



## Introduction

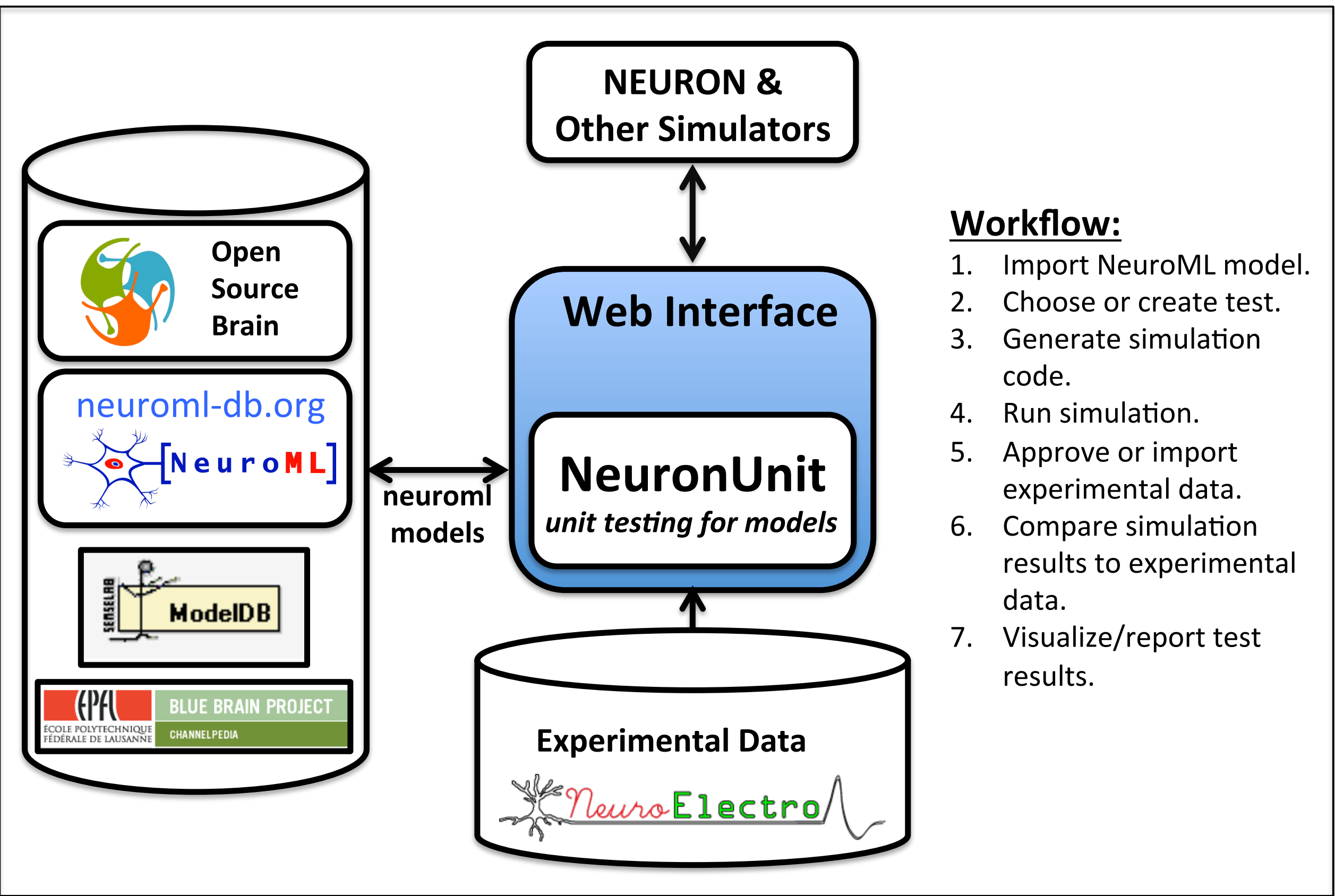
Computational models of neurons and neural circuits are abundant, but their agreement with experimental data is usually assessed informally, incompletely, or not at all. In principle, everyone wants models to accord with experimental data, but it can be tedious to make this a priority.

Where is all the good, machine-readable experimental data located?  
[neuroelectro.org](http://neuroelectro.org) and the [Allen Cell Types](http://allen-cell-types.org) databases

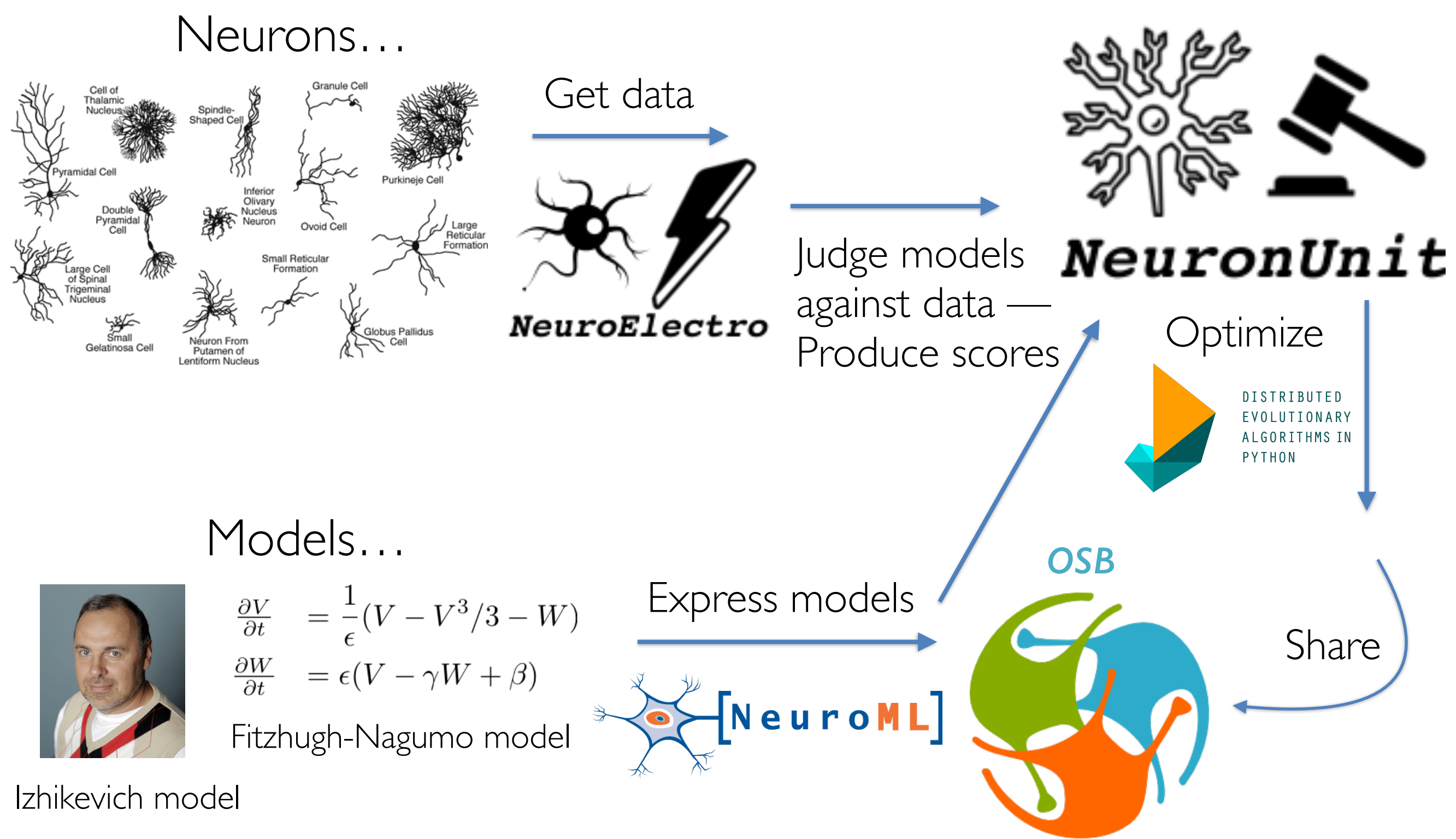
Who is going to digitize it and write “tests” for the model?  
[neuronunit](http://neuronunit.org) uses these resources to auto-generate tests using emerging standards

How can these tests be written to generalize across models?  
[sciunit](http://sciunit.org) separates interface from implementation

## Workflow



## A validated, custom model for every neuron in the brain



## Tools



## Approach

We apply the “unit-testing” approach of software development to the validation of scientific models against experimental data.

### Virtues:

- Write a test once and test all the models
- Implement model capabilities once and take all the tests
- Continuously measure the agreement between model and data

### Example

```
import quantities as pq
from neuronunit import tests as nu_tests, neuroelectro
neuron = {'nlex_id': 'nifext_50'} # Layer V pyramidal cell
tests = []

dataset_id = 354190013 # Internal ID that AIBS uses for a particular Scnn1a-Tg2-Cre
# Primary visual area, layer 5 neuron.
observation = aibs.get_observation(dataset_id, 'rheobase')
tests += [nu_tests.RheobaseTest(observation=observation)]

test_class_params = [(nu_tests.InputResistanceTest, None),
                      (nu_tests.TimeConstantTest, None),
                      (nu_tests.CapacitanceTest, None),
                      (nu_tests.RestingPotentialTest, None),
                      (nu_tests.InjectedCurrentAPWidthTest, None),
                      (nu_tests.InjectedCurrentAPAmplitudeTest, None),
                      (nu_tests.InjectedCurrentAPThresholdTest, None)]

for cls, params in test_class_params:
    observation = cls.neuroelectro_summary_observation(neuron)
    tests += [cls(observation, params=params)]

def update_amplitude(test, tests, score):
    rheobase = score.prediction['value']
    for i in [5, 6, 7]:
        print(tests[i])
        # Set current injection to just suprathreshold
        tests[i].params['injected_square_current']['amplitude'] = rheobase*1.01

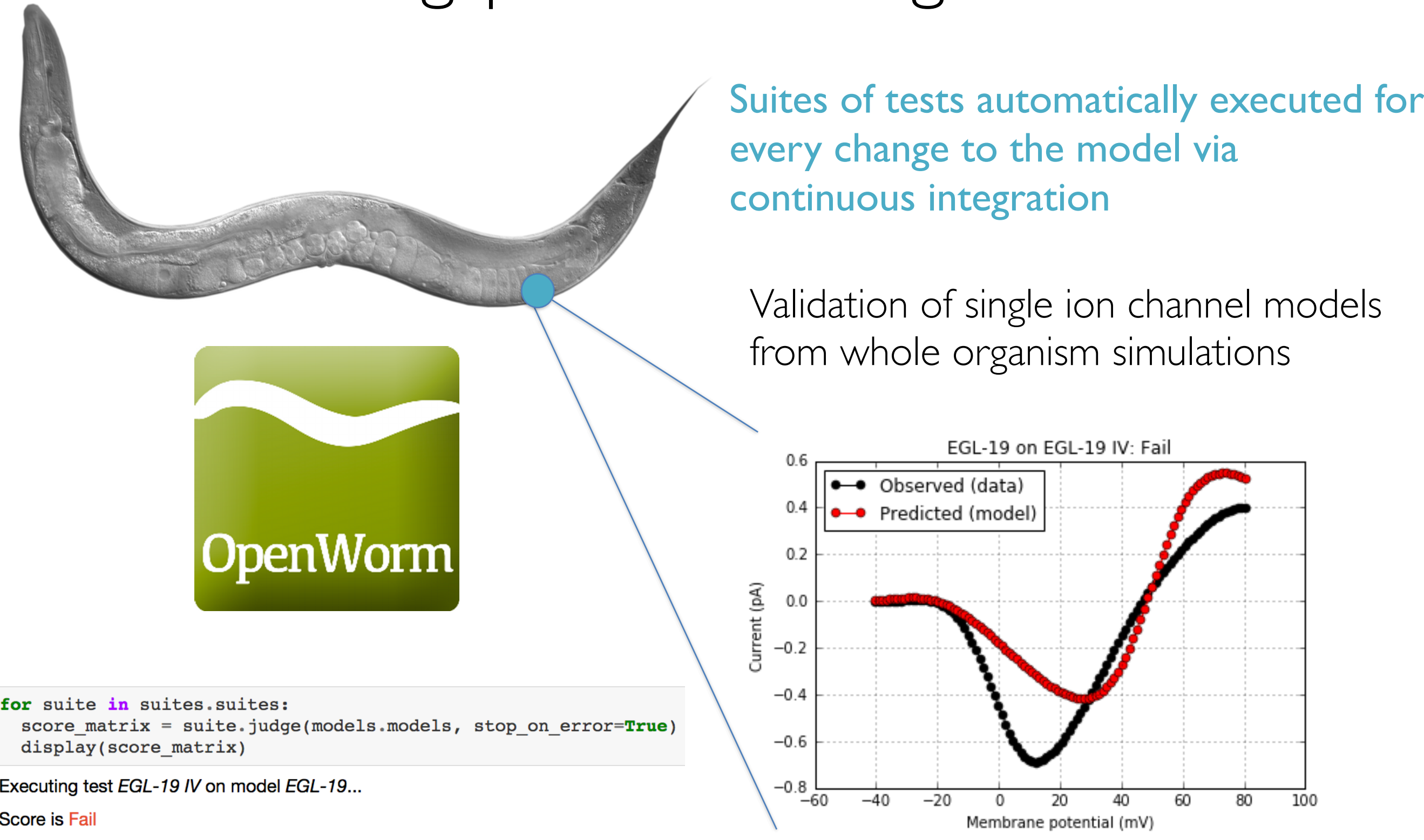
hooks = {tests[0]: {'f': update_amplitude}}
suite = sciunit.TestSuite("vm_suite", tests, hooks=hooks)
```

Test scores can be visualized as dynamic HTML outputs—in Jupyter notebooks or on the web—or from a command line.

	Mean	Rheobase	InputResistance	TimeConstant	Capacitance	RestingVm	APwidth	APAmplitude	APThreshold
V_rest=-75mV	0.426	Ratio = 1.39	Z = -0.94	Z = -0.23	Z = 0.97	Z = -0.99	Z = -0.73	Z = -1.30	Z = 1.76
V_rest=-70mV	0.532	Ratio = 1.00	Z = -0.85	Z = -0.12	Z = 0.78	Z = -0.23	Z = -0.69	Z = -1.43	Z = 1.98
V_rest=-65mV	0.510	Ratio = 0.66	Z = -0.71	Z = 0.04	Z = 0.59	Z = 0.54	Z = -0.64	Z = -1.56	Z = 2.19
V_rest=-60mV	0.440	Ratio = 0.40	Z = -0.51	Z = 0.28	Z = 0.41	Z = 1.31	Z = -0.60	Z = -1.70	Z = 2.40
V_rest=-55mV	0.393	Ratio = 0.20	Z = -0.20	Z = 0.69	Z = 0.25	Z = 2.07	Z = -0.55	Z = -1.83	Z = 2.61
V_rest=-50mV	0.311	Ratio = 0.07	Z = 0.30	Z = 1.42	Z = 0.13	Z = 2.84	Z = -0.49	Z = -1.96	Z = 2.82

Test suites can be run automatically every time a model is updated through free cloud infrastructure (Travis-CI) or local clusters.

## Validating pieces of larger models



## Discussion

[Neuronunit](http://neuronunit.org) is a Pythonic library for rapidly testing neuron, circuit, and ion channel models against ever-increasing quantities of available experimental data.

We automate test construction using neuron physiology data obtained from open databases or your computer.

We aim test every model that can be expressed in NeuroML.

Visualize test scores for every parameterization of every model against all relevant experimental data: both during development and after publication. Formal comparison of models becomes possible!

## References

1. Omar et al. "SciUnit: Collaborative Infrastructure for Test-Driven Scientific Model Validation", 2014 (10.1145/2591062.2591129).
2. Tripathy et al. "NeuroElectro: A Window to the World's Neuron Electrophysiology Data", 2014 (10.3389/fninf.2014.00040).
3. Gleeson et al. "A Language for Describing Data Driven Models of Neurons and Networks with a High Degree of Biological Detail", 2010 (10.1371/journal.pcbi.1000815, <http://neuronunit.org>).
4. <http://sciunit.scidash.org>
5. <http://neuronunit.scidash.org>
6. <http://opensourcebrain.org>
7. <http://celltypes.brain-map.org>
8. <http://openworm.org>

This research was funded by NIMH (R01MH106674) and NIBIB (R01EB021711).