
ChannelWorm Documentation

Release 0.1

Vahid Ghayoomi

May 01, 2015

1	Introduction to ChannelWorm	3
1.1	Introduction	3
1.2	ChannelWorm	4
1.3	Objectives	4
1.4	Some use cases	4
1.5	Implementation	5
2	Information Management	7
2.1	Data Collection	7
2.2	Data Management	8
2.3	Data Representation	8
2.4	Update Management	8
3	Indices and tables	9

Contents:

Introduction to ChannelWorm

1.1 Introduction

Ion channels are pore-forming proteins, found in the membranes of all cells. Their known functions include establishing a resting membrane potential, shaping electrical signals like action potentials, gating the flow of ions across the cell membrane, controlling cell volume, and regulating the net flow of ions across epithelial cells of secretory and resorptive tissues. Ion channels are considered to be one of the two traditional classes of ionophoric proteins, with the other class known as ion transporters (including the sodium-potassium pump, sodium-calcium exchanger, and sodium-glucose transport proteins, amongst others)¹.

Most plasma membrane ion channels in the nervous system come in four distinct topologies which likely evolved independently²:

- The voltage-gated family of potassium, sodium and calcium channels
- The cysteine-loop family of ligand-gated ion channels
- Ionotropic glutamate receptors
- P2X and ASIC channels

The *C. elegans* genome codes for members of all the main families of ion channels. Within specific families, there are some individual members missing, the most notable being sodium-gated ion channels, P2X channels and HCN channels.

Ion channels, and associated regulatory machinery have been identified in genetic screens of various processes in *C. elegans* such as locomotion, gonad development, mechanosensory and osmotic avoidance behavior, chemotaxis and thermotaxis, resistance to fluoride ions, defecation, egg laying, toxin and heavy metal sensitivity, programmed cell death, neuronal degeneration, and abnormal catecholamine levels.

A wealth of powerful molecular tools and online databases are available to study ion channels and transporter functional genomics in *C. elegans*. Conceptually, functional genomic studies can be divided into two approaches.

The gene-driven approach focuses on identification, cloning, and characterization of genes. Gene characterization involves expression and structure-function studies and determination of where and when a gene is expressed in the organism.

The phenotype-driven approach seeks to determine the physiological roles of genes from the cellular to whole animal level, and to identify the mechanisms that regulate protein function³.

¹ Hille, Bertil (1984). *Ionic Channels of Excitable Membranes*.

² Hobert O. The neuronal genome of *Caenorhabditis elegans* (August 13, 2013), WormBook, ed. The *C. elegans* Research Community, WormBook, doi/10.1895/wormbook.1.161.1, <http://www.wormbook.org>.

³ Kevin Strange. From Genes to Integrative Physiology: Ion Channel and Transporter Biology in *Caenorhabditis elegans*, *Physiological Reviews*, Apr 2003, 83 (2) 377-415; DOI: 10.1152/physrev.00025.2002

1.2 ChannelWorm

Due to its simplicity and wealth of data available, *C. elegans* is an ideal organism for modeling ion channels and related functionalities. The aim of the **ChannelWorm** is to integrate information and tools related to modeling ion channels in *C. elegans* for the [OpenWorm Project](#). Here we use the Hodgkin Huxley approach for modeling ion channel kinetics and simulating action potentials in excitable cells in *C. elegans* (see [this tutorial](#)).

1.3 Objectives

- **Information management**
 - Integrate and structure data related to ion channels in *C. elegans*, from genotype to phenotype
 - Develop APIs for accessing data
 - Keep data up-to-date
- **Ion channel modeling**
 - Build Hodgkin-Huxley models for ion channels based on experimental patch clamp studies
 - Estimate kinetics and build models for ion channels with no patch clamp data available (based on homologous channel types)
 - Create verification & validation tests to prove matching of the models with experimental data
- **Setting up a simulation environment**
 - Simulate the computational analysis phase of a patch clamp experiment
 - Simulate and run customized versions of ion channel(s) in cell(s)
 - Check if the simulation fits the biological boundaries

1.4 Some use cases

- I want to know which type of ion channels we have in *C. elegans*!
- I want to know in which cell which gene encodes which protein/subtype of which ion channel!
- I want to view, build, or validate a model of an ion channel
- I want to check references that data or models are based on
- I have some graphs/data from a patch clamp experiment and want to build a HH model
- I want to have a customized version of an ion channel model (e.g. of a mutant)
- I want to run my customized model of ion channel(s) (along with other ion channels) in a simulation environment
- I want to simulate a patch clamp experiment with known mutations in genes encoding ion channels in *C. elegans*
- I want to simulate ion channel diseases (channelopathies), and investigate defects in neuromuscular transmission, *C. elegans* movement, etc.

1.5 Implementation

All the models are generated in [NeuroML2](#) format and all the simulations in [LEMS](#), and the verified models can be run in the [Geppetto](#) simulation platform.

1.5.1 References

Information Management

An information management system is needed in order to:

- Integrate data related to ion channels in *C. elegans* from different sources
- Structure data for better representation
- Ease access to the data through scripts/APIs
- Keep data up-to-date

Data includes:

- Description about the ion channel
- Genetics, gene ontology, sequences, and locations
- Proteins, structures, and homology information
- Expression information and patterns in neurons/cells
- Phenotypes and functionalities
- Interaction network
- Channelopathies (ion channel diseases), known mutations, and pharmacological studies
- Available models and simulations
 - Model type (from patch clamp studies, or estimation from known homologs)
 - Experimental conditions (cell type, temperature, Reversal potential, age of the organism, etc.)
 - Channel Properties (conductance, activation/inactivation parameters, etc.)
 - Cell and synapse properties (membrane capacitance, surface area, external ion concentration, etc.)
 - Graphs demonstrating kinetics
 - Representation files for each model/simulation (NeuroML and LEMS respectively)
- Evidence(s) for all the assertions

2.1 Data Collection

Biological information about each ion channel was obtained from the [WormBase](#) database which is available in [this spread sheet file](#). Every tuple in the spread sheet has some links to its WormBase entry that could be used for more information.

In the *ion channels with properties* sheet, you can find properties, such as type, subtype, kinetic properties, etc. for every ion channel.

The *references* column in this table contains links which reference publications on *Mendeley*.

2.2 Data Management

To keep data in a structured and easy-to-access format, we use the [PyOpenWorm](#), from the OpenWorm project. The database stores data for generating model files along with annotations describing the origins of the data. PyOpenWorm utilizes [RDF](#) which facilitates integration of data from disparate sources. For example, it is possible to ask PyOpenWorm to list all the ion channels of a muscle cell named MDL08, with evidence:

```
muscle = PyOpenWorm.Muscle('MDL08')
muscle.channels()
['EGL-19', 'SHK-1', 'SHL-1']
#look up what reference says this muscle has an ion channel EGL-19
muscle.get_reference(0, 'EGL-19')
['http://dx.doi.org/10.1083%2Fjcb.200203055']
```

PyOpenWorm allows easy access to data about ion channels, either for human or machine reading.

For full usage of the PyOpenWorm API, take a look at [this document](#)

2.3 Data Representation

Final models are represented in [NeuroML2](#) format which is an XML based description language that provides a common data format for defining and exchanging descriptions of neuronal cell and network models. You can find an example for this representation [here](#). Single channel kinetics are described by a NeuroML file, which could then be integrated with other channels and properties in a [LEMS](#) format for final simulations ([LEMS sample file](#)). For more information see [this tutorial](#).

In addition to [jNeuroML](#), these representation files could be used in [different tools](#) that support neuronal simulation. [Here](#) you can find an example for a simulation of muscle cell ion channels in *C. elegans*.

Finally an [XML file](#) could be generated for running the simulation under the [Geppetto simulator](#).

2.4 Update Management

In order to get informed about new publications about ion channels in *C. elegans* and also any update on existing data, an update management system should be implemented.

2.4.1 Current State

You can find open/closed issues for this topic [here](#)

Indices and tables

- *genindex*
- *modindex*
- *search*