NeuroML update

NeuroML update

Ankur Sinha 21/09/2022

☐ What we've been up to on the NeuroML front

What we've been up to on the NeuroML front

- Google summer of code (GSoC)
- Paper and general improvements

Google Summer of Code

Google Summer of Code

└**GSoC**

- Under the INCF organisation
 - 40 accepted this year, in total
 - No longer limited to students

- Under the INCF organisation
 - 40 accepted this year, in total
 - No longer limited to students
- Us: 4 candidates
 - 3 working on NeuroML/modelling
 - 1 on NWB
 - 175 hours each, over 3 months (medium projects)

NeuroML update -Google Summer of Code

-GSoC

- · Under the INCF organisation - 40 accepted this year, in total · No longer limited to students · Us: 4 candidates - 1 on NWB
- 175 hours each, over 3 months (medium projects)

- Under the INCF organisation
 - · 40 accepted this year, in total
 - No longer limited to students
- · Us: 4 candidates
 - 3 working on NeuroML/modelling
 - 1 on NWB
 - 175 hours each, over 3 months (medium projects)
- Goals:
 - Conversions of models to NeuroML (to allow them to be re-used and featured on OSB)
 - · Get folks looking into and using NeuroML in different use cases

NeuroML update -Google Summer of Code

-GSoC

- 40 accepted this year, in total · No longer limited to students · Us: 4 candidates

· Under the INCF organisation

- 1 on NWB · 175 hours each, over 3 months (medium projects

. Conversions of models to NeuroML (to allow them to be re-used and featured on OSB) - Get folks looking into and using NeuroML in different use

- Get folks to improve NeuroML where possible

-GSoC

-Google Summer of Code

- Under the INCF organisation
 - 40 accepted this year, in total
 - No longer limited to students
- · Us: 4 candidates
 - 3 working on NeuroML/modelling
 - 1 on NWB
 - 175 hours each, over 3 months (medium projects)
- Goals:
 - Conversions of models to NeuroML (to allow them to be re-used and featured on OSB)
 - Get folks looking into and using NeuroML in different use cases
 - Get folks to improve NeuroML where possible

 Masters student at the Bernstein Centre for Computational Neuroscience, Berlin.

2024-02-2 GSoC: Anuja: convert Allen Institute models to

NeuroML

NeuroML

-GSoC: Anuja: convert Allen Institute models to

SoC: Anuia: convert Allen Institute models to NeuroM

- Masters student at the Bernstein Centre for Computational Neuroscience, Berlin.
- Source models: Allen institutes cell type database (example)

GSoC: Anuja: convert Allen Institute models to NeuroML

- Masters student at the Bernstein Centre for Computational Neuroscience, Berlin.
- Source models: Allen institutes cell type database (example)
- Steps:
 - Automate download of models using the Allen SDK
 - · Automate conversion to NeuroML
 - Plot comparison graphs to validate conversions (LIF models, Detailed)
 - · Document comparison, usage, update OMV tests for CI

NeuroML update

2024-02-2

-Google Summer of Code

-GSoC: Anuja: convert Allen Institute models to NeuroML

SSoC: Anuja: convert Allen Institute models to NeuroML

Neuroscience, Berlin.

Source models: Allen institutes cell type database

 Source models: Allen institutes cell type database (example)

Automate download of models using the Allen SDK
 Automate conversion to NeuroML
 Plot comparison graphs to validate conversions (LI

Automate conversion to NeuroM.

Plot comparison graphs to validate conversions (LIF models, Detailed)

GSoC: Anuja: example figure I

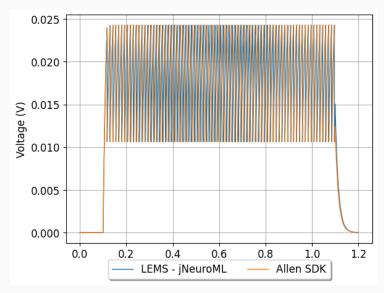
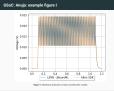


Figure 1: Membrane potentials of Allen and NeuroML models

NeuroML update 2024-02-21 —Google Summer of Code

—GSoC: Anuja: example figure I



GSoC: Anuja: example figure II

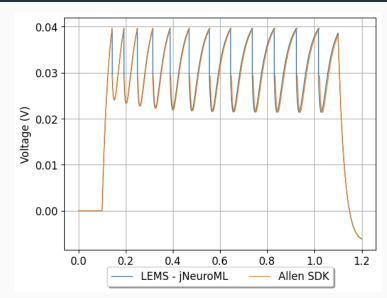
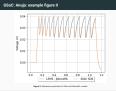


Figure 2: Membrane potentials of Allen and NeuroML models

NeuroML update 2024-02-21 Google Summer of Code

☐GSoC: Anuja: example figure II



GSoC: Anuja: example figure III

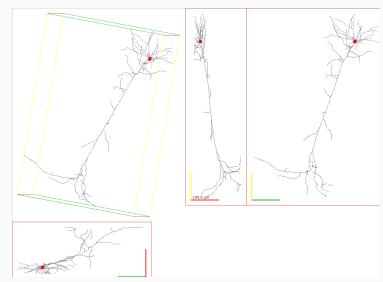


Figure 3: Morphology of example cell

NeuroML update

Google Summer of Code

☐GSoC: Anuja: example figure III



GSoC: Anuja: example figure IV

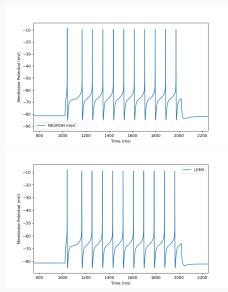


Figure 4: NEURON vs NeuroML output



└─GSoC: Anuja: example figure IV



NeuroML update

-Google Summer of Code

☐GSoC: Shayan: convert BahlEtAl2012 (Reduced L5 Pvr Cell) to NeuroML

MSc/BSc from IIT Kharagpur (Maths and Computing

9/34

• Source models: OSB/BahlEtAl2012

NeuroML update

Google Summer of Code

GSoC: Shayan: convert BahlEtAl2012 (Reduced

L5 Pyr Cell) to NeuroML

2024-02-21

- Source models: OSB/BahlEtAl2012
- Steps:
 - Convert ion channels to NeuroML, validate, compare with NEURON mod files
 - Implement single compartment cell model with passive channels
 - incrementally add ion channels
 - compare with NEURON model
 - Implement multi-compartmental cell, repeat
 - · Document comparison, usage, update OMV tests for CI
 - · Interactive notebook to reproduce figures from paper using NeuroML models

NeuroML update

Google Summer of Code

-GSoC: Shayan: convert BahlEtAl2012 (Reduced L5 Pvr Cell) to NeuroML

MSc/BSc from IIT Kharagour (Maths and Computing Source models: OSB/BahlEtAl2012

Implement single compartment cell model with passive

 Document comparison, usage, update OMV tests for CI Interactive notehook to reproduce figures from paper usin

GSoC: Shayan: example figures

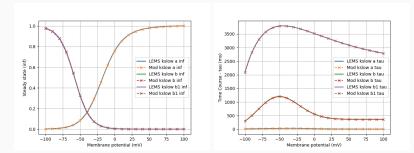


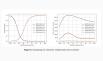
Figure 5: Comparing ion channels: steady state, time constant

NeuroML update

Google Summer of Code

2024-02-21

GSoC: Shayan: example figures



GSoC: Shayan: example figures

GSoC: Shayan: example figures II

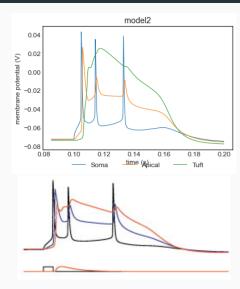


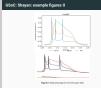
Figure 6: Replicating figures from the paper (8d)

NeuroML update

Google Summer of Code

2024-02-21

☐GSoC: Shayan: example figures II



· Aerospace engineer (MTech, IIT Kharagpur), working in South Korea

GSoC: Rahul: convert HH tutorial to Jupyter notebook

notebook

GSoC: Rahul: convert HH tutorial to Jupyter

GSoC: Rahul: convert HH tutorial to Jupyter notebook

- Aerospace engineer (MTech, IIT Kharagpur), working in South Korea
- Source models: OpenWorm/HH tutorial

GSoC: Rahul: convert HH tutorial to Jupyter notebook

- Aerospace engineer (MTech, IIT Kharagpur), working in South Korea
- Source models: OpenWorm/HH tutorial
- · Steps:
 - Investigate Jupyter widgets
 - Convert pure Python HH tutorial to use Jupyter widgets
 - · Investigate NeuroML based tutorial
 - Investigate conversion of NeuroML to Jupyter widgets
 - Update sphinx documentation for ReadTheDocs site
 - · Document usage.

NeuroML update -Google Summer of Code

> -GSoC: Rahul: convert HH tutorial to Jupyter notebook

GSoC: Rahul: convert HH tutorial to Jupyter notebook

- Aerospace engineer (MTech, IIT Kharagpur), working in
- · Source models: OpenWorm/HH tutoria
- Investigate Jupyter widgets
- Convert pure Python HH tutorial to use Jupyter widgets
- Investigate NeuroML based tutorial Investigate conversion of NeuroML to Jupyter widgets Undate onliny documentation for ReadTheDocs site

2024-02-2

GSoC: Rahul: example I

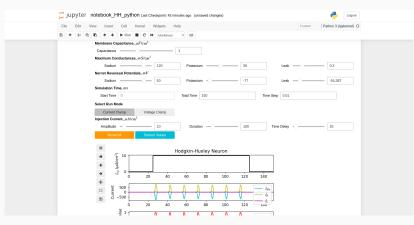


Figure 7: Pure python tutorial converted to use Jupyter Widgets

NeuroML update 2024-02-21 —Google Summer of Code

GSoC: Rahul: example I



GSoC: Rahul: example II

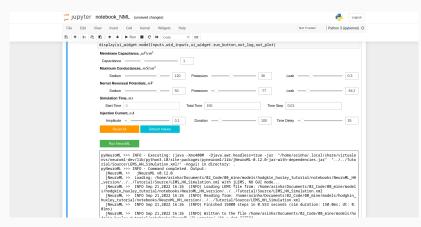


Figure 8: NeuroML tutorial converted to use Jupyter Widgets

NeuroML update

☐Google Summer of Code

2024-02-2

—GSoC: Rahul: example II



Paper and general updates

2024-02-21

Convince readers (research community) to use NeuroML for their modelling work*.

—Paper and general updates

Paper: NeuroML is the best thing since sliced bread

bread

Paper: NeuroML is the best thing since sliced

er: NeuroML is the best thing since sliced bread

Convince readers (research community) to use NeuroML for their modelling work*.

*instead/ahead of other tools

Paper: NeuroML is the best thing since sliced bread

Convince readers (research community) to use NeuroML for their modelling work*.

- *instead/ahead of other tools
- *on a daily basis
 - not as an afterthought for standardisation—once the paper has been published no one has time to re-write model (or re-process data!) to standardise

2024-02-2

Paper: NeuroML is the best thing since sliced bread

Paper: NeuroML is the best thing since sliced bread

ce readers (research community) to use NeuroML their modelling work*.

*Instead/ahead of other tools
 * on a daily basis
 * not as an afterthought for standardisation—once the paper has been published no one has time to re-write model (or

Paper: NeuroML is the best thing since sliced bread

Convince readers (research community) to use NeuroML for their modelling work*.

- *instead/ahead of other tools
- *on a daily basis
 - not as an afterthought for standardisation—once the paper has been published no one has time to re-write model (or re-process data!) to standardise
 - the carrot, "standardisation is good for science", isn't enough in a mostly resource limited academic/research system
 - requires stick: "for this grant, you must ..."; "for this journal, you must ..."

NeuroML update

—Paper and general updates

Paper: NeuroML is the best thing since sliced bread

Paper: NeuroML is the best thing since sliced bread

lers (research community) to use NeuroM

Instead/ahead of other tools
on a daily basis
not as an afterthought for standardisation

as been published no one has time to re-write model (or process datal) to standardisse the carrot, "standardisation is good for science", isn't enoug in a mostly resource limited academic/research system requires stick: "for this grant, you must"; "for this journal,

2024-02-2

You should use NeuroML because

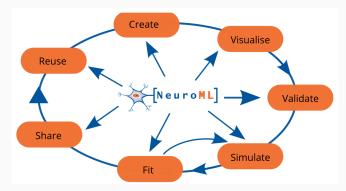


Figure 9: NeuroML overview figure from paper

euroN -Pap

2024-02-21

NeuroML update

Paper and general updates

Fast hindiams by shores

You should use NeuroML because

—You should use NeuroML because

You should use NeuroML because

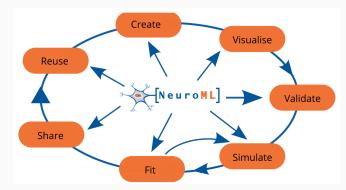


Figure 9: NeuroML overview figure from paper

These claims are all true, and have been for quite a while.

2024-02-21

NeuroML update

—Paper and general updates

These claims are all true, and have been for quite a while

You should use NeuroML because

└─You should use NeuroML because

2024-02-21

2024-02-2

Irrespective of all its features, NeuroML needs to be easier to use than other tools.

Paper and general updates

—So many advantages: why isn't everyone using it?

So many advantages: why isn't everyone using it?

Irrespective of all its features, NeuroML needs to be easier to use than other tools.

- information on NeuroML needs to be easy to find:
 - website
 - point of entry for completely new users: first impression
 - replaced with modern looking static page that redirects to individual pages in docs
 - · all information migrated to docs

NeuroML update

Paper and general updates

information on NeuroML needs to be easy to find:

- website
- point of entry for correlately new users: first impression
- replaced with modern looking static page that redirects to
individual pages in docs
- all information migrated to docs

So many advantages: why isn't everyone using it?

-So many advantages: why isn't everyone using it?

2024-02-2

So many advantages: why isn't everyone using it?

Irrespective of all its features, NeuroML needs to be easier to use than other tools.

- information on NeuroML needs to be easy to find:
 - website
 - point of entry for completely new users: first impression
 - replaced with modern looking static page that redirects to individual pages in docs
 - all information migrated to docs
 - docs
 - re-organised, modernised
 - include tutorials, interactive tutorials via Jupyter note books, how-tos
 - complete searchable schema docs
 - still not yet fully complete

NeuroML update -Paper and general updates -So many advantages: why isn't everyone using

many advantages: why isn't everyone using it?

information on NeuroML needs to be easy to find replaced with modern looking static page that redirects to · all information migrated to docs

include tutorials, interactive tutorials via Jupyter note books

2024-02-2

it?

So many advantages: why isn't everyone using it?

Irrespective of all its features, NeuroML needs to be easier to use than other tools.

- usability: so much can be done, but can it be done easily?
 - has not been clearly easier to use than other tools
 - Python API exists, but we haven't taken advantage of it enough to make life easier for users yet

NeuroML update

2024-02-2

-Paper and general updates

has not been clearly easier to use than other tools
 Python API exists, but we haven't taken advantage of it enough to make life easier for users yet

So many advantages: why isn't everyone using it?

-So many advantages: why isn't everyone using it?

Example: create: single neuron Izhikevich network (from docs)

```
nml_doc = NeuroMLDocument(id="IzhSingleNeuron")
izh0 = Izhikevich2007Cell(
    id="izh2007RS0", v0="-60mV", C="100pF", k="0.7nS_per_mV", vr="-60mV",
    vt="-40mV", vpeak="35mV", a="0.03per_ms", b="-2nS", c="-50.0mV", d="100pA")
nml_doc.izhikevich2007_cells.append(izh0)
net = Network(id="IzhNet")
nml_doc.networks.append(net)
size0 = 1
pop0 = Population(id="IzhPop0", component=izh0.id, size=size0)
net.populations.append(pop0)
pg = PulseGenerator(
    id="pulseGen_%i" % 0, delay="0ms", duration="1000ms",
    amplitude="0.07 nA"
nml_doc.pulse_generators.append(pg)
exp_input = ExplicitInput(target="%s[%i]" % (pop0.id, 0), input=pg.id)
net.explicit_inputs.append(exp_input)
nml_file = 'izhikevich2007_single_cell_network.nml'
writers.NeuroMLWriter.write(nml_doc, nml_file)
print("Written network file to: " + nml_file)
validate_neurom12(nml_file)
```

NeuroML update

2024-02-2

19/34

-Paper and general updates

Example: create: single neuron Izhikevich network (from docs) Example: create: single neuron Izhikevich network (from docs)

, vier-kinder, derdolger, berkindel, par yelft, verer-kindel miterialität, and Klaper, mit, berkinder, and district de MRT_realita appendicionality
**MRT_realita appendicionality
**Technologicality
**

na papatoina apasilpopi)

ge "Nalmineratori"

idi"pulmina (10" | 50, dalay "lim", daratime "liilim",

papitando "Alit ali"

papitando "Alit ali"

nalitando "Alit alitando "Alitando"

nalitando "Alitando"

nalitando"

nalitando

nal

Inspect/visualise network



Figure 10: Generated network graph

```
>>> nml_doc.summary()
* NeuroMLDocument: IzhSingleNeuron
* Izhikevich2007Cell: ['izh2007RS0']
* PulseGenerator: ['pulseGen_0']
* Network: IzNet
  1 cells in 1 populations
     Population: IzhPop0 with 1 components of type izh2007RS0
   O connections in O projections
   O inputs in O input lists
    explicit inputs (outside of input lists)
     Explicit Input of type pulseGen_0 to IzhPopO(cell 0), destination: unspecified
*****************
```

NeuroML update

—Paper and general updates

2024-02-2

20/34

└─Inspect/visualise network

Inspectivisasilise network

Figure 10 december of the december

- extremely declarative
 - · components (Izhikevich2007Cell, Network, Population, PulseGenerator, ExplicitInput) clearly visible
 - · components and dynamics fully, formally documented in schema docs
 - component parameters clearly visible
 - · units/dimensions explicitly mentioned

NeuroML update -Paper and general updates

2024-02-2

-Strengths

· extremely declarative - components (Izhikevich/2007Cell, Network, Population,

PulseGenerator. Explicitingut) clearly visible · components and dynamics fully, formally documented in

· units/dimensions explicitly mentioned

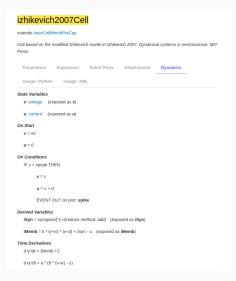


Figure 11: Schema docs on Izhikevich2007Cell

NeuroML update 2024-02-21 Paper and general updates

-Strengths

- validation without simulation (no simulator has this)
 - Level 1 validation: units/dimensions, structure of model checked against schema
 - · Level 2 validation: extra "logical" checks

NeuroML update

—Paper and general updates

-Strengths

2024-02-2

 validation without simulation (no simulator has this)
 Level 1 validation: units/dimensions, structure of mode checked against schema
 Level 2 validation: extra "logical" checks

- validation without simulation (no simulator has this)
 - Level 1 validation: units/dimensions, structure of model checked against schema
 - Level 2 validation: extra "logical" checks
- easy simulation with different simulators
- easy visualisation and inspection of model: network structure, connectivity matrices, LEMS simulation graph, morphology figures, model summary

NeuroML update -Paper and general updates

-Strengths

- · validation without simulation (no simulator has this) · Level 1 validation: units/dimensions, structure of mode checked against schema
- · Level 2 validation: extra "logical" checks
- easy visualisation and inspection of model: network structure, connectivity matrices, LEMS simulation graph

—Paper and general updates

-Weaknesses

- information on components: must switch back and forth between docs and code
 - Not too bad: required for most simulators/programming languages

Weaknesses

- information on components: must switch back and forth between docs and code
 - Not too bad: required for most simulators/programming languages
- unclear how components fit together
 - how do I know that Network \rightarrow Populations \rightarrow ?

Weaknesses

· Not too bad: required for most simulators/programmin unclear how components fit together how do I know that Network — Populations —?

between docs and code

2024-02-2

- information on components: must switch back and forth
 - Not too bad: required for most simulators/programming languages
- · unclear how components fit together

between docs and code

- how do I know that Network → Populations →?
- Too many "under the hood" bits that users are expected to know:
 - nml_doc.izhikevich2007_cells.append(izh0)
 - how do users know this?
 - read the schema docs?
 - read the NeuroML Python API documentation?
 - read the NeuroML Python API source? (currently 64,000 lines of code)

-Weaknesses

- validation upon model completion
 - using jnml, so requires Java
 - better than errors on run, but still quite late

Paper and general updates

Weaknesses

· validation upon model completion · using jast, so requires Java

- validation upon model completion
 - using jnml, so requires Java
 - · better than errors on run, but still quite late
- NeuroML generated NEURON code is less performant than native NEURON code
 - · because hand-written mod files can be optimised, while we rely on a template to generate them
 - TODO: optimisation of template

Example: create: single neuron Izhikevich network (devel)

```
nml_doc = component_factory("NeuroMLDocument", id="IzhSingleNeuron")
     nml doc.info(show contents=True)
     izh0 = nml_doc.add(
         "Izhikevich2007Cell".
         id="izh2007RS0", v0="-60mV", C="100pF", k="0.7nS_per_mV", vr="-60mV",
         vt="-40mV", vpeak="35mV", a="0.03per_ms", b="-2nS", c="-50.0mV", d="100pA")
     izh0.info(show contents=True)
     net = nml_doc.add("Network", id="IzNet", validate=False)
     size0 = 1
     pop0 = net.add("Population", id="IzhPop0", component=izh0.id, size=size0)
     pg = nml_doc.add(
         "PulseGenerator".
         id="pulseGen_%i" % 0, delay="0ms", duration="1000ms",
         amplitude="0.07 nA"
19
     exp_input = net.add("ExplicitInput", target="%s[%i]" % (pop0.id, 0), input=pg.id)
     nml doc.validate(recursive=True)
     nml_file = 'izhikevich2007_single_cell_network.nml'
     writers.NeuroMLWriter.write(nml doc, nml file)
     print("Written network file to: " + nml_file)
```

NeuroML update

2024-02-2

-Paper and general updates

Example: create: single neuron Izhikevich network (devel)

Example: create: single neuron Izhikevich network (devel)

requests (servey("lear#linement", id="iddingialness") alo(idea_menterless)

- milyter, mild [**Telephone (**Telephone (**Telephone

stond = 1 pp(= na. ndf(**Ppdiesin*, id=*Inft*pf*, componentials.id, sine-sine) pp(= na. dos. ndf(**Psisoline star*, id=*psisoline, 10* 1.0, doi:py**On*, does in=*100in*, splitseline, 10* 1.0, doi:py**On*, does in=*100in*, splitseline, 10* 10.

aplicate/4.07 al'

sep.inpa = nor.nid/Sepliciologue*, nagon/Se[ii]* 5 (sepl.id, 0), inpurgg.id)
and,der.militae(recenter/out)

The component factory

- single factory function to create new components
- runs extra checks
 - are all arguments (parameters) correct?
 - is the component valid (level 1 validation for each component at build-time)?

```
>>> izh0 = component_factory("Izhikevich2007Cell")
ValueError: Validation failed:
- Izhikevich2007Cell (None): Required value v0 is missing
- Izhikevich2007Cell (None): Required value k is missing
- Izhikevich2007Cell (None): Required value vr is missing
- Izhikevich2007Cell (None): Required value vt is missing
- Izhikevich2007Cell (None): Required value vpeak is missing
- Izhikevich2007Cell (None): Required value a is missing
- Izhikevich2007Cell (None): Required value b is missing
- Izhikevich2007Cell (None): Required value c is missing
- Izhikevich2007Cell (None): Required value d is missing
- Izhikevich2007Cell (None): Required value C is missing
- Izhikevich2007Cell (None): Required value id is missing
```

NeuroML update -Paper and general updates

-The component factory

The component factory

· runs extra checks

· is the component valid (level 1 validation for each component at build-time)?

The component factory

```
>>> izh0 = component_factory(
    "Izhikevich2007Cell",
    id="izh2007RSO", v0=".60mV", C="100pF", k="0.7nS_per_mV", vr="-60mV",
    vt="-40ms", vpeak="35mV", a="0.03per_ms", b="-2nS", c="-50.0mV", d="100pA")

ValueError: Validation failed:
    - Izhikevich2007Cell (izh2007RSO): Value "-40mS" does not match xsd pattern restrictions:
    \[ [['^(-?([0-9]*(\\.[0-9]+)?)([eE]-?[0-9]+)?[\\s]*(V|mV))$']]]
```

NeuroML update

Paper and general updates

The component factory

The component factory

-New add method

· smart enough to know where the new element needs to go

New add method

 smart enough to know where the new element needs to go in the parent

```
izh0 = Izhikevich2007Cell(
   id="izh2007RS0", v0="-60mV", C="100pF", k="0.7nS_per_mV", vr="-60mV",
   vt="-40mV", vpeak="35mV", a="0.03per_ms", b="-2nS", c="-50.0mV", d="100pA")
nml_doc.izhikevich2007_cells.append(izh0)
```

VS

```
izh0 = nml doc.add(
   "Izhikevich2007Cell",
   id="izh2007RS0", v0="-60mV", C="100pF", k="0.7nS_per_mV", vr="-60mV",
   vt="-40mV", vpeak="35mV", a="0.03per_ms", b="-2nS", c="-50.0mV", d="100pA")
```

29/34

Inspect each component individually

```
izh0.info(show_contents=True)
Izhikevich2007Cell -- Cell based on the modified Izhikevich model in Izhikevich 2007. Dvnamical

→ systems in neuroscience, MIT Press

Please see the NeuroML standard schema documentation at
→ https://docs.neuroml.org/Userdocs/NeuroMLv2.html for more information.
Valid members for Izhikevich2007Cell are:
* b (class: Nml2Quantity_conductance, Required)
        * Contents ('ids'/<objects>): -2nS
* C (class: Nml2Quantity_capacitance, Required)
        * Contents ('ids'/<objects>): 100pF
* c (class: Nml2Quantity_voltage, Required)
        * Contents ('ids'/<objects>): -50.0mV
* d (class: Nml2Quantity_current, Required)
        * Contents ('ids'/<objects>): 100pA
* neuro_lex_id (class: NeuroLexId, Optional)
* metaid (class: MetaId, Optional)
* v0 (class: Nml2Quantity_voltage, Required)
        * Contents ('ids'/<objects>): -60mV
* id (class: NmlId, Required)
        * Contents ('ids'/<objects>): izh2007RS0
                                                                                               30/34
```

NeuroML update

2024-02-2

-Paper and general updates

Inspect each component individually

Inspect each component individually

nor, contumberDrus) HTGALL — Call based on the modified labilitatick model in labilitatich 2007, in nanounisane, RIT Press.

Emminer for InhibertableCOLL area (class: SollQuantity_conductance, Repaired) • Consume ('ide'/colperate): -265

Consume ("ide"/tobjertel): 500

tion: Sel Sport by yolnge, Septime
 Consume ("ide"/tobjertel): 30

Consume ("ide"/tobjertel): 30

* Corneta ('ida'/tokjurtar): 105pi r.lar.id (rlam: Heardarld, Sprimal) id (rlam: Hetald, Sprimal) rlam: Hel2(sarring.yolnaga, Repaired)

* Corners ("ide//cdgarter): ide2007800

Component Type info without creating a new component: ctinfo

```
>>> ctinfo("ExpOneSynapse")
ExpOneSynapse -- Ohmic synapse model whose conductance rises instantaneously by ( **gbase** *
→ **weight** ) on receiving an event, and which decays exponentially to zero with time course
→ **tauDecay**
Please see the NeuroML standard schema documentation at
→ https://docs.neuroml.org/Userdocs/NeuroMLv2.html for more information.
Valid members for ExpOneSynapse are:
* neuro_lex_id (class: NeuroLexId, Optional)
* gbase (class: Nml2Quantity_conductance, Required)
* metaid (class: MetaId, Optional)
* erev (class: Nml2Quantity_voltage, Required)
* notes (class: xs:string, Optional)
* id (class: NmlId, Required)
* properties (class: Property, Optional)
* annotation (class: Annotation, Optional)
* tau_decay (class: Nml2Quantity_time, Required)
```

NeuroML update

2024-02-2

-Paper and general updates

Component Type info without creating a new component: ctinfo

Component Type info without creating a new component, clinfo

we contribute the component of the component clinfo

we contribute the component of the component of the component

which is a component of the component of the component of the component

which is a component of the component of the

Where does this component type fit?

```
>>> ctparentinfo("HHRate")
Please see the NeuroML standard schema documentation at

→ https://docs.neuroml.org/Userdocs/NeuroMLv2.html for more information.

Valid parents for HHRate are:
* GateHHRates
        * forward_rate (class: HHRate, Required)
        * reverse_rate (class: HHRate, Required)
* GateHHRatesInf
        * forward_rate (class: HHRate, Required)
        * reverse_rate (class: HHRate, Required)
* GateHHRatesTau
        * forward_rate (class: HHRate, Required)
        * reverse_rate (class: HHRate, Required)
* GateHHRatesTauInf
        * forward_rate (class: HHRate, Required)
        * reverse_rate (class: HHRate, Required)
* GateHHUndetermined
        * forward_rate (class: HHRate, Optional)
        * reverse_rate (class: HHRate, Optional)
```

NeuroML update

L—Paper and general updates

Where does this component type fit?

Where does this component type fit?

per organización (Blanch).

Total promo de la companización del la companización de la companización del la companización d

Additions to make multi-compartment cell building easier

- set_init_memb_potential() instead of cell.biophysical_properties.membrane_properties.init_memb_potential
- add_channel_density()...
- add_segment, add_unbranched_segment_list...
 - also takes care of NeuroLex (now InterLex) Ids, used by NEURON to create "sections".
 - another hidden feature of NeuroML

NeuroML update

2024-02-2

-Paper and general updates

—Additions to make multi-compartment cell building easier

Sitions to make multi-compartment cell building easier

init_memb_potential() INSCEAD OF opposion in a properties and properties in a properties in a

_channel_density()...

 add_segment, add_unbranched_segment_list...
 also takes care of NeuroLex (now InterLex) ids, used by NEURON to create "sections".
 another hidden feature of NeuroML

oing/future work: not all to be done before paper

- more utils for other common, repeated tasks (model factory/templates?)
- update docs, add more tutorials/recipes

2024-02-2 -On-going/future work: not all to be done before paper

On-going/future work: not all to be done before paper

- more utils for other common, repeated tasks (model factory/templates?)
- update docs, add more tutorials/recipes
- GUI model building
 - Jupyter widgets based model inspection/modification/creation
 - NeuroMLLite GUI (NeuroML v3?)
 - · Will absorb all neuroConstruct functionality

NeuroML update

2024-02-2

-Paper and general updates

On-going/future work: not all to be done

going/future work: not all to be done before paper

ils for other common, repeated tasks (mode templates?)

date docs, add more tutorials/

GUI model building
 Jupyter widgets based model

NeuroMLLite GUI (NeuroML v37)
- Will absorb all neuroConstruct functionality

On-going/future work: not all to be done before paper

- more utils for other common, repeated tasks (model factory/templates?)
- update docs, add more tutorials/recipes
- GUI model building
 - Jupyter widgets based model inspection/modification/creation
 - · NeuroMLLite GUI (NeuroML v3?)
 - · Will absorb all neuroConstruct functionality
- · Complete NetPyNE UI interactions with NeuroML
 - · import works, but cannot modify model
 - · export needs to be implemented

NeuroML update

Paper and general

2024-02-2

-Paper and general updates

On-going/future work: not all to be done before paper

going/future work: not all to be done before paper

or other common, repeated tasks (miplates?)

tory/templates?) date docs, add more tutorials/

 Jupyter widgets based model inspection/modification/creation
 NeuroMLLite GUI (NeuroML v37)

Will absorb all neuroConstruct functionality
 Complete NetPyNE UI interactions with NeuroML import works, but cannot modify model
 export needs to be implemented

On-going/future work: not all to be done before paper

- more utils for other common, repeated tasks (model factory/templates?)
- · update docs, add more tutorials/recipes
- · GUI model building
 - Jupyter widgets based model inspection/modification/creation
 - NeuroMLLite GUI (NeuroML v3?)
 - Will absorb all neuroConstruct functionality
- Complete NetPyNE UI interactions with NeuroML
 - · import works, but cannot modify model
 - · export needs to be implemented
- Migrate all code to Python (so no more Java required)
 - · long term, multi-year project, requires another grant

On-going/future work: not all to be done before paper

oing/future work: not all to be done before paper

ils for other common, repeated tas 'templates?')

ry/templates?) ie docs. add more tutorials/re

 Jupyter widgets based model inspection/modification/creati

inspection/modification/creation
NeuroMLLite GUI (NeuroML v37)
Will absorb all neuroConstruct functionality

Complete NetPyNE UI interactions with NeuroML
 import works, but cannot modify model
 export needs to be implemented
 Migrate all code to Python (so no more Java required)