Chapter 8

Data Models in Neuroinformatics

Elishai Ezra Tsur

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.73516

Abstract

Advancements in integrated neuroscience are often characterized with data-driven approaches for discovery; these progressions are the result of continuous efforts aimed at developing integrated frameworks for the investigation of neuronal dynamics at increasing resolution and in varying scales. Since insights from integrated neuronal models frequently rely on both experimental and computational approaches, simulations and data modeling have inimitable roles. Moreover, data sharing across the neuroscientific community has become an essential component of data-driven approaches to neuroscience as is evident from the number and scale of ongoing national and multinational projects, engaging scientists from diverse branches of knowledge. In this heterogeneous environment, the need to share neuroscientific data as well as to utilize it across different simulation environments drove the momentum for standardizing data models for neuronal morphologies, biophysical properties, and connectivity schemes. Here, I review existing data models in neuroinformatics, ranging from flat to hybrid object-hierarchical approaches, and suggest a framework with which these models can be linked to experimental data, as well as to established records from existing databases. Linking neuronal models and experimental results with data on relevant articles, genes, proteins, disease, etc., might open a new dimension for data-driven neuroscience.

Keywords: databases, hierarchy-based data models, integrated neuroscience, LEMS, layer-oriented data models, NeuroML, object-based data models

1. Introduction

Integrated neuroscience (IN) is an emerging field of research with implications that range from the derivation of neural networks motifs [1] to approaching one of the most important questions ever tackled: the nature of consciousness [2]. IN has emerged from the aspiration for insights, which could only be inferred from data obtained across multiple spatial scales

(Ångströms to centimeters) and temporal scales (milliseconds to years). An integrated approach toward neuroscience requires multiscale neural data—from molecular regulations (S1) and the dynamics of individual synapses (S2) to information processing in neural networks (S3) and to the orchestrated function of brain maps (S4) and systems (S5) (**Figure 1**).

In their seminal paper "Neuroscience on the NET" [3], Peter Fox and Jack Lancaster draw parallels between neuroinformatics and the "genome informatics community" that have gained remarkable insights leveraging the Web to generate federated frameworks for "collective wisdom." Fox and Lancaster called the "prospective developers of neuroscience databases" to "absorb the collective wisdom of these network pioneers," handle the challenge of "sematic compatibility," and develop a neuroscientific database federation to realize the field's potential of "scientific exploration." The increased attention over the past decade to data-driven neuroscience is attested by the number of published papers having these terms as keywords. Tracking the number of published papers on the subject (retrieved from PubMed) follows an exponential curve, where the "knee" of the curve is in 2010 (**Figure 2**, left). A combination of an integrated approach to neuroscience with the establishment of a federated framework for "collective wisdom" of neuroscientists and engineers can fuel the celebration of the "era of the brain."

1.1. The data tail

Neuroscientific data flow from various resources, ranging from government funded consortiums of laboratories, to individual laboratories spread worldwide.

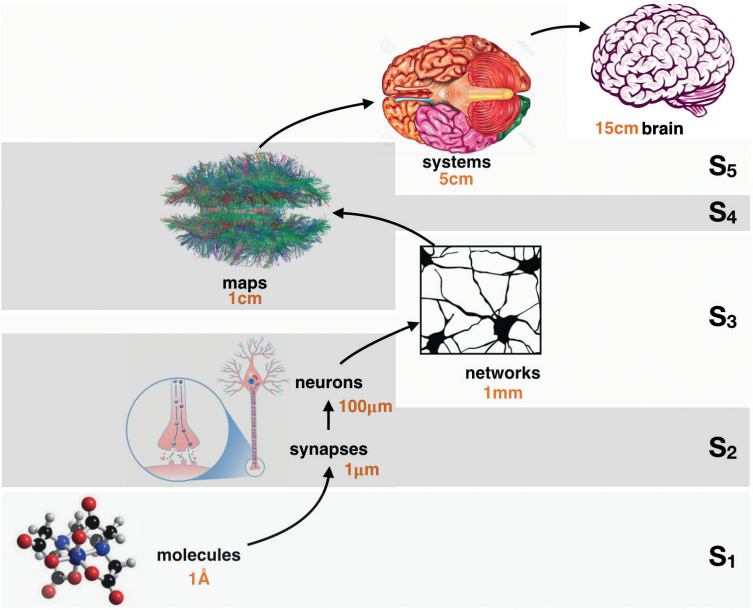

Figure 1. Schematics of the spatial scales (molecules to complete nervous systems) of integrated neuroscience.

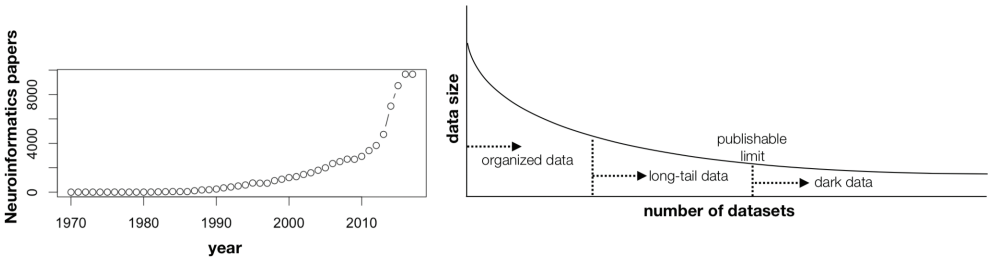

Figure 2. Number of papers linking neuroscience and data across the last 4 decades (right), the emergence of the long-tail and dark data volumes from exploring the size and number of neuroscientific data sets.

1.1.1. "Big science" initiatives

Today, one of the most ambitious endeavors aiming at integrated neuroscience is the human brain project (HBP) [4]. HBP is a multinational EU-funded research initiative, aimed at advancing multiscale brain-inspired information technology. Neuroinformatics lies at the core of HBP and orchestrated by COLLAB, a Web-based collaborative cloud-based system, developed within HBP's neuro informatics platform (NIP). COLLAB is fueling the project's other platforms (brain simulation, neurorobotics, medical informatics, and neuromorphic computing) with immense upstream and downstream data flows. It is distributed as a software as a service (SaaS) by HBP's high-performance analytics and computing platform (HPAC), enabling massive data archiving and distribution of virtual machines (VM) to collaborators, empowering them with high-end supercomputing capabilities for simulation and data analytics. COLLAB's mission is not a trivial one: it must be interfaced with heterogeneous data types and ontologies to manage metadata storage and provide a query system with which rodent and human brain atlases can be constructed and populated using different data modalities (anatomy, physiology). Moreover, COLLAB should link its data with foreign maps, databases, and atlases. HBP precedent is the human genome project (HGP) [5], a project that radically changed the ways research in molecular biology is carried out and how we perceive it. HGP has new disciplines as heirs, ranging from personalized genomic-based medicine to comparative genomics. It has established innovative approaches to biological database creation and maintenance, such as the construction of public small-molecule libraries with which biological pathways can be standardized. HBP approach aims to do the same for neuroscience.

Inspired by HGP and HBP, a new scientific endeavor termed "BRAIN" was initiated in the US by the White House, "aimed at revolutionizing our understanding of the human brain" [6] and like the other initiatives to "empower individual labs by providing…open-access databases." Another ambitious project is the NIH-funded human connectome project (HCP), which aims to characterize the human brain connectivity and functions. In this project, colossal amount of data will be gathered from many hundreds of patients with state-of-the-art 3D fMRI machines, EEG and MEG. Full-genome sequencing from all subjects will be performed as well. Behavioral measures in different domains (cognition, emotion, perception, and motor function) will also be recorded [7]. Other governmentally funded integrated neuroscience programs include the "Brain Canada" [8] and the "China Brain Project" [9]. All aforementioned acknowledge the

fact that establishing standardized data collection and processing, as well as mechanisms for data sharing and credit allocation, are fundamental to their project's success.

1.1.2. The long tail data

Enormous "big-science" initiatives such as the HBP, HGP, and the BRAIN have large coordination teams, and as mentioned above, great emphasis is given within their scope to data and copyrights. Moreover, they are usually required (by the funding agency) to share their results with the community. However, routine scientific work in individual labs or small consortiums generates the majority of scientific data. Although each lab produces relatively limited amount of data, together they constitute the bulk of neuroscientific information. These granular, individually assembled data sets (usually given as publishable units) are referred to in the literature as "long-tail data." The tail of data also includes "dark-data," which is comprised of unpublished information, sitting aimlessly in personal hard drives or in restricted shared folders (**Figure 2**, right). Within this tail of neuroscientific data lies a unique opportunity—the possibility of assembling these scattered pieces of knowledge into "deep" data collections [10]. Ferguson and colleagues reviewed "data sharing" in the long tail of neuroscience [11]. While describing the limitations of data sharing among individual labs, they demonstrated the impact such an attempt would make through the success of the IMPACT consortium [12]. IMPACT collected tailed clinical data from over 43,243 patients who have suffered from traumatic brain injury (TBI) over the span of 20 years into a "deep" database. Their data were mined to derive a prognostic model with unprecedented precision for predicting recovery, ushering a new era for TBI precision medicine [13]. IMPACT demonstrated the way "deepening" long-tail data can provide incredible insights and even revolutionize treatment. Another example is the recently established data sharing community for spinal cord injury (SCI) research [14].

1.1.3. Deepening the long tail data

The main challenges of deepening tailed neuroscientific data encompass all levels of data handling and association including acquisition, quality control, representation, system implementation, user interface and documentation, data analysis, budget and maintenance, and federation [15]. Among all these dimensions, data representation is the most extensively discussed, as it stands as a prominent bottle neck in the definition of data sharing standards. Recently, a group of thought leaders, comprised of scholars, librarians, archivists, publishers and research funders, came together to provide the research community with guidelines toward the creation of standards for data sharing, which they termed the "FAIR Data Principles" [16]. The FAIR guidelines dictate that data should be (1) findable, with a rich assigned standardized metadata and persistent identifier; (2) accessible, via an identifier and an open, free, and universally implementable communications protocol; (3) interoperable, via broadly applicable language for knowledge representation; and (4) reusable, via domain-relevant community standards. A great emphasis is therefore given in the FAIR guidelines to carefully constructed metadata.

Following the importance of data standardization in computational modeling in biology, and particularly in neuroscience, the COMBINE consortium has been initiated in 2009 [17]. COMBINE aims to coordinate and facilitate different community-based standardization efforts in the field of computational biology. COMBINE's neuro-related standardization efforts include

computational neuroscience ontology (CNO) [18], NeuroML [19], and spiking neural markup language (SpineML) [20].

One of the most prominent database federations for the neuroscientific community is the neuroscience information framework (NIF) [21], which has been cataloging and surveying the neuroscience resource landscape since 2006. NIF currently gives access to over 250 data sources categorized to different subjects ranging from software tools to funding resources. NIF provides a distributed query engine to tailed data, which is independently created and curated. This type of distributed search among independent databases is enabled through NIF's DISCO registry tool with which a Web resource can send automatic, or manual, data updates to the NIF system [22].

1.2. Models for computational neuroscience

Linking neuroscientific data with simulation environments has deep roots in the origins of neuronal modeling and databases. Starting with the seminal works of Alan Hodgkin, Andrew Huxely, and Wilifrid Rall during the 1970s, which established today's most utilized models for neuronal dynamics, the scale of simulating neural networks has picked up. As computing resources became abundant, neuronal simulations began to be carried out by an increasing number of labs, creating the need for a database in which already established models could be realized and build upon.

Models of neuronal dynamics span over all scales abstraction, where each abstraction level encapsulates an increasing amount of details (**Figure 3**) [23].

Increasing level of complexity entails increasing amount of required data. Databases for computational models are therefore well integrated with simulation platforms such as NEURON [24] and GENESIS [25].

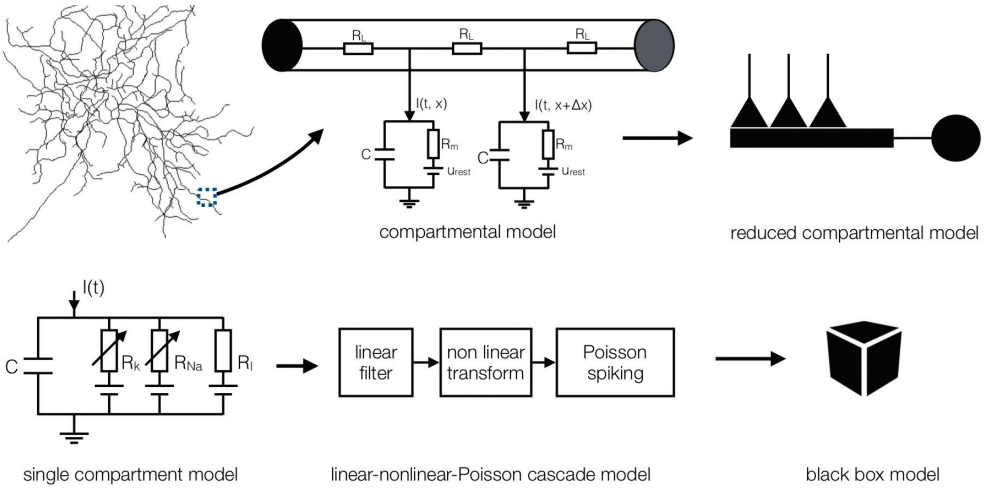

Figure 3. Models' schematics in computational neuroscience.

PyNN [26] and NeuroML are independently developed approaches to allow standardization of neuronal modeling, enabling models' utilization across simulators. While NeuroML took the declarative approach for modeling, explicitly specifying the model using in a structured format (with XML), PyNN took the procedural approach, specifying the models using functions and procedures, in this case, executing python scripts on different simulators.

Neuronal modeling usually requires morphological, connectivity, and physiological data. Neuromorph.org is the largest federated collection of 3D neuronal reconstructions and associated metadata [27]. For each neuron, a rich metadata is gathered, including miscellaneous information ranging from file format to the source specie, sex, age, weight, etc. Forward automatic analysis, ranging from size to topology, is also made for each morphology, leading to a range of morphological insights [28]. NeuroMorpho.Org is carefully curated and administrated, with a team responsible for file transfer, conversion, annotation and curation, minimizing the burden on the data submitter. A model can be submitted to NeuroMorpho.Org only when it is associated with published results—a curation decision that on one hand ensures data quality but on the other hand rejects the wealth of information residing on the dark side of the data tail. Since neuronal modeling incorporates morphological data, as well as physiological data, interoperability between the two is essential. Indeed, the NeuroMorpho.Org database can be utilized with other complementary resources such as the CellPropDB, NeuronDB [29], ModelDB [30], and MicrocircuitDB (all four are curated by SenseLab at Yale university). While CellPropDB is comprised of data regarding receptors, channels, and transmitters, NeuronDB distributes these elements across a specific neuron. ModelDB comprised of computational models of neurons derived from NeuronDB. MicrocircuitDB contains circuit modeling, which was built upon data from ModelDB. Today, all SenseLab databases are tightly coupled with Neuron [31].

2. Data models

Neuroscientific data models must encompass the different levels of neuronal scales: starting at the molecular regime, going up to the membrane and synapse levels, moving through the dendritic tree and axonal branches, and finishing at the circuit and system levels. Each level encapsulate further details. For example, at the circuit level, data on proteins and ions is 'hidden' at the encapsulated lower levels of representation. Various data models exist for each scale—here I chose a representative for each model, which in my opinion reflects its main properties. Please note that the schematics shown below for each data model, particularly for Neuron's object-based representation schemes, do not aim to accurately specify the objects hierarchy scheme in terms of inheritance or composition. They are given here to purely illustrate the general approach for modeling.

2.1. Structuring data

Following samples acquisition, data must be structurally organized. It can be structured in either a "flat file," a tabular formation, a structured file (such as XML), an object based, or a layer-oriented scheme (**Figure 4**). Data in a flat file are stored in an unstructured manner and therefore manipulating it would require reading it entirety into memory. Data can be

Figure 4. Schematics of data structuring paradigms.

structured as a table, where each value is headed with a type and usually also with a size identifier. eXtensible markup language (XML) is a different approach for data structuring, in which data is arranged in schemes, where each subsequent level increases the scope of the previous one. XML gained industry momentum due to its simplicity and flexibility, enabling declarative specifications rather than coding. This facilitates automated transformation of model specifications into multiple other formats. One of the main alternatives to data modeling is object-based representation of information, in which entities are defined with a set of properties and connected as attributes. Object-based representation allows the encapsulation of internal details of the data associated with the heterogeneity of the underlying data sources. Another approach is the layer-oriented approach (LOA), in which interlinked declarative languages (or layers) specify the model. The rationale behind the LOA is the premise that computational models are not a "flat collection of equations" but rather a hierarchical structure from which the underlying biological concept is reflected.

2.2. Models of morphological data

Before data can be modeled, it needs to be abstracted. The level of neuromorphological details with today's advanced imaging techniques, such as the two photons microscopy, is staggering. Moreover, since image stacks cannot be directly used for computational modeling due to their nontrivial interpretability and size, morphology must be reconstructed from them. Encapsulation of the details of neuromorphological data needs to consider its application, which in our case is computational modeling. Since different environments such as Neurolucida, NEURON, and GENESIS use a different representation of morphological data (**Figure 5**), a generalized representation, such as the MorphoML, is required to enable easy conversion to each format.

2.2.1. Flat structuring of morphological data

Neuromantic [32] is a semiautomatic stand-alone freeware reconstruction application, in which serial image stacks (JPEG to TIFF) are used to reconstruct dendritic trees. Reconstructions are stored in the SWC data format. SWC is one of the most widely used data models for neuromorphological data, for which a standardized version is used by Neuromorpho.org (not to be confused with Adobe file format). It is ASCII encoded text, where each line represents a single morphological sample point, which is represented by seven data items: id, structure identifier, 3D location, radius, and parent id. For example, the data entry:

```
2 1 -2 -3.33 0 7.894 1
```

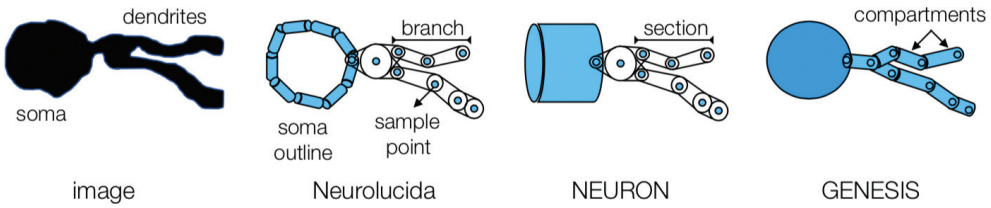

Figure 5. Representation of morphological data across different environments.

signifies a sample point with id number 2, connected to sample point number 1, identified as being located at the soma (structure identifier 1), located at (x = −2, y = −3.33, z = 0), in a compartment with a 7.894 radius. SWC files are generally small in size, trivial to read, and widely adopted across applications.

2.2.2. *Hierarchical structuring of morphological data*

Another approach for neuromorphological data modeling is using XML. One example is the MorphoML [33], which is a part of NeuroML. For example, defining soma and a dendrite can be written as:

```
<cells>
   <cell name = "Example">
     <meta:notes>A Simple cell</meta:notes>
     <segments>
       <segment id = "0" name = "Soma" cable = "0">
         <proximal x = "0.0" y = "0.0" z = "0.0" diameter = "16.0"/>
         <distal x = "0.0" y = "0.0" z = "0.0" diameter = "16.0"/>
       </segment>
       <segment id = "1" name = "Dend" parent = "0" cable = "1">
         <proximal x = "8.0" y = "0.0" z = "0.0" diameter = "5.0"/>
         <distal x = "28.0" y = "2.0" z = "0.0" diameter = "6.0"/>
     </segment>
     </segments>
     <cables>
       <cable id = "0" name = "SomaCable" />
       <cable id = "1" name = "DendriteCable" />
     </cables>
   </cell>
</cells>
```

This XML-based neuromorphological specification can be verified using a dedicated software, as well as be converted to GENESIS or NEURON readable formats. Schematics of hierarchy-based representations of neuromorphological data are illustrated in **Figure 6** (left).

2.2.3. *Object-based structuring of morphological data*

NEURON, one of the dominant players in computational neuroscience, has a dedicated file type termed "HOC." It has C-like syntax with an additional object-oriented expressability. One

of the uses for "HOC" is defining a neuronal morphology by constructing an array of "section" objects, each defined by a series of four points (using neuron's "pt3dadd" function): three coordinates and a radius. Sections can be connected to one another (using neuron's "connect" function). For example, two connected sections can be characterized by sample points: (109.72, 125.39, 19.28) and (109.93, 125.85, 19.01) with radiuses 3.96136 and 3.88, respectively, for the first section and (115.42, 125.23, 15.19) and (115.69, 125.16, 15.05) with radiuses 0.752 and 0.64, respectively, for the second section:

```
create section[703]

section[0] {
  pt3dclear()
  pt3dadd(109.721,125.39,19.2812,3.96136,0)
  pt3dadd(109.93,125.285,19.0172,3.88406,0)
}

section[1] {
  pt3dclear()
  pt3dadd(115.427,125.239,15.19,0.752,0)
  pt3dadd(115.695,125.161,15.0518,0.649936,0)
}
connect section[1](0.0), section[0](1.0)
```

A list of sections can be linked as attributes in a "cell" class, enabling treating them in a unified (abstracted) manner. Schematics of object-based representation of neuromorphological data are illustrated in **Figure 6** (right).

2.2.4. Tabular structuring of morphological data

One of the prevalent platforms for morphological reconstruction is Neurolucida (http://www.mbfbioscience.com/neurolucida), which provides different data models for representation, including ASC, DAT, and XML, for which format specification is not publicly available. However, reversed engineered specification for Neurolucida's DAT data format (available

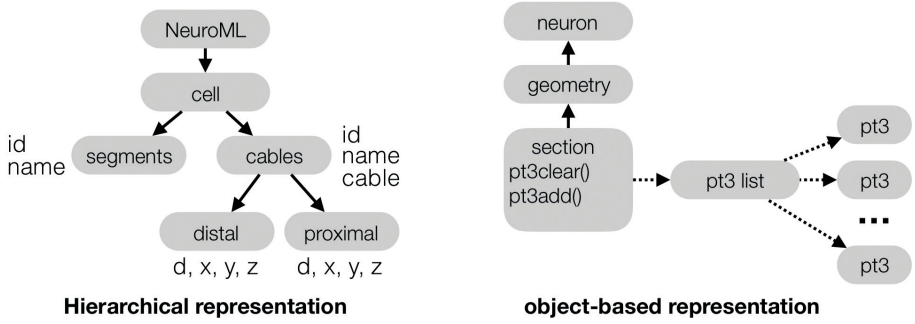

Figure 6. Hierarchical and object-based representation of neuromorphological data.

through: neuronland.org) reveals a hierarchy of data blocks, each identified by a Hexadecimal-encoded header (specifying the block type and size), followed by ASCII encoded data. For example, name and sample data are encoded using:

```
% header : size       : String
  0x0001 : 0x0000000A : 'Name'
% header : size       : x    : y     : z     : d    : section id
0x0101   : 0x00000018 : 2.15 : -3.25 : 18.55 : 0.54 : 0x0000
```

The type of block determines the data which follow the header including the Tree and Sub-Tree types to define the topology and connections of the samples. Data is therefore organized as a table.

Frameworks such as neuroconstruct [34] can import morphology files in all of the above formats and use them in conjunction with network specification and cellular mechanisms to generate script files for various simulation platforms, such as NEURON, GENESIS, and PyNN. While Neuromorpho.org adopted SWC and NEURON's data model as their data-sharing standard, the Human Brain Project adopted the Neurolucida data model as the format of choice.

2.3. Models of biophysical data

The establishment of the Hodgkin–Huxley-type compartments modeling and the development of experimental methods such as patch-clamp recording and imaging techniques are two complementary advancements which have transformed the field of neuroscience. Molecular aspects of neuroscience could be precisely measured and then used for computational modeling. Modeling neuronal behavior at the molecular level is a crucial aspect of modern neuroscience. Standardizing and modeling neurophysiological data, which often include mechanisms as a set of nonlinear equations, differential equations, or kinetic reaction schemes, are critical for utilization of computational models across simulators.

2.3.1. Object-based structuring of biophysical data

Over the years, NEURON has been extended to include a library of biophysical mechanisms, which were developed using its dedicated high-level programming language: NMODL (which was also adopted later by GENESIS). For example, a model for a leak current using the canonical electrical model of a current channel, with i (leak current), e (equilibrium potential), and g (conductance) can be defined using NMDOL with [35]:

```
NEURON {                  % interface
  SUFFIX leak             % density mechanism
  NONSPECIFIC_CURRENT I % i in charge of the balance equations
  RANGE i, e, g           % are functions of position
}
PARAMETER {
  g = 0.001 (siemens/cm2) < 0, 1e9 >
```

```
   e = -65 (millivolt)
}
ASSIGNED {
   i (milliamp/cm2)
   v (millivolt)
}
BREAKPOINT {       % to be incrementally executed by the simulator
   i = g * (v - e)
}
```

In this modeling paradigm for physiological data, its type is encapsulated with a "template" class (following the object-based data structuring) and instantiate as objects where appropriate. For example, to instantiate a leakage current (with specific values for i and g) and attribute it to a NEURON's cable segment, one can write:

```
cable {
   nseg = 5
   insert leak
   g_leak = 0.002 % S/cm2
   e_leak = -70 % mV
}
print cable.i_leak(0.1) % show leak current density near 0 end of cable
```

Schematics of object-based representation of biophysical data are illustrated in **Figure 7** (right).

2.3.2. Hierarchical structuring of biophysical data

ChannelML is the second layer of NeuronML, enabling specifying biophysical data with XML. For example, specifying a Na + channel in ChannelML can be written as:

```
<channelml>
    <channel_type name="NaChannel" density="yes">
        <current_voltage_relation
          cond_law="ohmic" ion="na" default_erev="50" default_gmax="120">
            <gate name="m" instances="3">
                <closed_state id="m0"/>
                <open_state id="m"/>
                <transition name="h" from="m0" to="m" expr_form="exp_linear"
                            rate="1" scale="10" midpoint="-40"/>
                <transition name="beta" from="m" to="m0" expr_form="exponential"
                            rate="4" scale="-18" midpoint="-65"/>
            </gate>
            <gate name="h" instances="1">
                <closed_state id="h0"/>
                <open_state id="h"/>
                <transition name="alpha" from="h0" to="h" expr_form="exponential"
                            rate="0.07" scale="-20" midpoint="-65"/>
                <transition name="beta" from="h" to="h0" expr_form="sigmoid"
                            rate="1" scale="-10" midpoint="-35"/>
            </gate>
        </current_voltage_relation>
    </channel_type>
</channelml>
```

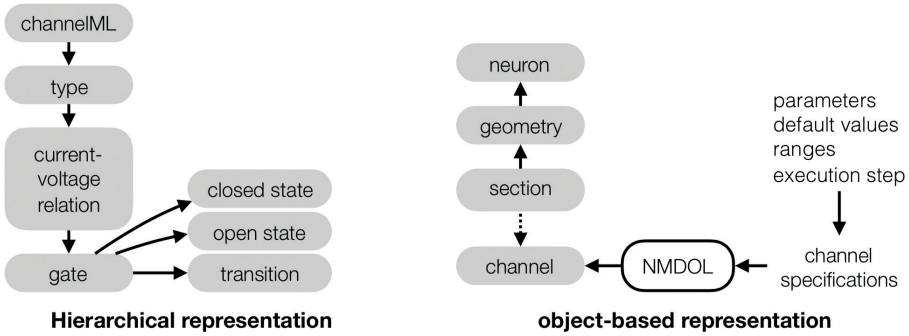

Figure 7. Hierarchical and object-based representation of biophysical data.

Neuroconstruct support both data models. Moreover, scripts for converting ChannelML specification to NEURON are also available. Schematics of hierarchy-based representation of biophysical data are illustrated in **Figure 7** (right).

2.3.3. Layer-oriented structuring of biophysical data

Another approach for physiological modeling is the layer-oriented approach (LOA) [36], in which the mathematical model (usually a set of differential equations) is governed by interlinked aspects of its structure. The LOA rationale is that biophysiological models such as the Hodgkin–Huxley model for ion channels have a hierarchical structure from which the underlying biological concept is reflected. Layer structure and relations are described in **Figure 8**.

By structuring mathematical behavior in a layered-structure manner, modules can be reused where different parameters are incorporated. One can utilize for example the same computational mechanism for membrane potential with either Hodgkin-Huxley model or GHK model or utilize the same gating dynamics for different dynamic models. Here, each layer is defined using a XML-like definition language (similarly to what was shown above), where connections between layers are defined separately in a meta-data file.

2.4. Models of network data

A model of a neural network must indicate at the very least the following specifications: connectivity scheme, as well as neuron and synapse models (typically by a set of differential equations, spike generation criteria, and refractory periods) [37].

2.4.1. Hierarchical structuring of network data

NetworkML is NeuroML's third specification level, which allows positioning neurons in 3D, as well as defining their connectivity pattern, and synaptic specifications to other neurons. It uses three core elements for network description: population (cells of a specific type),

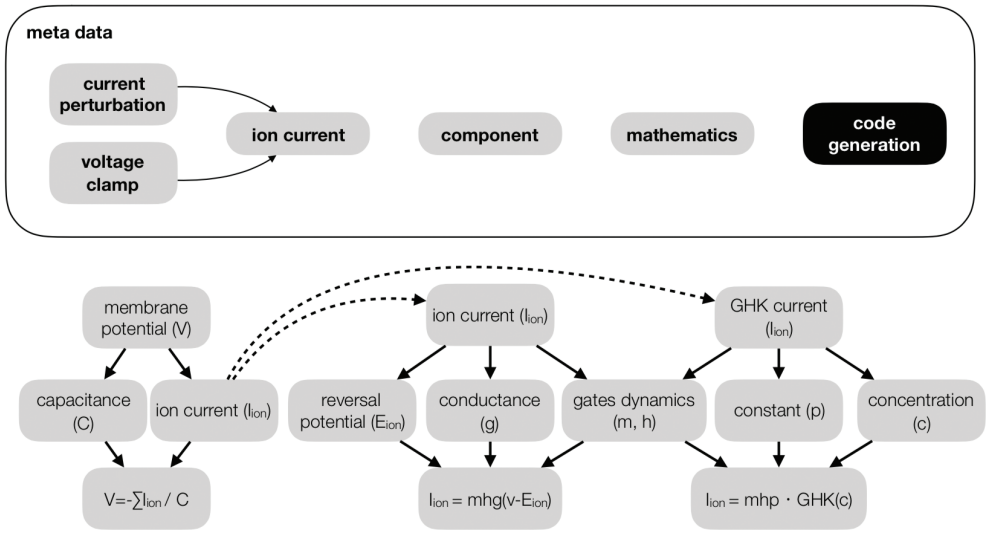

Figure 8. Layer-oriented representation of biophysical data.

projection (set of synaptic connections between populations), and input (describes an external electrical input into the network). Networks can be described with either instance-based (explicit list of positions and synaptic connections) or template-based (e.g., placing and connecting N cells randomly in a particular rectangular region) representation. For example, placing two populations of neuron PopA and PopB in 3D can be specified in NetworkML with [19]:

```
<populations>
  <population name="PopA" cell_type="CellA">
    <instances size="2">
       <instance id="0"> <location x="0" y="0" z="0"/> </instance>
       <instance id="1"> <location x="10" y="0" z="0"/> </instance>
    </instances>
  </population>
  <population name="PopB" cell_type="CellB">
    <instances size="3">
      <instance id="0"> <location x="0" y="100" z="0"/> </instance>
      <instance id="1"> <location x="10" y="100" z="0"/> </instance>
      <instance id="2"> <location x="20" y="100" z="0"/> </instance>
    </instances>
  </population>
</populations>
```

PopA and PopB can be connected with "projection":

```
<projections units="Physiological Units">
  <projection name="NetworkConnection" source="PopA" target="PopB">
    <synapse_props synapse_type="DoubExpSynA" internal_delay="5" weight="1" threshold="-20"/>
      <connections>
        <connection id="0" pre_cell_id="0" pre_segment_id = "1"
        post_cell_id="1" post_segment_id = "0" post_fraction_along = "0.25"/>
        <connection id="1" pre_cell_id="1" pre_segment_id = "1"
        post_cell_id="0 post_segment_id = "0" post_fraction_along = "0.25"/>
        </connection>
      </connections>
  </projection>
</projections>
```

Schematics of hierarchy-based representation of network data are illustrated in **Figure 9** (left).

```
for i in range(N):                    % N cells
    src = cells[i]                    % select source cell
    tgt = cells[(i + 1) % N]          % select target cell
    syn = h.ExpSyn(tgt.dend(0.5))     % place a synapse in the middle of the target
    nc = h.NetCon(src.soma(0.5)._ref_v, syn, sec=src.soma)
% Connect source soma to target synapse
    nc.weight[0] = .05
    nc.delay = 5
```

2.4.2. Object-based structuring of network data

In NEURON, neurons can be interconnected to form networks using the object-based approach. For example, giving an array of "cell" objects (each encapsulates its defining sections, such as a soma and dendrites), they can be connected (e.g., circle topology) using Neuron's ExpSyn and NetCon object using (written in NEURON-Python):

Schematics of object-based representation of network data are illustrated in **Figure 9** (right).

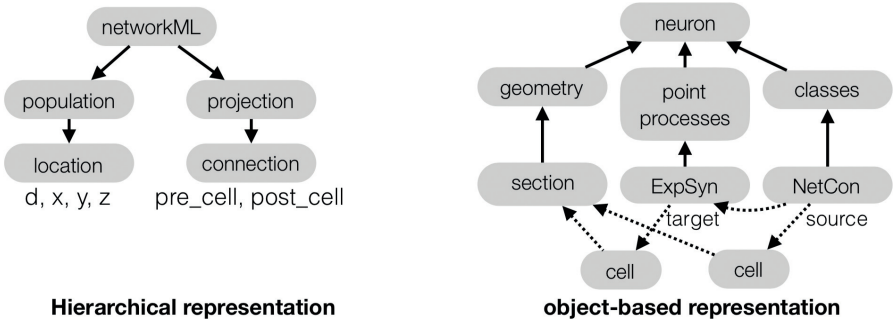

Figure 9. Hierarchical and object-based representation of network data.

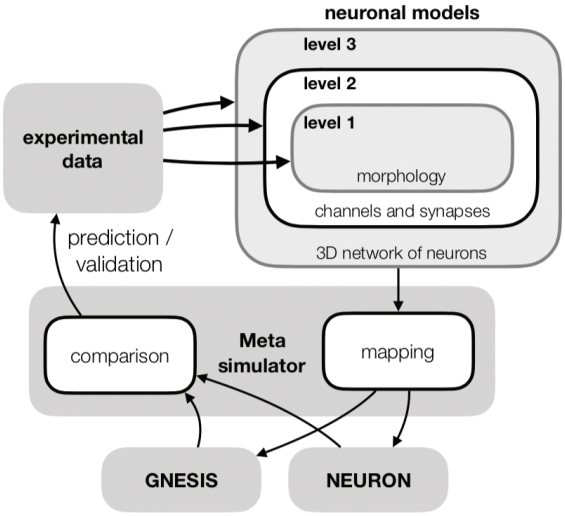

Figure 10. NeuroML 1 integrated approach to morphological, biophysical and network modeling.

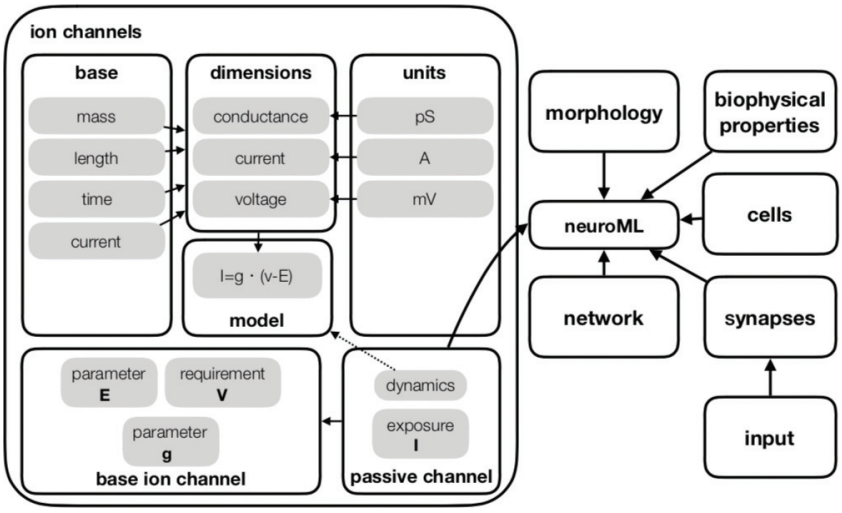

Figure 11. NeuroML 2 hybrid approach to morphological, biophysical and network modeling.

2.5. Integrated models

When it comes to integrated structuring of neuromorphic data, NeuroML is a prominent standard. It is defined using MorphML, ChannelML, and NetworkML, as they were described above. This integrated approach for neuroinformation standardization enables

such models to be directly converted and mapped into different simulation frameworks. When integrating standard representation models with a "Meta Simulator" such as the NeuroConstruct or PyNN, a powerful framework is established. With such an approach, data can be distributed across multiple simulators, compared, and then validated with experimental data (**Figure 10**) [19].

In the second version of NeuroML, a new holistic approach is being developed for modeling, termed Low Entropy Model Specification (LEMS). LEMS is a hierarchical, XML-based language in which ion channels, synapses, neurons, and networks can be specified together. It combines a hybrid hierarchical object-based approach to modeling. An illustration is given in **Figure 11**. Detailed example is given in [38].

3. Rapid development of specialized neurocentric databases

In contrary to primary and secondary databases, specialized databases are mostly curated by individual laboratories or consortiums. They are characterized with a research-specific relational schema and specialized data types. Specialized databases are under constant development, aiming at supporting the rapid advancements in experimental techniques, which often produce vast amount of heterogeneous data. Most specialized databases are comprised of both new results and datasets–derived entries, constituting a hybrid approach of the new and the established. This stands as a major challenge to specialized data base designer, which have to support data querying, acquiring, and parsing from established data sources, as well as to integrate (or link) the results, with their own data model.

Specifically, the curation of specialized databases for neuroinformatics is an ever-growing challenge due to the need for organizing, structuring, and interconnecting vast amount of data, with standardized data structures. Here, an open-source framework for the curation of specialized databases is proposed. Our framework has the potential of realizing two complementary needs in the context of neuroinformatics: (1) structuring experimental data with standardized models which can be used for cross-simulations and (2) incorporating the experimental data and models with other data such as relevant diseases, articles, and biological models.

3.1. Framework

Databases often use a stable URL syntax, which renders a standard set of input parameters into the information needed to search and fetch the requested data. The proposed framework supports the generation of URL structured interface to local and remote data sets, including NCBI's databases, Malacards, and Biomodels. It was implemented with Java, extended to support objects' persistency with EclipseLink. I chose Apache Derby (part of the Apache DB Project) for data management. Derby is written in Java, and it is suitable for code embedding due to its small footprint and ease of use. Syntactic analysis was based on the w3c.dom open libraries, Apache Commons, J3D, and jsoup. The framework is described in length and exemplified for the curation of a database dedicated for aneurysms in [39].

In the context of neuroinformatics, the user can therefore take her morphology, biophysical, and connectivity experimental data, encapsulate them into interconnected classes (thus, creating a schema), and then link each of them to a structured data model (such as the ones described above). Each data model can be connected to articles, biological models, and diseases, which can be derived from existing databases and deposited in a specialized local database. Data can be retrieved later for further analysis. See schematics in **Figure 12**.

The proposed framework can be implemented with different packages and programming environments. For example, Java was utilized to map data entities to NCBI's PubChem schema and to provide functions to invoke NCBI eUtilities and PubChem web services [40]. Similarly, objects persistency can be attained with either Python, Java, or C++. Python's standard library for example supports a family of hash-based file formats and objects serialization. The Java Persistence API (JPA) was also implemented by various development groups, including Apache OpenJPA, Hibernate, and EclipseLink, offering metadata-based automatic creation of data models. Providers of database management frameworks are likewise varied and include Apache Derby and the cloud-based MongoDB.

3.2. Implementation

I have recently proposed a framework for the development of specialized databases [39]. In this framework, Java was chosen as the development environment, with which interfaces to online databases such as MalaCards (to retrieve disease information), Biomodels (to retrieve biological models), and NCBI's databases (to retrieve gene, taxonomy, protein, and articles data) were designed. By integrating these interfaces with EclipseLink (JPA provider), Apache Derby (database

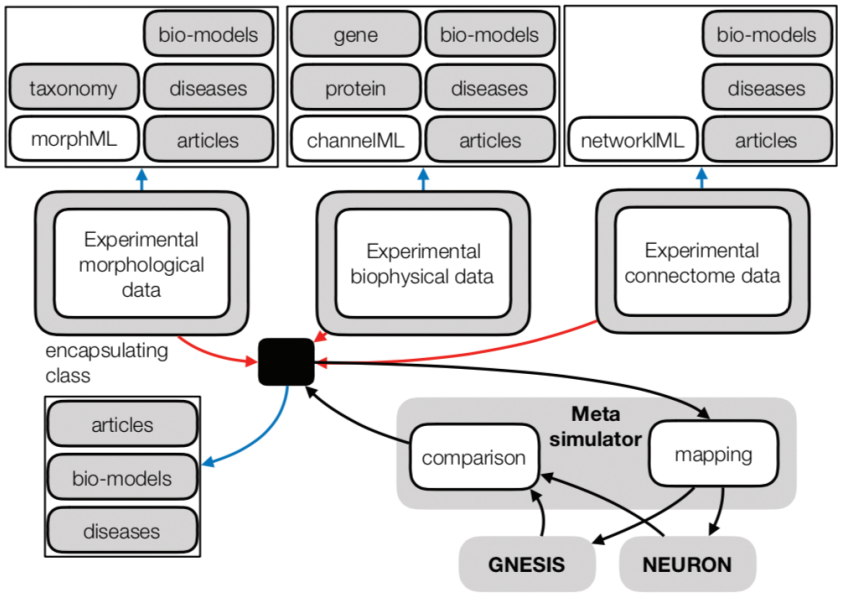

Figure 12. Database integrated approach to morphological, biophysical and network modeling.

manager), and a range of data parsers, a versatile framework for the curation of specialized databases is provided. This framework can be used to integrate new data and database-derived information into a user-defined data model. A schematic of the implementation is presented in **Figure 13**.

In the framework's main data flow, structured URL interfaces are used to establish connections between the user-defined data model to online data sets. Here, I used Entrez to interface with NCBI's data sets. NCBI's Entrez Programming Utilities provide a structured URL interface to their dozens of databases covering a variety of biomedical data, including gene and protein sequences, gene records, three-dimensional molecular structures, and biomedical literature [41].

Efforts to provide a similar utility for the neuroscientific community were also made. For example, Samwald and colleagues developed the "Neuron Entrez" [42], which integrates several neuroscientific ontologies: NeuronDB and ModelDB, subcellular anatomy ontology (SAO), and an OWL conversion of the cell centered database (CCDB). Once matured, this type of integrated neurocentric retrieval of data can greatly enhance frameworks, such as the one being proposed here.

A series of data processing tools were utilized to implement parsers for syntactic analysis of the retrieved data. The w3c.dom package provides the document object model (DOM) interfaces, which were used as the API for XML processing. This is essential for handling NeuroML structured data. The Apache Commons' libraries, the jsoup library, and the org.j3d library of the Java 3D Community were utilized for CSV, html, and STL parsing, respectively.

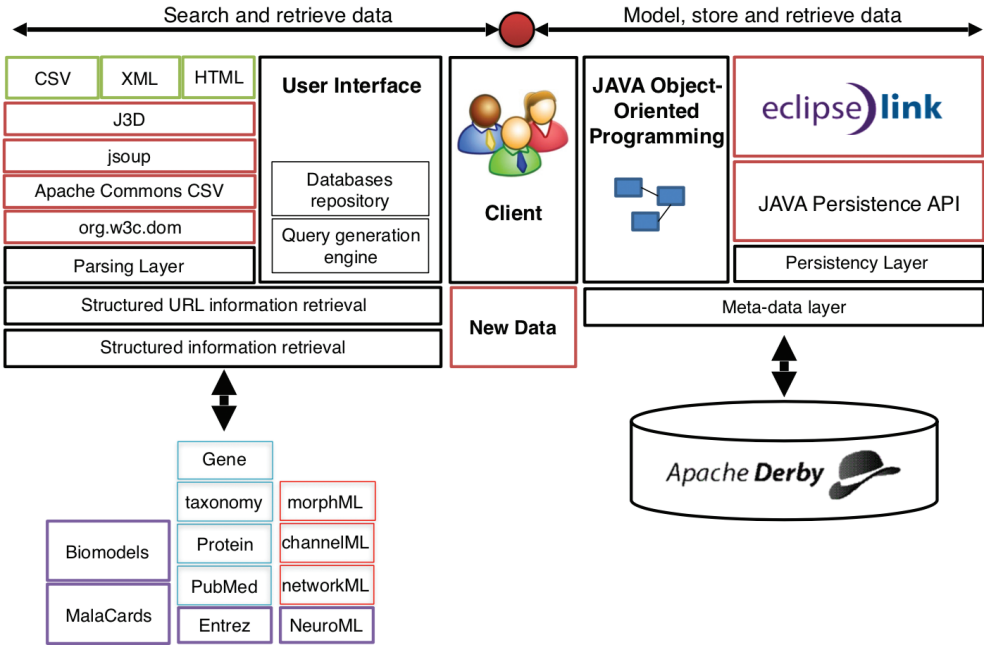

Figure 13. Realization of the database integrated approach to morphological, biophysical and network modeling.

The user utilizes Java object-oriented approach to encapsulate the retrieved data and to integrate it with her own data model. Object-relational mapping (converting Java objects to relational tables) is defined via persistence metadata. Metadata is defined via annotations embodied in the Java class and with an accompanying XML file. This allows EclipseLink to statically and dynamically query the database with SQL-like syntax. Apache Derby supports SQL data storing and querying in a client/server operation mode (commonly used database architecture). Suggested implementation for the above is provided via NBEL-lab.com and distributed under the creative common agreement.

4. Conclusions

Recent developments in Integrated Neuroscience (IN) are often characterized with efforts to up-scale data production and to provide frameworks from which new insights can emerge [43]. Since insights from integrated neuronal models often rely on the combination of experimental and computational approaches [44], simulations and modeling have a key role. Moreover, sharing neuroscientific data in the heterogeneous environment of IN drove the momentum for standardizing data models for neuronal morphologies, biophysical properties, and connectivity. Here, I propose a framework with which standardized models can be structured with experimental data, as well as with established data from existing databases. A combination of an integrated approach to neuroscience with the establishment of a federated framework for "collective wisdom" of neuroscientists and engineers might open a new dimension for data-driven neuroscience and fuel the celebration of the "era of the brain."

Author details

Elishai Ezra Tsur

Address all correspondence to: elishai85@gmail.com

Neuro-Biomorphic Engineering Lab, Jerusalem College of Technology, Israel

References

[1] Kashtan N, Alon U. Spontaneous evolution of modularity and network motifs. Proceedings of the National Academy of Sciences of the United States of America. 2005;**102**(39): 13773-13778

[2] Crick F, Koch C. A framework for consciousness. Nature Neuroscience. 2003;**6**(2):119-126

[3] Fox PT, Lancaster JL. Neuroscience on the net. Science. 1994;**266**(5187):994-997

[4] Amunts K, Ebell C, Muller J, Telefont M, Knoll A, Lippert T. The human brain project: Creating a European research infrastructure to decode the human brain. Neuron. 2016; **92**(3):574-581

[5] Collins FS, Morgan M, Patrinos A. The human genome project: Lessons from large-scale biology. Science. 2003;**300**(5617):286-290

[6] Insel TR, Landis SC, Collins FS. The NIH brain initiative. Science. 2013;**340**(6133):687-688

[7] Van Essen DC, Ugurbil K, Auerbach E, Barch D, Behrens TE, Bucholz R, et al. The human connectome project: A data acquisition perspective. NeuroImage. 2012;**62**(4): 2222-2231

[8] Jabalpurwala I. Brain Canada: One brain one community. Neuron. 2016;**92**(3):601-606

[9] Poo MM, Du JL, Ip NY, Xiong ZQ, Xu B, Tan T. China brain project: Basic neuroscience, brain diseases, and brain-inspired computing. Neuron. 2016;**92**(3):591-596

[10] Heidorn PB. Shedding light on the dark data in the long tail of science. Library Trends. 2008;**57**(2):280-299

[11] Ferguson AR, Nielson JL, Cragin MH, Bandrowski AE, Martone ME. Big data from small data: Data-sharing in the 'long tail' of neuroscience. Nature Neuroscience. 2014;**17**(11): 1442-1447

[12] Marmarou A, Lu J, Butcher I, McHugh GS, Mushkudiani NA, Murray GD, Steyerberg EW, Maas AI. IMPACT database of traumatic brain injury: Design and description. Journal of Neurotrauma. 2007;**24**(2):239-250

[13] Steyerberg EW, Mushkudiani N, Perel P, Butcher I, Lu J, McHugh GS, Murray GD, Marmarou A, Roberts I, Habbema JDF, Maas AIR. Predicting outcome after traumatic brain injury: Development and international validation of prognostic scores based on admission characteristics. PLoS Medicine. 2008;**5**(9):e165

[14] Callahan A, Anderson KD, Beattie MS, Bixby JL, Ferguson AR, Fouad K, Jakeman LB, Nielson JL, Popovich PG, Schwab JM, Lemmon VP. Developing a data sharing community for spinal cord injury research. Experimental Neurology. 2017;**295**:135-143

[15] Kötter R. Neuroscience databases: Tools for exploring brain structure–function relationships. Philosophical Transactions of the Royal Society of London B: Biological Sciences. 2001;**356**(1412):1111-1120

[16] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, et al. The FAIR guiding principles for scientific data management and stewardship. Scientific Data. 2016;**3**:160018

[17] Hucka M, Nickerson DP, Bader GD, Bergmann FT, Cooper J, Demir E, Garny A, et al. Promoting coordinated development of community-based information standards for modeling in biology: The COMBINE initiative. Frontiers in Bioengineering and Biotechnology. 2015;**3**

[18] Yann LF, Davison AP, Gleeson P, Imam FT, Kriener B, Larson SD, Ray S, Schwabe L, Hill S, Schutter ED. Computational neuroscience ontology: A new tool to provide semantic meaning to your models. BMC Neuroscience. 2012;**13**(1):P149

[19] Gleeson P, Crook S, Cannon RC, Hines ML, Billings GO, Farinella M, Morse TM, et al. NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail. PLoS Computational Biology. 2010;**6**(6):e1000815

[20] Richmond P, Cope A, Gurney K, Allerton DJ. From model specification to simulation of biologically constrained networks of spiking neurons. Neuroinformatics. 2014;**12**(2):307-323

[21] Gardner D, Akil H, Ascoli GA, Bowden DM, Bug W, Donohue DE, Goldberg DH, et al. The neuroscience information framework: A data and knowledge environment for neuroscience. Neuroinformatics. 2008;**6**(3):149-160

[22] Marenco LN, Wang R, Bandrowski AE, Grethe JS, Shepherd GM, Miller PL. Extending the NIF DISCO framework to automate complex workflow: Coordinating the harvest and integration of data from diverse neuroscience information resources. Frontiers in Neuroinformatics. 2014;**8**

[23] Herz AV, Meier R, Nawrot MP, Schiegel W, Zito T. G-node: An integrated tool-sharing platform to support cellular and systems neurophysiology in the age of global neuroinformatics. Neural Networks. 2008;**21**(8):1070-1075

[24] Hines M. NEURON – A program for simulation of nerve equations. Neural Systems: Analysis and Modeling. 1993;**127**:136

[25] Wilson MA, Bhalla US, Uhley JD, Bower JM. GENESIS: A system for simulating neural networks. Advances in Neural Information Processing Systems. 1989:485-492

[26] Davison AP, Brüderle D, Eppler JM, Kremkow J, Muller E, Pecevski D, Perrinet L, Yger P. PyNN: A common interface for neuronal network simulators. Frontiers in Neuroinformatics. 2009;**2**:11

[27] Ascoli GA, Donohue DE, Halavi M. NeuroMorpho.Org: A central resource for neuronal morphologies. The Journal of Neuroscience. 2007;**27**(35):9247-9251

[28] Shepherd GM, Stepanyants A, Bureau I, Chklovskii D, Svoboda K. Geometric and functional organization of cortical circuits. Nature Neuroscience. 2005;**8**(6):782-790

[29] Marenco L, Nadkarni P, Skoufos E, Shepherd G, Miller P. Neuronal database integration: The Senselab EAV data model. Proceedings of the AMIA Symposium. 1999:102

[30] Hines ML, Morse T, Migliore M, Carnevale NT, Shepherd GM. ModelDB: A database to support computational neuroscience. Journal of Computational Neuroscience. 2004;**17**(1): 7-11

[31] McDougal RA, Morse TM, Carnevale T, Marenco L, Wang R, Migliore M, Miller PL, Shepherd G, Hines ML. Twenty years of ModelDB and beyond: Building essential modeling tools for the future of neuroscience. Journal of Computational Neuroscience. 2017;**42**(1):1-10

[32] Myatt DR, Hadlington T, Ascoli GA, Nasuto SJ. Neuromantic–from semi-manual to semi-automatic reconstruction of neuron morphology. Frontiers in Neuroinformatics. 2012;**6**

[33] Crook S, Gleeson P, Howell F, Svitak J, Silver RA. MorphML: Level 1 of the NeuroML standards for neuronal morphology data and model specification. Neuroinformatics. 2007;**5**(2):96-104

[34] Gleeson P, Steuber V, Silver RA. neuroConstruct: A tool for modeling networks of neurons in 3D space. Neuron. 2007;**54**(2):219-235

[35] Hines ML, Carnevale NT. Expanding NEURON's repertoire of mechanisms with NMODL. Neural Computation. 2000;**12**(5):995-1007

[36] Raikov I, Schutter ED. The layer-oriented approach to declarative languages for biological modeling. PLoS Computational Biology. 2012;**8**(5):e1002521

[37] Nordlie E, Gewaltig M-O, Plesser HE. Towards reproducible descriptions of neuronal network models. PLoS Computational Biology. 2009;**5**(8):e1000456

[38] Cannon RC, Gleeson P, Crook S, Ganapathy G, Marin B, Piasini E, Silver RA. LEMS: A language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. Frontiers in Neuroinformatics. 2014;**8**

[39] Tsur EE. Rapid development of entity-based data models for bioinformatics with persistence object-oriented design and structured interfaces. BioData Mining. 2017;**10**(1):11

[40] Southern MR, Griffin PR. A Java API for working with PubChem datasets. Bioinformatics. 2011;**27**(5):741-742

[41] NCBI. "Entrez programming utilities help," 2009. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK25501/

[42] Samwald M, Lim E, Masiar P, Marenco L, Chen H, Morse T, Mutalik P, Shepherd G, Miller P, Cheung K-H. Entrez neuron RDFa: A pragmatic semantic Web application for data integration in neuroscience research. Studies in Health Technology and Informatics. 2009;**150**:317-321

[43] Narasimhan K. Scaling up neuroscience. Nature Neuroscience. 2004;**7**:425

[44] Markram H. The blue brain project. Nature Reviews Neuroscience. 2006;**7**(2):153-160