a data structure similar to the one he uses to store his simulation results, allowing him to easily access both datasets for further processing. Jake then uses the textual description of the experimental task and the queryable odML metadata information of the experiment to select, from the experimental and/or simulated data, sets of spiketrains from individual neurons that match – as well as possible – in number, spatial location, and neuron type. Using this sub-sampled and matching set of data from model and experiment represented in the common Neo representation, Jake can now directly apply tools from the Elephant analysis library to analyse the two data sets using the identical tools, for example, to calculate the distribution of pair-wise zero-lag correlation coefficients between all pairs of simultaneously recorded data sets in each of the recordings. These analyses then act as a benchmark to validate in how far the model results match the experimental observations, both on the single-neuron level, as well as on the network level. Jake assembles a number of such comparisons in the Jupyter notebook and uses this tool in the future to guide his parameter fitting procedure and validate future changes.

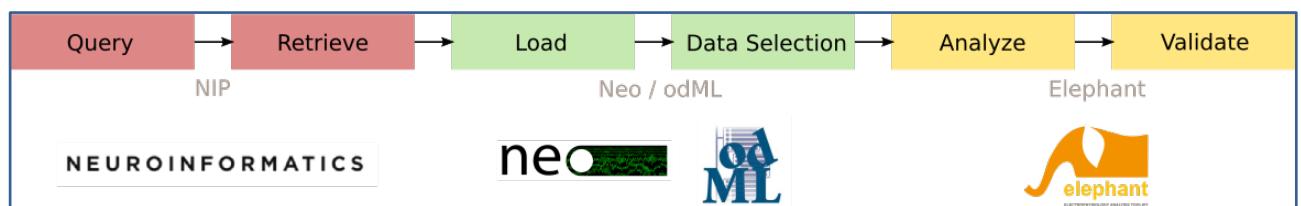


Figure 40: Sample of Elephant NIP use case

## 12. Brain Atlases (WP5.5)

### 12.1 General comment

In T5.1.6 EPFL developed the core Volumetric Imaging Service and BBIC atlas data format, the KnowledgeGraph extensions necessary for Atlas Building, the necessary ontologies and the Multiresolution Atlas Viewer. AlignII and QuickNI were developed by UiO and were used to anchor / spatially register high resolution microscopic images of experimental data to the reference atlases for mouse and rat brain. The tools were thus used for producing the different atlases of experimental data reported under under T5.5.1. At the time of the review, the tools were registered in the Data Information Cards and available to interested researchers within or outside

HBP. Information about the tools and their use is now clearly communicated through the Collaboratory > Platform Collaboratories > Neuroinformatics > Tools (Brain Atlas tools, Documentation). The standalone tool, QuickNII, is distributed to users based on email contact with UiO. The AlignII tool, which functions as a service running on UiO servers, is now available through the use of an application form available at the project web page. UiO also developed a suite of tools to inspect the reference atlases at the level of whole brain or individual brain structures, as well through creation of user defined cut planes at any angle of orientation chosen by the user. These tools (MeshGen, MeshView, and CutNII) are all fully shared. They were intended to facilitate the use of the reference atlases and although providing only basic viewing and slicing, they are among the very few tools of its kind for direct use online (MeshView: <http://www.nesys.uio.no/wiki/meshview.html>) or downloading (MeshGen, used to generate the meshes from the reference atlases as shown in MeshView; and CutNII, used to view and cut slices through the volumetric versions of the reference atlases).

The NIP provides users with structural data analysis and functional analysis, as well as brain atlases. Tools that have been specifically developed in order to help neuroscientists to retrieve information and visualise dataset can be easily found within the NIP (<https://collab.humanbrainproject.eu/#/collab/47/nav/6641>).

The different tools are listed within our [Neuroinformatics Platform Collaboratory section](#).

## 12.2 The Mouse Brain Atlas (T5.5.1)

### 12.2.1 AlignII

Web-based tool for anchoring 2D image data into 3D atlas space.

#### Metadata

Category	Service
Maintainers	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Dmitri DARINE <a href="mailto:d.a.darine@medisin.uio.no">d.a.darine@medisin.uio.no</a>
Contributors	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Dmitri DARINE <a href="mailto:d.a.darine@medisin.uio.no">d.a.darine@medisin.uio.no</a> Trygve B. LEERGAARD <a href="mailto:t.b.leergaard@medisin.uio.no">t.b.leergaard@medisin.uio.no</a> Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
Homepage	<a href="http://www.rodentbrainworkbench.org">http://www.rodentbrainworkbench.org</a> <a href="http://www.rbwb.org">www.rbwb.org</a>
Documentation	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>
Support	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>

License	AlignII is embedded in the Rodent Brain Navigator data system. Access for partners in the Human Brain Project. Contact: Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
---------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Current Version	1.0
-----------------	-----

All Versions	1.0
--------------	-----

## Description

AlignII is web-based tool for user guided registration (anchoring) of 2D experimental image data, typically high resolution microscopic images, to 3D atlas reference space (Waxholm Space Mouse / Rat Atlas or Allen Mouse Brain Atlas), thus facilitating data integration through standardised coordinate systems. AlignII is integrated in the web-based Rodent Brain Navigator data system — a data system for internal use in HBP, developed and maintained in collaboration with the INCF Norwegian Node.

A key feature in the tool is the capability to generate user-defined cut planes through the atlas templates, matching the orientation of the cut plane of the 2D experimental image data, as a first step towards anchoring of images to the relevant atlas template.

AlignII supports multi-modality 3D atlas packages and one-click switching between image modalities (e.g., MRI and DTI) while preserving the view location and UI settings in general, and in-depth numerical control over cut plane settings. The tool has been tested extensively and is improved based on user feedback. A series of research projects are currently using the tool for registration of project data to standardised atlas space.

### 12.2.2 QuickNII

Standalone tool for anchoring 2D image data into 3D atlas space.

## Metadata

Category	Application
----------	-------------

Maintainers	<a href="#">Gergely CSUCS</a>
-------------	-------------------------------

Contributors	Jan BJALLIE Trygve LEERGAARD Gergely CSUCS
--------------	--------------------------------------------------

Homepage URL	<a href="http://www.rodentbrainworkbench.org">http://www.rodentbrainworkbench.org</a> <a href="http://www.rbwb.org">www.rbwb.org</a>
--------------	-----------------------------------------------------------------------------------------------------------------------------------------

Documentation	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>
---------------	-------------------------------------------------------------------------------

---

Support <http://www.nesys.uio.no/AlignII>

---

License Access for partners of the Human Brain Project. Contact: Jan G. BJAALIE

---

Current Version 1.0

---

All Versions 1.0

---

## Description

QuickNII is a standalone tool for user guided registration (anchoring) of 2D experimental image data, typically high resolution microscopic images, to 3D atlas reference space (Waxholm Space Mouse / Rat atlas or Allen Mouse Brain Atlas ), thus facilitating data integration through standardised coordinate systems.

A key feature in the tool is the capability to generate user-defined cut planes through the atlas templates, matching the orientation of the cut plane of the 2D experimental image data, as a first step towards anchoring of images to the relevant atlas template.

QuickNII supports multi-modality 3D atlas packages and one-click switching between image modalities (e.g., MRI and DTI) and structural parcellations, while preserving the view location and UI settings in general and in-depth numerical control over cut plane settings. The tool has been tested extensively and is improved based on user feedback. A series of research projects are currently using the tool for registration of project data to standardised atlas space.

QuickNII is a standalone iteration of AlignII where the volumetric templates for the rat brain (MRI, DTI and structural parcellations for the Waxholm Space Rat Atlas) and mouse brain (volumetric histology and structural parcellations for the Allen Mouse Brain Atlas) are brought to the user's computer. This practice eliminates the network roundtrip, and thus results in near real-time operation with high refresh rate of the navigational and user-defined cut plane renderings. Other aspects of user experience are preserved with the same controls, views and general look-and-feel as provided by AlignII. While QuickNII is a standalone tool, steps are taken in order to integrate it in the workflow provided by the Rodent Brain Navigator data system as well as in other HBP image services.

### 12.2.3 MeshGen

Mesh generator for volumetric segmentation data.

## Metadata

---

Category Application

---

Maintainers [Gergely CSUCS](#)

---

Contributors Gergely CSUCS [gergely.csucs@medisin.uio.no](mailto:gergely.csucs@medisin.uio.no)  
Jan G. BJAALIE [j.g.bjaalie@medisin.uio.no](mailto:j.g.bjaalie@medisin.uio.no)

---

Homepage URL	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
Documentation	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
Support	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
License	CC-BY-NC-SA
Current Version	1.0
All Versions	1.0

## Description

MeshGen generates separate mesh files for each label present in a segmentation volume. The generation employs a variant of Constrained Elastic Surface Nets [CESN], which has been modified in order to avoid intersection between neighbouring meshes.

Supported input formats:

- NIFTI for segmentation volume, with integer elements (8/16/32 bits, signed/unsigned)
- ITK label file (ITK-Snap compatible, does not contain hierarchy)
- ILF - Integrated Label File (MBAT compatible, hierarchical)
- Brain-Map XML file (ontology from brain-map.org, hierarchical)

When label hierarchy is available, MeshGen can follow containment relations, e.g. eliminating generation of extra inner contours around contained labels. The tool can optionally eliminate contained labels if their colour matches with the colour of their parent, e.g. when rendering or not rendering the meshes, using their original colours would make no visual difference.

In order to provide compact mesh output, MeshGen can apply edge elimination algorithms on generated data. One algorithm eliminates edges based on the curvature of neighbouring surface patch. A second algorithm eliminates edges based on their length.

Online results/examples using the companion viewer tool, MeshView:

- Waxholm Space Atlas of the Sprague Dawley rat brain (meshes for 79 anatomical structures - 80 with inner ear)[WHSRat]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSRatV2.bitlas>
- Waxholm Space Atlas of the adult C57BL/6J mouse brain (meshes for 26 anatomical structures - 27 with inner ear) [WHSMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSMouseHier.bitlas>
- Allen Mouse Brain Atlas reference atlas (meshes for 106 anatomical structures including hierarchy)[ABAMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=ABAMouseHier.bitlas>

### 12.2.4 MeshView

Web-based mesh viewer with user-defined cut plane support.



## Metadata

Category	Service
Maintainers	<a href="#">Gergely CSUCS</a>
Contributors	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
Homepage URL	<a href="http://www.rodentbrainworkbench.org">http://www.rodentbrainworkbench.org</a> <a href="http://www.rbwb.org">www.rbwb.org</a>
Documentation	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
Support	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
License	MeshView is presented through the HBP services. Open access. Contact: Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
Current Version	0.9
All Versions	0.9

## Description

Web application for real-time 3-D display of mesh-data representing structural parcellations, and generation of user defined cut planes.

MeshView is provided with different atlases, including the Allen Mouse Brain Atlas and Waxholm Space atlases for mouse and rat:

- Waxholm Space Atlas of the Sprague Dawley rat brain (meshes for 79 anatomical structures - 80 with inner ear) [WHSRat]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSRatV2.bitlas>
- Waxholm Space Atlas of the adult C57BL/6J mouse brain (meshes for 26 anatomical structures - 27 with inner ear) [WHSMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSMouseHier.bitlas>
- Allen Mouse Brain Atlas reference atlas (meshes for 106 anatomical structures including hierarchy) [ABAMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=ABAMouseHier.bitlas>

In the final release, the aim is to support viewing of anchored 2D image data, and viewing of the locations of any data set.

## 12.2.5 CutNII

Standalone viewer for volumetric atlases with custom-angle slicing support.

### Metadata

Category	Application
Maintainers	<a href="#">Gergely CSUCS</a>
Contributors	Jan BJALLIE Trygve LEERGAARD Gergely CSUCS
Homepage URL	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>
Documentation	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>
Support	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>
License	CC-BY-NC-SA
Current Version	1.0
All Versions	1.0

### Description

CutNII is a limited, standalone version of QuickNII where the volumetric templates for the rat brain (MRI, DTI and structural parcellations for the Waxholm Space Rat Atlas) and mouse brain (volumetric histology and structural parcellations for the Allen Mouse Brain Atlas) are brought to the user's computer.

A key feature in the tool is the capability to generate user-defined cut planes through the atlas templates, matching the orientation of the cut plane of the 2D experimental image data. CutNII supports multi-modality 3D atlas packages and one-click switching between image modalities (e.g., MRI and DTI) and structural parcellations, while preserving the view location and UI settings in general, and in-depth numerical control over cut plane settings.

Two software packages are available:

- CutNII for Waxholm Space atlas of the Sprague Dawley rat brain: This package contains the CutNII software bundled with pre-processed data from Waxholm Space atlas of the Sprague Dawley rat brain. T2\* MRI, DTI, and delineation volumes are provided.

- CutNII for Allen Mouse Brain Atlas reference atlas: This package contains the CutNII software bundled with pre-processed data from Allen Mouse Brain Atlas reference atlas. Grayscale Nissl volume and annotation volume are provided.

## 12.2.6 Scalable Brain Atlas

The Scalable Brain Atlas (SBA) is a fully web-based display engine for brain atlases, imaging data and topologies. It allows client websites to show brain region related data in a 3D interactive context, and provides services to look up regions, generate thumbnails or download nomenclature- and delineation data.

### Metadata

Category	Service
Maintainers	<a href="#">Rembrandt BAKKER</a>
Homepage	<a href="http://scalablebrainatlas.incf.org/main/index.php?">http://scalablebrainatlas.incf.org/main/index.php?</a>
License	<a href="http://scalablebrainatlas.incf.org/main/citationpolicy.php">http://scalablebrainatlas.incf.org/main/citationpolicy.php</a>
Current Version	1
All Versions	1

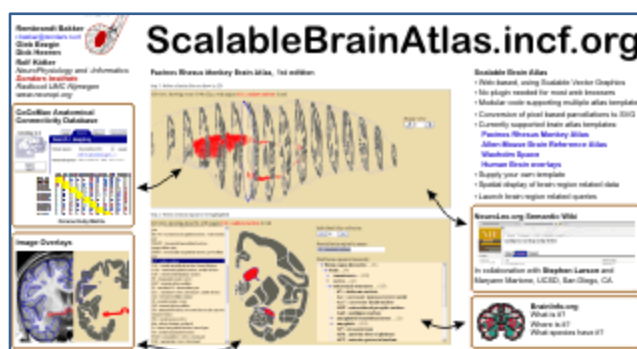


Figure 41: The Scalable Brain Atlas

### Description

The Scalable Brain Atlas (SBA) is a fully web-based display engine for brain atlases, imaging data and ontologies. It is described in the following publication:

Bakker, R., Tiesinga, P., & Kötter, R. (2015). The Scalable Brain Atlas: instant web-based access to public brain atlases and related content. *Neuroinformatics*, 13: 353-366. doi: 10.1007/s12021-014-9258-x.

Specifically added in the context of the Human Brain Project is a standardised set of anatomical landmarks for registration of whole brain experimental image data from the mouse and rat brain to Waxholm Space. The use of standardised landmarks in combination with a common coordinate system for the rodent brain increases speed and accuracy of data integration.

Internal landmarks of the C57BL/6J mouse brain and the Sprague Dawley rat brain (Waxholm Space templates) have been identified across different MRI contrasts (T1, T2, T2\*) and other modalities in different specimens and at different slice acquisition angles and different image resolutions. The use of these landmarks increases the accuracy of registration of experimental volumetric datasets to atlas templates and thereby facilitates comparisons of data from different experiments and data analysis requiring use of standardised atlas space.

The use of landmarks for registration is described in the following publication:

Sergejeva, M., Papp, E.A., Bakker, R., Gaudnek, M.A., Okamura-Oho, Y., Boline, .J, Bjaalie, J.G., & Hess, A. (2015). Anatomical landmarks for registration of experimental image data to volumetric rodent brain atlasing templates. *Journal of Neuroscience Methods*, 240:161-9. doi: 10.1016/j.jneumeth.2014.11.005.

## 12.3 The Human Brain Atlas (T5.5.2)

### 12.3.1 JuBrain workflow

Tools listed below are used during the Data workflow in order to create Dataset that will be recorded in the NIP.

Some of the tools presented are not HBP derived but are pre-requisit to this workflow:

- 1) The web-based JuBrain cytoarchitectonic atlas browser, hosted at <https://jubrain.fz-juelich.de/apps/cytoviewer/cytoviewer-main.php>. It contains supporting information under the "help" menu item.
- 2) The SPM Anatomy toolbox, including an MRI-scale version of the JuBrain atlas, available at [http://www.fz-juelich.de/inm/inm-1/DE/Forschung/docs/SPMAatomyToolbox/SPMAatomyToolbox\\_node.html](http://www.fz-juelich.de/inm/inm-1/DE/Forschung/docs/SPMAatomyToolbox/SPMAatomyToolbox_node.html). It contains a set of matlab functions to combine probabilistic cytoarchitectonic maps and functional imaging data in the context of SPM. A manual can be retrieved from <http://www.fz-juelich.de/SharedDocs/Downloads/INM/INM-1/DE/Toolbox/Manual.pdf>, and is also linked from the above page.
- 3) The official web portal for the Big Brain dataset, hosted by our partner Alan Evans (MNI), hosted at <https://bigbrain.loris.ca/main.php>.
- 4) The „microdraw“ collaborative vectorial annotation tool for ultra high resolution data, originally developed by Roberto Toro, with recent contributions from developers in JUELICH, available at <https://github.com/r03ert0/microdraw>. It is a JavaScript-based web-application for efficient rendering and annotation of high-resolution 2D images, that can be hosted on standard web servers. We use it in Jülich for mapping cytoarchitectonic structures.
- 5) The freely available CTVox volume renderer for mobile devices, available from <http://bruker-microct.com/products/ctvox.htm>. It has not been developed within HBP, but has been successfully applied for visualization of medium-sized 3D datasets, like 3D histological volumes of rat brain. This refers to multimodal atlas resources that have been produced using the 3D Brain Atlas Building tools AlignNII and QuickNII.

We have explained above how these tools have been used. The tutorials provided for these tools explains how large series of images, representing sections cut at different angles of orientation, are anchored / spatially registered to the reference atlases. Further coordinated efforts in terms of registering large number of experimental datasets is a key task in SGA1.

## 12.3.2 JuBrain Cytoviewer

JuBrain Cytoarchitectonic Atlas Viewer.

### Metadata

Category	Service
Maintainers	<a href="#">Katrin AMUNTS</a> <a href="#">Hartmut MOHLBERG</a>
Contributors	
Homepage	<a href="http://jubrain.fz-juelich.de">http://jubrain.fz-juelich.de</a>
Documentation	<a href="#">JuBrain Website</a> <a href="#">List of supported client technologies</a>
Support	<a href="#">Hartmut MOHLBERG</a>
License	<a href="#">Copyright information</a>
Current Version	Atlas data Version 1.0, Webviewer application version 2.1
All Versions	

### Description

The JuBrain CytoViewer shows cytoarchitectonic probabilistic maps, which have been developed in the teams of Katrin AMUNTS and Karl ZILLES in Juelich and Duesseldorf (INM-1). Cytoarchitectonic areas were analysed in histological sections of ten human *post mortem* brains. The maps are based on image analysis and statistical criteria for localising areal borders. Cytoarchitectonic maps have been developed during the past 20 years as a joint effort of many doctoral students, post docs and guests.

### References for the JuBrain CytoViewer

Mohlberg, H., Eickhoff, S.B., Schleicher, A., Zilles, K., & Amunts, K. (2012). A new processing pipeline and release of cytoarchitectonic probabilistic maps - JuBrain, OHBM 2012, Peking, China.

## References for probabilistic cytoarchitectonic mapping

Amunts, K., Schleicher, A., & Zilles, K. (2007). Cytoarchitecture of the cerebral cortex - more than localization. *Neuroimage*, 37: 1061-1065.

Zilles, K., & Amunts, K. (2010). Centenary of Brodmann's map - conception and fate. *Nature Reviews, Neuroscience*, 11: 139-145.

Zilles, K., Schleicher, A., Palomero-Gallagher, N., & Amunts, K. (2002). Quantitative analysis of cyto- and receptor architecture of the human brain. In: *"Brain Mapping: The Methods"*, Mazziotta, J.C. & Toga, A. (eds.), USA: Elsevier, 2002, p. 573-602.

### 12.3.3 Papaya

Medical research web-based image viewer, with extended functionality to retrieve context-sensitive 2D graphical and textual context information for brain areas.

## Metadata

Category	Application
Maintainers	<a href="#">Research Imaging Institute, University of Texas Health Science Center</a>
Contributors	<a href="#">T. DICKSCHEID</a> <a href="#">S. EICKHOFF</a> <a href="#">K. AMUNTS</a>
Homepage	<a href="http://ric.uthscsa.edu/mango/papaya.html">http://ric.uthscsa.edu/mango/papaya.html</a>
Documentation	<a href="#">Online Userguide</a> <a href="#">Online Developer Guide</a>
Support	
License	<a href="https://github.com/rii-mango/Papaya/blob/master/LICENSE">https://github.com/rii-mango/Papaya/blob/master/LICENSE</a>
Current Version	v0.8

## All Versions

## Description

Papaya is a pure JavaScript medical research image viewer, supporting DICOM and NIFTI formats, compatible across a range of web browsers. This orthogonal viewer supports overlays and atlases. For HBP, the viewer is being extended by 2D overlay functionality to enable retrieval and display



of rich context-sensitive information for brain areas, like receptor plots, quantitative data in the form of tables, and 2D linear and polar plots. The feature developments are anchored in SP2. The viewer will be integrated by SP5 as an app into the Collaboratory for exploring human brain atlas data.



## Annex B: Software and Services Included in this Platform Release

### Product/Software Package/Service name:

#### *3DSomaMS*

3DSomaMS is an R package for soma reparation, segmentation and characterisation.

#### Metadata

Category	Library
Maintainers	<a href="#">Luis RODRIGUEZ</a>
Contributors	Pedro LARRAÑAGA Concha BIELZA Sergio Luengo SÁNCHEZ Luis Rodríguez LUJÁN
Homepage	<a href="http://vps136.cesvima.upm.es:3838/hbp/somaMS/">http://vps136.cesvima.upm.es:3838/hbp/somaMS/</a>
Documentation	<a href="http://vps136.cesvima.upm.es/Docs/somaMS/">http://vps136.cesvima.upm.es/Docs/somaMS/</a>
Support	<a href="mailto:luis.rodriguezl@upm.es">luis.rodriguezl@upm.es</a>
License	GPL 2 or later



---

Current Version	1
-----------------	---

---

All Versions	1
--------------	---

### ***Description***

The 3DSomaMS component is designed to characterise quantitatively a 3D soma, according to morphological features taken from its image reconstruction, to build a mathematical model that also will allow simulation of artificial somas.

### ***Functionality***

Selection of soma; 3D representation; generation of table with quantitative characterisation; reconstruction of the soma.

### *DATA Repository service (ZENODO)*

Our repository service is based on Zenodo. It builds and operates a simple and innovative service that enables researchers, scientists, EU projects and institutions to share, preserve and showcase multidisciplinary research results (data and publications).

#### Metadata

Category	Service
Maintainers	Zenodo is developed by CERN under the EU FP7 project OpenAIREplus (grant agreement no. 283595).
Homepage	<a href="https://zenodo.org/">https://zenodo.org/</a>
Documentation	<a href="https://zenodo.org/features">https://zenodo.org/features</a>
Support	<a href="https://zenodo.org/contact">https://zenodo.org/contact</a>
Source Code	<a href="https://zenodo.org/dev">https://zenodo.org/dev</a>
License	Missing License
Current Version	REST API – includes support for uploading your research outputs. OAI-PMH API – allows you to harvest all or parts of Zenodo via the OAI-PMH protocol.
All Versions	The REST API is versioned. We strive not to make backward incompatible changes to the API, but if we do, we release a new version. Changes to the API are documented on this

page, and advance notification is given on our Twitter account.

## Description

(Zenodo enables researchers, scientists, EU projects and institutions to:

- easily share the long tail of small research results in a wide variety of formats including text, spreadsheets, audio, video, and images across all fields of science.
- display their research results and receive credit by making the research results citable and integrating them into existing reporting lines to funding agencies such as the European Commission.
- easily access and reuse shared research results.

Deliverables:

- An open digital repository for everyone and everything not served by a dedicated service; the so called “long tail” of research results.
- A modern look and feel in line with current trends in state-of-the-art online services.
- Integration with OpenAIRE infrastructure and assured inclusion in OpenAIRE corpus.
- Easy upload and semi-automatic metadata completion by communication with existing online services such as DropBox for upload, Mendeley/ORCID/CrossRef/OpenAIRE for upload and pre-filling metadata.
- Easy access to research results via an innovative viewing option, open APIs, integration with existing online services, and the preservation of community-independent data formats.
- A safe and trusted service by combining community based curation with short- and long-term archival and digital preservation strategies in accordance with best practices.
- Persistent identifiers, Digital Object Identifiers (DOIs), for sharing research results.
- Service hosting according to industry best practices in CERN’s professional data centres.

An easy way to link research results with other results and products, funding sources, institutions, and licenses.



### ***3DSynapsesSA***

SynapsesSA is a tool designed to process and analyse patterns in the three-dimensional spatial distribution of cortical synapses.

#### **Metadata**

Category	Library
Partners	Universidad Politécnica de Madrid
Maintainers	<a href="#">Luis Rodríguez LUJÁN</a>
Contributors	<a href="#">Pedro LARRAÑAGA</a> <a href="#">Concha BIELZA</a> <a href="#">Laura Antón SÁNCHEZ</a> <a href="#">Luis Rodríguez LUJÁN</a>
Homepage	<a href="http://vps136.cesvima.upm.es:3838/hbp/synapsesSA/">http://vps136.cesvima.upm.es:3838/hbp/synapsesSA/</a>
Documentation	<a href="http://vps136.cesvima.upm.es/Docs/synapsesSA/">http://vps136.cesvima.upm.es/Docs/synapsesSA/</a>
Support	<a href="mailto:luis.rodriguezl@upm.es">luis.rodriguezl@upm.es</a>
Source Code	<a href="http://vps136.cesvima.upm.es/R/">http://vps136.cesvima.upm.es/R/</a>
Download Page	<a href="http://vps136.cesvima.upm.es/R/">http://vps136.cesvima.upm.es/R/</a>





---

License	GPL 2 or later
---------	----------------

---

Current Version	1.2
-----------------	-----

---

All Versions	1.2
--------------	-----

---

SynapsesSA is a tool designed to process and analyse patterns in the three-dimensional spatial distribution of cortical synapses. It brings a variety of both innovative and well-known techniques from the spatial statistics field and a web-based graphical interface compatible with most common browsers.

### Functionality

- Process and visualise data from cortical synapses
- Model the spatial distribution of the synapses
- Replicate, via simulation, samples of cortical synapses
- Compare several indicators obtained from data of different layers

***Elephant***

Electrophysiology Analysis Toolkit.

**Metadata**

Category	Library
Maintainers	<a href="#">Michael DENKER</a> <a href="#">Alper YEGENOGLU</a> <a href="#">Andrew DAVISON</a>
Contributors	The Elephant Community
Homepage	<a href="http://elephant.readthedocs.org">http://elephant.readthedocs.org</a>
Documentation	<a href="http://elephant.readthedocs.org/en/latest/index.html">http://elephant.readthedocs.org/en/latest/index.html</a>
Support	<a href="https://github.com/NeuralEnsemble/elephant/issues">https://github.com/NeuralEnsemble/elephant/issues</a>
Source Code	<a href="https://github.com/NeuralEnsemble/elephant/">https://github.com/NeuralEnsemble/elephant/</a>
License	BSD
Current Version	<i>Release of analysis tools:</i> Elephant 0.3.0



---

*Anticipated: March 8, 2016 Integration on Collab: March 20, 2016*  
*(incl. spike sorting SpikeSort (0.12) or SpyKING CIRCUS (latest))*

---

All Versions 0.2.0, 0.1.0

### Description

Elephant (Electrophysiology Analysis Toolkit) is an open-source library for the analysis of electrophysiological data in the Python programming language. The focus of Elephant is on generic analysis functions for spike train data and time series recordings from electrodes, such as local field potentials (LFP) or intracellular voltages. In addition to providing a common platform for analysis code from different laboratories, the Elephant project aims to provide a consistent and homogeneous analysis framework built on a modular foundation.

The Python software is available for use on the Collaboratory using the Task system or as a Jupyter notebook.



### *EspINA*

EspINA is a user-friendly tool which performs segmentation and analysis of elements in a reconstructed 3D volume of the cerebral cortex, and greatly facilitates and accelerates these processes. It allows visualisation and segmentation of large image stack datasets, both from electron microscopy (e.g. FIB/SEM) and light microscopy.

#### Metadata

Category	Application
Tags	image-segmentation microscopy analysis
Partners	Universidad Politécnica de Madrid École Polytechnique Fédérale de Lausanne
Maintainers	<a href="#">Felix de las POZAS</a>
Contributors	Jose María (Chema) PEÑA Felix DE LAS POZAS Jorge PEÑA Pol MONSO Carlos BECKER Juan MORALES Pablo Marquez NEILA, Antonio LATORRE and Pascal FUA



---

Homepage	<a href="http://cajalbbp.cesvima.upm.es/espina/">http://cajalbbp.cesvima.upm.es/espina/</a>
Documentation	<a href="https://www.youtube.com/channel/UCN3kLTMxaJXkJrvsJbx-ww">https://www.youtube.com/channel/UCN3kLTMxaJXkJrvsJbx-ww</a>
Support	<a href="https://plus.google.com/115332585861057402895/posts">https://plus.google.com/115332585861057402895/posts</a>
Source Code	<a href="https://launchpad.net/~cbbp/+archive/ubuntu/main">https://launchpad.net/~cbbp/+archive/ubuntu/main</a>
Source Code	<a href="https://launchpad.net/~cbbp/+archive/ubuntu/main">https://launchpad.net/~cbbp/+archive/ubuntu/main</a>
License	GPL v3
Current Version	2.1.2
All Versions	1 and 2.1.2

---

## Description

### *Explore*

**Natural 3D navigation over the orthogonal planes.** Fluent navigation through the different sections of the stack using three planar views. Simultaneous different channel visualisation can be performed in a customised staining and intensity/contrast properties. Navigation bookmarks helps to easily find stack spots.



### *Segment*

**Brain structures can be individually segmented to later analysis.** Identify as many elements as you want using our classification or expand it as needed. Reconstruct them using our manual or semi-automated tools.

To this end we have introduced a new method for synapse detection that leverages available 2D or 3D context information to provide high-quality segmentations. For mitochondria, we used a similar approach combined with structured prediction to enforce common geometric constraints, such as connectivity, and relationships, such as being surrounded by a membrane.

In parallel, we have been working on Active Learning techniques to reduce the amount of effort required to train the Statistical Machine Learning algorithms they rely on. We have presented a novel approach to Active Learning that is geared towards segmenting 3D image volumes and also applicable to ordinary 2D images. By design, it takes into account geometric constraints to which regions should obey and makes the annotation process more user-friendly.

### *Edit*

**Extensive toolkit to improve segmentations.** If the result of the segmentation is not completely satisfactory, we provide a set of tools to enhance their morphology automatically or manually.

### *Visualise*

**Different 2D and 3D representations.** Geometrical features of segmented elements are relevant to their function. Different 3D visualisations provide detailed views for every segmented element.

### *Analyse*

**Obtain multiple measures from segmented structures.** The final step of the working session is the extraction of measurements for each object segmented. A list of morphometric features and parameters can be selected to be easily exported to standard .csv or .xls spreadsheet format.

## **Functionality**

- Image stack navigation.
- Multi channel visualisation.
- Metadata information: Acquisition protocol, specimen, region, provenance information, etc.
- Manual, assisted and automatic segmentation of structures.
- Editing and refinement tools.



- Visualisation in 2D and 3D representations.
- Analysis and reporting of segmented objects.
- Plugin-based extensibility.

## Technical aspects

### *Development*

EspINA is implemented in C++ as a standalone client-based tool. It required a series of external libraries:

- Qt 4.8
- Boost
- VTK (Visualization Toolkit)
- ITK (Insight Toolkit)

We developed an EspINA plug-in that integrates our methods for synapse and mitochondria segmentation. This plugin was improved following the requests and recommendations of the UPM neuroscientists. We also upgraded it for two subsequent versions of EspINA that had a new, backward-incompatible API.





## *Papaya*

Medical research web-based image viewer, with extended functionality to retrieve context-sensitive 2D graphical and textual context information for brain areas.

### Metadata

Category	Application
Maintainers	<a href="#">Research Imaging Institute, University of Texas Health Science Center</a>
Contributors	<a href="#">T. DICKSCHEID</a> <a href="#">S. EICKHOFF</a> <a href="#">K. AMUNTS</a>
Homepage	<a href="http://ric.uthscsa.edu/mango/papaya.html">http://ric.uthscsa.edu/mango/papaya.html</a>
Documentation	<a href="#">Online Userguide</a> <a href="#">Online Developer Guide</a>
Support	
License	<a href="https://github.com/rri-mango/Papaya/blob/master/LICENSE">https://github.com/rri-mango/Papaya/blob/master/LICENSE</a>
Current Version	v0.8



---

All Versions

## Description

Papaya is a pure JavaScript medical research image viewer, supporting DICOM and NIFTI formats, compatible across a range of web browsers. This orthogonal viewer supports overlays and atlases. For HBP, the viewer is being extended by 2D overlay functionality to enable retrieval and display of rich context-sensitive information for brain areas, such as receptor plots, quantitative data in the form of tables, and 2D linear and polar plots. The feature developments are anchored in SP2. The viewer will be integrated by SP5 as an app into the Collaboratory for exploring human brain atlas data.



## *JuBrain Cytoviewer*

JuBrain Cytoarchitectonic Atlas Viewer.

### Metadata

Category

Service

Maintainers

[Katrin AMUNTS](#)

[Hartmut MOHLBERG](#)

Contributors

Homepage

<http://jubrain.fz-juelich.de>

Documentation

[JuBrain Website](#)

[List of supported client technologies](#)

Support

[Hartmut MOHLBERG](#)

License

[Copyright information](#)

Current Version

Atlas data Version 1.0, Webviewer application version 2.1

All Versions



## Description

The JuBrain CytoViewer shows cytoarchitectonic probabilistic maps, which have been developed in the teams of Katrin AMUNTS and Karl ZILLES in Juelich and Duesseldorf (INM-1). Cytoarchitectonic areas were analysed in histological sections of ten human post-mortem brains. The maps are based on image analysis and statistical criteria for localising areal borders. Cytoarchitectonic maps have been developed during the past 20 years as a joint effort of many doctoral students, post-docs and guests.

## References for the JuBrain CytoViewer

Mohlberg, H., Eickhoff, S.B., Schleicher, A., Zilles, K., & Amunts, K. (2012). A new processing pipeline and release of cytoarchitectonic probabilistic maps - JuBrain, OHBM 2012, Peking, China.

## References for probabilistic cytoarchitectonic mapping

Amunts, K., Schleicher, A., & Zilles, K. (2007). Cytoarchitecture of the cerebral cortex - more than localization. *Neuroimage*, 37: 1061-1065.

Zilles, K., & Amunts, K. (2010). Centenary of Brodmann's map - conception and fate. *Nature Reviews, Neuroscience*, 11: 139-145.

Zilles, K., Schleicher, A., Palomero-Gallagher, N., & Amunts, K. (2002). Quantitative analysis of cyto- and receptor architecture of the human brain. In: "Brain Mapping: The Methods", Mazziotta, J.C., & Toga, A. (eds.), USA: Elsevier, 2002, p. 573-602.



### *Morphology Viewer*

Web-based interactive 3D visualisation of neuron morphologies, downloaded from local hard drive or via the NIP. Selected parts of the neuron can be switched on/off. Supported formats are SWC and Neurolucida (asc, dat, nrx or xml).

#### Metadata

Category	Service
Maintainers	<a href="#">Rembrandt BAKKER</a>
Contributors	<a href="#">Rembrandt BAKKER</a> , <a href="#">Paul TIESINGA</a>
Homepage	<a href="https://neuroinformatics.nl/HBP/morphology-viewer/">https://neuroinformatics.nl/HBP/morphology-viewer/</a>
Documentation	<a href="https://neuroinformatics.nl/HBP/morphology-viewer/">https://neuroinformatics.nl/HBP/morphology-viewer/</a>
Support	hbp@scalablebrainatlas.org
Source Code	<a href="https://github.com/rbakker/HBP-morphology-viewer">https://github.com/rbakker/HBP-morphology-viewer</a>
License	GPLv3
Current Version	1
All Versions	1



## Description

The morphology viewer is a web-based service that interactively displays a 3D visualisation of neuron morphologies, and allows parts of the neuron to be switched on/off to allow a detailed inspection of the morphology. Supported formats are SWC and NeuroLucida (asc, dat, nrx or xml).

There are two user scenarios. The first is to use the service as a standalone viewer, whereby the user visits the website, loads a morphology file from his own computer, and views the neuron. The second is to use the service from within the NIP, whereby search queries that return neuron morphologies link directly to the service.

Example: [direct link](#).



## *AlignII*

Web-based tool for anchoring 2D image data into 3D atlas space.

### Metadata

Category	Service
Maintainers	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Dmitri DARINE <a href="mailto:d.a.darine@medisin.uio.no">d.a.darine@medisin.uio.no</a>
Contributors	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Dmitri DARINE <a href="mailto:d.a.darine@medisin.uio.no">d.a.darine@medisin.uio.no</a> Trygve B. LEERGAARD <a href="mailto:t.b.leergaard@medisin.uio.no">t.b.leergaard@medisin.uio.no</a> Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
Homepage	<a href="http://www.rodentbrainworkbench.org">http://www.rodentbrainworkbench.org</a> <a href="http://www.rbwb.org">www.rbwb.org</a>
Documentation	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>
Support	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>
License	AlignII is embedded in the Rodent Brain Navigator data system. Access for partners in the Human Brain Project. Contact: Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>





---

Current Version     1.0

---

All Versions        1.0

## Description

AlignII is web-based tool for user guided registration (anchoring) of 2D experimental image data, typically high resolution microscopic images, to 3D atlas reference space (Waxholm Space Mouse / Rat Atlas or Allen Mouse Brain Atlas), thus facilitating data integration through standardised coordinate systems. AlignII is integrated in the web-based Rodent Brain Navigator data system - a data system for internal use in the HBP, developed and maintained in collaboration with the INCF Norwegian Node.

A key feature in the tool is the capability to generate user-defined cut planes through the atlas templates, matching the orientation of the cut plane of the 2D experimental image data, as a first step towards anchoring of images to the relevant atlas template.

AlignII supports multi-modality 3D atlas packages and one-click switching between image modalities (e.g., MRI and DTI) while preserving the view location and UI settings in general, and in-depth numerical control over cut plane settings. The tool has been tested extensively and is improved based on user feedback. A series of research projects are currently using the tool for registration of project data to standardised atlas space.

***QuickNII***

Standalone tool for anchoring 2D image data into 3D atlas space.

**Metadata**

Category	Application
Maintainers	<a href="#">Gergely CSUCS</a>
Contributors	Jan BJALLIE Trygve LEERGAARD Gergely CSUCS
Homepage URL	<a href="http://www.rodentbrainworkbench.org">http://www.rodentbrainworkbench.org</a> <a href="http://www.rbwb.org">www.rbwb.org</a>
Documentation	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>
Support	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>
License	Access for partners of the Human Brain Project. Contact: Jan G. BJAALIE
Current Version	1.0
All Versions	1.0



## Description

QuickNII is a standalone tool for user-guided registration (anchoring) of 2D experimental image data, typically high resolution microscopic images, to 3D atlas reference space (Waxholm Space Mouse / Rat Atlas or Allen Mouse Brain Atlas ), thus facilitating data integration through standardised coordinate systems.

A key feature in the tool is the capability to generate user-defined cut planes through the atlas templates, matching the orientation of the cut plane of the 2D experimental image data, as a first step towards anchoring of images to the relevant atlas template.

QuickNII supports multi-modality 3D atlas packages and one-click switching between image modalities (e.g., MRI and DTI) and structural parcellations, while preserving the view location and UI settings in general and in-depth numerical control over cut plane settings. The tool has been tested extensively and is improved based on user feedback. A series of research projects are currently using the tool for registration of project data to standardised atlas space.

QuickNII is a standalone iteration of AlignII where the volumetric templates for the rat brain (MRI, DTI and structural parcellations for the Waxholm Space Rat Atlas) and mouse brain (volumetric histology and structural parcellations for the Allen Mouse Brain Atlas) are brought to the user's computer. This practice eliminates the network roundtrip, and thus results in near real-time operation with high refresh rate of the navigational and user-defined cut plane renderings. Other aspects of user experience are preserved with the same controls, views and general look-and-feel as provided by AlignII. While QuickNII is a standalone tool, steps are taken in order to integrate it in the workflow provided by the Rodent Brain Navigator data system as well as in other HBP image services.

***MeshGen***

Mesh generator for volumetric segmentation data.

**Metadata**

Category	Application
Maintainers	<a href="#">Gergely CSUCS</a>
Contributors	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
Homepage URL	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
Documentation	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
Support	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
License	CC-BY-NC-SA
Current Version	1.0
All Versions	1.0



## Description

MeshGen generates separate mesh files for each label present in a segmentation volume. The generation employs a variant of Constrained Elastic Surface Nets (CESN), which has been modified in order to avoid intersection between neighbouring meshes.

Supported input formats:

- NIfTI for segmentation volume, with integer elements (8/16/32 bits, signed/unsigned)
- ITK label file (ITK-Snap compatible, does not contain hierarchy)
- ILF - Integrated Label File (MBAT compatible, hierarchical)
- Brain-Map XML file (ontology from brain-map.org, hierarchical)

When label hierarchy is available, MeshGen can follow containment relations, e.g. eliminating generation of extra inner contours around contained labels. The tool can optionally eliminate contained labels if their colour matches with the colour of their parent, e.g. when rendering or not rendering the meshes using their original colours would make no visual difference.

In order to provide compact mesh output, MeshGen can apply edge elimination algorithms on generated data. One algorithm eliminates edges based on the curvature of neighbouring surface patch. A second algorithm eliminates edges based on their length.

Online results/examples using the companion viewer tool, MeshView:

- Waxholm Space Atlas of the Sprague Dawley rat brain (meshes for 79 anatomical structures - 80 with inner ear) [WHSRat]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSRatV2.bitlas>
- Waxholm Space Atlas of the adult C57BL/6J mouse brain (meshes for 26 anatomical structures - 27 with inner ear) [WHSMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSMouseHier.bitlas>
- Allen Mouse Brain Atlas reference atlas (meshes for 106 anatomical structures including hierarchy) [ABAMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=ABAMouseHier.bitlas>



### *MeshView*

Web-based mesh viewer with user-defined cut plane support.

#### Metadata

Category	Service
----------	---------

Maintainers	<a href="#">Gergely CSUCS</a>
-------------	-------------------------------

Contributors	Gergely CSUCS <a href="mailto:gergely.csucs@medisin.uio.no">gergely.csucs@medisin.uio.no</a> Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Homepage URL	<a href="http://www.rodentbrainworkbench.org">http://www.rodentbrainworkbench.org</a> <a href="http://www.rbwb.org">www.rbwb.org</a>
--------------	-----------------------------------------------------------------------------------------------------------------------------------------

Documentation	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
---------------	---------------------------------------------------------------------------------------------------

Support	<a href="http://software.incf.org/software/meshgen">http://software.incf.org/software/meshgen</a>
---------	---------------------------------------------------------------------------------------------------

License	MeshView is presented through the HBP services. Open access. Contact: Jan G. BJAALIE <a href="mailto:j.g.bjaalie@medisin.uio.no">j.g.bjaalie@medisin.uio.no</a>
---------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

Current Version	0.9
-----------------	-----



---

All Versions      0.9

## Description

Web application for real-time 3-D display of mesh-data representing structural parcellations, and generation of user defined cut planes.

MeshView is provided with different atlases, including the Allen Mouse Brain Atlas and Waxholm Space atlases for mouse and rat:

- Waxholm Space Atlas of the Sprague Dawley rat brain (meshes for 79 anatomical structures - 80 with inner ear) [WHSRat]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSRatV2.bitlas>
- Waxholm Space Atlas of the adult C57BL/6J mouse brain (meshes for 26 anatomical structures - 27 with inner ear) [WHSMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=WHSMouseHier.bitlas>
- Allen Mouse Brain Atlas reference atlas (meshes for 106 anatomical structures including hierarchy) [ABAMouse]:  
<http://www.nesys.uio.no/MeshGen/MeshView.html?bitlas=ABAMouseHier.bitlas>

In the final release, the aim is to support viewing of anchored 2D image data, and viewing of the locations of any data set.



***CutNII***

Standalone viewer for volumetric atlases with custom-angle slicing support.

**Metadata**

Category	Application
Maintainers	<a href="#">Gergely CSUCS</a>
Contributors	Jan BJALLIE Trygve LEERGAARD Gergely CSUCS
Homepage URL	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>
Documentation	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>
Support	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>
License	CC-BY-NC-SA
Current Version	1.0
All Versions	1.0



## Description

CutNII is a limited, standalone version of QuickNII where the volumetric templates for the rat brain (MRI, DTI and structural parcellations for the Waxholm Space Rat Atlas) and mouse brain (volumetric histology and structural parcellations for the Allen Mouse Brain Atlas) are brought to the user's computer.

A key feature in the tool is the capability to generate user-defined cut planes through the atlas templates, matching the orientation of the cut plane of the 2D experimental image data. CutNII supports multi-modality 3D atlas packages and one-click switching between image modalities (e.g., MRI and DTI) and structural parcellations, while preserving the view location and UI settings in general, and in-depth numerical control over cut plane settings.

Two software packages are available:

- CutNII for Waxholm Space atlas of the Sprague Dawley rat brain: This package contains the CutNII software bundled with pre-processed data from Waxholm Space atlas of the Sprague Dawley rat brain. T2\* MRI, DTI, and delineation volumes are provided.
- CutNII for Allen Mouse Brain Atlas reference atlas: This package contains the CutNII software bundled with pre-processed data from Allen Mouse Brain Atlas reference atlas. Grayscale Nissl volume and annotation volume are provided.



### ***Scalable Brain Atlas***

The Scalable Brain Atlas (SBA) is a fully web-based display engine for brain atlases, imaging data and topologies. It allows client websites to show brain region-related data in a 3D interactive context and provides services to look up regions, generate thumbnails or download nomenclature- and delineation data.

#### **Metadata**

Category	Service
Maintainers	<a href="#">Rembrandt BAKKER</a>
Homepage	<a href="http://scalablebrainatlas.incf.org/main/index.php?">http://scalablebrainatlas.incf.org/main/index.php?</a>
License	<a href="http://scalablebrainatlas.incf.org/main/citationpolicy.php">http://scalablebrainatlas.incf.org/main/citationpolicy.php</a>
Current Version	1
All Versions	1

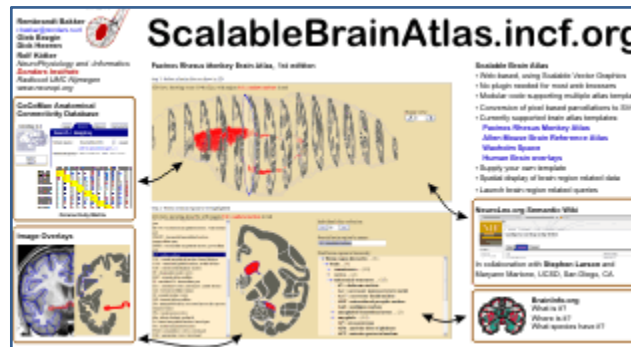


Figure 42: The Scalable Brain Atlas

## Description

The Scalable Brain Atlas (SBA) is a fully web-based display engine for brain atlases, imaging data and ontologies. It is described in this publication:

Bakker, R., Tiesinga, P., & Kötter, R. (2015). The Scalable Brain Atlas: instant web-based access to public brain atlases and related content. *Neuroinformatics*, 13: 353-366. doi: 10.1007/s12021-014-9258-x.

Specifically added in the context of the Human Brain Project is a standardised set of anatomical landmarks for registration of whole brain experimental image data from the mouse and rat brain to Waxholm Space. The use of standardised landmarks in combination with a common coordinate system for the rodent brain increases speed and accuracy of data integration.

Internal landmarks of the C57BL/6J mouse brain and the Sprague Dawley rat brain (Waxholm Space templates) have been identified across different MRI contrasts (T1, T2, T2\*) and other modalities in different specimens, and at different slice acquisition angles and different image resolutions. The use of these landmarks increases the accuracy of registration of experimental volumetric datasets to atlas templates and thereby facilitates comparisons of data from different experiments and data analysis requiring use of standardised atlas space.

The use of landmarks for registration is described in this publication:

Sergejeva, M., Papp, E.A., Bakker, R., Gaudnek, M.A., Okamura-Oho, Y., Boline, J., Bjaalie, J.G., & Hess, A. (2015). Anatomical landmarks for registration of experimental image data to volumetric rodent brain atlasing templates. *Journal of Neuroscience Methods*, 240:161-9. doi: 10.1016/j.jneumeth.2014.11.005.



## *SynapseGenerator*

SynapseGenerator is a pipeline used to predict connectivity at the synapse level.

### Metadata

Category	Library
Maintainers	<a href="#">Rembrandt BAKKER</a>
Contributors	<a href="#">Max BAKKER</a> <a href="#">Rembrandt BAKKER</a> <a href="#">Paul TIESINGA</a>
Homepage	<a href="http://www.neuroinformatics.nl/SynapseGenerator/index.html">http://www.neuroinformatics.nl/SynapseGenerator/index.html</a>
Documentation	<a href="http://www.neuroinformatics.nl/SynapseGenerator/index.html">http://www.neuroinformatics.nl/SynapseGenerator/index.html</a>
Source Code	<a href="https://collab.humanbrainproject.eu/#/collab/47/nav/2784?state=uuid%3D79042414-6074-4daf-a4c0-6b9ead269cfe">https://collab.humanbrainproject.eu/#/collab/47/nav/2784?state=uuid%3D79042414-6074-4daf-a4c0-6b9ead269cfe</a>
License	GPLv3
Current Version	0.9+4
All Versions	0.9, 0.9+1, 0.9+2, 0.9+3



## Description

SynapseGenerator is a pipeline used to predict connectivity at the synapse level based on statistical properties of all neuron populations that co-exist in the given brain region of interest. For each population, the spatial distribution of the dendritic and axonal arborisations needs to be either specified as a probability density function (e.g. a mixture of Gaussians), or can be estimated from samples of reconstructed dendritic/axonal arborisations.

The pipeline returns the microconnectivity between individual neurons, thereby specifying the branch order at which they connect.

SynapseGenerator is written in Matlab, source files are available [for download](#).



## *Knowledge Space*

KnowledgeSpace (KS) is a community-based encyclopaedia that provides living review articles, literature, neuroscience ontologies and the links between neuroscience concepts and a diversity of federated data sources. These include HBP KnowledgeGraph, Neuromorpho.org, Allen cell type morphologies, Allen cell type electrophysiology, Neuroelectro.org, Cell Image Library, NIF Integrated Connectivity, ChannelPedia.net, Ion Channel Genealogy, Blue Brain Neocortical Microcircuit, ModelDB, OpenSourceBrain, GenSat. This list will be extended in an ongoing manner.

### Metadata

Category	Resource
Maintainers	The KnowledgeSpace consortium ( <a href="http://www.knowledge-space.org">www.knowledge-space.org</a> )
Contributors	Sean HILL, Maryann MARTONE, Tom GILLESPIE, Mathew ABRAMS, Linda LANYON, Jeff GRETHE, Willy WONG, Catherine ZWAHLEN, Renaud RICHARDET
Homepage	<a href="https://www.knowledge-space.org">https://www.knowledge-space.org</a>
Documentation	<a href="https://www.knowledge-space.org/NeuroKS/index.php/Documentation">https://www.knowledge-space.org/NeuroKS/index.php/Documentation</a>
Source Code	<a href="https://github.com/NeuroscienceKnowledgeSpace/KnowledgeSpace">https://github.com/NeuroscienceKnowledgeSpace/KnowledgeSpace</a>
License	CC-by-4.0
Current Version	First public release





---

All Versions

## Description

KnowledgeSpace (KS) is a community-based encyclopaedia for neuroscience, which links a common vocabulary for neuroscience concepts with data, models and literature available from sources around the world. KS will allow neuroscience community contributions and will support data curation, annotation, and text mining.

KS is the result of recommendations from a community workshop held by the INCF Program on Ontologies of Neural Structures in 2012; the report of this workshop can be found here:

[2012 INCF Workshop Report](#)

## Annex C: Summary - Platform Use Case Status

In Appendix II, you will find a listing of all the names of Use Cases and Use Case IDs.

- 1) Organize 2D serial microscopic images in web-atlas (Tool = Navigator3):
  - a) User Daniel is a data provider and has produced data for the purpose of visualising brain architecture and mapping the morphology of specific cellular elements in a 3D brain atlas. X wants to organise and analyse a series of microscopic images, and publicly share a selection of images in a web-based virtual microscopy viewing tool.
  - b) Daniel prepares images files in tiff format (convention = case#serial#\*.tif) and experimental metadata.
  - c) Daniel requests a user account on an HBP data management system (Rodent Brain Navigator3), uploads images (ftp) to Navigator through a web interface, resulting in automatic image conversion and storage of primary and derived image data in the Navigator.
  - d) In Navigator, Daniel has access to:
    - i) Native tiff images, pyramidal files for web viewing, thumbnails for preview
    - ii) Annotation editor for full resolution images (text, lines)
    - iii) Metadata editor
    - iv) 2D filmstrip viewers for virtual microscopy (single, dual, triple viewers available)
    - v) Anchoring tool (AlignII) for spatial registration of preview images to reference atlas and dual display of image and atlas diagrams [\[usecase 2\]](#)
  - e) Now Daniel can:
    - i) Enter metadata
    - ii) Add users to the project with different levels of privileges
    - iii) Make the project publicly (anonymously) read-only available. Unique URLs are available for providing web access to filmstrip viewers and download of files.
    - iv) Annotate images in web GUI or import XML graphics created in external software
    - v) Register images to reference atlas to create metadata coordinates describing image location in reference space [\[usecase 2\]](#)
    - vi) Inspect and analyse images in virtual microscopy viewers and atlasing tools
- 2) Anchor 2D serial images to 3-D standard brain reference atlas (Navigator3/ AlignII / QuickNII):



- a) User Arthur wants to register serial microscopic images to a standard 3-D reference atlas, and has two options:
  - i) Use web-based tool IigNII embedded in the data management system Navigator.
    - (1) Advantages: works in web-browser, metadata are directly coupled to images in the data management system.
    - (2) Disadvantage: some response time due to communication with server.
  - ii) Use standalone anchoring tool QuickNII.
    - (1) Advantages: images are stored locally, quick response time. Efficient procedure for registering large image series. Future integration with other data management systems.
    - (2) Disadvantages: Requires downloading of thumbnail images, and spatial metadata need to be submitted to data management system.
- b) Arthur chooses a registration tool and performs a step-wise registration process:
  - i) Select appropriate 3-D reference atlas package
  - ii) Identify spatial orientation of 2-D slices relative to 3-D reference template and assign orientation parameter to series of images
  - iii) Determine linear scaling and positioning of each image providing best fit to 3-D reference template
  - iv) Save positional parameters for each image as metadata in data management system
- c) Arthur can now:
  - i) Submit data to curation and uploading to the HBP portal [\[usecase 3\]](#)
  - ii) Analyse images [\[usecase 4\]](#)
- 3) Validate quality of image and spatial registration before sharing via HBP portal:
  - a) Christin is a data curator assigned the task of validating the quality of images and spatial registration of images provided by user X for the purposes of sharing via the HBP portal.
  - b) Data user Tom provides curator Christin with anchored image data with a request of validating and approving images for submission to the HBP portal.
  - c) Curator Christin performs the following validation steps:
    - Completeness of data / metadata registration using metadata editor available in data management system
    - Accuracy of anchoring to atlas using standalone tool QuickNII to evaluate positional parameters for image series in graphical plots, and spatial registration accuracy in selected images



Approves registration without corrections or after implementing minor adjustments OR rejects registration and provides corrective feedback to user X

At the moment these tasks are semi-automated in our prototype Platform. They will be automated during the SGA1 Phase.

- 4) Analyse spatial distribution of specific cellular elements across the brain (Filmstrip viewer / AlignII):
  - a) Data user Tom is an investigator interested in mapping the distribution of labelled cellular elements across the brain (e.g. axonal connections, cellular / molecular markers, e.g. neurotransmitter proteins, pathological markers, specific gene products etc.).
  - b) Tom surveys full-resolution images in filmstrip viewer tool and selects images with labelling of interest for analysis using web tool AlignII ([step c](#)) or computational image analysis of downloaded files ([step d](#)).
  - c) Anchoring tool AlignII is used to co-display experimental image and spatially anchored custom atlas plates, and qualitative analysis is performed:
    - i) Atlas delineations may be adjusted if necessary to match local anatomical boundaries for regions of interest.
    - ii) For each anatomical region of interest, the spatial distribution of relevant cellular markers are noted and amount / density of markers registered using a semi-quantitative scoring system.
  - d) Full resolution images and corresponding custom atlas plates are downloaded and co-displayed. Atlas delineations are used to define regions-of-interest. Suitable software (e.g. ImageJ, espINA, etc.) is used to extract and quantify labelling of interest (signal) from background (noise).
  - e) Derived images may optionally be added to the data management system for sharing along with original data and metadata.
- 5) Share full-resolution images and custom atlas diagrams in HBP portal:
  - a) Data curator Christin wants to share full resolution image together with anchored atlas delineations. Christin imports high-resolution native images, with corresponding custom atlas diagrams and metadata to HBP portal and makes these available in interactive viewer tool for combined display of full resolution images and atlas plates.
- 6) 3D web visualisation of complex brain anatomy in 3-D atlas volumes (MeshView):
  - a) Data user Tom wants to visualise the anatomy of a complex anatomical region (e.g. the hippocampus) in a 3-D spatial reference atlas to facilitate assignment of anatomical location in experimental image data.
  - b) Tom uses the MeshView web tool to inspect 3-D anatomical delineations from the Allen Mouse Brain Atlas or the Waxholm Space atlas of the mouse or rat brain. This tool provides 3-D surface rendering of anatomical structures, and user-defined arbitrary slices through the atlas volume.

- c) Tom identifies a slice orientation of interest (e.g. matching a series of 2D images) and generates a series of custom atlas plates through the entire brain with corresponding orientation and spacing.
- 7) Identification of expert-defined boundaries in the rat hippocampal region (Rat hippocampus atlas application):
  - a) Data user Tom needs to identify the boundaries of specific hippocampal subregion in the rat brain in experimental histological material.
  - b) Tom makes use of the Rat Hippocampus Atlas resource, which provides access to a repository of serial histological images (cut in standard coronal, sagittal, and horizontal orientations) showing salient aspects of brain cyto- and chemoarchitecture, to find a reference image with orientation and position matching the experimental image of interest.
  - c) Tom inspects the atlas images in an interactive web-microscope viewer allowing zoom, pan, and optional delineation overlay.
  - d) Tom looks up expert descriptions of the delineated structures in an associated web-encyclopaedic resource, including descriptions of cyto- and chemoarchitectonic criteria used to identify the boundaries between hippocampal subregions.
- 8) 3D analysis of volumes (CutNII):
  - a) Data provider Daniel has generated a volumetric data set and wants to compare 3-D cellular marker distributions with data available in 2-D sectional images registered to a brain atlas template.
  - b) Daniel performs a 3D non-linear registration of the image volume to a 3D reference space.
  - c) Daniel uses the stand-alone tool CutNII to slice the volumetric histology volume in arbitrary slices, and inspects marker distributions.
  - d) Daniel makes use of previously defined spatial parameters (defined when 2-D sectional images were anchored to the reference atlas template [\[see usecase 2\]](#)) to slice the experimental 3-D volume in slices matching the 2-D histological series.
  - e) Images from corresponding atlas levels are visually compared (side-by-side or by overlay in a suitable image software).

**Table 1: Use cases and success scenarios (1)**

Use Case ID	Success Scenario	Status/comments	ID of Related Functional Requirement	Related Product/Software Package/Service	TRL at end of Ramp-Up Phase	Contributors
SP5-WP5.1_001: Interpretation of anatomical location in rodent brain image	Users achieve improved guidance to analysis of anatomical location in mouse and rat brains.	Tools have been extensively tested and are publicly shared. MeshView tools provide	SP5NIP-FR 012 SP5NIP-FR 013	CutNII MeshView	5-6	UIO

material	Interactive web-based tools (CutNII and MeshView) are used to slice standard 3D rat / mouse reference atlases in user-defined slices corresponding with experimental image material. A web-application provides interactive access to expert delineations of histological section images and descriptions of boundary defining criteria.	access to unique 3D versions of the Allen Mouse Brain Atlas, and the Waxholm Space atlas of the mouse / rat brain. The rat hippocampus atlas provides access to delineated images and expert defined boundary descriptions.				
SP5-WP5.1_002: Brain-wide connectivity mapping in rat somatosensory system	A web-based repository of 2-D and 3-D histological images showing the brain-wide distribution of axons labelled by tracer injection in the rat primary somatosensory cortex has been established. Images have been anchored to the Waxholm Space atlas of the rat brain and co-displayed in virtual microscopy viewers, facilitating assignment of anatomical location to observed axonal labelling.	The image repository is publicly shared and provides combined visualization of images and correspondingly sliced reference atlas plates. Semi-quantitative analyses provide a complete overview of S1 efferent projections.	SP5NIP-FR 012 SP5NIP-FR 013	Publicly available web-based tools and data packages accessible via HBP portal	3	UiO / EPFL
SP5-WP5.1_003: Viewing	Expression of	The image repositories	SP5NIP-FR 012	Publicly available web-	5	UiO

and interpreting the brain-wide spatial distribution of cellular labelling in transgenic mouse models	tetracycline-dependent (tTA) promoter genes is visualised in histological sections from different tTA driver lines, and assembled in web-based atlas repositories. Section images are anchored to the standard Allen mouse brain reference space using the AlignII or QuickNII tools and the spatial distribution and amount of labelling semi-quantitatively analysed across entire brains.	comprise five different driver lines. Three of these are publicly shared, while data from two additional lines are available to HBP collaborators. Knowledge of the spatial distribution of different tTA-promoter expression patterns is important for generating new conditional transgenic mouse models, and for interpreting data from such models. Semiquantitative analyses provide brain-wide overview of promoter expression distribution frequently used tTA driver lines (Prion protein; Calmodulin-dependent protein kinase II; Neuropsin; Pituitary homeobox 3 protein; Purkinje cell protein 2)	SP5NIP-FR 013	based tools and data packages accessible via HBP portal		
SP5-WP5.1_004: Multimodal, volumetric atlas of rat brain receptor- and fibre architecture	High-resolution, volumetric images showing receptor densities and fibre orientations are spatially registered to the Waxholm Space atlas of the rat brain using non-linear transformation, constituting a multiscale	Volumetric data are publicly accessible through the HBP portal. HBP viewer tools (CutNII) allow arbitrary slicing of the data volume, facilitating inspection of spatial distribution patterns and comparison with other	SP5NIP-FR 012 SP5NIP-FR 013	Publicly available web-based tools and data packages accessible via HBP portal	2-3	Juelich / UiO / EPFL



	and multimodal model of rat brain architecture.	section based image data registered to the same reference space.				
--	-------------------------------------------------	------------------------------------------------------------------	--	--	--	--



Table 2: Use cases and success scenarios (2)

Use Case ID	Success Scenario	Status/comments	ID of Related Functional Requirement	Related Product/Software Package/Service	TRL at end of Ramp-Up Phase	Contributors
SP5NIP-UC-001 WP5.1 Sean HILL	<p>Scientific group Ajax is producing data it would like to share with the HBP Consortium or other collaborators.</p> <p>The technical contact from Ajax works with the DataSpace team to install and configure DataSpace software.</p> <p>The Ajax team federates their data with the global DataSpace federation (managed by INCF).</p> <p>The Ajax team can set read/write privileges to allow global access to their data repository through the INCF login.</p> <p>Scientific user Bill can now download or uses Portal tasks to work with the morphologies.</p>	<p>DataSpace (v1) is developed and deployed, and useful for all elements of the use case except the integration with the Portal (Collaboratory) due to technical challenges/limitations integrating with HBP AAI.</p> <p>Zenodo.org serves as a permanent archive repository currently.</p>	<p>SP5NIP-FR 003</p> <p>SP5NIP-FR 004</p>	<p>dataspace.incf.org</p> <p>zenodo.org</p>	TRL 9	
SP5NIP-UC-002 WP5.1 Sean HILL	Scientific group Ajax would like to explicitly register an existing data	Completed	<p>SP5NIP-FR 001</p> <p>SP5NIP-FR 002</p>		TLR3	<p>Sean HILL</p> <p>Yury KATKOV</p>



	<p>set to make it available to the DataSpace federation.</p> <p>The Ajax team executes a register command on the data set, which adds the essential metadata to the global metadata catalogue.</p> <p>Other members of the data federation can now see and access the data set.</p>		SP5NIP-FR 004			<p>Lomig MÉGARD</p> <p>Samuel KERRIEN</p> <p>Catherine ZWAHLEN</p>
<p>SP5NIP-UC-003</p> <p>WP5.1 Sean HILL</p>	<p>Scientific group Ajax would like to annotate a data set with free text and metadata - providing the context of data acquisition and sufficient metadata to characterise measurement parameters.</p> <p>The Ajax team creates an XML template file containing the metadata in a standard form.</p> <p>A script is executed which parses the XML metadata file and populates a wiki page with a template page describing the data, using the metadata</p>	Obsolete use case				



	<p>tags.</p> <p>The Ajax team can further annotate the data with text in a wiki environment.</p>					
<p>SP5NIP-UC-004</p> <p>WP5.1 Sean HILL</p>	<p>Scientific group Ajax would like to annotate a data set with free text and metadata - providing the context of data acquisition and sufficient metadata to characterise measurement parameters.</p> <p>The Ajax team selects a location in an XML template file containing the metadata in a standard form.</p> <p>A script is executed which parses the XML metadata file and populates a wiki page with a template page describing the data, using the metadata tags.</p> <p>The Ajax team can further annotate the data with text in a wiki environment.</p>	Duplicate of above				
SP5NIP-UC-005	Data curation team Omega establishes a	Obsolete use case - KnowledgeSpace				



WP5.1 Sean HILL	<p>standardised curation workflow for data from scientific group Ajax.</p> <p>The Omega team defines a standard XML metadata template file for scientific group Ajax.</p> <p>Ajax submits initial annotated data to the wiki using an annotation workflow.</p> <p>Omega team reviews data for completeness and quality.</p> <p>Omega team edits KnowledgeSpace entries to provide rich annotation around key data sets and ensures all tags conform to standard ontological terminology, data types, previews, etc.</p>	provides an interface to curated datasets, it does not directly support the curation of specific datasets.				
SP5NIP-UC-006 WP5.1 Sean HILL	<p>Data curation team Omega curates data with a previously unused data type.</p> <p>Omega creates a data type definition in the Knowledge Graph.</p> <p>The data type is defined within a data type ontology (relating the</p>	Implemented and tested (internal to HBP)	SP5NIP-FR 004		TLR4	<p>Sean HILL</p> <p>Martin TELEFONT</p> <p>Emily CLARK</p>

	new data type to other key types - image, waveform, event, video, sound, etc.).					
SP5NIP-UC-007 WP5.1 Sean HILL	<p>Text mining team Alpha extracts facts, assertions, parameters, etc. from published literature.</p> <p>The text mining service uses common ontological terms to populate the KnowledgeSpace with the mined data.</p> <p>The mined data are annotated with the text mining technique, version and data source.</p>	The text mining output for mined connectivity statements has been integrated into the KnowledgeSpace as the BlueBrain Bluima Connectivity dataset. This appears under the Anatomy tab of the Dataspace.	SP5NIP-FR 001 SP5NIP-FR 002 SP5NIP-FR 004 SP5NIP-FR 012		TLR4	Sean HILL Renaud RICHARDET
SP5NIP-UC-008 WP5.1 Sean HILL	<p>A scientific user wishes to search for data, models or literature.</p> <p>The user provides free text or specific semantic tags to define the search.</p> <p>The unified search service returns a collection of objects consistent with the search query.</p>	Implemented and tested	SP5NIP-FR 001 SP5NIP-FR 002 SP5NIP-FR 004 SP5NIP-FR 008 SP5NIP-FR 009		TLR4	Sean HILL Renaud RICHARDET



SP5NIP-UC-009 WP5.1 Sean HILL	<p>Scientific user Abigail would like to build a model of a neuron type brain model of a cortical pyramidal neuron.</p> <p>She searches for all morphological data for cortical pyramidal neurons.</p> <p>She searches for all electrophysiological data for cortical pyramidal neurons.</p> <p>She searches for all gene expression data for cortical pyramidal neurons.</p> <p>She selects the data that has been curated and identified as ready for model building.</p> <p>With the search results she now can begin her workflow in the Brain Simulation Platform.</p> <p>She uses data that has been curated and identified as ready for validation use to validate the model.</p>	Implemented and tested	SP5NIP-FR 001 SP5NIP-FR 002 SP5NIP-FR 003 SP5NIP-FR 004 SP5NIP-FR 005 SP5NIP-FR 006 SP5NIP-FR 007 SP5NIP-FR 008 SP5NIP-FR 009 SP5NIP-FR 010 SP5NIP-FR 011 SP5NIP-FR 012		TLR4	Sean HILL François Mohamet SY Samuel KERRIEN Emily CLARK Martin TELEFONT Catherine ZWAHLEN
SP5NIP-UC-009 WP5.1 Sean HILL	Scientific user Abigail would like to build a model of a neuron type	Duplicate	SP5NIP-FR 001 SP5NIP-FR 002			

	<p>brain model of a cortical pyramidal neuron.</p> <p>She searches for all morphological data for cortical pyramidal neurons.</p> <p>She searches for all electrophysiological data for cortical pyramidal neurons.</p> <p>She searches for all gene expression data for cortical pyramidal neurons.</p> <p>She selects the data that has been curated and identified as ready for model building.</p> <p>With the search results she now can begin her workflow in the Brain Simulation Platform.</p> <p>She uses data that has been curated and identified as ready for validation use to validate the model.</p>		<p>SP5NIP-FR 003</p> <p>SP5NIP-FR 004</p> <p>SP5NIP-FR 005</p> <p>SP5NIP-FR 006</p> <p>SP5NIP-FR 007</p> <p>SP5NIP-FR 008</p> <p>SP5NIP-FR 009</p> <p>SP5NIP-FR 010</p> <p>SP5NIP-FR 011</p> <p>SP5NIP-FR 012</p>			
<p>SP5NIP-UC-010</p> <p>WP5.1 Sean HILL</p>	<p>Scientific user Csaba would like to build a model of a whole mouse brain.</p> <p>He searches for cell densities for each brain</p>	<p>The VoxelAPI partially addresses this use case. The Brain Simulation platform requirements are still being refined and the APIs will support those when ready. In</p>	<p>SP5NIP-FR 001</p> <p>SP5NIP-FR 002</p> <p>SP5NIP-FR 003</p> <p>SP5NIP-FR 004</p> <p>SP5NIP-FR 005</p>		<p>TLR4</p>	<p>Sean Hill</p> <p>Samuel KERRIEN</p> <p>Catherine ZWAHLEN</p>



	<p>region.</p> <p>He searches for cell-type gene expression markers for each brain region, from literature and gene-expression databases.</p> <p>He constructs a predictive model for cell-type densities for each brain region.</p> <p>He searches for whole brain DTI data for macro-scale connectivity.</p> <p>He searches for whole brain tracer injection data for fine-scale connectivity.</p> <p>He constructs a predictive model for whole brain neuron connectivity.</p> <p>He selects the data that have been curated and identified as ready for model building.</p> <p>He searches for literature-mined properties for each brain area to further parameterise or validate the model.</p>	<p>addition, the link between the Atlas concepts and federated data sources in KnowledgeSpace completes this use case.</p>	<p>SP5NIP-FR 006</p> <p>SP5NIP-FR 007</p> <p>SP5NIP-FR 008</p> <p>SP5NIP-FR 009</p> <p>SP5NIP-FR 010</p> <p>SP5NIP-FR 011</p> <p>SP5NIP-FR 012</p>			
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--



	With the search results he now can begin his workflow in the Brain Simulation Platform.					
SP5-UC-011 WP5.1 Sean HILL	<p>Daniel would like to create a brain atlas from a stack of high-resolution 2D images located on his computer, and share the brain atlas online.</p> <p>Daniel creates a project - Project1. He is now the owner of Project1.</p> <p>Daniel uploads 2D images to Project1.</p> <p>Daniel is requested to enter the metadata (HBP-CORE or PAB).</p> <p>Daniel starts the converter via Web GUI. The converter creates pyramids, previews and thumbnails from the 2D images. This might be a time consuming task, Daniel might log out and login later to check the progress.</p> <p>The conversion is finished. Now the following extra tools and features are automatically available</p>	Largely complete - not as fully automated as described.	SP5NIP-FR 001 SP5NIP-FR 002 SP5NIP-FR 003 SP5NIP-FR 004 SP5NIP-FR 005 SP5NIP-FR 006 SP5NIP-FR 007 SP5NIP-FR 008 SP5NIP-FR 009 SP5NIP-FR 010 SP5NIP-FR 011 SP5NIP-FR 012		TLR5	Sean HILL Stanislaw ADASZEWSKI Tsolmongerei PAPILLOUD Raphael DUMUSC



via unique URLs for Daniel only:

- Original 2D images, pyramids, thumbnails and previews.
- 2D image viewers for individual 2D images in full resolution (zoom and pan).
- 2D filmstrip viewers for a set of 2D images in full resolution (select image, zoom and pan).
- Annotation editor for individual 2D images in full resolution (draw an annotation, zoom and pan).

Via Web GUI Daniel selects which 3D template package he wants to anchor to and the 2D image to anchor.

Daniel uses AligNII, allowing a step-wise process to anchor the 2D images to the 3D template package. The steps include:

- Identification of the orientation of the 2D image stack in relation to the 3D template package.



	<ul style="list-style-type: none"> <li>- Global scaling and positioning of the individual images (3D best fit) to the 3D template package.</li> <li>- Local positioning (region - by - region) of regions of interest.</li> </ul> <p>Daniel saves the anchoring results.</p> <p>Now the 2D images are searchable using metadata and/or 3D spatial requests, and can be viewed individually, in filmstrips or in 3D viewer.</p> <p>Daniel can add other users to Project1 and give them read or read-write access OR make Project1 publicly (anonymously) read-only available. The tools inherit the access rights.</p> <p>Optional: the tools, original 2D images, pyramids, thumbnails and previews can be embedded into web projects.</p>					
SP5-UC-012	John would like to find all available brain	Completed	SP5NIP-FR 013		TLR4	Sean HILL



WP5.1 Sean HILL	<p>atlases and/or 2D images for a particular 3D location and/or a particular set of metadata.</p> <p>John logs in to the Collaboratory, defines the criteria and runs the search.</p> <p>The results can be displayed individually, in filmstrips or in 3D viewer.</p>		SP5NIP-FR 012			<p>Michael DUPONT</p> <p>TsolmongereI PAPILLOUD</p>
<p>SP5-UC-013</p> <p>WP5.1 Sean HILL</p>	<p>Bill would like to highlight certain areas on 2D images of the brain atlas created by Daniel (owner) as Project1.</p> <p>Daniel gives read-write access to Bill for Project1.</p> <p>Bill selects a 2D image and opens Annotation editor for it.</p> <p>Bill draws an annotation, saves it and repeats the process for every 2D image he wants to annotate.</p> <p>The 2D images with the annotations can be displayed individually or in filmstrips. The</p>	Available via AlignNII	<p>SP5NIP-FR 013</p> <p>SP5NIP-FR 012</p>			

Page 133 of 176



	<p>5) Maybe she wants to compare this subset with another subset extracted in a similar way. She will be able to compare both subsets using population charts (e.g. box plots) and statistical tests (e.g., t-test).</p> <p>6) When Ruth is happy with the comparisons made and decide to finish the working session, she can export the results to standard formats.</p>		<p>SP5-FR-024</p> <p>SP5-FR-032</p>			
<p>SP5-UC-015</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	<p>1) Angel has identified a structure that needs to be segmented which can be automatically identified by one of the tools provided by EspINA.</p> <p>2) He opens EspINA and adds the stack to the analysis. EspINA requests, if needed, the meta-information to identify the sample which it was obtained from and lets him modify it if necessary.</p>	All Completed	<p>SP5-FR-030</p> <p>SP5-FR-017</p> <p>SP5-FR-019</p> <p>SP5-FR-018</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>Juan MORALES</p> <p>Antonio LATORRE</p>

	<p>3) Once the stack is loaded, it is displayed using one slicing view for each orthogonal plane.</p> <p>4) Then, he selects the appropriate tool on the toolbar to identify the neuron structure.</p> <p>5) The tool may require some parameters to operate, providing default values for all of them.</p> <p>6) He uses the tool to identify all the structures on the stack and once it has finished, he decides which ones he wants to keep.</p>		<p>SP5-FR-030</p> <p>SP5-FR-029</p> <p>SP5-FR-022</p> <p>SP5-FR-027</p> <p>SP5-FR-028</p> <p>SP5-FR-022</p> <p>SP5-FR-024</p>			
<p>SP5-UC-016</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	<p>1) Angel has identified a structure that needs to be segmented which cannot be automatically identified by one of the tools provided by EspINA.</p>	All Completed	<p>SP5-FR-030</p> <p>SP5-FR-017</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>PoI MONSO</p> <p>Carlos BECKER</p> <p>Juan MORALES</p> <p>Pablo Marquez NEILA</p> <p>Antonio LATORRE</p> <p>Pascal FUA</p>

Page 136 of 176





	<p>7) The tool may require some parameters to operate, providing default values, which will be set depending on the neurological category he had already selected, for all of them.</p>		SP5-FR-022			
	<p>8) He uses the tool to indicate in one of the views the location of the structure.</p>		SP5-FR-022			
	<p>9) EspINA will start reconstructing a new segmentation with all the voxels with the same grey levels than the selected voxel.</p>		SP5-FR-022			
	<p>10) Once it has finished it is displayed in the segmentation explorer as well as in all the views.</p>		SP5-FR-024			
	<p>11) He repeats this process until all the structures have been identified and segmented.</p>		<p>SP5-FR-020 SP5-FR-018</p>			



<p>SP5-UC-017</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	<p>1) Angel has identified a structure that needs to be segmented, but the automated tools cannot obtain a correct morphology.</p>	All Completed	SP5-FR-030	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>PoI MONSO</p> <p>Carlos BECKER</p> <p>Juan MORALES</p> <p>Pablo Marquez NEILA</p> <p>Antonio LATORRE</p> <p>Pascal FUA</p>
	<p>2) Angel has identified a structure that needs to be segmented, which cannot be automatically or semi-automatically identified by one of the tools provided by EspINA.</p>		SP5-FR-030			
	<p>3) He opens EspINA and adds the stack to the analysis. EspINA requests, if needed, the meta-information to identify the sample from which it was obtained, and allows him to modify it if necessary.</p>		<p>SP5-FR-019</p> <p>SP5-FR-018</p> <p>SP5-FR-029</p>			
	<p>4) Once the stack is loaded, it is displayed using one slicing view for each orthogonal plane.</p>		<p>SP5-FR-030</p> <p>SP5-FR-029</p>			
	<p>5) Angel identifies a new structure to be segmented using the slice views.</p>		SP5-FR-030			



6) Then, he selects the appropriate tool (i.e. brush tool) on the toolbar to manually create the neuron structure.

7) The tool may require some parameters to operate, providing default values, which will be set depending on the neurological category he had already selected, for all of them.

8) He uses the tool to mark on the views the voxels belonging to the structure.

9) After the first voxel is annotated a new segmentation is displayed in the segmentation explorer as well as in all the views.

10) He repeats this process until all the structures have been identified and segmented.

SP5-FR-022

SP5-FR-027

SP5-FR-028

SP5-FR-022

SP5-FR-022

SP5-FR-024

SP5-FR-022

SP5-UC-018 WP5.2 Pascal FUA, Jose M PENA	<p>1) Angel is not satisfied with a segmentation obtained by any of the EspINA tools.</p> <p>2) He decides to manually edit the segmentation voxels.</p> <p>3) He selects the segmentation and the edition tool that best fits its needs (brush, close, open, erode, dilate, planar split, merge or subtract).</p> <p>4) Once he has made the modifications to the segmentation, it has the morphological appearance that he desires and is ready for a later analysis.</p>	All Completed	<p>SP5-FR-030</p> <p>SP5-FR-017</p> <p>SP5-FR-030</p> <p>SP5-FR-022</p> <p>SP5-FR-024</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>Juan MORALES</p> <p>Antonio LATORRE</p>
SP5-UC-019 WP5.2 Pascal FUA, Jose M PENA	1) Angel has segmented a number of synapses with EspINA and wants to obtain the information relative to their apposition surfaces.	All Completed	<p>SP5-FR-030</p> <p>SP5-FR-017</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>Juan MORALES</p> <p>Antonio LATORRE</p>



	<p>2) He selects the synapses he is interested in using the views and/or the segmentation explorer.</p> <p>3) Then he uses the automatic apposition surface extractor tool to generate the apposition surface of each selected synapse.</p> <p>4) Once the tool has finished its execution he selects the resulting apposition surfaces and opens the apposition surface analysis dialogue, where he can see all the apposition surface information of every synapse.</p>		<p>SP5-FR-031</p> <p>SP5-FR-026</p> <p>SP5-FR-026</p>			
<p>SP5-UC-020</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	<p>1) Once Angel has identified all the structures needed for his analysis, he shares the resulting segmentation file to his colleague, Javier.</p>	<p>All Completed</p>	<p>SP5-FR-020</p> <p>SP5-FR-018</p>	<p>All EspINA</p>	<p>All TRL 9 - Production</p>	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>PoI MONSO</p> <p>Carlos BECKER</p>

	<p>2) Javier reviews all the segmentations and in case of discordance he tags those segmentations as doubtful.</p> <p>3) In case he finds a segmented structure that Angel has missed, he adds it to the analysis himself using the segmentation tools.</p> <p>4) When Javier has finished, they discuss their opinions and finally they classify each segmented structure accordingly.</p>		<p>SP5-FR-021</p> <p>SP5-FR-022</p> <p>SP5-FR-023</p>			<p>Juan MORALES</p> <p>Pablo Marquez NEILA</p> <p>Antonio LATORRE</p> <p>Pascal FUA</p>
<p>SP5-UC-021</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	<p>1) Rose is learning to segment neural structures using EspINA.</p> <p>2) After a briefing from her teacher, Angel, she starts identifying all the structures of the stack.</p> <p>3) When she has finished, she sends the analysis to Angel for evaluation.</p>	All Completed	<p>SP5-FR-030</p> <p>SP5-FR-017</p> <p>SP5-FR-022</p> <p>SP5-FR-029</p> <p>SP5-FR-020</p> <p>SP5-FR-018</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>Juan MORALES</p> <p>Antonio LATORRE</p>



	<p>4) Angel reviews and adds tags or notes to the analysis where he thinks it is convenient.</p> <p>5) He then sends the analysis file back to Rose.</p> <p>6) Rose then opens the analysis and inspects the corrections and annotations Angel has made.</p>		<p>SP5-FR-021</p> <p>SP5-FR-020</p> <p>SP5-FR-018</p> <p>SP5-FR-021</p>			
<p>SP5-UC-022</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	<p>1) Angel decides to split his workload in the segmentation of the structures of a stack with his colleague Rodrigo.</p> <p>2) Each of them segments one half of the stack.</p> <p>3) When they have finished, they gather and open both analyses at the same time, so they get the segmentations of all the structures of the stack.</p>	All Completed	<p>SP5-FR-021</p> <p>SP5-FR-022</p> <p>SP5-FR-029</p> <p>SP5-FR-021</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>Juan MORALES</p> <p>Antonio LATORRE</p>



<p>SP5-UC-023</p> <p>WP5.2 Pascal FUA, Jose M PENA</p>	1) Angel has to extract some of the information contained within an EspINA file.		SP5-FR-020			<p>Jose María (Chema) PEÑA</p> <p>Felix DE LAS POZAS</p> <p>Jorge PEÑA</p> <p>Juan MORALES</p> <p>Antonio LATORRE</p>
	2) He starts EspINA in his machine and opens the corresponding file.		SP5-FR-018			
	3) Angel then selects the relevant segmentations that the information will be extracted from.		SP5-FR-020			
			SP5-FR-018			
	4) He then opens the segmentation information dialogue showing the representations and information of the relevant segmentations.		SP5-FR-030			
	5) He selects the properties he wants to extract, excluding all the information not relevant at the moment.		SP5-FR-024			
	6) Then he is asked to enter a file name and a format in which the information will be saved.		SP5-FR-024			



	7) EspINA extracts only the relevant information and saves it to disk.		SP5-FR-024			
SP5-UC-024 WP5.2 Pascal FUA, Jose M PENA	<p>1) Angel needs to verify some data generated with an older version of EspINA.</p> <p>2) He starts EspINA in his machine and opens the relevant file.</p> <p>3) EspINA imports all the relevant data contained in the old file format and starts a new session with the data.</p> <p>4) Angel can now inspect the data, modify or generate new data it if necessary and save the results in a new EspINA analysis.</p>	All Completed	<p>SP5-FR-020 SP5-FR-018</p> <p>SP5-FR-020 SP5-FR-018</p> <p>SP5-FR-020 SP5-FR-018</p> <p>SP5-FR-020 SP5-FR-018</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA Felix DE LAS POZAS Jorge PEÑA Juan MORALES Antonio LATORRE</p>
SP5-UC-025 WP5.2 Pascal FUA, Jose M PENA	<p>1) Angel has completed an experiment with EspINA in his machine and needs some images for a paper publication.</p> <p>2) He starts EspINA and loads the segmentation file containing the relevant information of the experiment.</p>	All Completed	<p>SP5-FR-032 SP5-FR-017</p> <p>SP5-FR-020 SP5-FR-018</p>	All EspINA	All TRL 9 - Production	<p>Jose María (Chema) PEÑA Felix DE LAS POZAS Jorge PEÑA Pol MONSO Carlos BECKER Juan MORALES Pablo Marquez NEILA</p>

<p>3) He configures the views until he decides he has the right configuration from which he wants to get images.</p> <p>4) Angel uses the 3D export tool to export several 2D images from the 3D view.</p> <p>5) He also wants to export the scene to a Blender-compatible format, so he uses the 3D export tool to obtain a VRML description of the scene he will later use to obtain additional images using an external tool.</p>		SP5-FR-031				Antonio LATORRE Pascal FUA
		SP5-FR-032				
		SP5-FR-032				

## Summary of use cases

- Assisted Neurological Structure Segmentation
- Semi-automated Neurological Structure Segmentation
- Manual Neurological Structure Segmentation
- Manual Edition of Segmented Structures
- Synaptic Apposition Surface (SAS) Analysis
- Analysis Review
- Training
- Collaborative Segmentation



- Data Extraction
- Import Data from Previous Versions
- Export Views

Table 3: Use cases and success scenarios (3)

Use Case ID	Success Scenario	Status/comments	ID of Related Functional Requirement	Related Product/Software Package/Service	TRL at end of Ramp-Up Phase	Contributors
SP5-UC-033	<p>Scientific Developer User Daniel would like to analyse the three-dimensional spatial distribution of a set of synapses.</p> <p>Daniel needs to use the R program and load some necessary packages (the <i>spatstat</i> package is essential but some others can be needed for certain processing operations).</p> <p>Daniel loads the (x, y, z) coordinates of each synapse and, if he has this information, its diameter.</p> <p>Daniel analyses whether the pattern defined by synapses coordinates follows a CSR process. To do this he can choose between different statistical tests.</p> <p>He runs three-dimensional CSR processes simulations with the same</p>	Implemented and validated	SP5-FR-037	3DSynapsesSA	6 - Technology demonstrated in relevant environment	Laura Antón SÁNCHEZ Luis Rodríguez LUJÁN Concha Bielza LOZOYA Pedro Larrañaga MÚGICA



	<p>characteristics of the synapses he is studying and he compares the simulations with the real samples using statistical tests.</p> <p>If he has the synapses diameters, he can do the same as in the previous point but with RSA processes simulations.</p> <p>Daniel graphically represents G, F, K and L functions of simulations patterns and real samples.</p>					
SP5-UC-034	<p>Scientific Developer User Daniel has a set of neurons in Neurolucida software format and he wants to analyse them in order to obtain a general descriptive analysis of the different neuron features.</p> <p>Daniel will change the Neurolucida format (.DAT) to ASC using the windows command prompt (cmd.exe).</p> <p>Daniel will select the different features to be measured.</p>	Implemented but not validated.	<p>SP5-FR-039</p> <p>SP5-FR-040</p>	3DPyrStructure	5 - Technology demonstrated in relevant environment	<p>Luis Rodríguez LUJÁN</p> <p>Concha Bielza LOZOYA</p> <p>Pedro Larrañaga MÚGICA</p>



	Finally, the results will pop up.					
SP5-UC-035	<p>Scientific Developer User Jane has a set of neurons in Neurolucida software format and she wants to get the conditional dependencies between the different features in order to get a fitted model.</p> <p>Jane will change the Neurolucida format (.DAT) to ASC using the windows command prompt (cmd.exe).</p> <p>Jane will select the different features (variables) to be included.</p> <p>She will obtain the model that connects the selected variables. A graphical representation (directed acyclic graph) could also be displayed.</p> <p>Jane asks queries to the model, as a probability of some variables given other variables fixed to some values (evidence).</p> <p>Jane will receive these probabilities in numerical and graphical</p>	Final implementation stage	<p>SP5-FR-039</p> <p>SP5-FR-040</p> <p>SP5-FR-041</p>	3DPyrStructure	5 - Technology demonstrated in relevant environment	<p>Luis Rodríguez LUJÁN</p> <p>Concha Bielza LOZOYA</p> <p>Pedro Larrañaga MÚGICA</p>



	formats attached to each node in the directed acyclic graph.					
SP5-UC-036	<p>Alex has 3D data in his local machine corresponding to a representation of a soma.</p> <p>He wants to analyse soma's morphology and decided to download the R source with its dependencies (R packages).</p> <p>Alex considers interesting for his research to analyse the soma representation and he runs the script with the file as input.</p> <p>Alex supervises the input file for a correct pre-processing of the soma (cut of dendrites, filling holes, convex hull and smoothing).</p> <p>When pre-processing is finished Alex, selects a temporal path to save the pre-processed data and an additional path and name for the reconstructed mesh.</p> <p>Next a file path and</p>	Implemented and validated	<p>SP5-FR-043</p> <p>SP5-FR-044</p> <p>SP5-FR-045</p> <p>SP5-FR-046</p> <p>SP5-FR-047</p>	3DSomaMS	6 - Technology demonstrated in relevant environment	<p>Sergio Luengo SÁNCHEZ</p> <p>Luis Rodríguez LUJÁN</p> <p>Concha Bielza LOZOYA</p> <p>Pedro Larrañaga MÚGICA</p>



	<p>name are requested to store the quantitative characteristics.</p> <p>Alex repeats the process iteratively for each soma to save data results in the output file and obtain a table with all of the soma characteristics.</p> <p>Finally, an automatic typology of the soma shapes is obtained according to the cluster method selected by Alex.</p> <p>Alex can click on some group of somas to see its members as images.</p>					
SP5-UC-037	<p>Once Alex has passed through the previous scenario he can perform this use case.</p> <p>Alex chooses a file with the soma characteristics and the model obtained from the clustering process. Alex starts the simulation of artificial somas of some prefixed typology.</p> <p>Alex has to set the parameters for the optimisation process,</p>	Implemented and validated	SP5-FR-047	3DSomaMS	6 - Technology demonstrated in relevant environment	<p>Sergio Luengo SÁNCHEZ</p> <p>Luis Rodríguez LUJÁN</p> <p>Concha Bielza LOZOYA</p> <p>Pedro Larrañaga MÚGICA</p>





	<p>which transforms the simulated soma characteristics into a 3D soma image.</p> <p>He selects the path and file name to save the results.</p>					
SP5FDAT-UC-001	Full-scale simulation and analysis of a multi-layered local cortical column	Functionality implemented into new New Use Case Demos 1 and 2.	Neo, Elephant	n/a	See new Use Case 1 and 2	SP5FDAT-UC-001 Michael DENKER Alper YEGENOGLU Andrew DAVISON
SP5FDAT-UC-002	Web-based analysis of simulated data	All functionality available, implementation covered in part by New Use Case Demos 1-3	Neo, Elephant	n/a	2	Michael DENKER Alper YEGENOGLU Andrew DAVISON
SP5FDAT-UC-003	Estimating complex stimulus-response relationships	All functionality available, implementation covered in part by New Use Case Demos 1-3	Neo, Elephant	n/a	2	Michael DENKER Alper YEGENOGLU Andrew DAVISON
SP5FDAT-UC-004	Simulator development	All functionality available, implementation covered in part by New Use Case Demos 1-3	Neo, Elephant	n/a	2	Michael DENKER Alper YEGENOGLU Andrew DAVISON
SP5FDAT-UC-005	LFP analysis using data from an electrophysiological experiment	All functionality available, covered in part by New Use Case Demos 1-3	Neo, Elephant	n/a	2	SP5FDAT-UC-005 Michael DENKER Alper YEGENOGLU



						Andrew DAVISON
SP5FDAT-UC-006	Spike sorting	All functionality available	Spike Sorting Package	n/a	2	SP5FDAT-UC-006 Michael DENKER Alper YEGENOGLU Andrew DAVISON
SP5FDAT-UC-007	Detecting higher-order spike patterns in stochastic model data	Functionality implemented (with alternative methods: CuBIC and Unitary Events instead of “frequent item set mining”) into new New Use Case Demos 1-3	Neo, Elephant	n/a	see new Use Case 1 and 2	SP5FDAT-UC-007 Michael DENKER Alper YEGENOGLU Andrew DAVISON

## New Use Cases Platform Use Case Status

Table 4: New use cases Platform use case status

Use Case ID	Description	Status/Comments	Related Tools	Services/Applications called	TRL at end of Ramp-Up Phase	Test scenarios (in-/Output data)
1	Analysis of simulated data (NEST-Elephant workflow demo SP5-SP6 at Summit 2015 and SfN)	Collab with corresponding tasks available on Collaboratory	Elephant	n/a	4-5	NEST simulated data (microcircuit model)
2	Comparison of simulated and Neuromorphic data (NEST-SpiNNaker-Elephant demo SP5-SP6-SP7-SP9 prepared for release day)	Collab with corresponding tasks under development	Elephant	n/a	4-5	NEST and SpiNNaker simulated data (microcircuit model)
3	Interactive analysis of experimental data using Neo and Elephant (Neo-Elephant tutorial; available on Collaboratory)	Jupyter Notebook available on Collab	Elephant	n/a	5	Example data set

## Summary of use cases

In the course of the Ramp-Up Phase multiple use cases are being implemented to guide the development and integration of functional data analysis capabilities into the Collaboratory.

*NEST-Elephant demo at the HBP summit Madrid 2015:* In this use case, we demonstrated how a user is able to run a simulation of a microcircuit model on the Collaboratory using the NEST simulator, convert and read the data using the NEO API, and analyse the data using the Elephant framework. The individual simulation and analysis components are implemented as Collaboratory tasks.

*Neo-Elephant tutorial:* A special tutorial based on a Jupyter notebook is hosted on the Collaboratory that demonstrates the use case of interactively loading and analysing a specific data set.

*NEST-SpiNNaker-Elephant demo:* A demo is currently being developed by SP5, SP6, SP7, and SP9 to show how a user can compare the results of HPC based (NEST) and neuromorphic hardware based (SpiNNaker) simulations of a spiking neuron network model using the Elephant analysis framework running on the Juelich HPC infrastructure.

Table 5: Use cases and success scenarios (4)

Use Case ID	Success Scenario	Status/comments	ID of Related Functional Requirement	Related Product/Software Package/Service	TRL at end of Ramp-Up Phase	Contributors
SP5BAS-UC-001	<p>Scientific developer user Daniel would like to generate an instantiation of the projection of thalamocortical relay cells from the ventral posterior medial (VPM) nucleus of the thalamus to layer 4 spiny stellate cells (L4 SS) in barrel cortex (S1) in the rodent.</p> <p>Daniel will select the source area (VPM) and cell type (relay cell) and the destination area (S1) and cell type (L4 SS) in the atlas tool and download a <i>projection</i> data object with the corresponding statistics: size of area, cell density,</p>	<p>Daniel will use a combination of tools to carry out the use case: cell densities and morphologies are available from the Neuroinformatics Platform, whereas the connectivity engine is available for download as a Matlab library under the name SynapseGenerator. This tool, with cell densities and morphologies as input, executes the two use cases (SP5BAS-UC-001 and SP5BAS-UC-002). The tool is in a proof of concept phase.</p> <p>To streamline the workflow, we have implemented a web-based morphology</p>	SP5BAS-FR-001	SynapseGenerator MorphologyViewer	TRL 3 PoC Implementation	Rembrandt BAKKER, Max BAKKER, Paul TIESINGA (SKU)



	<p>axonal/dendritic morphology.</p> <p>Daniel will call the connectivity engine to build connectivity based on the downloaded statistics, which will generate indices for each cell, morphologies, and a connection matrix. For each instantiated connection, it will list the axonal branch on which the presynaptic part of the synapse is located, and the dendritic branch on which the presynaptic part is located. This information will be collected in a <i>pathway</i> data object.</p> <p>Daniel wants to validate the connectivity. He uploads the data object to the <i>pathway validator</i> tool—which calculates relevant statistics such as distribution of synapses across branch order—which he compares to experimental data in the atlas. The outcome of the validation is placed in a <i>pathway validation</i> data object.</p>	<p>viewer and an accompanying ipython notebook that downloads morphologies, analyses them with NeuroM (SP6), downloads cell densities and saves the result as a projection data object, ready for consumption by the SynapseGenerator.</p> <p>The use case (SP5BAS-UC-001) has been performed and is documented as part of the Master thesis of Max BAKKER. Data for the VPM to L4SS projection has been obtained from SP1.</p> <p>SynapseGenerator converts morphologies into neurite densities using a Gaussian mixture model, on the basis of which synaptic element densities are generated. These are then mapped to source and target neurons. We have implemented and evaluated four different versions of the connectivity engine and have identified the selection model (this will be described in</p>				
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	--



	Finally, he incorporates the <i>pathway</i> data object into the circuit building workflow to include the synthesis task in the appropriate location.	more detail in the Milestone report) as the fastest, with lowest bias. The resulting connectivities are validated in terms of the branch and multiplicity statistics and the resulting connectivity-based reconstruction of morphologies. The resulting connectivity data has not yet been used in the circuit builder.				
SP5BAS-UC-002 WP5.4 TIESINGA	<p>Scientific Developer User Jezebel has built a mouse brain using the brain builder and would like to validate the thalamocortical pathway in her model.</p> <p>Jezebel will select the source area (VPM) and cell type (relay cell) and the destination area (S1) and cell type (L4 SS) in the atlas tool and download a <i>projection</i> data object with the corresponding statistics: the branch order statistics of the projection.</p> <p>Jezebel collects the relevant part of her</p>	This use case (SP5BAS-UC-002) is also based on the SynapseGenerator (see preceding comments), specifically the validation tool that calculates the branch statistics etc. based on the connectivity matrix provided by Jezebel. The necessary experimental comparison statistics still need to manually obtained from the Neocortical Microcircuit Collaboration Portal.	SP5BAS-FR-002 SP5BAS-FR-004	SynapseGenerator MorphologyViewer	TRL 3 PoC Implementation	Rembrandt BAKKER, Max BAKKER, Paul TIESINGA (SKU)



<p>connection matrix into a <i>pathway</i> data object</p> <p>She then uploads the <i>pathway</i> data object to the <i>pathway validator</i> tool, which returns the <i>pathway validation</i> data object.</p> <p>She inspects the <i>pathway validation</i> data object using the visualisation tool and concludes that the instantiated projection does not match the recently updated experimental data.</p>					
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	--	--

## Response times and numbers of users

EspINA is a standalone tool that requires installation on the user's computer. The final performance depends on the hardware on which it is installed. In high-end workstations EspINA provides fully interactive use even when using large image stacks.



## Annex D: Summary - Service IT Resource Planning

Table 6: Service IT resource planning

Product/Software Package/Service	TRL	Data Storage Capacity used by this Product	Data Storage Capacity Allocated for this Product	Location(s) of Data Storage	Data Access Protocol(s)*	Compute Resource(s) Allocated	Location(s) of Compute Resource(s) Allocated	Compute Access Protocol(s)**
Elephant	2-5							
Neo								
Spike								
AlignII	7	13 TB	80 TB	UIO	HTTP	Server at UIO	UIO	
QuickNII	8	600 MB		End-user				
CutNII	9 N/A	600 MB		End-user				
MeshGen	6	N/A		N/A				
MeshView	5	256 MB		UIO	HTTP			
EspINA	TRL 9 - Production	Local	Local	Local	Local	Local	Local	Local
3DSomaMS	6	< 1GB	<1GB	External server - VPS CESVIMA		1 shared VPS (2 cores 2GB RAM)	External server - VPS CESVIMA	
3DSynapsesSA	6	< 1GB	< 1GB	External server - VPS CESVIMA		1 shared VPS (2 cores 2GB RAM)	External server - VPS CESVIMA	
SynapseGenerator	TRL3	4MB		Collab Storage	Matlab			
Morphology	TRL3	12MB		SKU Storage	Neuroinformatics			web-based, client





Viewer					Platform			side processing
3DPyrStructure	3 (6 at the end of Ramp-Up phase)	TBD	TBD	External server - VPS CESVIMA		1 shared VPS (2 cores 2GB RAM)	External server - VPS CESVIMA	
Search API	TRL4	N/A						
Index API	TRL4	10GB	<100GB	BBP infrastructure	HTTP			
Data submission API	TRL4	1T	40T	Zenodo, BBP document service	HTTP			
Ontology API	TRL5	<1GB		GitHub	HTTP			
Image service API	TRL4	5T	40T	EGI, BBP GPFS	HTTP	EGI, BBP VIZ	EGI, BBP VIZ	http, server-side processing
Search app	TRL4	N/A						
Atlas viewer app	TRL4	N/A						
Data submission app	TRL3	N/A						
Voxel Brain API	TRL4	1T	10T	BBP swift				
Zenodo	TRL9		1 pb	Zenodo	HTTP			
Text mining	TLR4	500GB	2T	BBP infrastructure				

## Elephant

Table 7: KPI status

KPI	Status	Comment
The number of completed <i>functional requirements</i> listed in the Specification Document (" <i>functions</i> ")	<p>Completed: 15 Final preparation: 6 Open: 0 Total: 21</p> <p><i>(The set of functional requirements of WP5.3 that are termed "integrated" in the Specification Document are automatically fulfilled due to a modification in the way Elephant is integrated on the Collaboratory, and are not included in this count.)</i></p>	<p>In final preparation:</p> <ul style="list-style-type: none"> <li>• Unitary Events (replaces "FIM adv.")</li> <li>• Spike sorting integration</li> <li>• Kernels / Spike train metrics</li> <li>• Spike Field Coherence</li> <li>• Signal phase and amplitude</li> <li>• Correlated stochastic process</li> </ul>
The number of completed (python) functions in the source code.	<p>Public: 49 (Public + Private: 68)</p>	<p>Additional functionality (outside specification):</p> <ul style="list-style-type: none"> <li>• addtl. surrogate methods</li> <li>• Inverse current source density (iCSD) method (in prep)</li> <li>• CuBIC/UE</li> </ul>



		<ul style="list-style-type: none"> <li>• pandas bridge</li> <li>• threshold detection</li> </ul>
The total test coverage	98% (coveralls)	

## Interfaces to data providers and other service

The Neo library, which acts as a data model backend to the Elephant analysis library, features a broad range of I/O modules for commonly used file formats for electrophysiological data to enable easy access to multiple data sources.

## Response times & number of users

Estimated number of users: 30-50

Number of monthly downloads of Elephant library (0.2.1) on PyPI: about 300

## Overall list of tools and services offered by platform

Electrophysiology Analysis Toolkit (Elephant): Python library providing tools for the analysis of electrophysiological data.

Neo: Library used by Elephant that provides data models and file I/O operations for electrophysiological data.

Spike sorting package: Library with tools to perform spike sorting on extracellularly recorded time series data.

**Table 8: Milestones**

Milestone	Name	Delivery (M)	Status
MS93	Specifications	6	Document submitted
MS94	Initial Release	12	Elephant 0.1.0 April 8, 2015
MS95	Alpha Release (Internal release)	18	Elephant 0.2.0 Sept. 22, 2015
MS96	Spike sorting and LFP analysis (Public release)	30	<i>Release of analysis tools:</i> Elephant 0.3.0 <i>Anticipated:</i> March 8, 2016 <i>Integration on Collab:</i> March 20, 2016 <i>(incl. spike sorting SpikeSort (0.12)</i> <i>or SpyKING CIRCUS (latest)</i>

\* Data Access Protocols such as GPFS, N.FS, S3, Collab storage, etc.

\*\* Compute Access Protocols such as EC2, Task Framework, Unicore, OCCI, Slurm, ssh, gLite, Condor, etc.

### ***EspINA***

**EspINA** may use a local iRods-based repository to store both image stacks and common metadata records. The maximum data storage capacity depends on the iRods repository limitations rather than EspINA itself.

EspINA is a standalone tool that requires installation on the user's computer. The final performance depends on the hardware on which it is installed. In high-end workstations EspINA provides fully interactive use even when using large image stacks.

## Annex E: Summary - Service Technology Readiness Levels (TRLs) Metrics

Documentation URL - User, Developer and/or Administrator documentation is available at this URL. Strong preference should be given for publicly available documentation services.

Target User Count (TRL6+) - Target user counts (concurrent service users).

SLA Defined - The software documentation defines some Quality of Service metrics in the service documentation. These metrics may or may not be enforced by the service itself. The service has not been tested to adhere to the documented QoS metrics.

SLA Monitored - The Quality of Service metrics are monitored by a monitoring service.

SLA Enforced - The Quality of Service metrics are enforced by implementing service. If the SLA Definition indicates on 3 API/request/sec/user, there are suitable mechanisms implemented in the service to ensure these limits are not exceeded.

**Table 9: Service technology readiness levels metrics**

Product/Software Package/Service	Technology Readiness Level (TRL1-9)	Documentation URL	Target User Count (TRL6+)	SLA Defined (TRL7+)	SLA Monitored (TRL7+)	SLA Enforced (TRL7+)	Comments
Elephant	2-5	<a href="http://elephant.reaidthedocs.org/en/latest/index.html">http://elephant.reaidthedocs.org/en/latest/index.html</a>					
Neo							
Spike							
AlignII	7	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>	100				
QuickNII	8	<a href="http://www.nesys.uio.no/AlignII">http://www.nesys.uio.no/AlignII</a>	500				
CutNII	9 N/A	<a href="http://software.incf.org/software/cutnii">http://software.incf.org/software/cutnii</a>	1000				
MeshGen	6	<a href="http://software.incf.org/software/meshg">http://software.incf.org/software/meshg</a>	500				



		<a href="#">en</a>					
MeshView	5	<a href="http://software.incfn.org/software/meshgen">http://software.incfn.org/software/meshgen</a>	1000				
EspINA	TRL 9 - Production	<a href="http://cajalbbp.cesvima.upm.es/espina/">http://cajalbbp.cesvima.upm.es/espina/</a>	N/A	N/A	N/A	N/A	Local tool
3DSomaMS	6	To be released	Not available				Target user count not measured
3DSynapsesSA	6	To be released	Not available				Target user count not measured
3DPyrStructure	3 (6 at the end of Ramp-Up phase)	To be released					
SynapseGenerator	TRL3	<a href="https://www.neuroinformatics.nl/SynapseGenerator/index.html">https://www.neuroinformatics.nl/SynapseGenerator/index.html</a>	2				
Morphology Viewer	TRL3	<a href="https://neuroinformatics.nl/HBP/morphology-viewer">https://neuroinformatics.nl/HBP/morphology-viewer</a>	20				
Search API	TRL4	<a href="https://nip.humanbrainproject.eu/api/kssearch/swagger/index.html">https://nip.humanbrainproject.eu/api/kssearch/swagger/index.html</a>					
Index API	TRL4						
Data submission API	TRL4	<a href="https://nip.humanbrainproject.eu/documentation/user-manual.html#registr">https://nip.humanbrainproject.eu/documentation/user-manual.html#registr</a>					



		ation-app					
Ontology API	TRL5	<a href="https://nip.humanbrainproject.eu/api/scigraph/docs/">https://nip.humanbrainproject.eu/api/scigraph/docs/</a>					
Image service API	TRL4	<a href="https://bbpcode.epfl.ch/browse/code/databanking/specifications/tree/image_services">https://bbpcode.epfl.ch/browse/code/databanking/specifications/tree/image_services</a>					
Search app	TRL4						
Atlas viewer app	TRL4						
Data submission app	TRL3	<a href="https://nip.humanbrainproject.eu/documentation/contribute/requirements.html">https://nip.humanbrainproject.eu/documentation/contribute/requirements.html</a>					
Voxel Brain API	TRL4	<a href="https://nip.humanbrainproject.eu/api/analytics/swagger/index.html">https://nip.humanbrainproject.eu/api/analytics/swagger/index.html</a>					
Text mining	TRL4	<a href="https://github.com/sherlok/sherlok">https://github.com/sherlok/sherlok</a>					
Zenodo	TRL9	<a href="https://zenodo.org/dev">https://zenodo.org/dev</a>	N/A	N/A	N/A	N/A	Zenodo



## Annex F: Backlog (Remaining bugs and new features to be added)

### Product/Software Package/Service name

#### *Current limitations, problems and bug list*

EspINA is a complex tool and thus many tasks take a long time to be developed. Currently, the following requests have been identified from users and are scheduled for future implementation:

- Update the “Refined Auto-Segmenter” plug-in to the latest version.
- Implement a skeleton-based segmentation algorithm.
- Integrate EspINA with graphCuts (EPFL), a tool for interactive segmentations splitting.
- Integrate EspINA with ccboost (EPFL), a tool for automatic segmentation of structures.
- Port EspINA to the Windows x64 platform.

#### *Usage levels of data storage and compute resources*

EspINA may use a local iRods-based repository to store both image stacks and common metadata records. The maximum data storage capacity depends on the iRods repository limitations rather than EspINA itself.

#### *Main technologies used*

EspINA is implemented in C++ as a standalone client-based tool. It required a series of external libraries:

- Qt 4.8
- Boost
- VTK (Visualization Toolkit)
- ITK (Insight Toolkit)

#### *Remaining Bugs*




**Table 10: Remaining bugs**

Bug ID	Related Use Case(s) or FR(s)	Short Description	Bug tracker URL (if available)	Comments

## Features

**Table 11: Features**

Feature ID	Necessary for Use Case(s) or FR(s)	Short Description	Bug tracker URL (if available)	Comments
SynapseGenerator Import tools	SP5BAS-UC-001 and SP5BAS-UC-002	Currently SynapseGenerator is standalone and implemented in Matlab. Matlab is not provided as a service in the Neuroinformatics Platform, which means that importing data from the Platform is not possible from within the tool and has to be done manually. One solution is to re-implement the tool (when completely tested and best algorithm is chosen) in python. The other solution is to run the tool locally and call APIs from within Matlab. This integration needs interaction and		



		the attention of the core programmers for the Platform.		
Reduce SynapseGenerator memory footprint	SP5BAS-UC-001 and SP5BAS-UC-002	The tool can do multiple overlapping projections at the same time, but has a too large memory footprint (number of cubic-micron-sized voxels times the number of neurons). This needs to be addressed by an efficient divide and conquer strategy in combination with sparse matrix implementations.		
	SP5BAS-UC-001 and SP5BAS-UC-002	The tool relies on generating parameterised density distributions based on reconstructed morphologies. When the tool is working correctly, the morphology of the neurons used as input, should be reproduced by the tool on output. Currently that is not the case. Our hypothesis is that we need improve the Gaussian density model and the minimal spanning tree algorithm used for turning synapses into morphologies. This requires research and development activity prior to a robust implementation.		

## Annex G: IPR Status, Ownership and Innovation Potential

Table 12: IPR status, ownership and innovation potential

Product/Software Package/Service	IPR Status*	Owner(s)	Non-HPP users**	Innovation Potential***
Elephant		Michael DENKER Alper YEGENOGLU Andrew DAVISON		
Neo				
Spike				
AlignII	Copyright	UIO	Laboratories in USA, Europe and Australia, provided with user credentials at UIO data system.	
QuickNII	Copyright	UIO		
CutNII	Copyright	UIO	Shared through INCF Software centre	
MeshGen	Copyright	UIO	Shared through INCF Software centre	
MeshView	Copyright	UIO	Accessed through INCF Software centre	
EspINA	Open Source	UPM	7 institutions	Commercial use (TBD)
3DSomaMS	Open Source (GPL 2/3)	Luis Rodríguez LUJÁN		
3DSynapsesSA	Open Source (GPL 2/3)	Luis Rodríguez LUJÁN		
3DPyrStructure	Open Source (GPL 2/3)	Luis Rodríguez LUJÁN		



Search API		Sean HILL		
Index API		Sean HILL		
Data submission API		Sean HILL		
Ontology API		Sean HILL		
Image service API		Sean HILL		
Search app		Sean HILL		
Atlas viewer app		Sean HILL		
Data submission app		Sean HILL		
Voxel Brain API		Sean HILL		
Text mining	Open Source	Sean HILL		
Zenodo	CCO.1.0	CERN	N/A	N/A

\* IPR Status: Open Source, Copyright, Patent, Trade Secret, pre-IPR (i.e. you intend to obtain some form of IPR in the future)

\*\* If this product/software package/service is currently being used outside HBP (e.g. donated, loaned, licensed, sold), please specify by whom.

\*\*\* Innovation Potential: Potential practical applications beyond HBP, commercial and/or non-commercial.



## Appendix I: Report on the Data Hosting Pilot

See attached file SP5 D5.8.4 Resubmission Appendix I FINAL.pdf



## Appendix II: Use Cases and Use Case IDs

Tables below refer to Use cases in Annex C; O = obsolete. The full version of the table can be seen in the Accompanying Excel Spreadsheet (SP5 D5.8.4 Resubmission Appendix II FINAL.xlsx).

**Table 13: SP5NIP-FR Use Cases and Use Case IDs**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47				
SP5NIP-UC-001			X	X																																															
SP5NIP-UC-002	X	X		X																																															
SP5NIP-UC-003	O																																																		
SP5NIP-UC-004	O																																																		
SP5NIP-UC-005	O																																																		
SP5NIP-UC-006				X																																															
SP5NIP-UC-007	X	X		X								X																																							
SP5NIP-UC-008	X	X		X				X	X																																										
SP5NIP-UC-009	X	X	X	X	X	X	X	X	X	X	X	X																																							
SP5NIP-UC-010	X	X	X	X	X	X	X	X	X	X	X	X																																							
SP5NIP-UC-011	X	X	X	X	X	X	X	X	X	X	X	X																																							
SP5NIP-UC-012												X	X																																						
SP5NIP-UC-013											X	X																																							
SP5NIP-UC-014																	X	X		X				X						X	X	X																			
SP5NIP-UC-015																	X	X	X		X		X				X	X	X	X																					
SP5NIP-UC-016																	X	X	X	X		X	X	X			X	X	X	X																					
SP5NIP-UC-017																		X	X		X		X				X	X	X	X																					
SP5NIP-UC-018																	X				X		X							X																					
SP5NIP-UC-019																	X									X				X	X																				
SP5NIP-UC-020																		X		X	X	X	X																												
SP5NIP-UC-021																	X	X		X	X	X							X	X																					
SP5NIP-UC-022																				X	X								X																						
SP5NIP-UC-023																		X		X				X						X																					
SP5NIP-UC-024																		X		X				X																											
SP5NIP-UC-025																	X	X		X											X	X																			
SP5NIP-UC-033																																																			
SP5NIP-UC-034																																																			
SP5NIP-UC-035																																																			
SP5NIP-UC-036																																																			
SP5NIP-UC-037																																																			
SP5-WP5.1-001												X	X																																						
SP5-WP5.1-002												X	X																																						

[illegible]

### Table 14: SP5DAT-FR Use Cases and Use Case IDs

SP5DAT-FR-XXX		1	2	3
		FDTA acceptable	FDTA necessary	FDTA file format
Full-scale simulation and analysis of a multi-layered local cortical column	SP5DAT-UC-001	X	X	X
Web-based analysis of simulated data	SP5DAT-UC-002			
Estimating complex stimulus-response relationships	SP5DAT-UC-003	X	X	X
Simulator development	SP5DAT-UC-004	X	X	X
LFP analysis using data from an electrophysiological experiment	SP5DAT-UC-005	X	X	X
Spike sorting	SP5DAT-UC-006	X	X	X
Detecting higher order spike patterns in stochastic model data	SP5DAT-UC-007	X	X	X

### Table 15: SP5BAS-FR Use Cases and Use Case IDs

	SP5DBAS-FR.XXX	1	2	3	4
		Authentication and authorisation	Selection of projection	Generation of projection	Validation of projection
Building the projection from thalamocortical relay cells in the ventral posterior medial nucleus of the thalamus to layer 4 spiny stellate	DP5BAS-UC_001	X			
Validating the thalamocortical projection generated by the brain builder	DP5BAS-UC_002		X	X	X

## Appendix III: NIP Listing of Available Tools

Table 16: NIP listing of available tools

Tool type	Tool name	Description	Status
Structural data analysis tools	3DSomaMS EspiNA		Tools extended as part of the Project activities Tools extended as part of the Project activities
Functional analysis tools Brain Atlas tools	Elephant AlignII QuicknII CutNII MeshView MeshGen Scalable Brain Atlas Morphology Viewer Synapse Generator  CATMAID PAPAYA	synapseGenerator morphology preprocessor	Tools extended as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools extended as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools extended as part of the Project activities Tools extended as part of the Project activities
Service API	Data Registration KnowledgeGraph Search API Voxel Atlas API Ontology Service API Dataspace API Search		Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities Tools developed as part of the Project activities
External sites	Zenodo repository KnowledgeSpace		Tools developed as part of the Project activities Tools extended as part of the Project activities