Neuroinformatics Platform Documentation

Data Integration Manual

Last update: 2016-09-15

Table of Content

Data Integration Manual

Intended audience

This manual is intended at people who are either trying to understand what the process of integrating data into the Neuroinformatics platform entails, as well as for data integration specialists to use as a reference when integrating data into the platform.

Our approach to data integration

The Neuroinformatics approach to integrating data is centered around the capture of the Neuroscience specific datasets into the KnowledgeGraph. Due to the broad nature of the field, we are exposed to heterogeneous data types (e.g. electrophysiology, volumetric data, imaging data, computer models…). All these datasets share some commonalities such as how they were generated (data provenance) and descriptive metadata (MINDS). Furthermore, each of these data types requires special attention to capture specialised metadata as well as ensuring that the data is integrated in a standard format while retaining the original dataset submission format.

The integration of the data in such a fashion enables users to search the KnowledgeGraph using our Search web application / KSearch API while leveraging both common and specialised metadata to find required data.

Finally, it is important to note that while the data integration plan is well defined, the actual task of formalising the integration of specific data types is at various level of maturity.

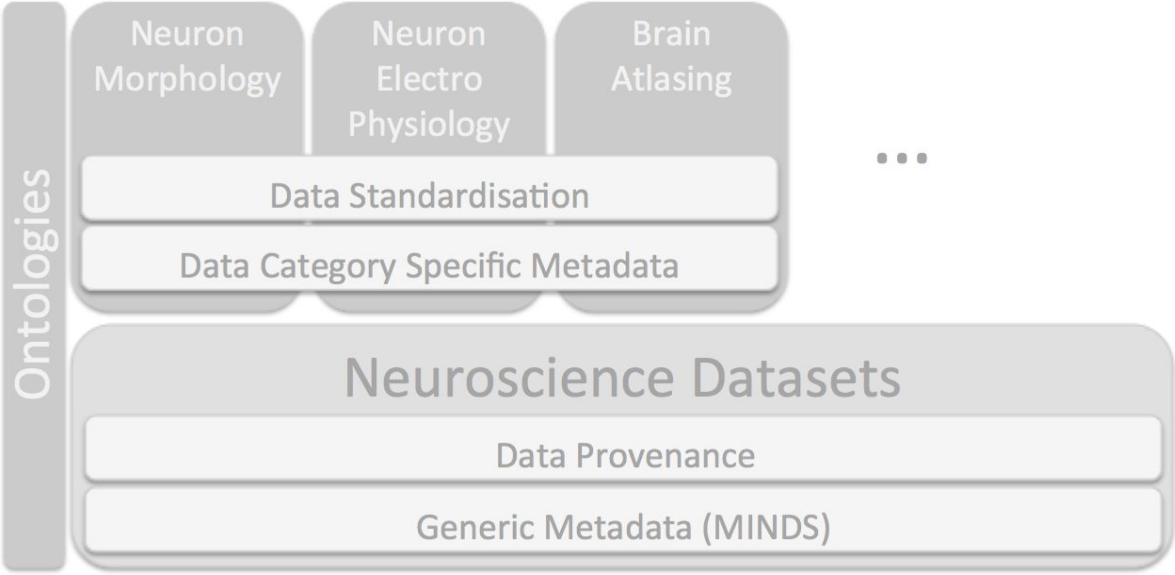


Fig. A high level view of the integration of datasets in the platform

Ontologies

What is an ontology?

Ontologies are used in various fields (artificial intelligence, NLP,...) and each one defines it differently. As different as those definitions are, it is agreed that an ontology is a representation of a domain that is:

- **formal**: the ontology expressiveness (i.e. the richness of the description of the domain knowledge) should be balanced with the complexity (mainly in time) it takes to perform reasoning tasks.
- **normalized** and **shared**: It is fundamental that people member of a targeted domain agree as much as possible on the ontology meaning and commitments,
- **partial:** not all concepts or all properties of a domain are represented

So an ontology is made of:

- **concepts**: sometimes called **ontology terms** in this documentation. A concept is described:
    - with a certain level of expressiveness (see what ontology expressiveness is supported in the NIP platform),
    - independently from the vocabulary used to name or express the reality (entities, process, events,...) it represents,
    - independently from a particular occurrence of the reality it represents

- **instances or individuals:** having concepts as types

- **properties**: that can be data properties (attributes or annotations) or object properties (relations)

For further details about what ontologies are good for in the Neuroscience field, please check this paper.

Overview of NIP Ontology Service

Many ontologies are used in the Neuroinformatics Platform (NIP) to annotate datasets during the curation phase. Most of the metadata in MINDS indeed take their values from an ontology. To store and retrieve those ontologies, an ontology service is deployed as part of the Neuroinformatics Platform. It consists of three main parts:

- Ontologies storage,
- Ontologies ingestion mechanism,
- Ontology service implemented as a REST API

The following picture gives an overview of how the NIP ontology service is implemented and which tools and libraries are used:

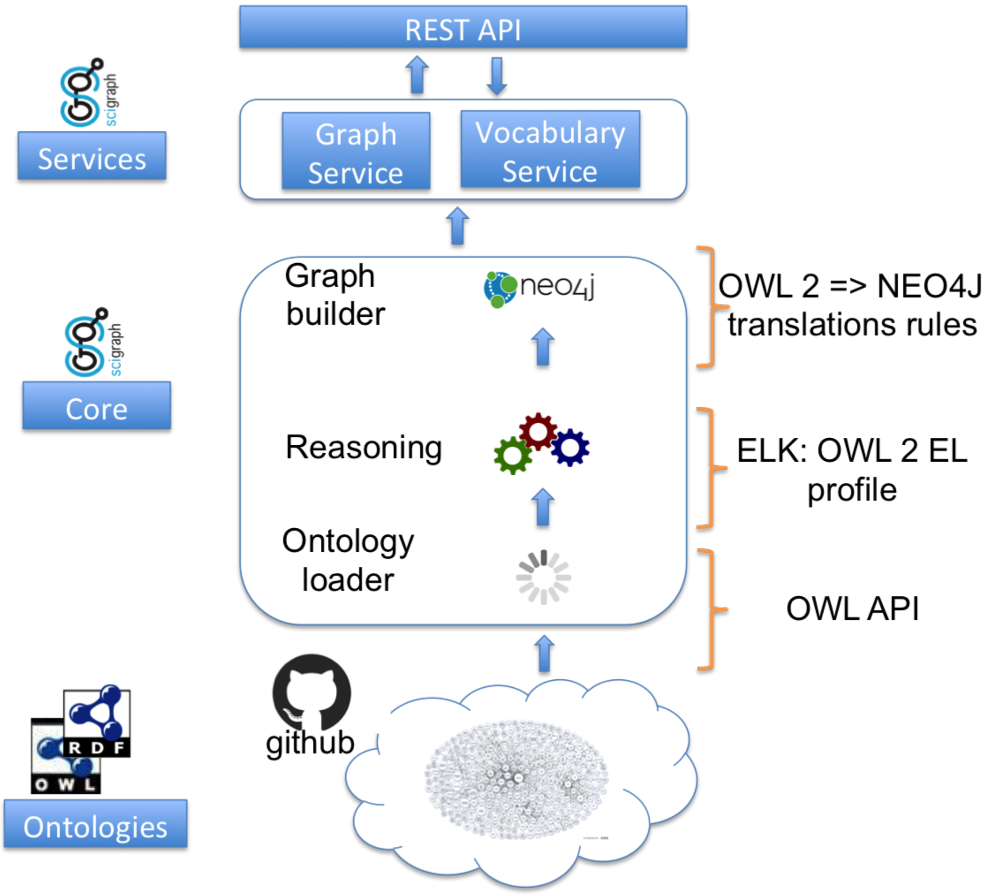


Fig. Overview of NIP Ontology Service

Ontology storage and edition

The various ontologies used in NIP are stored in a Github repository. Even if Github is not a proper ontology management system, it allows (among other things and just as for code bases):

- a multi-user ontology edition
- a basic ontology versions management
- to share NIP ontologies

To submit a new ontology entity, to report ontology related issues or just to give feedback, please submit a pull request or contact data.nip@humanbrainproject.eu.
Once ontologies are available in the Github repository, they have to be ingested and made available for clients (curators,indexer,...) by a REST API, which is the role of Scigraph.

Ontology ingestion mechanism using Scigraph

Scigraph, developed by the KnowledgeSpace team, is used as the main tool to implement the NIP ontology service. For further details about how to programmatically interact with it, please see the Scigraph API page. This section will specifically describe how Scigraph ingests ontologies.

Scigraph represents ontological data as graphs stored in an embedded NEO4J (in memory). To achieve that, several steps are needed:

- Ontology loading: OWL API library is used to load ontologies of different expressivity (the NIP platform supports RDF and OWL 2 EL) and serialized using different formats (Turtle, RDF/XML,...).

- Reasoning: The ELK reasoner is used to generate valid conclusions (entailments) as determined by the OWL 2 EL profile semantics. A common and simple reasoning task is classification:
  - it usually generates the transitive closure of the subsumption (subClassOf or isa) relation. An example of classification is shown below to briefly explains the reasoning process.

- OWL 2 to Neo4J Graph building: ontology entities and properties are respectively transformed into graph nodes and edges using a subset of OWL 2 semantics as translation rules. Many examples of those translations can be found here.

Let's go through the ontology ingestion mechanism with a simple example. The aim here is not to fully define it but to give an overview of the mechanism.
Let take an ontology made of the following subclass assertions (written here in OWL Functional Syntax for simplicity):

```
# Taxon is the ontology prefix name
# homo_sapiens is subclass of primates
# primates is subclass of mammalia

Prefix(:=<http://some.taxonomy.namespace/>)
Ontology(:Taxon
SubClassOf(:homo_sapiens :primates)
SubClassOf(:primates :mammalia)
)
```

The ingestion of this simple taxonomy takes place as in the following picture:

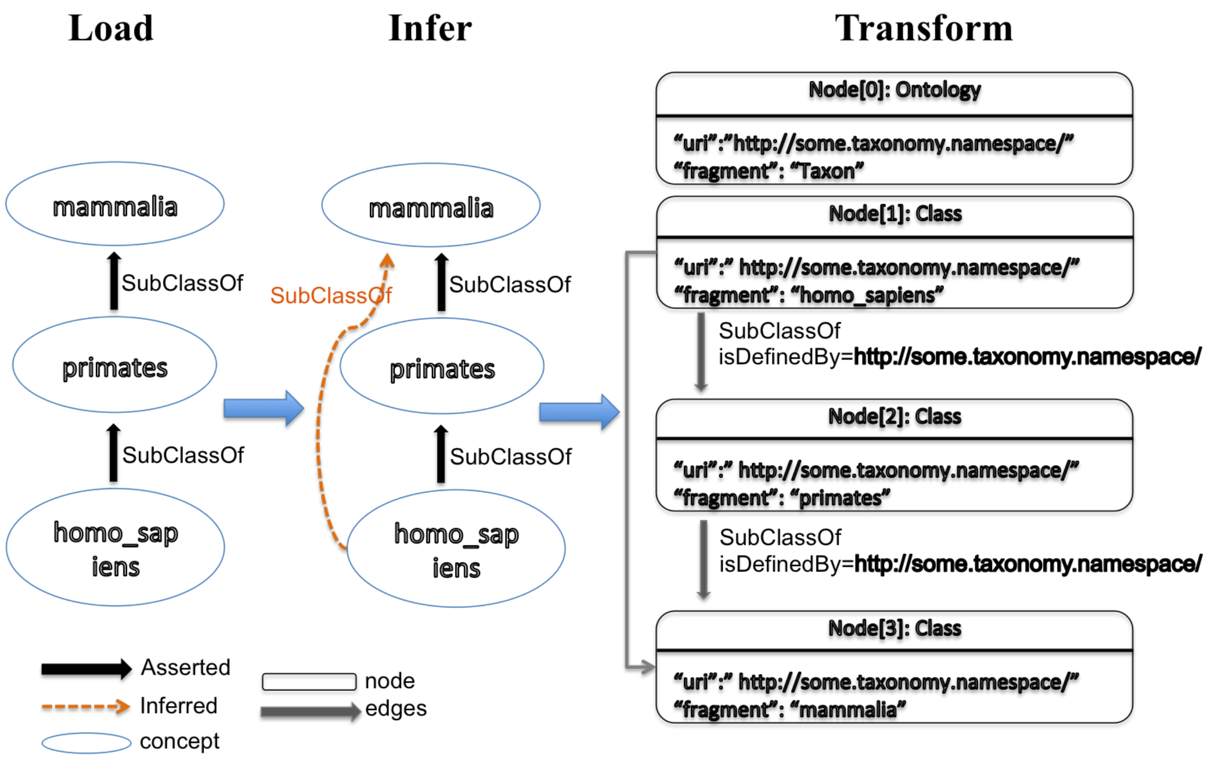


Fig. From OWL 2 ontology to NEOJ Graph

As shown in the previous picture:

- The relation "SubClassOf" being transitive, the reasoner will infer that "homo_sapiens" is a "SubClassOf" "mammalia",
- There is no way to reconstruct the original or the inferred ontology from the graph.
- The graph needs to be rebuilt whenever changes happened in the original ontology.
- The reasoning tasks only take place during the ingestion phase which means that there is no inline reasoning when querying the ontology service. Query response time is not impacted by reasoning.
- Graph operators can be used to access ontological data: e.g. traversal of the taxonomic direct acyclic graph (DAG).

Curation manual

General Information

What is a curation

Curation is a process to collect, annotate and validate scientific information that will be stored in a database. In the context of the NIP, curation means to extract and organize experimental data from the data providers using controlled terms and ontologies to enable queries across datasets and inter-operability.

Intended audience

- Data providers: as an introduction to the data provenance (HBP-PROV) and to identify the metadata that is required for the integration of their datasets.
- Data curators: as an introduction to the data provenance (HBP-PROV), the curation pipeline, the controlled terms and ontologies used and the different types of datasets and data formats currently supported.

Generic Dataset Metadata

Data Provenance

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.

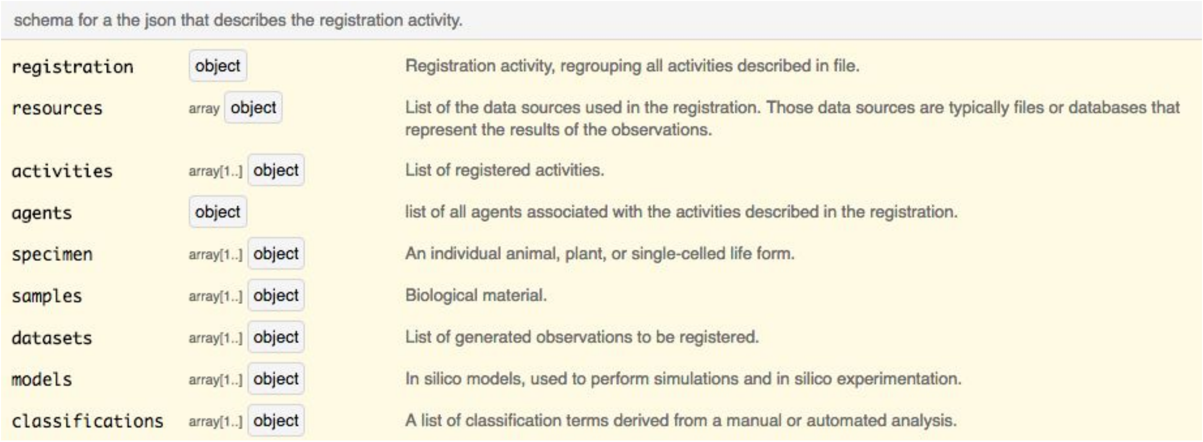
It allows to answer questions such as:

- 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 organizations and people were involved in the creation of the dataset?

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.

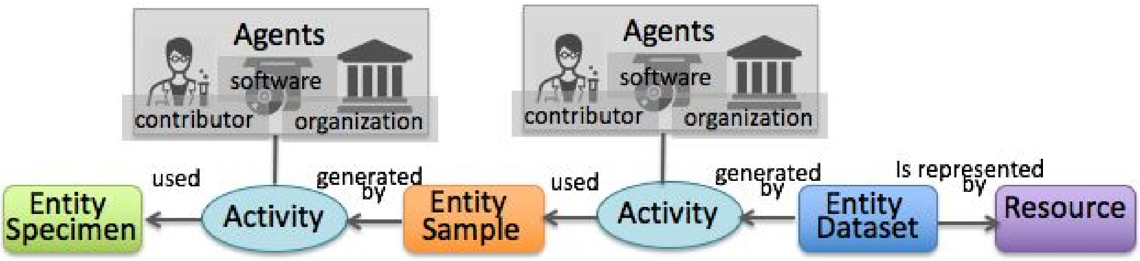


Fig. Simple example of provenance structure containing one dataset

Data storage

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

a) Web storage - storage accessible for web download on a dedicated web page indicated by the data provider

b) Archival storage (long term) - at one of these storage places:

- Zenodo (HBP community)
- HBP document service
- BBP GPFS

Privacy

The NIP currently supports two types of data :

1) HBP only (stored in the HBP document service or in Zenodo as “restricted”), metadata is visible. Download of the dataset requires HBP login.
2) Public (stored in Zenodo as “open access”)
   If public, the submitter has to choose one of the two licenses depending on the commercial potential use of your dataset. We strongly recommend you to carefully read the informations related to the following licenses in our ToS:
   - Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) <http://creativecommons.org/licenses/by-sa/4.0/>
   - Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Visibility

	Public	***HBP_only***	***Private***
View metadata	*yes*	*yes*	*no*
Download from search client	*yes*	*yes (after login)*	*no*
View from atlas viewer	*yes*	*yes (after login)*	*no*

Minimum Information about a Neuroscience DataSet (MINDS)

Data shared on the Neuroinformatic Platform (NIP) are enriched with minimal metadata to provide essential information in order to ensure traceability of any data artefact.
The metadata captured high-level description of experimental procedures, essential details about biological samples and experimental results. Metadata also registered scientists involved in data production. Structured metadata facilitates the integration and retrieval of data by defining a common language across many laboratories and experiment types. A large part of the metadata are specified using ontologies and dictionaries. The amount of mandatory metadata is kept to a minimum, applicable to all data sets. A standard metadata data model is defined based on W3C provenance standard and serves as the minimum specification required for all data, models and literature to be accessed via the NIP.

For a given data set the metadata would typically record the following:

- Specimen: age, sex, species, strain (when applicable)
- Classification properties, such as cell type
- Brain region either using ontological term or spatial coordinates
- Contributors and their affiliation
- Methods and parameters used to generate the data and the date they were created

- Access to the data and the format. Only Raw DATA or links to Raw DATA will be processed.
- License
- Publications

Table 1. Description of the usage of the metadata, the location on the HBP-PROV 3.0 structure and link to a dedicated ontology if it exists.

Type of metadata	*MINDS*	*HBP-PROV 3.0 location*	*Existing ontologies*
Specimen_taxonomy		*Specimen*	__
Specimen_strain		*Specimen*	__
Specimen_age		*Specimen*	
Specimen_sex		*Specimen*	__
Specimen_name		*Specimen*	
Brain_region		*Sample*	__ __ __
Protocol_title		*Activity*	
Protocol		*Activity*	
Measurement_methods		*Activity*	__
Experiment_date		*Activity*	
Contributors		*Activity*	
Affiliations		*Activity*	
Software		*Activity*	
Software version		*Activity*	
Data_category		*Entity dataset*	__
Data_type		*Resource*	__
File_size		*Resource*	
Checksum		*Resource*	
Original_file_name		*Resource*	
Publications		*Dataset*	
Reference_atlas		*Dataset*	__
Resolution		*Dataset*	

Cell_type		*Dataset*	__
Cell_name		*Dataset*	
Stimulus		*Dataset*	
Receptor_type		*Dataset*	__
License		*Dataset*	__

Description of the information needed for MINDS

1. Specimen

Describes an individual specimen or a group of specimens that were used for the experiment.

1.1 Taxonomy: NCBI taxonomy terms are used to describe the specimen and the NCBI ID is used as in the taxonomy ontology.
e.g. Mus musculus, obo:NCBITaxon_10090

1.2 Strain*: Experimental Factor Ontology (EFO) terms are used to describe the strain and EFO ID is used in the taxonomy ontology. If the term does not exist in EFO we create the term.
e.g. C57BL/6J, efo:EFO_0004472

1.3 Age: The age must be an integer, it can be expressed in days, weeks, months or years. For a group of specimens it can be added as an age range.
e.g.. post-natal day 14, P17-P22, between 25 - 40 years

1.4 Sex: The values can be male, female or hermaphrodite (intersex) and are found in the sex ontology.

1.5 Name: Holds the laboratory name given to the specimen.
e.g. R602

1.6 animalID: Holds the laboratory identification if there is a systematic identification system.
e.g. 344, 345, 346…

2. Brain region

Describes the brain anatomical region where the data came from. It can be the whole brain as for atlases, or a specific region. The most specific term should be used.
Each species has a specific ontology derived from parcellation schemes.

Species	*Parcellation scheme*
Homo sapiens	Allen Human Brain Atlas
Mus musculus	Allen Mouse Brain Atlas
Rattus norvegicus	Waxholm Space Atlas

3. Protocol title and Protocol

The protocol describes the details of the experimental process that was used to acquire and process the data. It contains the equivalent of the "materials and methods" of a publication. e.g.. title: "Generation of astrocyte single-cell transcriptomics from mouse hippocampus"

4. Agents:

4.1 Contributors, roles and affiliations

The name and surname of the contributors should be specified as well as their professional email. If it already exists in the database (persons.json), the ID should be retrieved and added to avoid any duplication of names.
The role of each contributor should be specified using the role ontology.
For the affiliations, the name of the laboratory and the organization should be retrieved.
If the organization already exists in the database (organisations.json), the ID should be retrieved and added to avoid any duplication of names.

4.2 Software

If a specific program was used to generate a new dataset as part of the submission, then the name and the version should be recorded as well as an URL where the software is stored so that other users are able to download it.

5. Dataset

Providers must give access to the data and the format. Only Raw DATA or links to Raw DATA will be processed.

6. Publication

If the work has been published on a scientific paper the PubmedID or DOI should be provided.

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating neuroscience datasets:

	Provenance Data Model	***Provenance Ontologies***	***Generic Metadata***
Maturity Level	*high*	*high*	*medium**

(*) while a set of metadata has been defined, it wis still actively being discussed with the community and it is still possible that it might evolve in the future.

Data Category Specific Metadata and Formats

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

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 and
- left_right resolution

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

- coronal,
- axial and
- 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 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.
e.g. 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 Tiff format in one orientation like sagittal. Tagged Image File Format, abbreviated 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 visualization and image processing applications.

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating brain atlas datasets:

	Standard Data Format	***Standard Dissemination Service***	***Specialised Metadata***
Maturity Level	*high**	*high**	*high*

(*) while the BBIC and Image Service API are already available and used in production for several years, we keep working with the community to adopt more open standards that fulfills a broader range of use cases.

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

It 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.

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating brain atlas datasets:

	Standard Data Format	***Standard Dissemination Service***	***Specialised Metadata***
Maturity Level	*medium**	*low***	*medium****

(*) So far, most data integration cases seem to use the IGOR data format. We are currently investigating the use of Neurodata Without Border as an open standard alternative to integrate such datasets into the platform.

(**) at this stage, no requirements has been formulated by the community to get a specialised access to morphology data. We will work toward building such data access with the requirement are better defined.

(***) working with the community, we have identified use cases related to how users might want to search electrophysiology data. This has helped us defined extra metadata to be captured to enable such queries to be executed through the KSeach API.

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) can contain single or multiple cell reconstructions.

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

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating brain atlas datasets:

	Standard Data Format	*Standard Dissemination Service*	*Specialised Metadata*
Maturity Level	*high*	*low**	*medium***

(*) at this stage, no requirements has been formulated by the community to get a specialised access to morphology data. We will work toward building such data access with the requirement are better defined.
(**) working with the community, we have identified use cases related to how users might want to search morphology data. This has helped us defined extra metadata to be captured to enable such queries to be executed through the KSeach API.

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.

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating receptor density datasets:

	Standard Data Format	*Standard Dissemination Service*	*Specialised Metadata*
Maturity Level	*low**	*low**	*low**

(*) the platform so far has received extremely few datasets of this type and it has prevented the task of standardisation from developing as fast as some other data types. We foresee that with new datasets being submitted to the platform we will be better able to standardise this kind of data.

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.

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating transcriptomics datasets:

	Standard Data Format	*Standard Dissemination Service*	*Specialised Metadata*
Maturity Level	*low**	*low**	*low**

(*) the platform so far has received extremely few datasets of this type and it has prevented the task of standardisation from developing as fast as some other data types. We foresee that with new datasets being submitted to the platform we will be better able to standardise this kind of data.

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 that.

Maturity Level

The table below describes the current level of progress of the Neuroinformatics Platform toward integrating kinetic model datasets:

	Standard Data Format	***Standard Dissemination Service***	***Specialised Metadata***
Maturity Level	*low**	*low**	*low**

(*) the platform so far has received extremely few datasets of this type and it has prevented the task of standardisation from developing as fast as some other data types. We foresee that with new datasets being submitted to the platform we will be better able to standardise this kind of data.

Metadata registration - manual curation pipeline

The metadata registration for the manual curation pipeline is currently done using an excel table that follows the HBP-PROV format. Each field is characterized by a node which can be an activity, an entity, an agent, a cross_reference, or an annotation. Each node has a type, a property_name, a property_value and a cross link to an ontology if it exists.

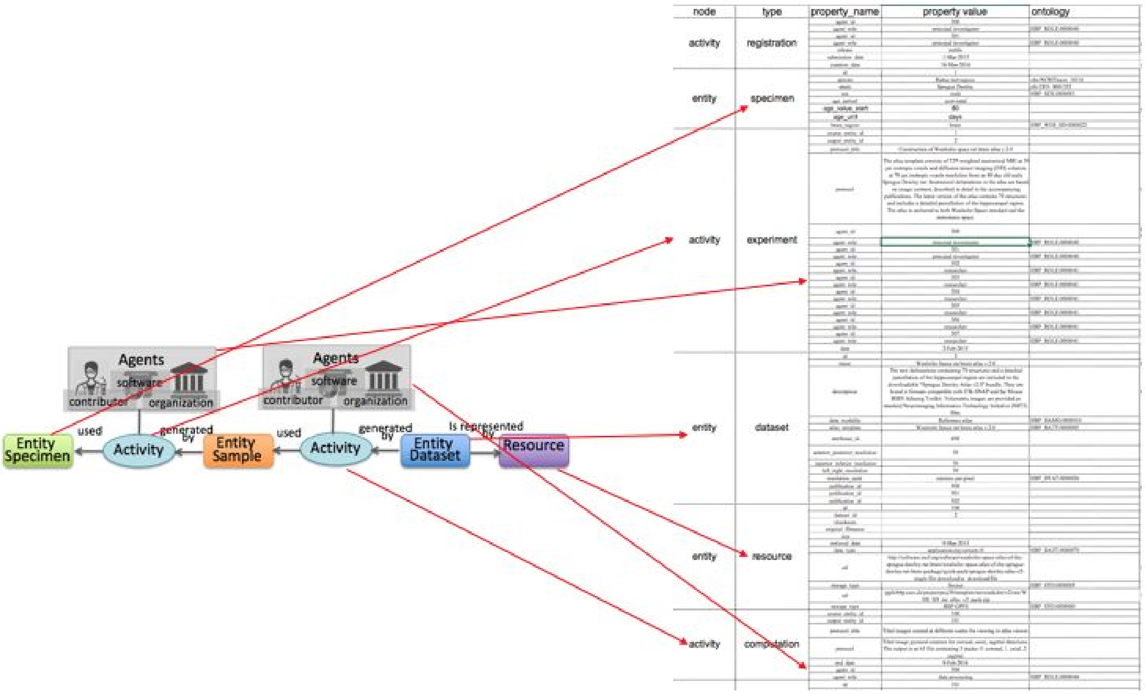


Fig. The detailed contents of the fields used in the excel table can be found in the XLS format section.

Please find here an example of how the metadata is added to the fields:
Convention for the use of temporary identifiers.

Entities – specimen – experiment - dataset: use temporary identifiers 1-99
Entity- resource: use temporary identifiers 100-299
Agents – contributor – software: use temporary identifiers 300–499
Cross-references – publications: use temporary identifiers 500–599
Annotation - attributes: use temporary identifiers 600–699

These numbers represent temporary identifiers which will be replaced by UUIDs during the conversion of the excel table into json.
If a contributor, organization or software already exists in the database, their UUID should be used instead of a temporary identifier.

Ontologies
Existing terms can be found in GitHub. New terms can be added to the existing ontologies in order to allow the registration of a dataset that requires them.

Activity – Registration

In the registration activity is captured the principal investigator(s) (**agent_id**) whose organization will appear as the main organization associated with the dataset. Here is also captured if the dataset is to be public or restricted to HBP (**release**) and the submission and curation **dates** as follows:

Property_name	***Property_value***	***Ontology/value***
agent_id*	*eb638fe9-e7a8-4c9c-8475-5201ec05066c*	
agent_role*	*e.g. principal investigator*	Role *e.g. HBP_ROLE:0000040*
release*		*public or HBP*
submission_date*	*DD-MM-YYYY*	
curation_date	*DD-MM-YYYY*	

Entity – Specimen

Here appears all the information related to the organism used during the experiment to generate the submitted dataset: the **species, strain, sex, age** (**age_value_start**) and the species specific brain region. In the case that a group of organisms was used, there is the possibility of registering an age **range** by filling in the **age_value_start** and the **age_value end**. The valid **age_period** values are pre- or post-natal, and the **age_units** can be days, weeks, months or years.

Property_name	***Property_value***	***Ontology/value***
id*	*1*	
species*	*e.g. Rattus norvegicus*	taxonomy *e.g. obo:NCBITaxon_10116*
strain	*e.g. Wistar*	taxonomy *e.g. efo:EFO_0001342*
sex	*e.g. female*	sex *e.g. HBP_SEX:0000002*
age_period	*e.g. pre-natal or post-natal*	
age_value_start		

age_value_end		
age_unit	days, weeks, months, years	
brain_region		brain (species specific)

Activity – Experiment

In the activity experiment is captured the **protocol** and the **protocol_title** of the experiment and the principal investigator and all contributors (**agent_id**) and their **roles**. The **date** at which the experiment was finished is also captured. The **source_entity_id** is the id of the Entity - Specimen and the **output_entity_id** is the id of the Entity - Dataset. There can be one or more datasets as output entities.

Property_name	Property_value	Ontology
source_entity_id*	1	
output_entity_id*	2	
protocol_title	free text	
protocol	free text	
agent_id*	e.g. eb638fe9-e7a8-4c9c-8475-5201ec05066c	
agent_role*	e.g. principal investigator	role e.g. HBP_ROLE:0000040
date	DD-MM-YYYY	

Entity – Dataset

In this section is captured all the information pertaining to the dataset. The **name** given here will appear as the visible name in the NIP. For atlases, the **description** is also displayed in the atlas viewer. The **data_modality** or **category** is used to group each dataset into a separate category, e.g. single cell transcriptomics, single cell morphology, electrophysiology etc...The **atlas_template** field has information on the space to which the dataset was registered. If the dataset is public, the **license** approved by the submitter is also added here. The **cell_type** is added for example for morphologies. The **attributes** are flexible placeholders that can currently be used for atlas resolution, for electrophysiology, receptor type and other technical terms associated with the dataset. These terms are not currently in the HBP-PROV structure but will be integrated in the future. The **attribute_id** points to the block Annotation Attributes.

Property_name	***Property_value***	***Ontology***
id*	*2*	
name*	*free text*	
description	*free text*	
data_modality*	*e.g. electrophysiology*	*categories* *e.g. HBP_DAMO:0000008*
atlas_template	*e.g. Waxholm Space rat brain atlas v.2.0*	*atlas* *e.g. HBP_BATT:0000003*
license	*e.g. Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International*	*license* *e.g. HBP_LICENSE:0000002*
cell_type	*e.g. Layer IV pyramidal cell*	*cell type* *e.g. HBP_CELL:0000029*
attributes_id	*600*	
anterior_posterior_resolution	*numerical value*	
superior_inferior_resolution	*numerical value*	
left_right_resolution	*numerical value*	
resolution_units	*e.g. microns per pixel*	*attributes* *e.g. HBP_DTAT:0000002*
publication_id	*500*	

Entity – Resource

The resource gives information about the dataset, what is the **original_filename**, the **size** of the file, the **data_type** (mime type), the **checksum** (generated using `md5sum)`, the url to be used to download the dataset and where it is stored (**storage**). The dataset can be stored at several storage places. In the example it is found both in Zenodo and the document service.

Property_name	***Property_value***	***Ontology***
id*	*100*	

dataset_id	*2*	
checksum	*e.g. b042e5317ce871366ae0a50e7d22c13d*	
original_filename	*e.g. expE1.txt*	
size	*e.g. 416172*	
retrieval_date	*DD_MM_YYYY*	
data_type	*e.g. application/vnd.hbp.traces.txt;version=0*	data type
url	*e.g. https://zenodo.org/record/51954/files/expE1.txt*	
storage_type	*e.g. Zenodo*	storage *e.g. HBP_STO:0000007*
url*	*e.g. 2c1efc42-857e-4c64-afd0-789ad6e17526*	
storage_type	*e.g. HBP Document service*	storage *e.g. HBP_STO:0000008*

Agent – Contributor

For agents can be contributors, software temporary identifiers 300–499 are used as **id** if there is not a UUID in the database. Otherwise the UUID should be added in this field. Contributors should have the following information: the **family_name, the given_name, the email, lab_name** if available and **the organization.**

Property_name	***Property_value***	***Ontology***
id*	*582c484b-386c-4989-98b2-69a675427536*	
family_name*	*Cherubini*	
given_name	*Enrico*	
email	*e.cherubini@ebri.it*	
lab_name		
organization_id	*d09ab56d-f2ae-495c-a389-93c965a85ebb*	

organization	*European Brain Research Institute, Rome, Italy*	

Software should contain the following information: the **name** of the software, the **version** and the **url** from where the software can be downloaded.

Property_name	***Property_value***	***Ontology***
id*	*500*	
name	*e.g. svgcreator/template_mouse_allen_ccfv3/allen_SvgCreate.py*	
version	*e.g. 43b80a77d852db584bd78cfd88f07bdca3aaf7f6*	
url	*e.g. https://bbpcode.epfl.ch/browse/code/datamining/datapreprocessing/commit/?h=refs/heads/master&id=43b80a77d852db584bd78cfd88f07bdca3aaf7f6*	

Cross-Reference – Publication

For publications we use temporary **id** identifiers 500–599. The **storage_type** should be captured, the most common ones are DOIs, PubMed IDs, or documentation published on the Web.

Property_name	***Property_value***	***Ontology***
id*	*500*	
pubmed_id*	*e.g. 23788795*	
storage_type*	*PubMed*	*storage* *e.g. HBP_STO:0000011*
id*	*501*	
doi*	*e.g. doi:10.1371/journal.pone.0008595*	
storage_type*	*DOI*	*storage* *e.g. HBP_STO:0000012*

id*	502	
url*		
storage_type*	Web	storage e.g.HBP_STO:0000002

Annotation - Attributes

The **attributes** are flexible placeholders that can currently be used for atlas resolution, for electrophysiology, receptor type and other technical terms associated with the dataset. These terms are not currently in the HBP-PROV structure but will be integrated in the future. Example of the attributes for an Atlas:

Property_name	Property_value	Ontology
id*	600	
HBP_DTAT:0000001	coronal	
HBP_DTAT:0000001	axial	
HBP_DTAT:0000001	sagittal	
id*	601	
HBP_DTAT:0000023	1250	
HBP_DTAT:0000024	1250	
HBP_DTAT:0000025	1250	
HBP_DTAT:0000002	microns per pixel	attributes e.g. HBP_DTAT:0000026

Example of the attributes for datasets containing receptor data:

Property_name	Property_value	Ontology
id*	600	
HBP_DTAT:0000021	AMPA receptor	receptor

		e.g. MESH:D018091
id*	601	
HBP_DTAT:0000021	NMDA receptor	receptor e.g. MESH:D016194
id*	602	
HBP_DTAT:0000021	Metabotropic glutamate receptor 2	receptor e.g. MESH:C108822

Metadata integration - manual curation pipeline

The integration of an excel table into the NIP requires 4 steps:

1. Converter: The converter takes an excel file as input and generates a json file consistent with HBP-PROV v3.0 json schema. It gives UUIDs to all the temporary identifiers found in the excel table.
2. Resolver: The resolver checks if the contributors (persons), organizations, protocols and software already exist in the database. If they exist, it retrieves the corresponding UUID. If they do not exist in the database, they are created as new entities and given a UUID.
3. Validation: The validation step does checks at **Syntactic** and **Semantic** levels:
    a. **Syntactic**: verifies that the format is correct against the corresponding Json schema, this encompasses the naming and cardinality of entities as well as their attributes.
    b. **Semantic**: we have the ability of implemented custom validation rules that can perform neuroscience oriented verifications as well as other aspects that a JSON schema cannot enforce.
4. Upload and Indexing: Uploads the validated json file into the NIP and makes it available though Ksearch.

The pipeline can be run in the following environments: development, staging and production.

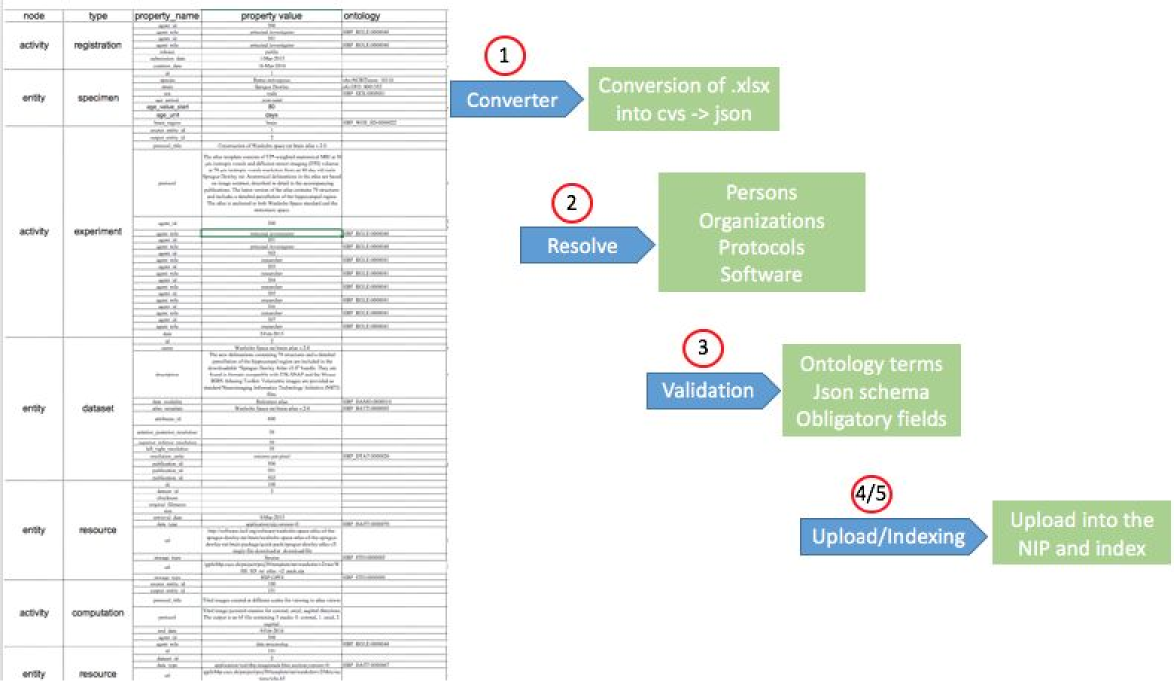
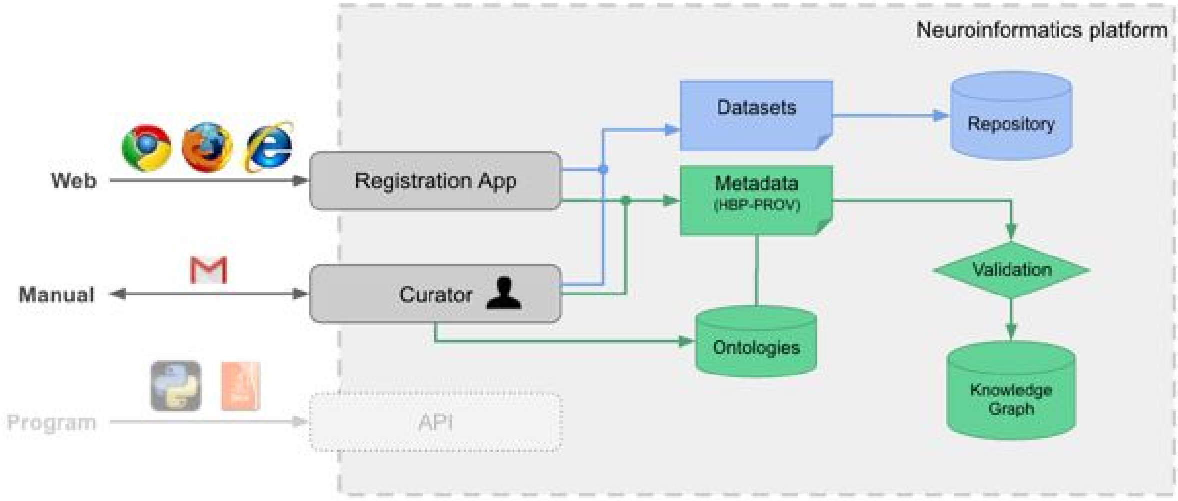


Fig. Steps of manual integration pipeline

Contribute Data

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



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 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.

Restricted Access Registration

In order to register your **HBP community onl**y within the NIP, the following information will be required:

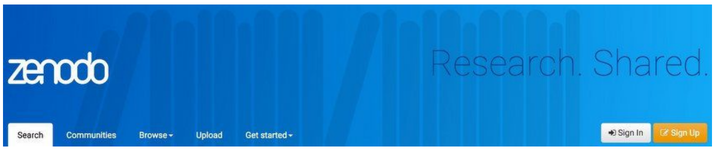
I Contribute data in an **Restricted Access Right** manner?

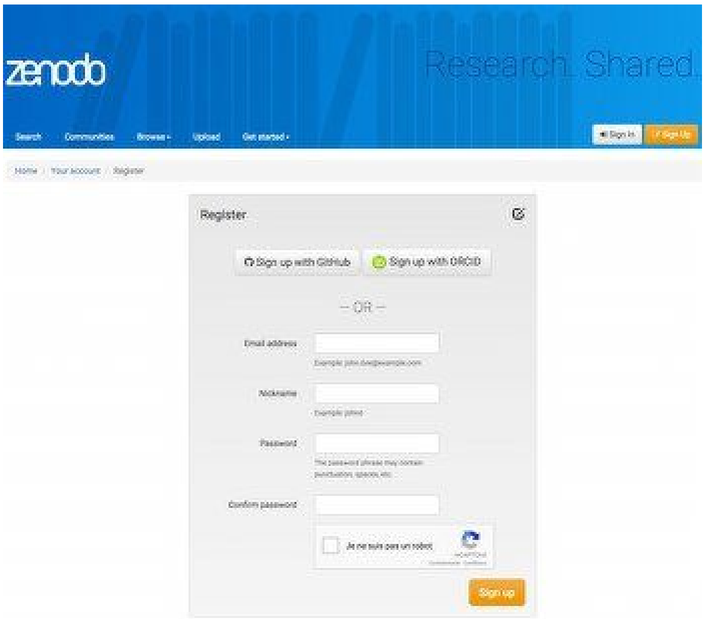
The Neuroinformatics Platform is using ZENODO service has a repository for your highly valuable data.

To be noticed:

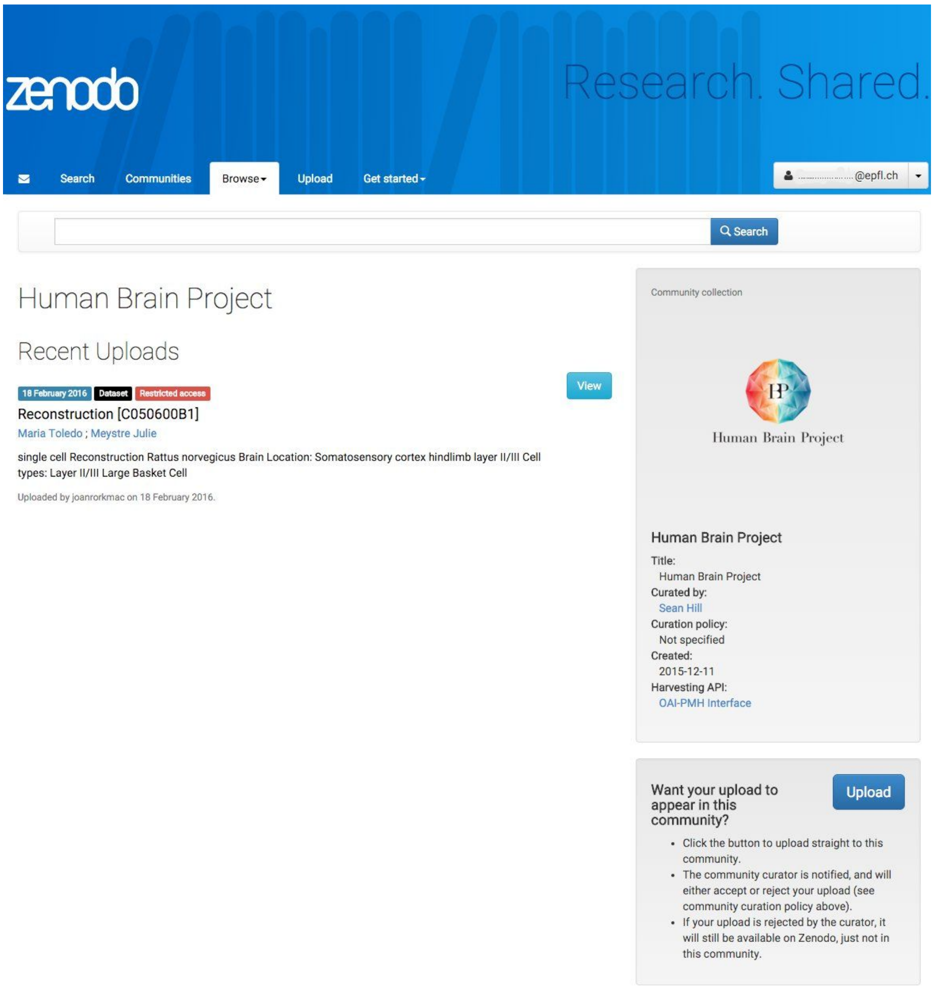
- the data you submit is published as-is (ie. not curated)
- we do not deal with pre-publication
- records are final, once published you cannot change them

1) Sign up into Zenodo using your institution e-mail





2) Join the Human Brain Project Community
Click on the following link to reach the Human Brain Project Community:https://zenodo.org/collection/user-hbp



Human Brain Project

Recent Uploads

18 February 2016 | Dataset | Restricted access

Reconstruction [C050600B1]

Maria Toledo ; Meystre Julie

single cell Reconstruction Rattus norvegicus Brain Location: Somatosensory cortex hindlimb layer II/III Cell types: Layer II/III Large Basket Cell

Uploaded by joanrorkmac on 18 February 2016.

Community collection

Human Brain Project

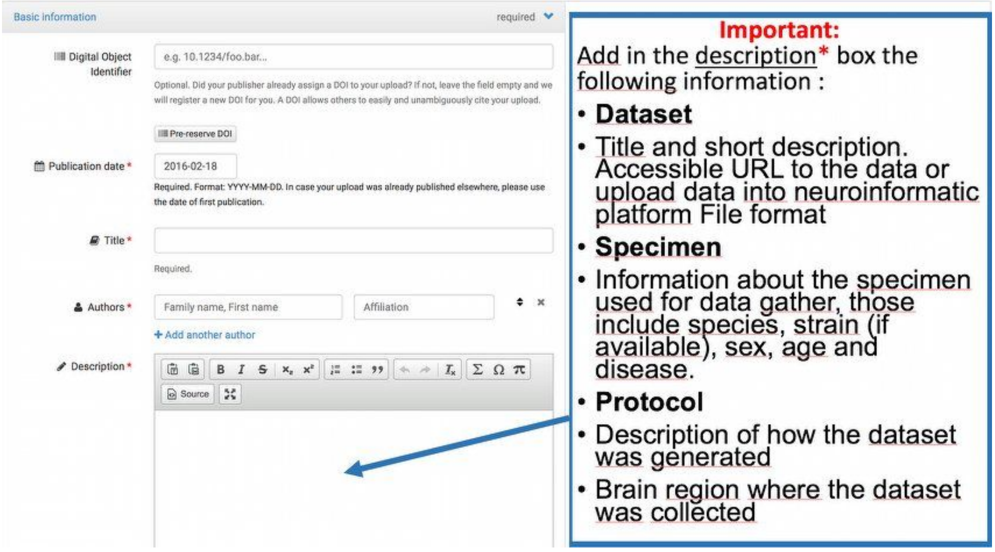
Title:
Human Brain Project
Curated by:
Sean Hill
Curation policy:
Not specified
Created:
2015-12-11
Harvesting API:
OAI-PMH Interface

Want your upload to appear in this community?

- Click the button to upload straight to this community.
- The community curator is notified, and will either accept or reject your upload (see community curation policy above).
- If your upload is rejected by the curator, it will still be available on Zenodo, just not in this community.

3) Fill-in up the Metadata

Submit your data from your local drive or using the DROPBOX service



Basic information — required

Digital Object Identifier: e.g. 10.1234/foo.bar...

Optional. Did your publisher already assign a DOI to your upload? If not, leave the field empty and we will register a new DOI for you. A DOI allows others to easily and unambiguously cite your upload.

Publication date *: 2016-02-18

Required. Format: YYYY-MM-DD. In case your upload was already published elsewhere, please use the date of first publication.

Title *:

Required.

Authors *: Family name, First name | Affiliation

Description *:

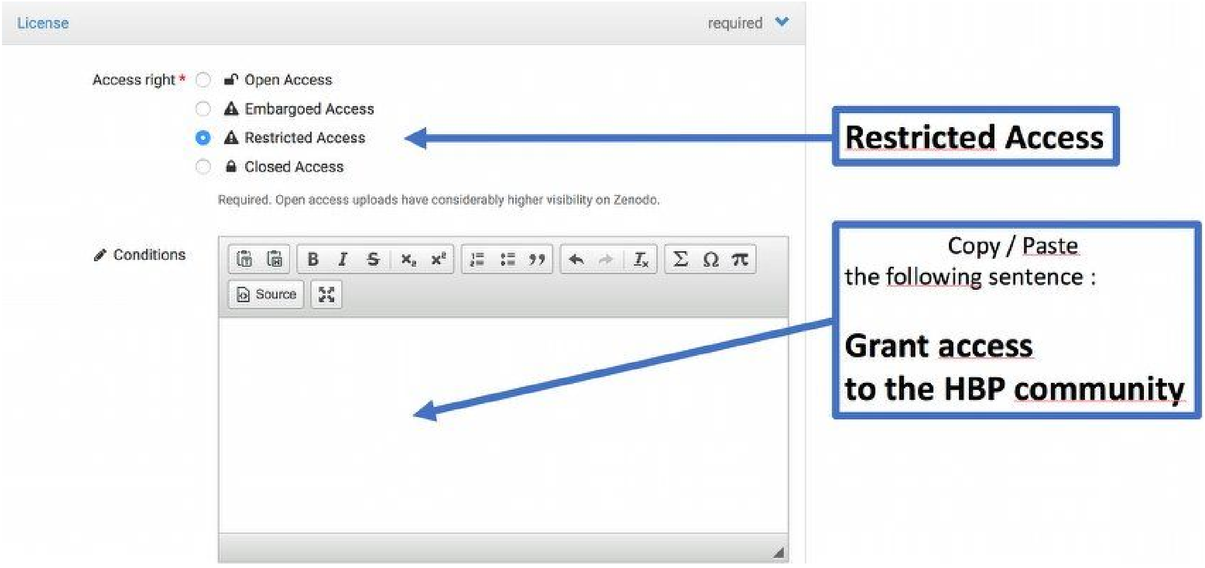
Important:

Add in the description* box the following information :

- **Dataset**
- Title and short description. Accessible URL to the data or upload data into neuroinformatic platform File format
- **Specimen**
- Information about the specimen used for data gather, those include species, strain (if available), sex, age and disease.
- **Protocol**
- Description of how the dataset was generated
- Brain region where the dataset was collected

4) Access Right

Please, select the Restricted access right option and choose a one year HBP community access.



License required

Access right *
○ Open Access
○ Embargoed Access
● Restricted Access
○ Closed Access

Required. Open access uploads have considerably higher visibility on Zenodo.

Conditions

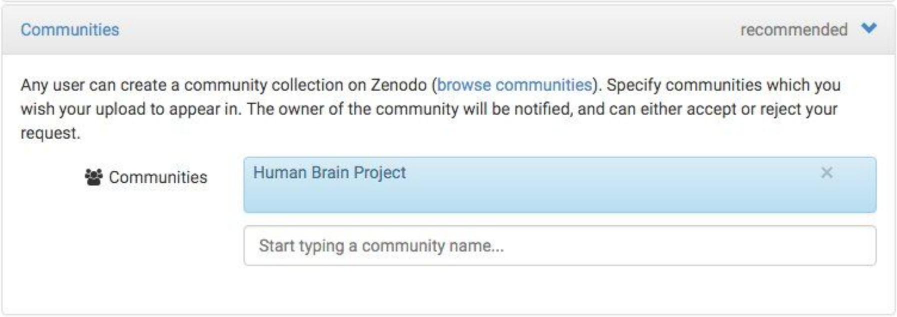
Restricted Access

Copy / Paste
the following sentence :

Grant access
to the HBP community

5) Choose the HBP community

Please, select the human Brain project community in order to visualized rapidly your data within the our community service.



Communities recommended

Any user can create a community collection on Zenodo (browse communities). Specify communities which you wish your upload to appear in. The owner of the community will be notified, and can either accept or reject your request.

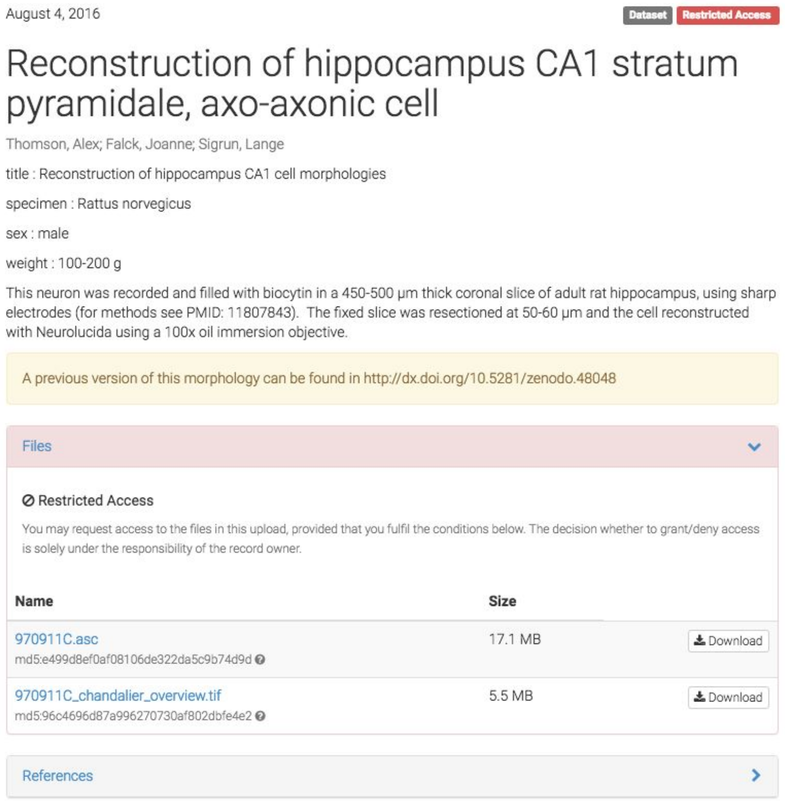
Communities: Human Brain Project ×

Start typing a community name...

6) Submit your File

To Note: By choosing the DROPBOX service, you will increase the Upload limitation up to 10 Giga.

sample of available information :



August 4, 2016

Reconstruction of hippocampus CA1 stratum pyramidale, axo-axonic cell

Thomson, Alex; Falck, Joanne; Sigrun, Lange

title : Reconstruction of hippocampus CA1 cell morphologies

specimen : Rattus norvegicus

sex : male

weight : 100-200 g

This neuron was recorded and filled with biocytin in a 450-500 µm thick coronal slice of adult rat hippocampus, using sharp electrodes (for methods see PMID: 11807843). The fixed slice was resectioned at 50-60 µm and the cell reconstructed with Neurolucida using a 100x oil immersion objective.

A previous version of this morphology can be found in http://dx.doi.org/10.5281/zenodo.48048

Files

⊘ Restricted Access

You may request access to the files in this upload, provided that you fulfil the conditions below. The decision whether to grant/deny access is solely under the responsibility of the record owner.

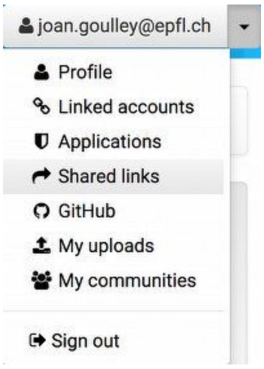
Name	Size	
970911C.asc md5:e499d8ef0af08106de322da5c9b74d9d	17.1 MB	Download
970911C_chandalier_overview.tif md5:96c4696d87a996270730af802dbfe4e2	5.5 MB	Download

References

II Retrieve your data?

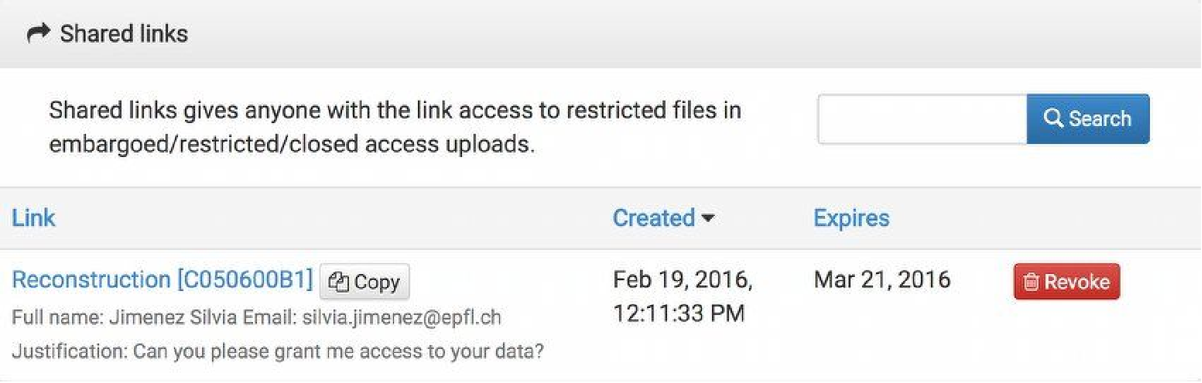
1) sign in in Zenodo using your institution e-mail

2) select the Shared links option



3) Select your Dataset from your list

4) Download the selected File



Shared links gives anyone with the link access to restricted files in embargoed/restricted/closed access uploads.

Link	Created ▾	Expires	
Reconstruction [C050600B1] Copy Full name: Jimenez Silvia Email: silvia.jimenez@epfl.ch Justification: Can you please grant me access to your data?	Feb 19, 2016, 12:11:33 PM	Mar 21, 2016	Revoke

Brain Atlasing

Background

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 capturing 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 **visualizable**, 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.

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 visualized in the Atlas viewer.

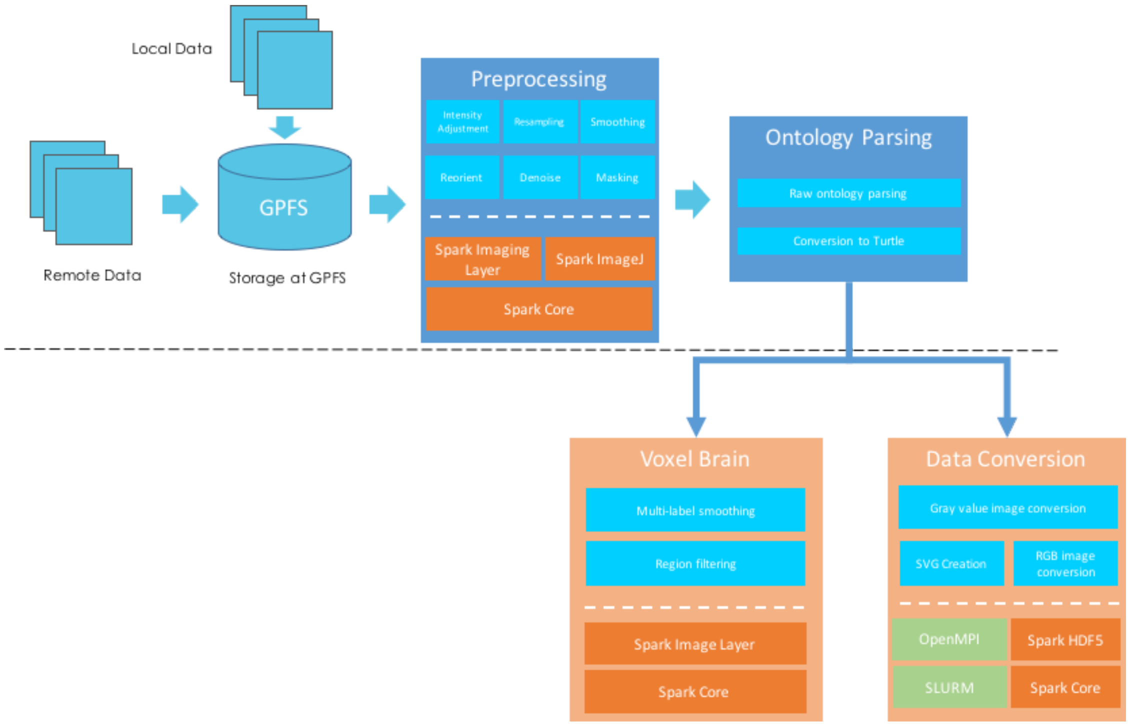


Fig. Schema of integrating existing atlas

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 visualized in the Atlas viewer.

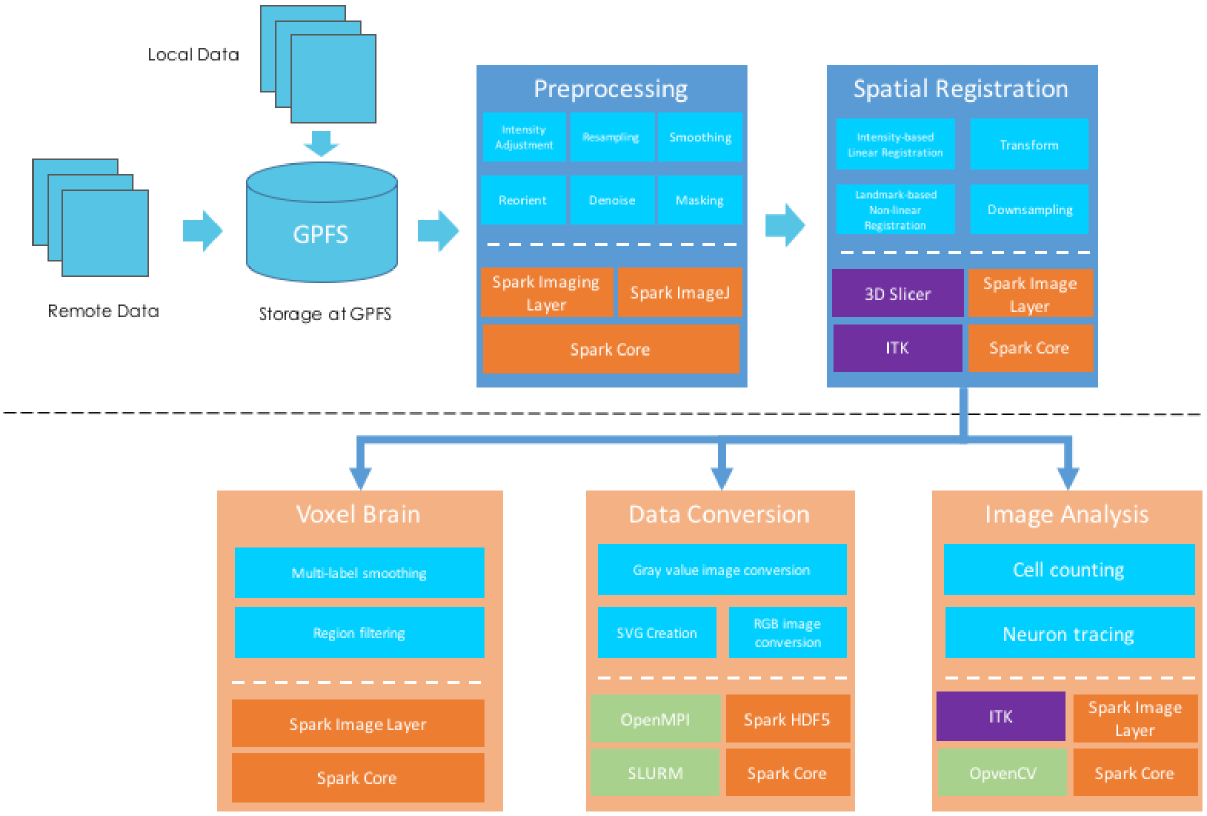


Fig. Pipeline of integration of new image data

Data processing modules

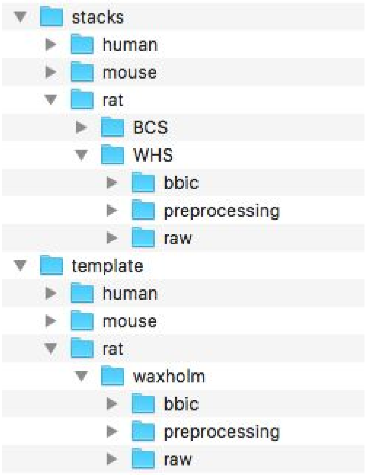
The above pipelines of integrating image data use the processing modules that are individually developed in-house. The tools are written in Python and C++. The Pythons tools aim at simple image operations which requires less computation and memory usage. Algorithms that need high-performance computing power are developed in C++ which can be run in both stand-alone 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 organized per species. In stacks, data is further categorized into the atlas space it maps to. Then, the subdirectory contains:

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

A typical data storage structure is shown as follow:



```
▼ stacks
    ▶ human
    ▶ mouse
    ▼ rat
        ▶ BCS
        ▼ WHS
            ▶ bbic
            ▶ preprocessing
            ▶ raw
▼ template
    ▶ human
    ▶ mouse
    ▼ rat
        ▼ waxholm
            ▶ bbic
            ▶ preprocessing
            ▶ raw
```

Data preprocessing

Raw image data that received from the data contributor usually does not conform with the requirement for further processing, analysis and visualization, therefore is required to apply data preprocessing.

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 visualization.

To correct the intensity inhomogeneity, one can use gamma correction tool and background subtraction tool available in ImageJ (https://imagej.nih.gov/ij/) or FIJI (https://fiji.sc/), together with the semi-automatic segmentation tool in Amira (https://www.fei.com/software/amira-3d-for-life-sciences/).

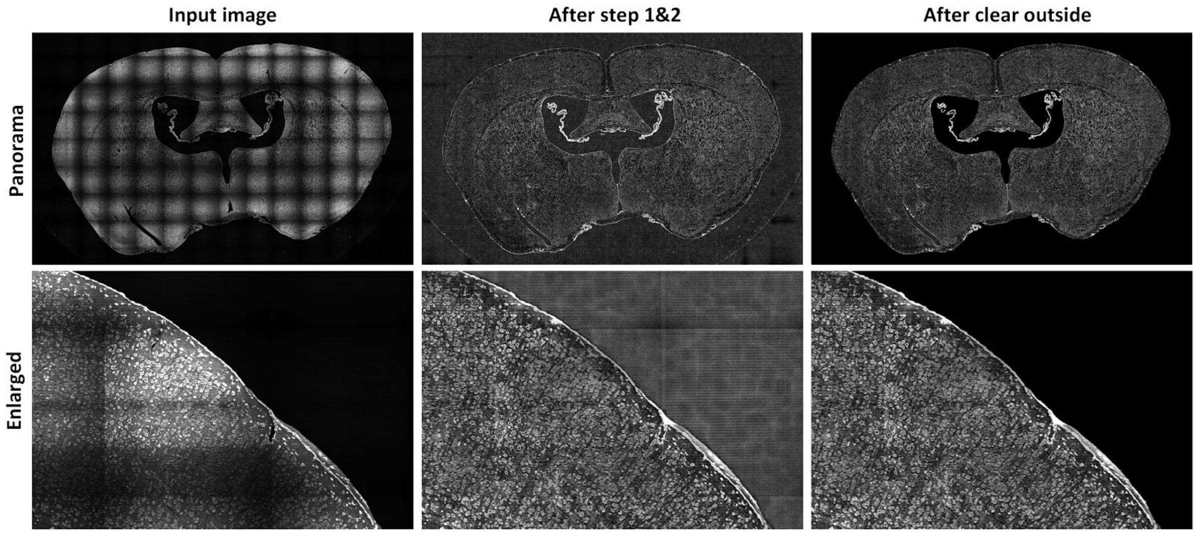


Fig. Intensity correction result from a GAD-stained mouse brain image

Isotropic Resampling

The atlas viewer treats the image data as isotropic at each orientation. For anisotropic data, especially microscopic images which has much larger inter-slice spacing than in-plane spacing, image plane whose orientation differs from the original scan would be stretched or squeezed in the viewer. To this end, the image stacks to be visualized in the atlas viewer need to be isotropic at all three orientations (axial, coronal, and sagittal), as indicated in the following steps:

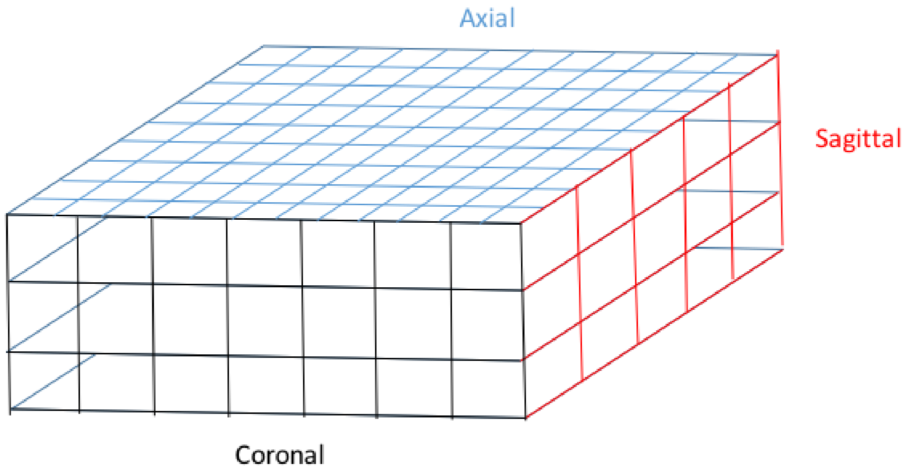


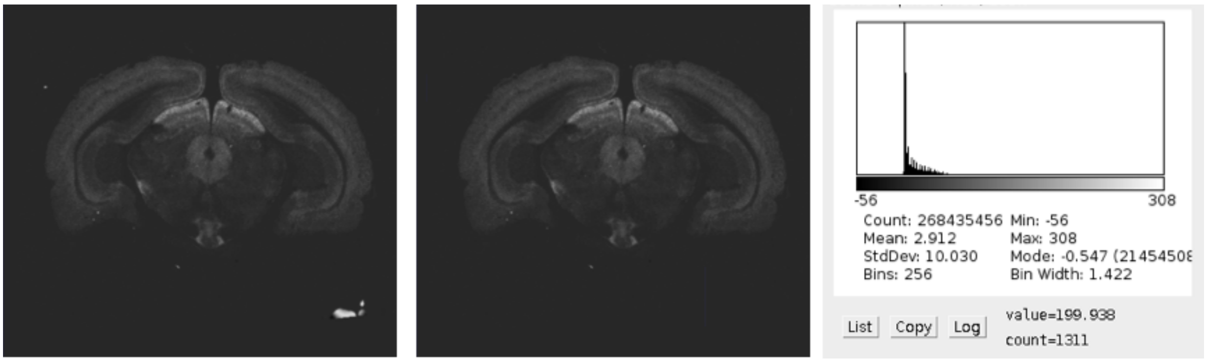
Fig. An example of an anisotropic image stack. The isotropic resampling resamples the coronal and sagittal planes into isotropic resolution with the pixel size of the slice spacing.

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, 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.

To use the Python tool, the user defines two thresholds—Upper and Lower—and two intensity values—Inside and Outside. For each pixel in the input image, the value of the pixel is compared with the lower and upper thresholds. If the pixel value is inside the range defined by [Lower,Upper] the output pixel is assigned the InsideValue. Otherwise the output pixels are assigned to the OutsideValue.



```
Count: 268435456 Min: -56
Mean: 2.912      Max: 308
StdDev: 10.030   Mode: -0.547 (21454508
Bins: 256        Bin Width: 1.422
value=199.938
count=1311
```

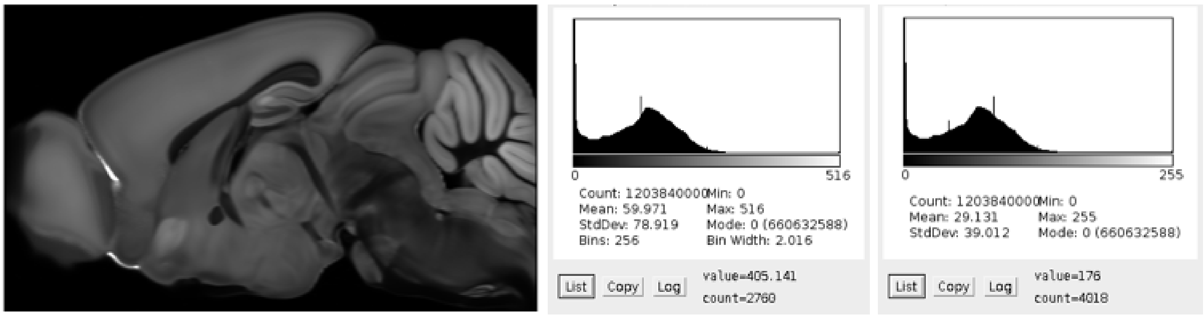
Fig. Threshold the hyper-intensity pixels (bottom right) in a rat m2 receptor image.

Intensity rescaling

Due to various imaging acquisition methods, the images usually have different intensity distribution, resulting in pixel value types being 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.

User can perform the intensity rescaling using Rescale.py, which is implemented based on the SimpleITK library. This tool applies pixel-wise a linear transformation to the intensity values of input image pixels. The linear transformation is defined by the user in terms of the minimum and maximum values that the output image should have (e.g. 0 and 255). The mapping of the intensity values is represented as:

outputPixel = (inputPixel - inputMin) x 255 / (inputMax - inputMin)



```
Count: 1203840000Min: 0
Mean: 59.971      Max: 516
StdDev: 78.919    Mode: 0 (660632588)
Bins: 256         Bin Width: 2.016
value=405.141
count=2760
```

```
Count: 1203840000Min: 0
Mean: 29.131      Max: 255
StdDev: 39.012    Mode: 0 (660632588)
value=176
count=4018
```

Fig. Intensity rescaling on Allen CCFv3. Intensity distribution is rescaled from 0-516 to 0-255.

Ontology parsing

Atlases usually comprise terminologies of brain regions or parcels. These terminologies are defined as ontology, which needs to submitted to a Github repository.

An ontology parse tool is developed to convert the raw ontology file in various format (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. The brain parcel id, synonym, and parent structure is captured by the conversion tool and is encoded as following in the Turtle file.

```
HBA:4007 rdf:type owl:Class ;
                rdfs:label "telencephalon"@en ;
                nsu:synonym "Tel"@en ;
                rdfs:subClassOf [ rdf:type owl:Restriction ;
owl:onProperty nsu:part_of ; owl:someValuesFrom HBA:4006 ] .
```

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 the 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 is 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 stand-alone 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.

Landmark registration module is used in the 3D Slicer to pick corresponding landmark in both subject image and atlas image. Then, a linear transformation is generated represented by a 4x4 matrix.

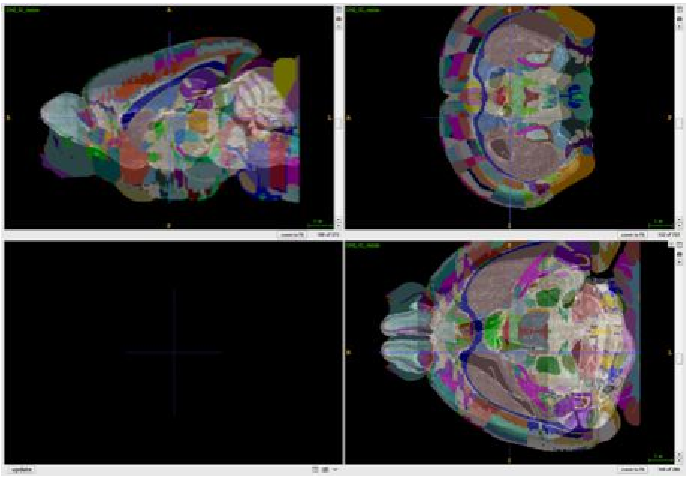


Fig. Linear alignment between the atlas and subject image

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.

Landmark registration module is used in the 3D Slicer to pick corresponding landmark in both subject image and atlas image. A Thin-plate-spline (TPS) interpolation model is then used to non-linearly transform the image. Voxel displacements, considered known in a restricted set of locations in the image, are interpolated for the rest of the image domain.

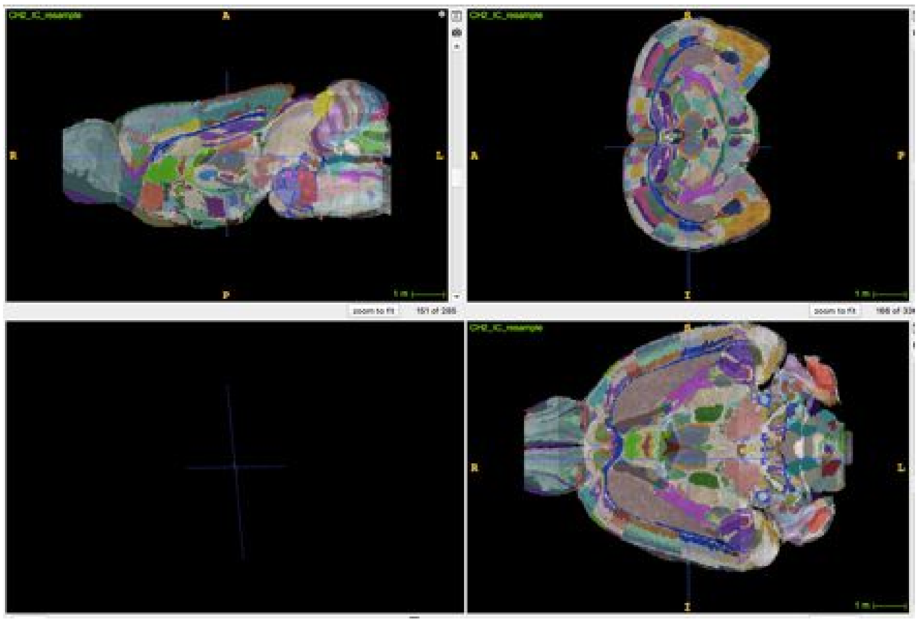


Fig. Deformable transformation between the subject image and the atlas.

Data conversion for atlas viewer

The size of processed data can be terabytes, therefore cannot be loaded into the memory for visualization in the Atlas Viewer. To this end, 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.

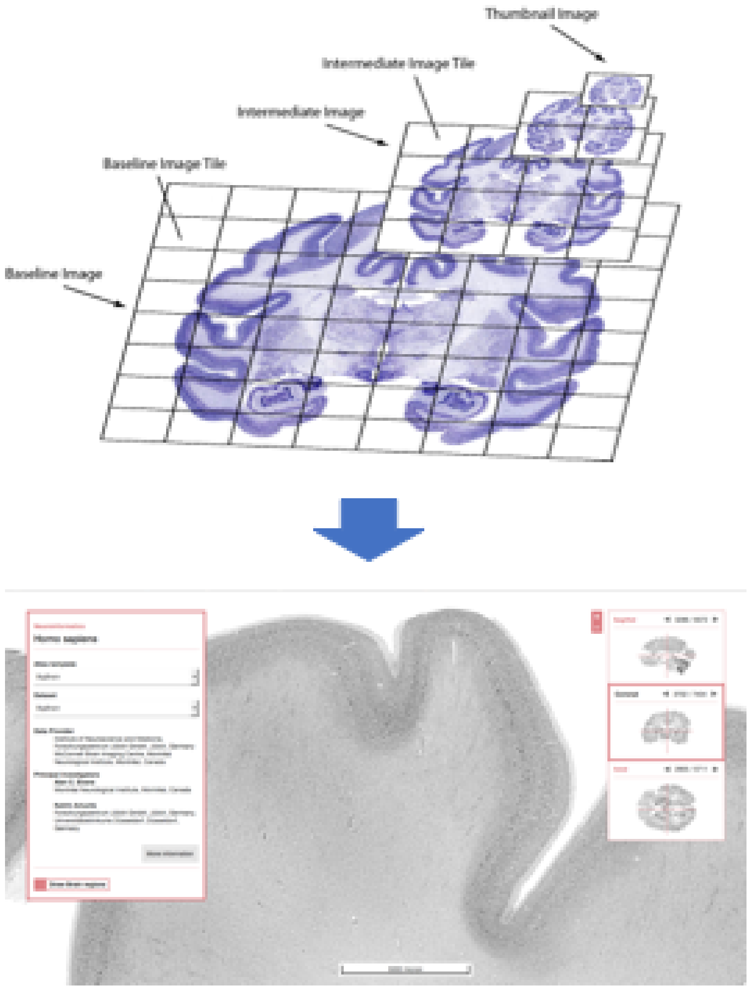


Fig. Images are converted into BBIC format and visualized in the Atlas viewer

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.

Example of SVG structure (below two polygons are part of brain regions on one slice):

```
<svg width="1320" height="800" xmlns="http://www.w3.org/2000/svg">
<g transform="scale(1.0)">
<g id="56" order="1" parent_id="" transform="scale(1.0, 1.0)"><path id="1"
order="0" structure_id="735" acronym="AUDp1" name="Primary auditory area,
layer 1" description="" d="M778.5,323.5 L778.5,324.5 L777.5,324.5
L776.5,324.5 L776.5,325.5 L775.5,325.5 L774.5,325.5 L773.5,325.5
L773.5,326.5 L772.5,326.5 L771.5,326.5 L770.5,326.5 L770.5,327.5
L769.5,327.5 L768.5,327.5 L767.5,327.5 L767.5,328.5 L768.5,328.5
L769.5,328.5 L770.5,328.5 L771.5,328.5 L771.5,327.5 L772.5,327.5
L773.5,327.5 L774.5,327.5 L774.5,326.5 L775.5,326.5 L776.5,326.5
L777.5,326.5 L778.5,326.5 L779.5,326.5 L780.5,326.5 L780.5,325.5
L781.5,325.5 L782.5,325.5 L783.5,325.5 L784.5,325.5 L785.5,325.5
L785.5,324.5 L786.5,324.5 L787.5,324.5 L788.5,324.5 L788.5,323.5
L787.5,323.5 L786.5,323.5 L785.5,323.5 L784.5,323.5 L783.5,323.5
L782.5,323.5 L781.5,323.5 L780.5,323.5 L779.5,323.5 L778.5,323.5Z"
style="stroke:none;fill:#019399"/><path id="15" order="14"
structure_id="97" acronym="TEa1" name="Temporal association areas, layer 1"
description="" d="M803.5,415.5 L803.5,416.5 L804.5,416.5 L804.5,415.5
L803.5,415.5Z" style="stroke:none;fill:#15B0B3"/></g></g></svg>
```

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 providing 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 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": "" }
           ]
        }
  }
```

Fig. 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’s 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 regularized the conflicting voxels as well as the gap voxels in the space.

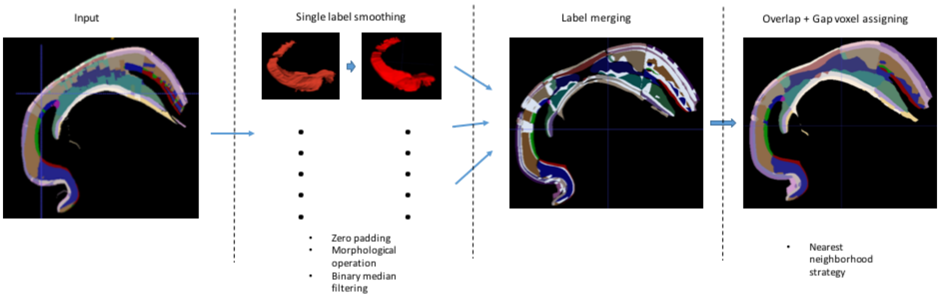

Input
Single label smoothing
Label merging
Overlap + Gap voxel assigning
• Zero padding
• Morphological operation
• Binary median filtering
• Nearest neighborhood strategy

Fig. Schema of multiple brain region smoothing

Glossary

Voxel Brain annotation layer
In the context of the Voxel Brain, an annotation layer is the dataset we attach to the voxels of a data volume.
Here are a few examples:

a. A brain region annotation layer would store an ontology term describing a given brain region or brain parcel per voxel in the atlas space.
b. a cell density Annotation layer would store a float value to every voxel of a given atlas space.
c. A gene expression annotation layer would store in each voxel of the atlas space a list of gene names with associated numerical expression level.

Atlas space
Atlas space refers to three-dimensional coordinate system of a given brain atlas.

Brain atlas
A brain atlas is a spatial representation of the brain that comprises ontology, parcellation, coordinate system, and multiple features/layers of the cerebral characteristics.

Brain parcel
A brain parcel is a specific brain region that is defined deterministically or probabilistically in the context of a given brain atlas.

Contributor
Individual or institution that produced the Data Set. The data will be accessible to the whole community. Therefore, experimental data are not permitted. Moreover, no personal information related to patients or de-anonymized data are accepted.

Data Registration

A unique set of information related to one specific dataset. It includes all piece of information required by the API or service registration pages as well as all piece of information about the Contributor registration, the contributors of the dataset and data user conditions.

Dataset

Digital data, either raw or derived, and its metadata provided through the data access portal interface, or through other media, including data and metadata derived from HBP monitoring protocols, field observations, collections, laboratory analysis, camera trap images, all written, recorded, graphic, audio, visual, and other materials in any media, whether or not subject to copyright protection, or the post-processing of existing data and identified by a unique identifier issued by the HBP.
All datasets provided by contributors should have been produced following EC ethical regulation.

Dataset Contact

Party designated in the accompanying metadata of the dataset as the primary contact for the dataset.

Data User

Individual to whom access to this dataset may be granted, subject to acceptance of these Terms and Conditions by such individual and his or her immediate collaboration sphere, defined here as the institutions, partners, students and staff with whom such individual collaborates and to whom access must be granted in order to fulfill the such individual's intended use of the dataset.

Human Brain Project

Human Brain Project (HBP) is a European Commission Future and Emerging Technologies Flagship that aims to achieve a multi-level, integrated understanding of brain structure and function through the development and use of information and communication technologies (ICT).

HBP-PROV

HBP-PROV 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.

KnowledgeGraph

KnowledgeGraph is a metadatabase built on a provenance data model HBP-PROV. In other words, it is a provenance graph to which data are registered for discovery, search, access and tracking. Currently, the KnowledgeGraph consists of Search API which is public and Indexer API which is private.

KnowledgeSpace

KnowledgeSpace (KS) is a community encyclopaedia that links brain research concepts with data, models and literature from around the world. It is an open project and welcomes participation and contributions from members of the global research community.

Ksearch

Ksearch is the search component of the Neuroinformatics Platform. It is a REST API allowing searching curated datasets using different filters that are mainly taken from MINDS.

MeSH

MeSH is the National Library of Medicine's controlled vocabulary thesaurus used for indexing articles for PubMed. It consists of sets of terms naming descriptors in a hierarchical structure that permits searching at various levels of specificity.

MINDS

MINDS stands for Minimum Information for Neuroscience DataSets. Data shared on the Neuroinformatic Platform are enriched with minimal metadata to provide essential information in order to ensure traceability of any data artefact.

Neuroinformatics Platform

Neuroinformatics Platform (NIP) is Sub-Project 5 of the Human Brain Project. The Neuroinformatics Platform and the Brain Atlases allow neuroscientists to collaboratively curate, analyse, share, and publish heterogeneous neuroscience data.

Parcellation

Brain parcellations define spatial or volumetric boundaries of brain region/structure. They could be represented in 2D and 3D depending on the use case.

Registration

For 2D or 3D volumetric datasets, we use different registration methods to convert one dataset from its own space to another space. Following registration methods are used: Linear registration registration is used to capture the global transformation between the subject image and the atlas image.

Landmark registration module is used in the 3D Slicer to pick corresponding landmark in both subject image and atlas image.

Deformable registration is a process consisting of establishing functional or spatial correspondences between two images.

NOTE: There are “Data registration” and “Registration App”, which are separate terms.

Voxel

The etymology of the work comes from a *blend* of ‘*volumetric’* and ‘*pixel’.* A voxel is the three-dimensional analogue of a pixel in a two-dimensional space.

Voxel Brain

Voxel brain is a REST service provides access to volumetric brain atlas and their annotation layers in the form of voxels.