NeuPrint Manual

An introduction to NeuPrint and the Cypher Query Language

Table of Contents

Introduction to Neo4j and NeuPrint

Neo4j

Neo4j is a graph database that Neuprint is built on. Stores massive amounts of data as linked lists and index tables. One can traverse relationships between elements in the data model by following links (memory pointers)

Nodes are entities in the graph
can hold properties
can be tagged with labels

Relationships are connections between nodes
have a type, direction, start and end node
can hold properties

Properties are key-value pairs stored on nodes or relationships

NeuPrint

NeuPrint - A set of tools for loading and analyzing connectome data into a Neo4j database

NeuPrint Explorer - A single page web application that provides simple interfaces to query an EM connectome stored in NeuPrint

NeuPrint Python - Python client utilities for interacting with the NeuPrint connectome analysis service. For information about installing and running neuprint-python see 4.1 neurpint-python Manual

NeuPrint API - REST interface for neuPrint. For information see 5.1 neuPrint API Manual

NeuPrint Datasets

Hemibrain - Dataset containing nearly half of a female adult Drosophila brain. Imaged to include central complex and right hemisphere mushroom body. This dataset continues to develop and be edited

Hemibrain:v1.0 - Dataset containing nearly half of a female adult Drosophila brain. Imaged to include central complex and right hemisphere mushroom body. This dataset has been frozen and reflects the data cited in the neuprint paper (PAPER)

Additional datasets may be added in the future

Important Properties

Body IDs

A **body** is generated by automatically predicting cell membranes in electron microscopy images of brain tissue. Bodies are manual proofread to correct errors made by the automatic segmentation algorithm and then assigned a **status** depending on their size/completeness. Statuses can be assigned automatically or manually by proofreaders and biologists.

Status	Description
Traced	A body more complete than 'Roughly traced' (usually traced by a lab) and validated by a biological expert (Shin-ya, Kazunori)
Orphan	A body that can't be traced and does not exit the volume
Assign	Small body that is within the set required for a 0.5 connectome - Has approximately ≥ 2 T-bars or ≥ 10 PSDs (starting bodies for OL and focused workflow)
Unimportant	A body irrelevant to reconstructing neurons and the connectome such as glial profiles and out-of-bounds bodies

Neuron Types and Instances

The **type** property contains the general neuron type. Neuron **instance** contains names more specific to the individual neuron or neuron subtype.

Nomenclature for neuron type and instances can be found in x section of the paper.
e.g. The format abc_X indicates neuron of morphology abc of connectivity subtype X.

Regions of Interest (ROI)

Overview. An ROI is a volumetric area of the dataset often marked by the boundaries of a neuropil, synaptically dense brain region. The hemibrain dataset also contains ROIs marked by fiber bundles, a collection of neuronal projections with a low density of synapses. The hemibrain dataset contains 109 manually annotated ROIs. For more information about creating ROIs, see section x of the paper *Paper Title*.

Organization. ROIs are organized into supercategories and may contain subregions. For hierarchical organization and abbreviations see the ROI explorer.

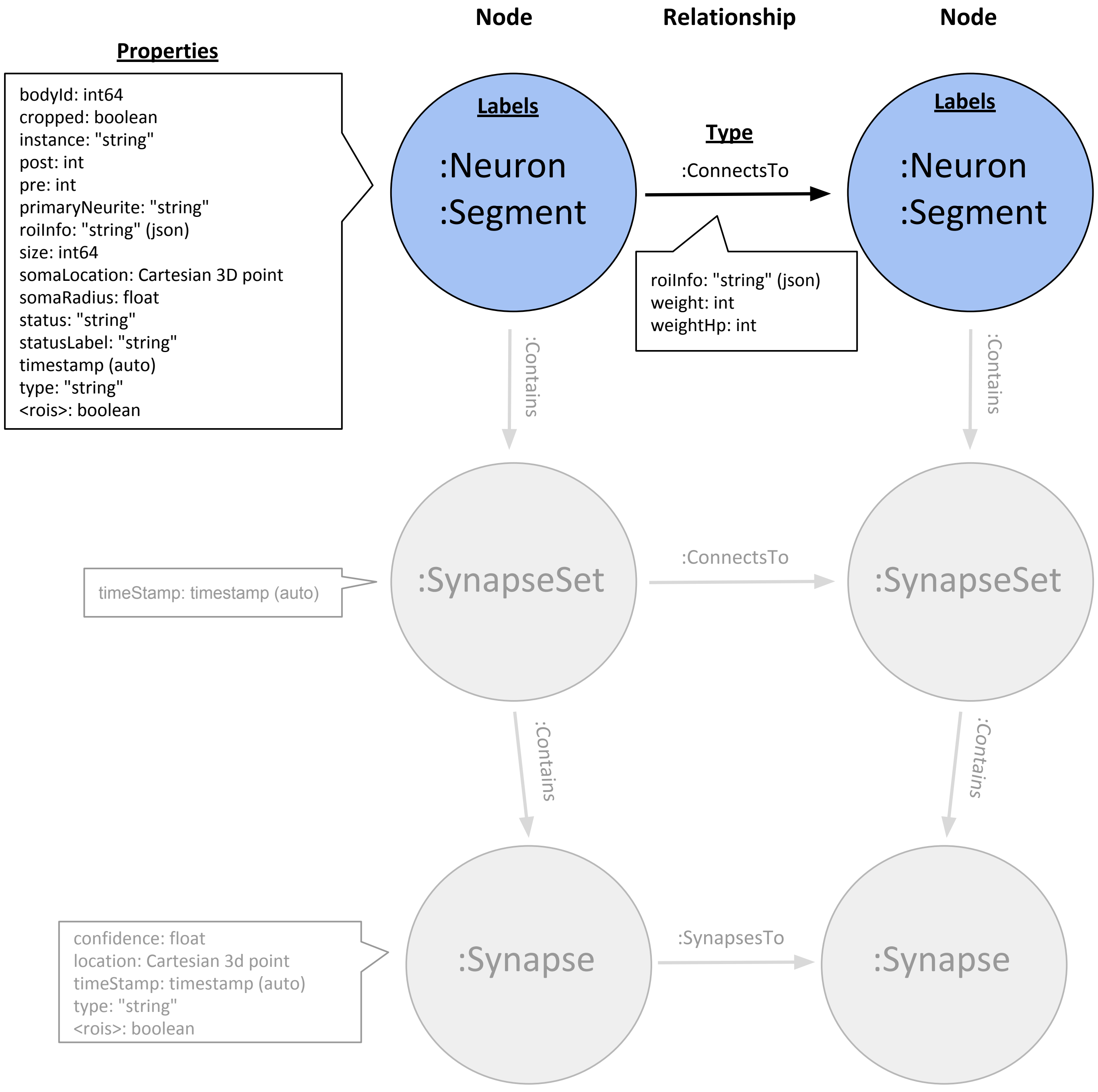
Nomenclature. The nomenclature of ROIs in the hemibrain follow *"A Systematic Nomenclature for the Insect Brain"*. For more information, see the paper

NeuPrint Data Model

Neuron nodes **connect to** other Neuron nodes

Neuron nodes hold neuron properties (body Id, instance, status, etc.)

ConnectsTo relationships hold connection properties (weight, ROI info)

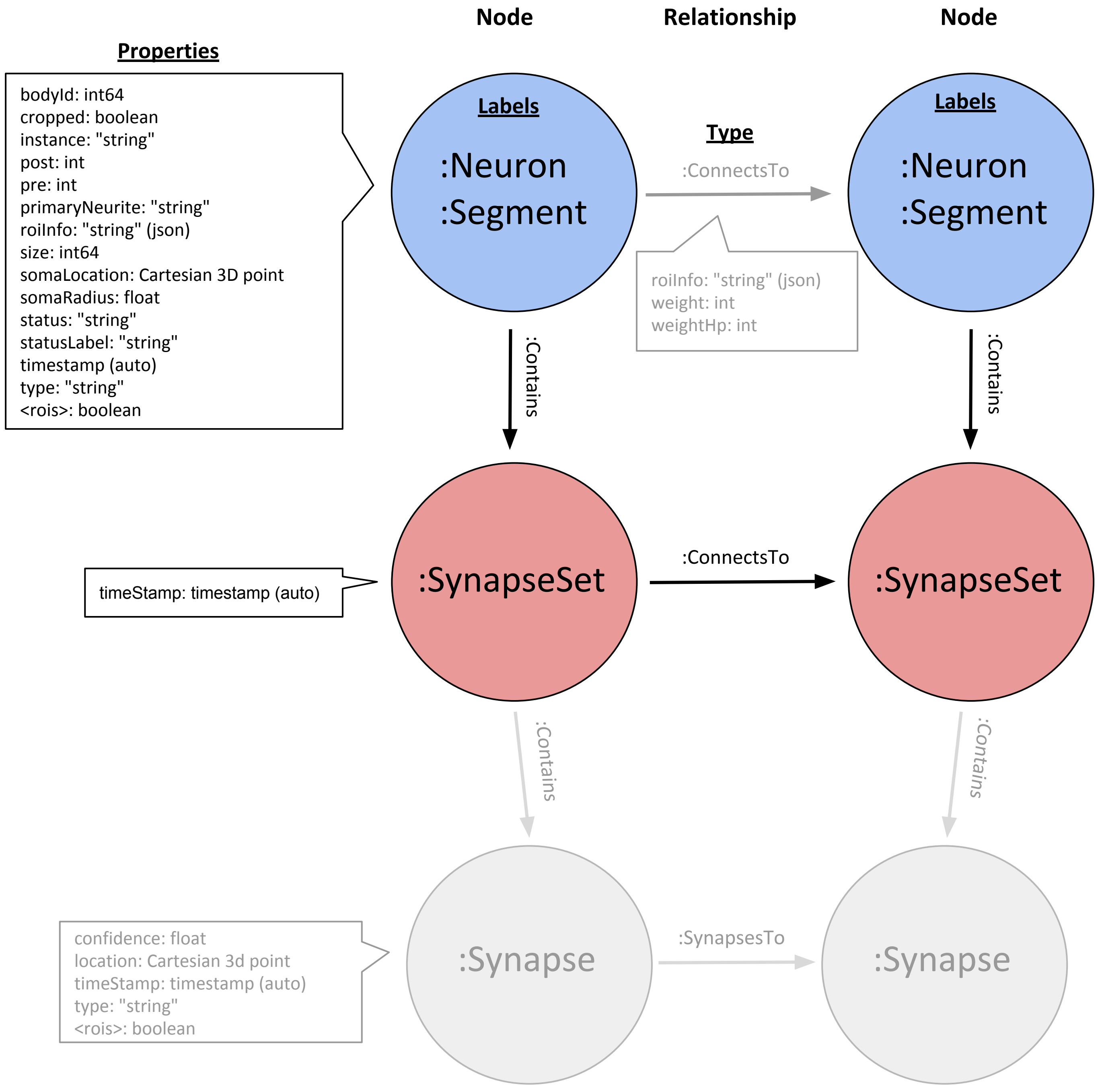


NeuPrint Data Model

Neuron nodes **contain synapse set** nodes

Synapse set nodes **connect to Synapse Set** nodes from other **Neuron** nodes

Synapse Set nodes hold **Synapse Set** properties (size, ROI info)

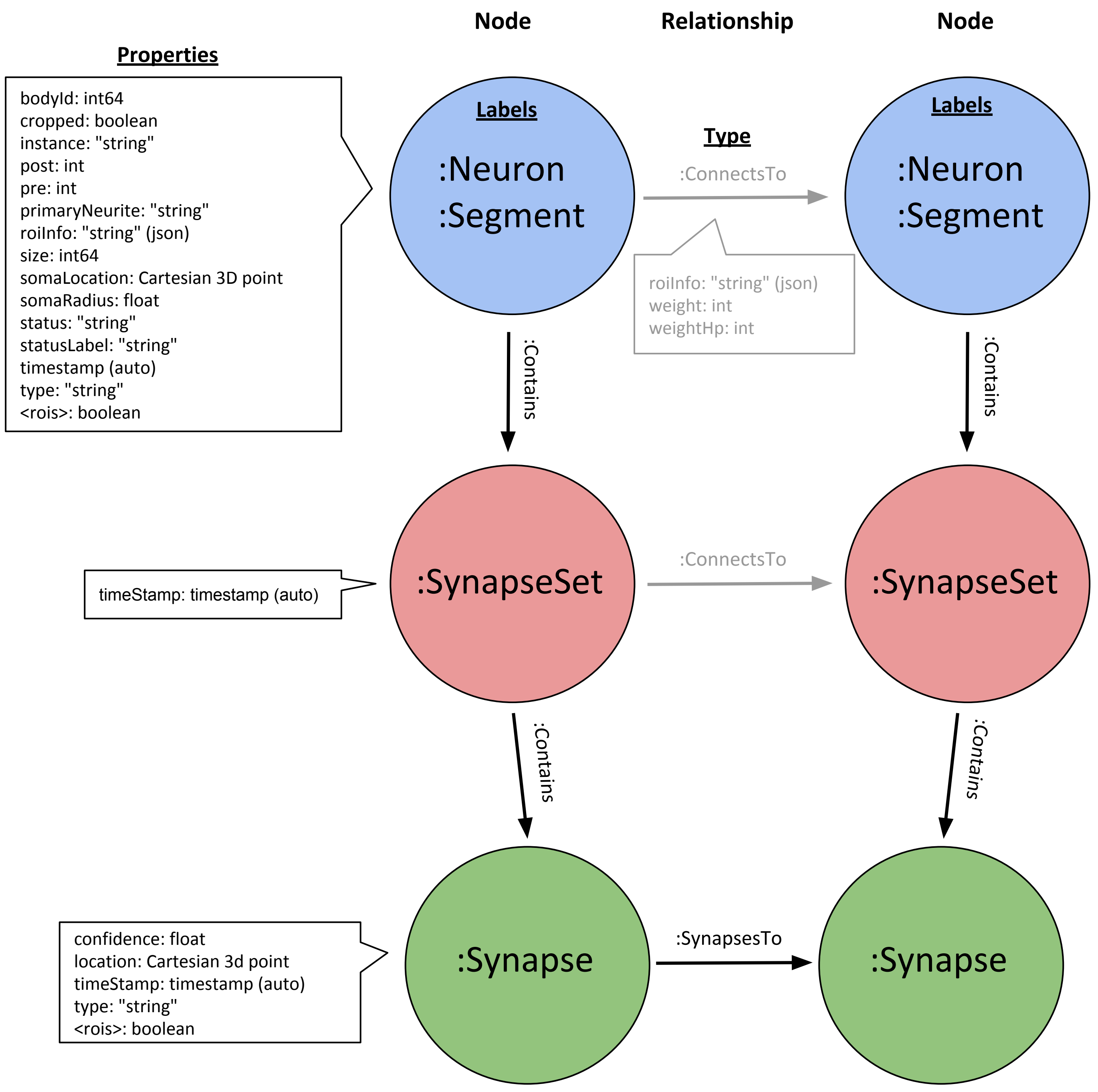


NeuPrint Data Model

Synapse set nodes **contain Synapse** nodes

Synapse nodes **connect to Synapse** nodes from other **Synapse Set** nodes

Synapse nodes hold **Synapse** properties (type, location, etc.)



Cypher Query Language

Cypher is Neo4j's query language. Queries are composed of multiple clauses. The following are the most common clauses:

```
MATCH - specifies the pattern Neo4j will search for in the graph
WHERE - adds content to the query and filters the search results
RETURN - defines what information to include in the results
```

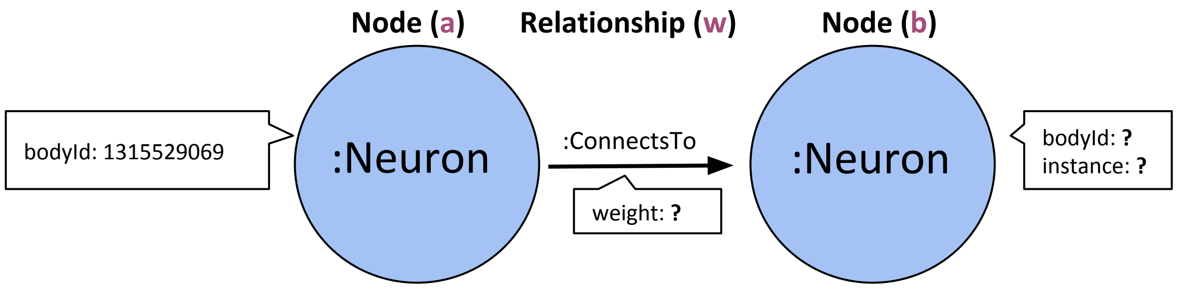
EXAMPLE

Return body ID, instance, and number of connections between input neuron 1315529069 and output neuron(s)

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE a.bodyId = 5813020698
RETURN b.bodyId, b.instance, w.weight
```

Use (:`dataset-Neuron`) when using neuPrint Python API i.e. (:`hemibrain-Neuron`)

NeuPrint Data Model



MATCH clause - *Input hemibrain neuron (a) connects to output hemibrain neuron (b)*

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
```

a, b, and w are variables assigned to nodes and relationships. Use variable.property in the WHERE or RETURN clause to reference a node or relationship property

(:`hemibrain-Neuron`) specifies hemibrain dataset label and neuron node type

-[:ConnectsTo]-> specifies relationship type and direction

WHERE clause - *The body ID (bodyId) of input neuron (a) is 1315529069*

```
WHERE a.bodyId = 1315529069
```

RETURN clause - *Return body ID (bodyId) and instance (instance) of output neuron (b). Return number of connections (w.weight) from input neuron (a) to output neuron (b)*

```
RETURN b.bodyId, b.instance, w.weight
```

Results Table

b.bodyId	b.instance	w.weight
294800293	FB07a(SFS1)_R	69
885262311	FB07c(SFS1)_R	64
850764542	Delta6e_04	55
⋮	⋮	⋮

NeuPrint Cypher - MATCH

The **MATCH** clause specifies the pattern Neo4j will search for in the graph. The MATCH clause syntax closely follows the graph data model. To write efficient queries, search the graph for as few nodes as possible

MATCH CLAUSE EXAMPLES - Search graph for:

Neuron nodes

```
MATCH (:Neuron)
```

(:`Label`) specifies a node in the graph tagged with a label

Synapse nodes contained in a hemibrain neuron node

```
MATCH(:Neuron)-[:Contains]->(:`SynapseSet`)-[:Contains]->(:Synapse)
```

-[:Type]-> specifies a relationship in the graph with a type and direction

Neuron nodes that connect to neuron nodes

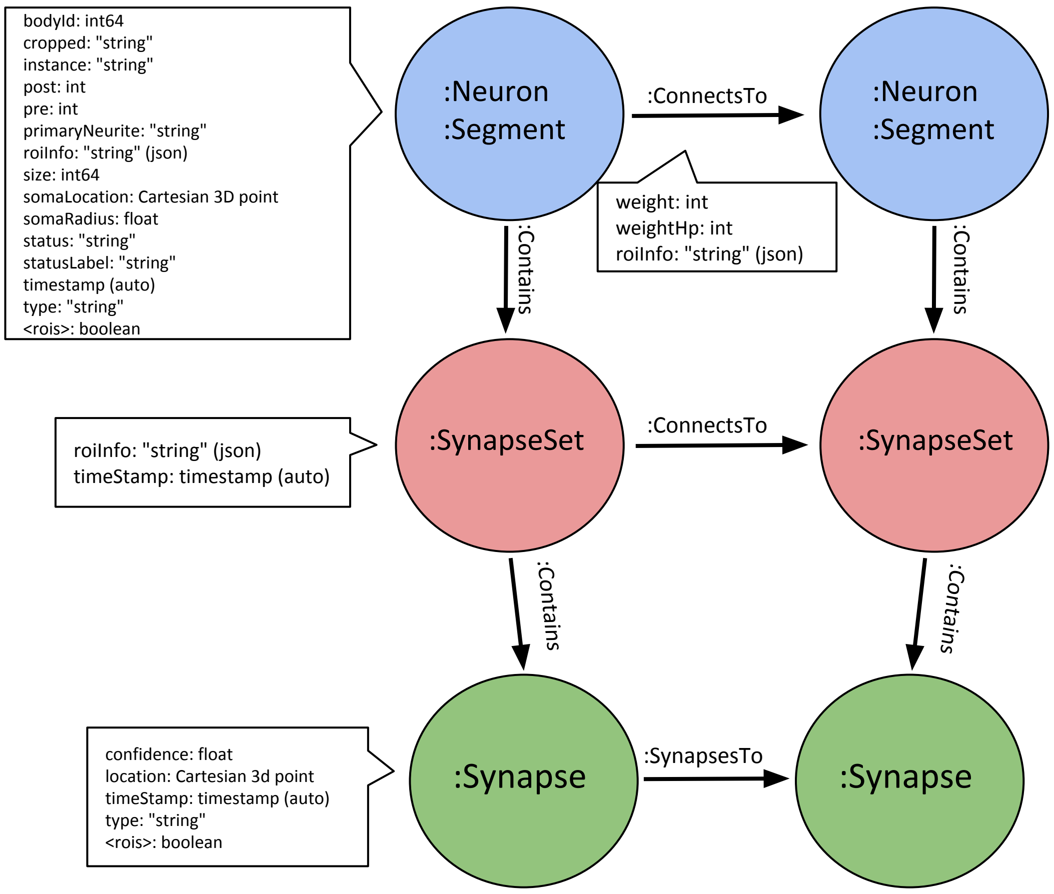
```
MATCH (:Neuron)-[:ConnectsTo]->(:Neuron)
```

Neuron nodes that connect to neuron nodes via synapse set nodes

```
MATCH (:Neuron)-[:Contains]->(:`SynapseSet`)-[:ConnectsTo]->
(:`SynapseSet`)<-[:Contains]-(:Neuron)
```

Neuron nodes that connect to neuron nodes via synapse nodes

```
MATCH (:Neuron)-[:Contains]->(:`SynapseSet`)-[:Contains]->(:Synapse)
-[:SynapsesTo]->(:Synapse)<-[:Contains]-(:`SynapseSet`)<-[:Contains]-(:Neuron)
```



NeuPrint Cypher - WHERE

The **WHERE** clause uses properties of nodes and relationships in the MATCH clause to filters the search results. Note: The WHERE clause is optional

WHERE CLAUSE EXAMPLES - Filter on:

Neuron node property (bodyId)

```
MATCH (a:Neuron)
WHERE a.bodyId = 707854989
```

```
MATCH (a:Neuron)
WHERE a.bodyId IN [707854989, 707863263, 707858790]
```

```
WITH [707854989, 707863263, 707858790, 1011797706, 917647959] AS bodyID_list
UNWIND bodyID_list AS bodyID
MATCH (a:Neuron)
WHERE a.bodyId = bodyID
```

Use IN operator to specify multiple values for a given property
Use WITH and UNWIND when

Neuron node property (instance)

```
MATCH (a:Neuron)
WHERE a.instance =~ 'MBON.*'
```

Use regular expressions to filter by strings.
See 'Regex and Special Characters' on page 13 for information about regular expressions

Neuron node property (bodyId, status) and ConnectsTo relationship property (weight)

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE a.bodyId = 707863263 AND w.weight > 5 AND NOT b.status = "Traced"
```

Use boolean operators (AND, OR, XOR, NOT) in the WHERE clauses to add filters
Use inequality operators (<, <=, >, >=) to check if value is inside certain range

ConnectsTo relationship property (roiInfo)

```
MATCH (:Neuron)-[w:ConnectsTo]->(:Neuron)
WHERE apoc.convert.fromJsonMap(w.roiInfo)["SMP(R)"] > 5
```

Use apoc.convert.fromJsonMap to access values within the roiInfo string. Can also use apoc.convert.fromJsonMap in the RETURN clause

```
MATCH (:Neuron)-[w:ConnectsTo]->(:Neuron)
WHERE apoc.convert.fromJsonMap(w.roiInfo)["SMP(R)"] IS NOT NULL
```

Use IS NULL or IS NOT NULL to check if property exist. Null is used to represent missing or undefined values

Synapse node property (<roi>)

```
MATCH (:Neuron)-[:Contains]->(:`SynapseSet`)-[:Contains]->(s:Synapse)
WHERE s.`CA(R)` OR s.`a'L(R)`
```

The <ROI> property on neuron and synapse nodes are boolean values. If the neuron node contains a synapse in an ROI or a synapse node is located in an ROI, the ROI exists as a property on the node. To check if the property exists, use property.ROI. Use back quotes (`) around the ROI.

NeuPrint Cypher - RETURN

The **RETURN** clause defines what information to include in the results

RETURN CLAUSE EXAMPLES - Return:

Neuron node properties (bodyId, instance, status)

```
MATCH (a:Neuron)
WHERE a.instance =~ "MBON.*"
RETURN a.bodyId, a.instance, a.status
```

Use comma to separate return items. Each item will be a column in the results table

Neuron node properties (bodyId, instance) and ConnectsTo relationship property (weight)

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE a.instance =~ 'MBON.*' AND w.weight > 10
RETURN a.bodyId AS InputID, w.weight AS weight, b.bodyId AS OutputID, b.instance AS OutputInstance
```

Use AS <new name> to rename a column in the results table

Neuron node properties (bodyId, instance) and calculated value (pre + post)

```
MATCH (a:Neuron)
WHERE a.instance =~ 'MBON.*'
RETURN a.bodyId, a.instance, (a.pre + a.post) AS total_synapses
```

Use mathematical operators (+, -, *, /, %, ^) to calculate value in results table

Number of neuron nodes

```
MATCH (a:Neuron)
WHERE a.instance =~ 'MBON.*'
RETURN count(a)
```

Use aggregating function (count(), sum(), avg(), min(), max()...) to aggregate data

Neuron node property (type), number of neuron nodes (count(type)), total number of connections (sum(weight))

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE b.bodyId = 5813020698 AND NOT a.type IS NULL
RETURN a.type, count(a.type), sum(w.weight) AS W
```

Synapse node property (location)

```
MATCH (a:Neuron)-[:Contains]->(:SynapseSet)-[:Contains]->(:Synapse)
-[:`SynapsesTo`]->(s:Synapse)<-[:Contains]-(:SynapseSet)<-[:Contains]-(b:Neuron)
WHERE a.bodyId = 5813020698 and b.bodyId = 294800293
RETURN s.type, s.location.x, s.location.y, s.location.z
```

Unique neuron node property (bodyId, instance) and total number of connections (sum(weight))

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE a.instance =~ '.*ring.*'
RETURN DISTINCT b.bodyId, b.instance, sum(w.weight)
```

Use DISTINCT operator to return unique values

Cypher - Additional Features

Additional Cypher Clauses. For more information see Cypher Manual: Clauses

ORDER BY - follows RETURN or WITH to sort output in ascending (**ASC**) or descending (**DESC**) order
WITH - allows manipulation of output before being passed to following parts of the query
UNWIND - transforms list into individual rows
SKIP - defines from which row to start including the rows in the output
LIMIT - limits the number of results returned
UNION - combines results of 2 or more queries and removes duplicates.
UNION ALL - same as UNION, but does not remove duplicates from the result set
OPTIONAL MATCH - specifies patterns to search for in the database while using nulls for missing parts of the pattern

EXAMPLES

ORDER BY and LIMIT. Returns top 20 outputs of given neuron. Results ordered by weight

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE a.bodyId = 949710555
RETURN b.bodyId, w.weight AS W
ORDER BY W DESC
LIMIT 20
```

UNION. Return presynaptic and postsynaptic connections to a neuron in a single results table

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE a.bodyId = 949710555 AND w.weight >= 5
RETURN a.bodyId AS InputID, w.weight AS W, b.bodyId AS OutputID
UNION
MATCH (a:Neuron)<-[w:ConnectsTo]-(b:Neuron)
WHERE a.bodyId = 949710555 AND w.weight >= 5
RETURN a.bodyId AS InputID, w.weight AS W, b.bodyId AS OutputID
```

Note: The number and names of columns must be identical in queries combined with UNION.
If a query times out, splitting it up into multiple queries and uniting them with UNION may make it run.

OPTIONAL MATCH. Returns outputs to a given neuron. If connection does not exist, returns null

```
MATCH (a:Neuron)
WHERE a.bodyId IN [1348653495, 1251032454, 5812990157]
OPTIONAL MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
WHERE b.bodyId = 1313038188 AND w.weight > 4
RETURN a.instance, w.weight, b.bodyId
```

Note: When using OPTIONAL MATCH, query must also include a MATCH clause

Multiple MATCH clauses. *Finds neurons both upstream of 1570922129 and downstream of 5813027266*

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)
MATCH (b:Neuron)-[w2:ConnectsTo]->(c:Neuron)
WHERE a.bodyId = 981194127 AND c.bodyId = 5813057404
RETURN b.instance, b.bodyId, w.weight, w2.weight, sum(w.weight + w2.weight) AS sumweight
ORDER BY sumweight DESC
```

Use multiple MATCH clauses or seperate MATCH clauses with a comma

Cypher - Additional Features

Cypher Syntax. There are often multiple ways to write the same query. For more information see Cypher Manual: Syntax

EXAMPLES

Search for hemibrain neuron node with body ID 5901213440

```
MATCH (a:Neuron)
WHERE a.bodyId = 5901213440
  OR
MATCH (a:Neuron{bodyId:5901213440})
  OR
MATCH (a:Neuron)
WHERE a.bodyId IN [5901213440]
  OR
UNWIND [5901213440] AS ID
MATCH (a:Neuron)
WHERE a.bodyId = ID
```

Filter by neurons with Delta6 in the instance

```
WHERE a.instance =~ '.*Delta6.*'
  OR
WHERE a.instance CONTAINS "Delta6"
```

Search for hemibrain neurons in the LO

```
MATCH (a:Neuron{`LO(R)`:true})
  OR
MATCH (a:Neuron)
WHERE a.`LO(R)`
```

Regex and Special Characters. A regular expression is a string of characters defining a search pattern. Specific characters are reserved for special use. To use these reserved characters as part of the search pattern, escape them.

EXAMPLES

Writing regular expressions

```
WHERE a.instance =~ 'KC.*'
```

Indicates 'KC' is at the beginning of the instance

```
WHERE a.instance =~ '.*KC.*'
```

Indicates 'KC' is at the beginning, middle, or end of the instance

```
WHERE a.instance =~ '.*KC'
```

Indicates 'KC' is at the end of the instance

Escape special characters ()

```
MATCH (a:Neuron)
WHERE a.instance =~ "ExR3\\(ring\\).*"
RETURN a.bodyId, a.instance
```

Use double backslashes (\\) to escape parentheses in a regular expression.

Cypher - Additional Features

Neuron Node Labels (Segment vs Neuron). All neuron nodes have a **Segment** label. However, only neuron nodes with 2 or more presynaptic sites or 10 or more postsynaptic sites have a **Neuron** label. Therefore, use the **Segment** label in the MATCH clause to query ALL neuron nodes

EXAMPLE

Search for all hemibrain neuron nodes with 1 presynaptic site in CA and under 300000 voxels

```
MATCH (a:Segment)
WHERE a.`CA(R)` AND apoc.convert.fromJsonMap(a.roiInfo)["CA(R)"].pre = 1 AND a.size < 30000
RETURN a.bodyId, a.size
```

Pathway Query. Use multiple ConnectsTo relationships in the MATCH clause to query pathways. Note: Result tables can get large quick. Use filters on nodes and connections

EXAMPLE

Find one hop pathways from MBONs to FB07 neurons

```
MATCH (a:Neuron)-[w:ConnectsTo]->(b:Neuron)-[w2:ConnectsTo]->
(c:Neuron)
WHERE a.instance =~ 'MBON.*' AND c.instance =~ 'FB07.*' AND w.weight >= 5 AND w2.weight >= 5
RETURN a.bodyId, a.instance, w.weight, b.bodyId, b.instance, w2.weight, c.bodyId, c.instance
ORDER BY w.weight DESC
```

META node. Each dataset stored in NeuPrint contains a META node. The META nodes holds dataset properties

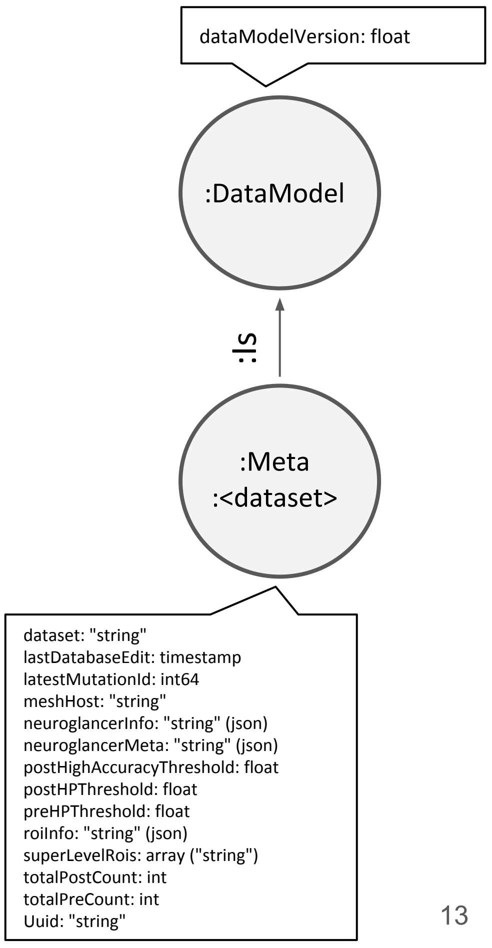
EXAMPLES

Total number of presynaptic and postsynaptic sites in the Hemibrain dataset

```
MATCH (m:Meta)
WHERE m.dataset = "hemibrain"
RETURN m.totalPreCount, m.totalPostCount
```

Threshold for high confidence (weightHP) synapses in the Hemibrain dataset

```
MATCH (m:Meta)
RETURN m.postHPThreshold, m.preHPThreshold
```



```
dataset: "string"
lastDatabaseEdit: timestamp
latestMutationId: int64
meshHost: "string"
neuroglancerInfo: "string" (json)
neuroglancerMeta: "string" (json)
postHighAccuracyThreshold: float
postHPThreshold: float
preHPThreshold: float
roiInfo: "string" (json)
superLevelRois: array ("string")
totalPostCount: int
totalPreCount: int
Uuid: "string"
```

Property Use Index - Neuron

bodyId: unique numerical identifier

```
WITH
  -   [123,456,789] AS TARGETS
MATCH
  -   (x:Neuron{bodyId:123})
WHERE
  -   x.bodyId = "123"
  -   x.bodyId IN [123,456,789]
  -
RETURN
  -   x.bodyId
```

cropped: boolean indicating if a significant portion of the body leaves the imaged area

```
MATCH
  -   (x:Neuron{cropped:True})
WHERE
  -   x.cropped = True
  -   x.cropped IS NULL
RETURN
  -   N/A (will return blank or NULL)
```

instance: individual neuron name

```
WHERE
  -   x.instance = "KCy-d"
  -   x.instance =~ '.*Delta6d.*'
  -   x.instance STARTS WITH "KC"
  -   x.instance CONTAINS "KC"
RETURN
  -   x.instance
```

post: number of postsynaptic sites

```
MATCH
  -   (x:Neuron{post:50})
WHERE
  -   x.post >= 50
RETURN
  -   x.post
  -   (x.post + x.pre)
```

pre: number of presynaptic sites

```
MATCH
  -   (x:Neuron{pre:50})
WHERE
  -   x.pre > 50
RETURN
  -   x.pre
  -   (x.pre + x.post)
```

size: volume of body in voxels

```
WHERE
  -   x.size > 30000
RETURN
  -   x.size
```

roiInfo: the ROIs this neuron is in

```
WHERE
  -   (apoc.convert.fromJsonMap(x.roiInfo)["C
      A(R)"]) IS NULL
  -   NOT
      (apoc.convert.fromJsonMap(x.roiInfo)["S
      LP(R)"]) IS NULL
RETURN
  -   (apoc.convert.fromJsonMap(x.roiInfo)["a'
      L"])
  -   (apoc.convert.fromJsonMap(x.roiInfo)["C
      A"]).post
  -   x.roiInfo
```

somaLocation: cartesian location of soma

```
WHERE
  -   x.location.x > 2000+500
  -   x.location.y <= 2000
  -   x.location.z >= 2000
RETURN
  -   x.location
  -   x.location.x, x.location.y, x.location.z
  -   [x.location.x, x.location.y, x.location.z]
```

somaRadius: radius of soma

```
WHERE
  -   x.somaRadius < 20000
RETURN
  -   x.somaRadius
```

status: tracing completion level of a neuron

```
WHERE
  -   x.status = "Traced"
  -   x.status CONTAINS "Orphan"
  -   x.status IS NULL
  -   NOT x.status IS NULL
RETURN
  -   x.status
```

type: neuron cell type

```
WHERE
  -   x.type = "KCg"
  -   x.type CONTAINS "KC"
RETURN
  -   x.type
```

<roi>: boolean for each ROI, true if neuron is in the ROI, false if not

```
MATCH
  -   (x:Neuron{`ROI`:true})
WHERE
  -   x.ROI
  -   x.ROI IS NULL
RETURN
  -   N/A (will return blank or NULL)
```

Property Use Index - ConnectsTo and Synapse

ConnectsTo

weight: number of connections

```
WHERE
    -    w.weight > 50
RETURN
    -    w.weight
    -    SUM(w.weight)
```

weightHP: high confidence number of connections

```
WHERE
    -    w.weightHP > 50
RETURN
    -    w.weightHP
    -    SUM(w.weightHP)
```

roiInfo: the ROIs connections are in

```
WHERE
    -    apoc.convert.fromJsonMap(n.roiInfo)["ROI"]
         .postHP IS NULL
    -    NOT
         apoc.convert.fromJsonMap(n.roiInfo)["ROI"]
         IS NULL
    -    apoc.convert.fromJsonMap(n.roiInfo)["ROI"]
         .pre IS NULL
RETURN
    -    apoc.convert.fromJsonMap(n.roiInfo)["ROI"]
         .postHP
    -    apoc.convert.fromJsonMap(n.roiInfo)["ROI"]
         .pre
```

Synapse

type: pre or postsynaptic

```
WHERE
    -    s.type = "pre"
    -    s.type = "post"
RETURN
    -    s.type
```

confidence: certainty that a predicted synapse is in fact a synapse

```
WHERE
    -    s.confidence > .5
RETURN
    -    S.confidence
```

location: cartesian location of a synapse

```
WHERE
    -    s.location.x > 5000,
         s.location.y < 5000,
         s.location.z => 5000
RETURN
    -    s.location
    -    s.location.x, s.location.y, s.location.z
    -    [s.location.x, s.location.y, s.location.z]
```

<roi>: boolean for each ROI, true if synapse is in the ROI, false if not.

```
WHERE
    -    s.ROI
RETURN
    -    N/A (will return blank or NULL)
```

NeuPrint Terms

Neo4j and Cypher

Neo4j - A graph database. For more information see https://neo4j.com/
Cypher - Neo4j's query language. For more information see https://neo4j.com/docs/cypher-manual/
Node - An entity in a graph that can hold properties and can be tagged with labels
Relationship - A connection between nodes that have a type, direction, start and end node, and can hold properties
Property - A key-value pair stored on a node or relationship

NeuPrint Terms

NeuPrint - A set of tools for loading and analyzing connectome data into a Neo4j database
NeuPrint Explorer - Single page web application that provides simple interfaces to query an EM connectome stored in NeuPrint
NeuPrintHTTP - A connectomics REST interface that leverages the NeuPrint data model
NeuPrint Python API - Python client utilities for interacting with the NeuPrint connectome analysis service. For more information see https://github.com/connectome-neuprint/neuprint-python

NeuPrint Datasets

Hemibrain - Dataset containing nearly half of a female adult Drosophila brain. Imaged to include central complex and right hemisphere mushroom body
Fib25 - Medulla 7 column dataset
From "Synaptic circuits and their variations within different columns in the visual system of Drosophila" (Takemura, et al. 2015)
Mb6 - Mushroom body dataset
From "A connectome of a learning and memory center in the adult Drosophila brain" (Takemura, et al. 2017)

Dataset Terms

Segmentation - Partitioning electron microscopy images of brain tissue into segments (bodies) by automatically predicting cell membranes
Body - A segment generated by automatically predicticting cell membranes in electron microscopy images of brain tissue. NeuPrint includes bodies with a soma or at least 1 synapse
Synapse - Structure that permits the passage of a signal from one neuron to neuron to one or more neurons. Can refer to presynaptic (t-bar) or postsynaptic site (PSD) on a body
T-bar - Presynaptic protein where vesicles bind and neurotransmitter is released into the synaptic cleft
PSD (postsynaptic density) - Postsynaptic receptor proteins that signify uptake of neurotransmitter
ROI (Region of interest) - Commonly referred to as brain region. Named for the most part using the Insect Brain Name Working Group nomenclature
Soma - Cell body
Interconnectivity - How a group interacts within itself, usually body ID or ROI interconnectivity
Interneuron - A neuron that serves as a connection bridge of two target neurons / ROIs

Reconstruction Statuses

Traced - A body more complete than 'Roughly traced' (usually traced by a lab) and validated by a biological expert
Orphan - Body that can't be traced and does not exit the volume
Orphan Artifact -Body that can't be traced due to imaging artifact and does not exit the volume
Orphan Hotknife -Body that can't be traced through a hotknife and does not exit the volume

NeuPrint Data Model Terms

Nodes

<dataset> - Node label on all nodes in NeuPrint. This label allows multiple datasets to be stored on the database, but partitioned when querying is targeted for one dataset. Add <dataset>_ (i.e. hemibrain_Neuron) before other node label. See 'NeuPrint Datasets' for datasets stored in NeuPrint

Neuron

Labels

- Segment - Body with <2 t-bars, < 10 PSD's
- Neuron - Body with >=2 t-bars, >=10 PSD's, an instance, a type, a status, or a soma

Properties

- bodyId - A randomly generated number that is unique to each body in the dataset. No overlap across datasets.
- cropped- boolean indicating if a body leaves the dataset x
- instance - a name assigned by biologist to indicate instance of cell type
- post - the number of postsynaptic sites (PSD) on a body
- pre - the number of presynaptic sites (t-bar) on a body
- primaryNeurite - ?x
- roiInfo - a string in json format ({"roiA":{"pre":1,"post":2},...}) containing the number of presynaptic and postsynaptic sites in each ROI
- size - the size of a body in voxels
- somaLocation - x, y, z coordinates in center of cell body
- somaRadius - radius of cell body in pixels
- status - a string used to designate completion level of a body (see 'Dataset Terms: Reconstruction Statuses')
- statusLabel - a string indicating the status of the body in a previous version of the data
- timestamp- ?x
- type - a name assigned by biologist to indicate cell type.
- <rois> - boolean indicating body is located in a particular ROI

SynapseSet

Labels

- SynapseSet - Contains all synapses between neuron nodes

Properties

- timestamp- ?

Synapse x

Labels

- Synapse - Structure that permits the passage of a signal from one neuron to neuron to one or more neurons. Can refer to presynaptic (t-bar) or postsynaptic site (PSD) on a body

Properties

- confidence - The certainty that an annotated synapse is correct and valid. Determined by algorithm
- location - x, y, z coordinates of a synaptic site
- type - a string that indicate synapse type (pre or post)
- <rois> - Boolean indicating synapse is located in a particular ROI (if present, always true)
- timestamp- ?x

Relationships

ConnectsTo - type of relationship between Neuron nodes or SynapseSet nodes

Properties

- roiInfo - a string in json format ({"roiA":{"pre":1,"post":2},...}) containing the number of connections between neuron nodes in each ROI
- weight - the number of connections between neuron nodes
- weightHP - the number of high confidence connections between neuron nodes. Confidence threshold can be checked on Meta node for each dataset

Contains - type of relationship between Neuron node and SynapseSet node or SynapseSet node and Synapse node

Biology Terms x

Neuron Anatomy

Dendrite - A branched extension of a nerve cell, along which impulses from other cells are received.
Axon - A branched extension of a nerve cell along which impulses are conducted to other cells.
Arbor - A branch of a neuron.
Neurite / neuronal fiber -
Neuronal projection -
Synapse - Structure that permits one neuron to pass an electrical or chemical signal to one or more neurons. Can send electrical or chemical signals. (*dataset here records only chemical synapses*) Chemical Synapses contain a presynaptic site, a postsynaptic site, a synaptic cleft between the two, that helps to release the signal, a synaptic cleft that is a gap between the two neurons that is the site of electrical or chemical transmission, and a postsynaptic protein that receives the signal.
Presynaptic neuron - The neuron that sends the signal in a given synapse
Postsynaptic neuron - The neuron that receives the signal in a given synapse
Receptor - A membrane protein on the postsynaptic neuron in a chemical synapse that binds to the neurotransmitter and creates an effect in the postsynaptic neurons
Vesicle - A phospholipid circular container in a presynaptic neuron that contains neurotransmitter
T-bar - Presynaptic membrane protein in drosophila that allows vesicles to release their neurotransmitter into the synaptic cleft
PSD - Stands for the postsynaptic density. Area on the membrane of the postsynaptic neuron that is dense with proteins. The EM image of a PSD shows a dense, dark membrane.
Synaptic cleft - The space between neurons that synapse with each other through which electrical or Neurotransmitter - A molecule that is the physical medium of the signal sent out by the presynaptic neuron in a chemical synapse
Electrical synapse - *This type of synapse is not recorded in these datasets.* A mechanical and electrically conductive link between two neighboring neurons that is formed at ta narrow gap between pre and postsynaptic neurons that is known as the gap junction. Here there is a connection between intercellular fluid of two neurons through hydrophilic channels that can open and close.
Dense core vesicle - A special type of vesicle that contains neurotransmitter. Does not need a t-bar to release the neurotransmitter to neighboring cells (unsure of this definition). Contain neuropeptides, dopamine and neuromodulators such as dopamine and serotonin. (uncertain about a lot of this)
Claw - Claw link branching structures with many PSD's that wrap around the bouton of another neuron
Bouton - A ballooning structure filled with vesicles and t-bars on a neuron. Sends information to many neurons, some of which form claws around the bouton.
Cell body fibers -
Membrane - The thin layer of tissue that acts as the boundary and lining of a cell. In neurons, this membrane is is a bilayer of lipid molecules with many proteins embedded inside.

Axes and planes - Dorsal, ventral, anterior, posterior, rostral, caudal. (add image)
Contralateral - A brain region that crosses from one hemisphere to the other.
Ipsilateral - A brain region that stays in one hemisphere and does not cross into the other.
medial, mediolateral, and lateral
Medial - Referring to the area of the brain that is near the center of the two hemispheres of the brain.

Projection neuron - Neurons whose axons extend from one area of the brain to another.
Local neuron/interneuron - Neurons that do not connect far regions of the brain. Interneuron - a neuron that serves as a connection bridge of two target neurons / ROIs (While working in neutu/neuprint, I have been using the term local neuron to refer to neurons that stay in the same ROI or in adjacent ROIs, and interneuron as a neuron that serves as a bridge between two other neurons by connecting with a strong weight to both. These two definitions are not the same, but wikipedia lists them as synonyms)

Biology Terms x

Sections of the Brain

Connectome - A comprehensive map of all the neurons and synapses within an organism's nervous system. It may be thought of as a wiring diagram.
Neuropil - Any area in the nervous system composed of mostly axons, dendrites, and glial cell processes that forms a synaptically dense region containing a low number of cell bodies. (add examples of what is a neuropil - is an ROI a neuropil?)
ROI - Region of interest. Can represent level 1, 2, or 3 neuropils, or a fiber bundle.
Supercategories - (Level 1 neuropil) Large neuropil blocks that can be further partitioned into smaller areas. No overlap between level 1 neuropils.
Unit neuropils - (Level 2 neuropil) Subdivisions of neuropil supercategories.
Subregions - (Level 3 neuropil) Subdivisions of unit neuropils.
Ganglia - A structure containing a number of nerve cell bodies, and often forming a swelling on a nerve fiber
Glomeruli - Clusters of nerve endings
Tract - A collection of fibers that travel together from one part of the brain to another.
Fascicles - A tract in the PNS?
Fiber bundles - A collection of fibers that travel together from one part of the brain to another. (is this in the brain or in the pns? Synonym of tract?)
Commissure - A band of nerve tissue that crosses from one side of the brain to the other. Connects two regions contralaterally
Neuromere - Transient segment of the developing brain.
Cell body layer - The layer on the outside of the fly brain that contains cell bodies
Neural circuits - A population of neurons interconnected by synapses to carry out a specific function when activates.
Glia - *This type of cell is not recorded in this dataset* Non-neuronal cells in the nervous system that maintain homeostasis and provides support and protection for neurons. (something about glial boundaries separating regions of the brain?
Clonally associated groups of neurons -
Cerebral ganglia - (is this in the hemibrain?)
Gnathal ganglia - (is this in the hemibrain?)
Hemisphere - The two halves of the brain representing the right and left side. The brain is roughly symmetrical across these two hemispheres.

Drosophila melanogaster - The most commonly used species in insect neurobiology, and the species from whence all data in neuprint came.
Arthropod - An arthropod is an invertebrate animal having an exoskeleton, a segmented body, and paired jointed appendages. Arthropods form the phylum Euarthropoda, which includes insects, arachnids, myriapods, and crustaceans.

Gal4/split gal4 - (I think it would be good to have the janelia split gal4 light page linked on neuprint)
Driver -
EM - Electron Microscope or Electron Microscopy. An imaging technique for obtaining high resolution images of biological and non-biological specimens. It is used in biomedical research to investigate the detailed structure of tissues, cells, organelles and macromolecular complexes
Fib-sem - Focused Ion Beam scanning electron microscope. Technique used to image the data in hemibrain (what else?) What imaging technique was used for the other datasets?