A Visual Web Query System for NeuronBank Ontology

Weiling Li, Rajshekhar Sunderraman, and Paul Katz
Georgia State University, Atlanta, GA
Outline

• Introduction
• Query sub-system overview
• Comparison with other visual query systems
• Conclusion
Understanding the brain requires understanding its circuitry
Problem: We are using publications as a method to catalog neurons and neural circuits

- Information is distributed and fragmented.
- No means to efficiently search this knowledge.
- No means to publish incremental knowledge without a functional story.
Our Approach

• Traditional databases are not a good fit for the problem of storing information about neural circuitry.
• Changes in representation would cause the database schema to change.
• *Ontology*: A formal representation of a set of concepts within a domain and the relationships between those concepts.
• We created a ontology for each species built upon the premise that some concepts are common across species.
NeuronBank.org: A Neuromics Tool

• NeuronBank is to neurons.
  – A place to publish knowledge about neurons and neural connectivity
  – A tool to search, analyze, and share knowledge of neurons and neural circuitry.
  – An ontology-based knowledge base system
Welcome to NeuronBank™ - Cataloging the "Neurome"

This is the homepage for the NeuronBank project, an effort to develop an online reference source and informatics tool for exploring our vast knowledge of identified neurons and the circuits they form. NeuronBank will organize information about identified neuron location, physiology, morphology, and connectivity, and help researchers to browse through known circuits, conduct comparative work between species, and classify new cell types. You can learn more about the rationale and design of the project [here](#).

**To get started with NeuronBank** - Go to [Branch Listing](#) and choose a branch to explore or go to [Using NeuronBank](#) to get specific instructions or visit the [Help](#), where you can also see a list of [known bugs](#)? NeuronBank also has its own [Wiki](#) for extended information about any entry. Registered users can contribute to this wiki.

The NeuronBank project is being developed as a collaborative effort across the [Biology](#) and [Computer Science](#) Departments at [Georgia State University](#). The project team is led by [Paul S. Katz](#), [Sushil Prasad](#), [Raj Sunderraman](#), and [Ying Zhu](#).

The project is funded by grant from the [Human Brain Project](#) (NIH, NIMH R21 MH076753). The project was initiated with a seed grant from the [GSU Brains and Behavior program](#). Additional funding was provided by [NSF](#), [NIH/NINDS](#), and the [Center for Behavioral Neuroscience](#).

Join the list serve for users and developers to get updates about NeuronBank and to participate in discussions about its development. [http://mailbox.gsu.edu/mailman/listinfo/neuronbank](http://mailbox.gsu.edu/mailman/listinfo/neuronbank)

**Note** - this application is compatible with [Firefox](#) and [Internet Explorer](#). There are problems with Safari.
NeuronBank.org: A Neuromics Tool

Branch Listing

<table>
<thead>
<tr>
<th>Branch</th>
<th>Prefix</th>
<th>Neurons</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tritonia</td>
<td>Tri</td>
<td>43</td>
<td>Guest</td>
</tr>
<tr>
<td>SandBox</td>
<td>Sbx</td>
<td>26</td>
<td>Guest</td>
</tr>
<tr>
<td>Melibe</td>
<td>Mel</td>
<td>2</td>
<td>Guest</td>
</tr>
</tbody>
</table>

Instructions

- Click on the name of a branch to begin searching that branch.
- Each branch represents knowledge about neurons and synaptic connections of a species.
- Cross branch searches can be initiated from within any branch.
- The Prefix column is the three letters that precede all accession IDs for that branch.
- The number of neurons currently represented in each branch is shown.
- The last column shows your permissions for each branch. If you are not logged in, then your permissions are set to Guest.
This is the Home page for the Tritonia branch of NeuronBank. It contains information about the nervous system of Tritonia dioecaea (Order: Nudibranchia, Subclass: Opisthobranchia, Class: Gastropoda, Phylum: Mollusca).

For more information about Tritonia, go to http://www.scholarpedia.org/article/Tritonia

The NeuronBank Wiki contains additional information about Tritonia neurons and connections including definitions for the ontology.

A simple URL for this page is http://www.neuronbank.org/branch/Tritonia

This knowledgebase is still under development. The project is funded by grant from the Human Brain Project (NIH, NIMH R21 MH076753). Previous support was from: NIH/NINDS, NSF, the GSU Brains and Behavior program, the Center for Behavioral Neuroscience, and a GSU RPE award. Join the Discussion Group for users to get updates about NeuronBank and to participate in discussions about its development.
NeuronBank.org: A Neuromics Tool
NeuronBank.org: A Neuromics Tool
NeuronBank.org: A Neuromics Tool
DSI

(Redirected from Tri0001043)

DSI is a Neuron in *Tritonia.*

**Contents** [hide]

1 Basic information
2 Identification
3 Homology
4 References

## Basic information

- **NeuronBank AccessionID** Tri0001043
- **Names and Aliases** Dorsal Swim Interneuron, DSI, DSI-A, DSI-B,C, or just swim interneuron.
- **Species:** *Tritonia diomedea*
- **Neurotransmitter:** Serotonin

There are three DSIs in each hemisphere of the cerebropleural ganglion. All three are serotonin-immunoreactive. They have axons that project contralaterally to the pedal ganglion. The DSIs play an important role in the swim central pattern generator. They have both synaptic and neuromodulatory actions on other neurons in the CPG. The DSIs also accelerate ciliary crawling by synapsing onto efferent neurons in the contralateral pedal ganglion.

The DSIs are members of the swim central pattern generator. They fire rhythmic bursts of action potentials during the swim motor pattern.

## Identification

**Anatomy:**

- Serotonergic Neuron in CeSP cluster with contralateral projecting axon.
- Generally the most lateral three neurons in the cluster are DSIs.
- In the dissection microscope, with some reflected illumination, the DSIs appear somewhat translucent with a small pigment spot.
- Cell fills show the DSIs exiting Pedal Nerve 5, which is the smaller of the two Pedal-Pedal connectives. But, electrophysiological data suggests that at least one DSI is present in Pedal Nerve 6.

Serotonin immunohistochemistry shows the locations of the DSIs (circled).

A drawing of a DSI with dendritic fields in the cerebral ganglia, projecting to the contralateral Pedal ganglion. Its axon projects out one of the Pedal connectives. Based on Getting et al.(1980)
Homology

The DSIs are homologous to the CeSP-A neurons in other Nudibranchs. They are also homologous to neurons in a similar position across all gastropods. Here are the names of identified homologues in other species:

- Pleurobranchaea californica: As1-3
- Hermissenda crassicornis: CPT

References

NeuronBank.org: A Neuromics Tool
NeuronBank.org: A Neuromics Tool
Architecture of NeuronBank

BranchKB
- Extensible Data Backend
- Webservices
  - Ontology Broadcast
- Algemon QL
- Data I/O
- Branch Admin

NB-Central
- Universal Authentication
- Species Server Directory
- Cross-species search
- Ontology Database
- User profiles / searches
- Central Admin

Clients
- Submit Authentication
- Browse to Branch
- Algemon QL Builder
- Custom control
  - QL build
- Data visualization
- Data entry

Web Client

Plug-in Clients

LocationViz

NetworkViz
Query component

- ontology-based Web query interface.
- Algernon system on frame-based knowledge bases.
- JavaServer Faces (JSF) technology.
- The form-based query is translated into a textual Algernon query.
Query system architecture

Client

Query Generation
User Interface

Request Response

Ontology Schema
Retriever

Sending form-based
Algernon query
expression

Text Algernon
Query Generator

Sending text-based
Algernon query
expression

Algernon Engine

Query Result Display
User Interface

Summary Page

Link to

Detail Page

Server

Retrieve
Return

Ontology
back-ends

Query
Return results
Query generation user interface

- Class list
  - Activate
  - Property list
    - A list of properties with primitive data types
  - Class list
    - A list of properties whose data types are class or a set of classes in the ontology

Query Criteria Panel

Construct form-based Algernon query expression
• Class lists

(a) The start dropdown menu
(b) Relationship Properties of Selected Class in next Column
• Property list boxes

Select attributes ...
:CREATOR
:MODIFIER
:MODIFICATION-TIMESTAMP
:CREATION-TIMESTAMP
-AccessionID
:NAME
-Alias
-Neuron_Part
-V_Rest
-Cell_Count
-Soma_diameter

Primitive Properties of Selected Class - in Property List-Box

• Query Criteria Panel
  – a form-based interface
  – construct Algernon query expressions
An Example Form-based Search

Find all neurons which are involved in chemical synapses satisfying the following two properties:
1. the connection probability of the synapse is greater than 2, and
2. the synapse has an article annotation which was published after year 2000.
Query results

- Cross Branch Query Results

<table>
<thead>
<tr>
<th>Branch</th>
<th>Query Status</th>
<th>No. of Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tritonia</td>
<td>Query Complete</td>
<td>3</td>
</tr>
<tr>
<td>SandBox</td>
<td>No results/Branch Unavailable</td>
<td>-</td>
</tr>
<tr>
<td>Melibe</td>
<td>No results/Branch Unavailable</td>
<td>-</td>
</tr>
</tbody>
</table>

- Summary Page

Query results (Contd)

- **Detail Page**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATOR</td>
<td>pxatz</td>
</tr>
<tr>
<td>MODIFIER</td>
<td>pxatz</td>
</tr>
<tr>
<td>MODIFICATION-TIMESTAMP</td>
<td>Jun 18, 2007</td>
</tr>
<tr>
<td>CREATION-TIMESTAMP</td>
<td>Mar 30, 2007</td>
</tr>
<tr>
<td>AccessionID</td>
<td>Trn0002367</td>
</tr>
<tr>
<td>NAME</td>
<td>S-Cell</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>-Cell_Count</td>
<td>70±45±10 [Annotations: 1]</td>
</tr>
<tr>
<td>-Soma_Laterality</td>
<td>Bilateral</td>
</tr>
<tr>
<td>-Activity_Reactivity</td>
<td>Silent</td>
</tr>
<tr>
<td>-Neuron_Type</td>
<td>Sensory_Neuron</td>
</tr>
<tr>
<td>-V_Rest</td>
<td>mv</td>
</tr>
<tr>
<td>-Molecule</td>
<td>Glutamate [Annotations: 2]</td>
</tr>
</tbody>
</table>

Location:

- Undo hide
- Show hidden neuron
- Help

Display All
- Hidden Neurons
- Display Grid
- Display All Neurons

Map: CNS-Dorsal

CNS-Dorsal

Neurons
Connections

Node-Link Tree Layout
- Help

Return URL:
http://test.neuronbank.org/Trn0002367

Wiki URL:
Algernon query generation

- create a query for a neuron

( (:INSTANCE -Neuron ?Col0_Returns) )
• Choose a chemical synapse (whose parent is Inputs), which has relationship with the neuron.

( (:INSTANCE -Neuron ?Col0_Returns) (-Inputs ?Col0_Returns ?Col1_Returns) (:INSTANCE –Chemical_Synapse ?Col1_Returns) )
• choose the -Connection Probability property of the Chemical Synapse, whose parent class is - My Properties.
• click the “Add” button
• set the value of that chosen property is larger than 2 in the query criterion.
• The updated Algernon query is:

```
(:INSTANCE -Neuron ?Col0_Returns)

(-Inputs ?Col0_Returns ?Col1_Returns)
(:INSTANCE -Chemical_Synapse ?Col1_Returns)

(-My_Properties ?Col1_Returns ?Col1_Cond.Prop6)
(:CHILD -Connection_Probability?Col1_Cond.Prop6 ?Col1_Cond6)
(-Value ?Col1_Cond6 ?Col1_Cond6.Values)
(:test (:lisp (> ?Col1_Cond6.Values 2)))
```

choose the article sub-class from the third pulldown menu of classes which is a relationship property of the chemical synapse. The parent class for the article sub-class is My_Annotations.

(:INSTANCE -Neuron ?Col0_Returns)
(-Inputs ?Col0_Returns ?Col1_Returns)
(:INSTANCE -Chemical_Synapse ?Col1_Returns)
(-My_Properties ?Col1_Returns ?Col1_Cond_Prop6)
(:CHILD -Connection_Probability?Col1_Cond_Prop6 ?Col1_Cond6)
(-Value ?Col1_Cond6 ?Col1_Cond6_Values)
(:test (:lisp (> ?Col1_Cond6_Values 2)))

(:My_Annotations ?Col1_Returns ?Col2_Returns)
(:INSTANCE -Article ?Col2_Returns)
<table>
<thead>
<tr>
<th>Query Criteria:</th>
<th>Class Name</th>
<th>Attribute</th>
<th>Operator</th>
<th>Value</th>
<th>Definitional</th>
<th>Remove</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-Inputs.-Chemical_Synapse</td>
<td>-Connection_Probability</td>
<td>&gt;</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-My_Annotations.-Article</td>
<td>-Year</td>
<td>&gt;</td>
<td>2000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- choose the primitive property -Year of the article class.
- Upon clicking the “Add” button, the system introduces a second row in the query criteria panel.
- enter the value of the year property as larger than 2000.
- The final Algernon query is generated:

\[
\begin{align*}
\text{(:INSTANCE } & -\text{Neuron } \?\text{Col0}_{\text{Returns}}) \\
\text{(-Inputs } & \?\text{Col0}_{\text{Returns}} \?\text{Col1}_{\text{Returns}}) \\
\text{(} & \text{:INSTANCE } -\text{Chemical_Synapse } \?\text{Col1}_{\text{Returns}}) \\
\text{(-My\_Properties } & \?\text{Col1}_{\text{Returns}} \?\text{Col1}_{\text{Cond\_Prop6}}) \\
\text{(:CHILD } -\text{Connection\_Probability}\?	ext{Col1}_{\text{Cond\_Prop6}} \?\text{Col1}_{\text{Cond6}}) \\
\text{(-Value } & \?\text{Col1}_{\text{Cond6}} \?\text{Col1}_{\text{Cond6\_Values}}) \\
\text{(} & \text{:test (:lisp (> ?Col1\_Cond6\_Values 2)))} \\
\text{(-My\_Annotations } & \?\text{Col1}_{\text{Returns}} \?\text{Col2}_{\text{Returns}}) \\
\text{(} & \text{:INSTANCE } -\text{Article } ?\text{Col2}_{\text{Returns}}) \\
\text{(-Year } & ?\text{Col2}_{\text{Returns}} \?\text{Col2}_{\text{Cond\_Prop12}}) \\
\text{(} & \text{:test (:lisp (> ?Col2\_Cond\_Prop12 2000)))}
\end{align*}
\]
Comparison

- Web-based.
- Retrieving ontology schema on demand and facilitating to construct a query expression with the minimal database knowledge.
- Returning not only the final results, but also all intermediate results.
Conclusion and future work

• Web query sub-system of NeuronBank.
  – Primitive properties of classes can be queried by the users as well as relationships with other classes.
  – The user can follow a chain of relationships to formulate complex queries.

• Future work:
  – query arbitrary Ontologies that are stored in Protege Frames.
  – modified to work with RDF/OWL Ontologies as well. SPARQL queries will have to be generated in this case.
Thank You for your time and attention

Questions?