Principles of computational modelling

Lubica Benuskova

Lecture 1

Outline of the course

- Introduction to basic methods of computational modelling of biological neurons and neural circuits.
- It is designed for people from a wide range of backgrounds from the biological, physical and computational sciences.
- We will learn basics of the programming in NEURON, which is a free software for biological neuron and neural networks modelling and simulation.
- Assessment consists of 2 programming assignments including the reports on results (40%), 1 written assignment (20%), and a final written exam (40%) – more info and download at the webpage.

http://dai.fmph.uniba.sk/courses/comp-neuro/

Lectures

PDF	Date	Topic	Reading
Lecture 01	20/02	Principles of computational modelling	Chap. 1 & Chap. 2, pp. 13-20
Lecture 02	27/02	NEURON: building a soma, Hodgkin-Huxley model	Chap. 3, pp. 47-58 & chap.3, pp. 60-71
Lecture 03	05/03	Theory of dendrites	Chap. 2, pp. 29-41 & Appendix B
Lecture 04	12/03	NEURON: adding dendrites to the soma	
Lecture 05	19/03	NEURON: creating more neurons	
Lecture 06	26/03	NEURON: creating dendritic tree(s)	One rule to grow them all by H. Cuntz et al (journal paper)
No lecture	02/04	Easter Holiday	
Lecture 07	09/04	NEURON: connecting neurons	Chap. 9, pp. 226-233
Lecture 08	16/04	Theory of synapses and synaptic plasticity	Chap. 7, pp. 172-176 and pp. 189-194
Lecture 09	23/04	NEURON and NMODL: ion channels	Chap. 5, pp. 96-105, Chap. 5, pp. 105-120 & 131
Lecture 10	30/04	Computational Neurogenetic Modelling	Book chapter and genetic disorders of the brain
Lecture 11	07/05	Revision: topics for exam	Latest Research and Reviews

Literature

- **TEXTBOOK:** Sterratt D., Graham B., Gillies A., and Willshaw D. (2011) Principles of Computational Modelling in Neuroscience, Cambridge University Press, Cambridge, U.K.
- Gillies A. and Sterratt D. (2012) NEURON Tutorial – available online - click here
- SCHOLARPEDIA - the free online encyclopedia of computational neuroscience - click here

Resources

- For the free download of software NEURON click here
- The Blue Brain Project
- ModelDB - computational neuroscience models

The third methodology of (neuro)science

- **Theory** is the results of abstract thinking and involves generalized explanations of how nature works. It is the way to interpret results of experiments and to design new experiments.

- Without **experiments** there is no point in theorizing--theory becomes speculation. Theory is corroborated/supported or falsified by experiments.

- **Computational modellig** allows more elaborate tests of theory than would be possible by experiment, thus enabling new ways to refine the theory. It also allows a much more sophisticated analysis of experiments than theory alone could provide.
  - Computational modelling has become the third methodology of science along with theory and experiment.

Basic steps in creating the model

- **Step 1**: Clearly state the assumptions, on which the model will be based. These assumptions should verbally describe the relationships among the quantities to be studied.

- **Step2**: Completely describe the quantities, i.e. variables and parameters to be studied in the model.

- **Step 3**: Use the assumptions formulated in Step 1 to derive equations relating the quantities in Step 2.

- **Step 4:** use mathematical knowledge and/or computer program to solve the equations and make predictions about the evolution of studied quantities in the future.

Step 1: assumptions

- Assumptions should describe what we think about **relationships** between variables we want to model.
  - E.g., we assume there are 2 types of inputs to each neuron: excitatory and inhibitory. They add up linearly, i.e., input = excitation – inhibition.

- The quality of assumptions determines the **validity** of the model and the situations to which the model is relevant.

- We must avoid "hidden assumptions" that make the model seem mysterious or magical.
  - i.e., we include something in the equation that makes the model work but corresponds to nothing in reality.

Step 2: defining the quantities

- Independent variables
  - The independent variable is almost always time (t). Time t is independent of any other quantity in the model.

- Dependent variables are the quantities that are functions of the independent variable.
  - For example, membrane voltage changes over time, $V = V(t)$, i.e. we say voltage is a function of time.

- Parameters are quantities that do not change with time (or any other independent variable), but their value have a profound influence on the behaviour of the dependent variables.

Variables and parameters

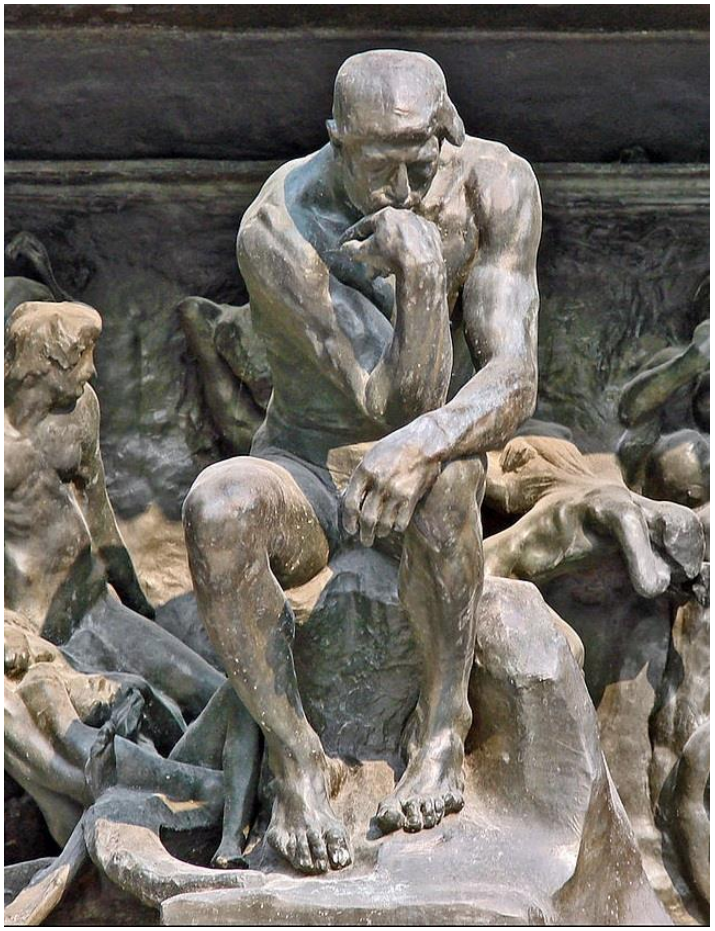
- The goal of a model is to describe the **behaviour of the dependent variable** as the independent variable changes.
  - For example, we may ask whether the dependent variable (e.g. voltage) increases or decreases with time, or whether it oscillates or tends to a limit.

- Observing how the behaviour of the dependent variable changes when we **change the values of parameters** can be the most important aspect of the study of a model.
  - For example, we may ask how the voltage evolves when the strength of excitatory inputs is the same as the strength of inhibitory inputs or how voltage will evolve when these strengths differ, *while the values of strengths of excitatory and inhibitory inputs are the parameters of the model*.

Step3: the most difficult part

- The hardest part in using the maths or computational algorithms to study phenomena is the translation from real life into **mathematical and / or computational formalism**.

- It is difficult because it involves
  - knowledge of maths and programming;
  - the **conversion of assumptions into mathematical equations** and to computational algorithms.



August Rodin: The Thinker

Equations, i.e., something = something

- We know that the dependent variable changes over time. Thus, we are looking for a "rate of change of …" or "rate of increase of …" a dependent variable over time.

- Mathematically, the rate of increase of something corresponds to the thing called "derivative", being written as:

$$\frac{dV}{dt}$$

  - Where that "something" is for example membrane voltage, *V*.

- In other words, the rate of change is synonymous with derivative.

Derivative = something

- We want to express what is the rate of change, i.e. what the derivative amounts to. That is, we want to know what is the "something" in the following equation:

$$\frac{dV}{dt} = \text{something}$$

- The phrase "A is proportional to B" means $A = k\,B$, where k is the so-called proportionality constant. So, to be mathematically correct we have to write the above equation as

$$\frac{dV}{dt} = k\ \text{something}$$

- i.e. the rate of change of voltage is proportional to something. Proportionality constant ***k* is the parameter** of the model.

Derivative = *k* something

- The equation: $$\frac{d(\text{dependent variable})}{dt} = k \text{ something}$$
- is at the core of all models in computational neuroscience.
- Dependent variable can be anything from voltage, current, field potential, number of synapses, concentration of neurotransmitter, speed of learning, number of memorized items, etc.
- The biggest challenge is to come up with "something"…

Concrete example: population growth

- We can develop and study mathematical models of systems that change (evolve) over time, for instance the number of rabbits.

- Except time itself, the number of rabbits depend on other variables too.
  - For example, the changes in population of rabbits depend on the amount of food, number of predators (hawks, foxes, etc.), rabbit diseases, etc.

- In order to make a model of evolution of rabbit population simple enough to understand, we have to make simplifying assumptions and neglect the things we know nothing about or not enough.

The simplest model of population growth

- **Step 1**: The assumption: the rate of growth of population depends only on the size of the population and nothing else. The more rabbits there are the more offspring they have.

- **Step 2**: definition of quantities:
  - t = time (independent variable)
  - N = population size (dependent variable), i.e., $N = N(t)$
  - k = proportionality constant (parameter)

- How about the units of these quantities? They obviously depend on species and environment. If we are talking about population of people, then t would be years and N would be millions. If talking about rabbits, then t would be months and N thousands.

Mathematical equation for the growth

- Step 3: let's express our assumption as an equation. The rate of growth of the population is the derivative:

$$\frac{dN}{dt}$$

- Being proportional to the population size is expressed as a product of the proportionality constant k and the population size N , i.e., $k\ N$.
  - Hence our assumption is expressed as the (differential) equation:

$$\frac{dN}{dt} = k\ N$$

- Read as: Derivative of N according to time t equals k times N.

Step 4: solution

- In order to be able to predict the change of rabbit population we must know the size of population at time zero N_0 , the so-called initial condition, i.e.

$$N(t_0) = N_0 > 0$$

- Next, we want to predict the value of N at various times in the future, e.g. $N(10)$ or $N(100)$, i.e. we want to know concrete values of quantity $N(t)$ for each value of t, that satisfy this equation:

$$\frac{dN}{dt} = k\ N$$

- To find a solution to this equation means to find a function $N(t)$ whose derivative is the product k with $N(t)$.

Analytical solution

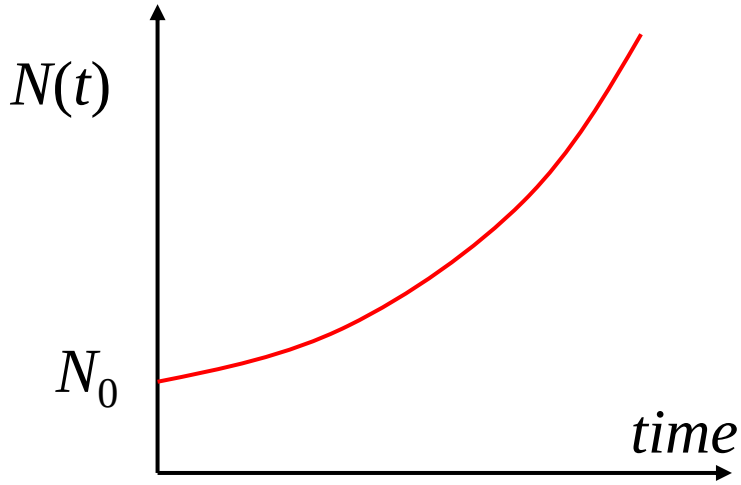
- The solution of the differential equation: $\frac{dN}{dt} = k\ N$

- is an exponential function

$$N(t) = N_0 e^{kt}$$

- Where N_0 is the initial population

$$N(t_0) = N_0 > 0$$

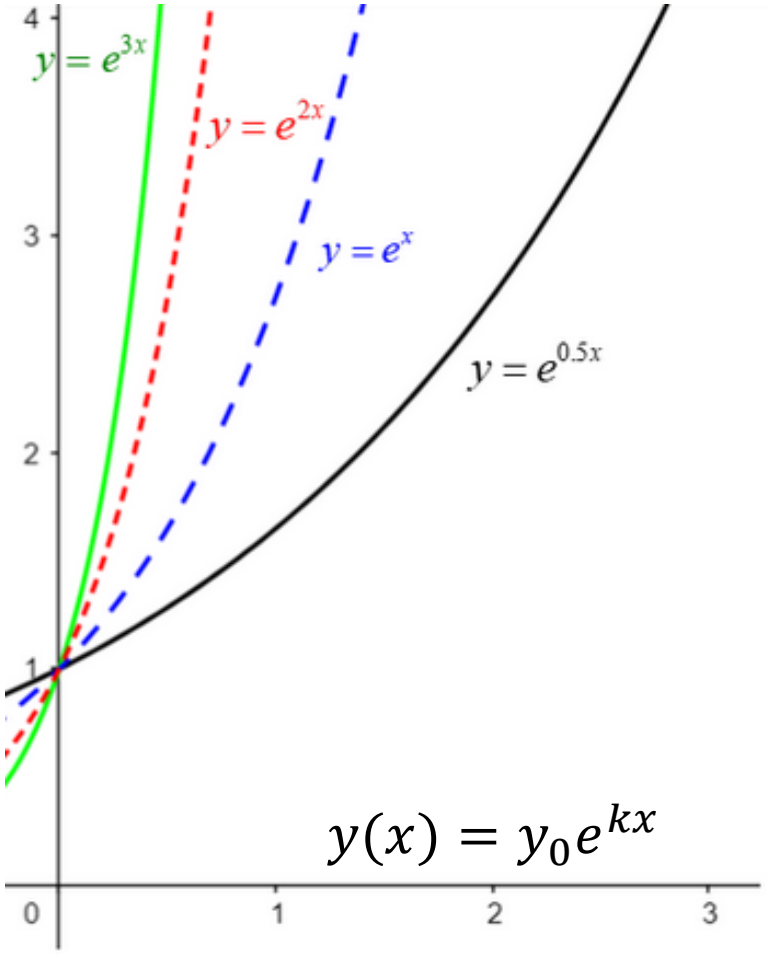


- Now we can make predictions about the size of the population in the future provided we know the value N_0 and k. The value of k can be derived by fitting the model to real data. We simply try different values of k and see which one yields the best fit with the data.

Computational model: changing parameter value

- Once we develop a mathematical or computational model, we have to compare predictions of the model with data from the system to check for empirical support or falsification.

- If the model prediction and the data disagree, then we first try different values of parameters, if this does not help then we change our assumptions and re-formulate the model.



Making the model more realistic

- What would happen if there are two species, one of which is a predator and one of which is the predator's prey?

- Our theory might state that:
  - (1) the prey population N grows in proportion to its size but declines as the predator population P grows and eats it; and
  - (2) the predator population P grows in proportion to its size and the amount of the prey N but declines in the absence of prey.

Prediction no. 1

- From this theory we can predict that the prey population N grows initially (provided there is enough food and no diseases).
- As the prey population N grows, the predator population P can grow faster.
- As the predator population P grows, this limits the rate at which the prey population N can grow.
- At some point, an equilibrium is reached when both predator P and prey sizes N are in balance.

Alternative prediction

- We might wonder whether there is a second possible prediction from the theory.
- Perhaps the predator population P grows so quickly that it is able to make the prey population N extinct. Once the prey has gone $N = 0$, the predator is also doomed to extinction $P = 0$.
- Now we are faced with the problem that there is one theory but two possible conclusions; the theory is logically inconsistent.
- What will happen to the population of prey N and predators P? Will they become extinct, or will they stabilize? Can the mathematical model help us?

Mathematical model (Lotka and Volterra 1925)

- For a prey population with size N(t) and a predator population with size P(t), it is assumed that

- (1) the prey population grows in proportion to its size N and declines in proportion to the rate at which predator and prey meet (assumed to be the product of the two population sizes, NP), i.e.:

$$\frac{dN}{dt} = a\,N - b\,NP$$

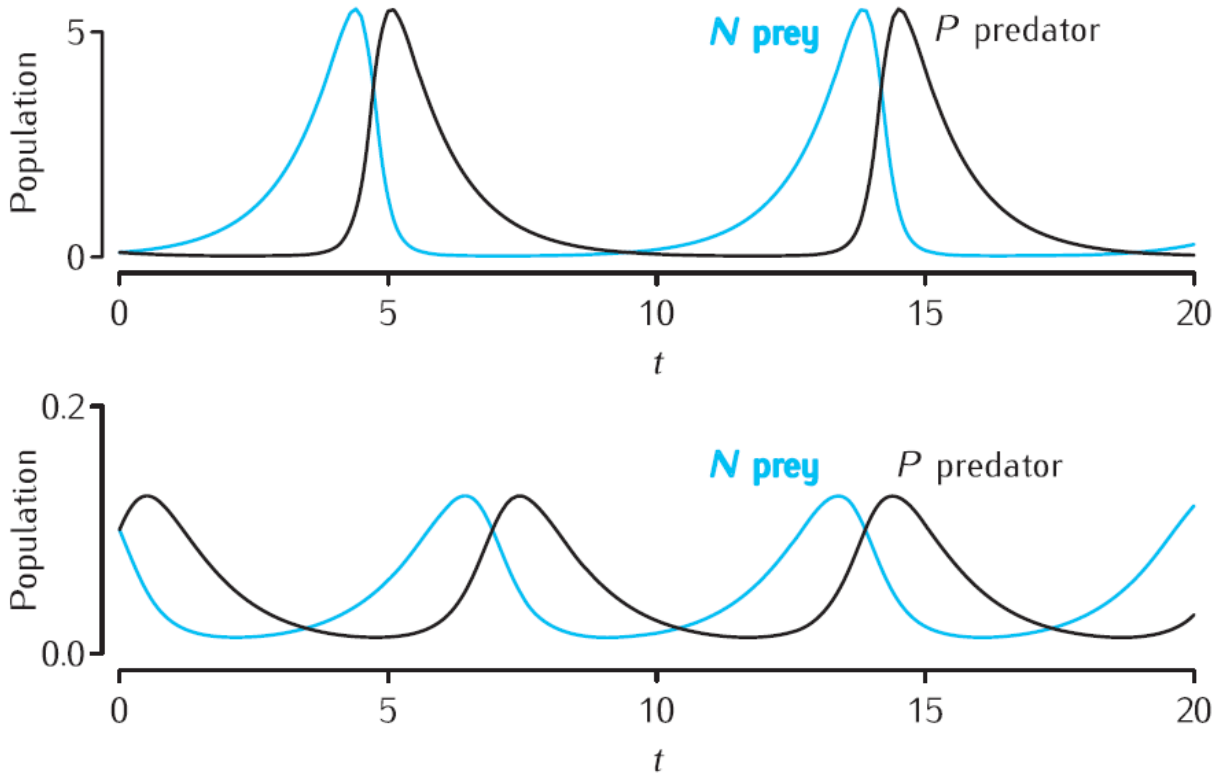
- (2) at the same time, there is an increase in the predator population size P in proportion to NP and decline in the absence of prey:

$$\frac{dP}{dt} = c\,NP - d\,P$$

- The parameters a, b, c and d are constants.

Mathematical model (Lotka and Volterra 1925)

- Behaviour of the Lotka–Volterra model of predator–prey interactions, with parameters $a = b = c = d = 1$ (upper graph), and $a = 1,\ b = c = 20, d = 1$ (lower graph).



Comments on the solution

- It turns out that neither of our guesses was correct. Instead of both species surviving in equilibrium or going extinct, the predator and prey populations oscillate over time.

- At the start of each cycle, the prey population grows. After a lag, the predator population starts to grow, due to the abundance of prey. This causes a sharp decrease in prey, which almost causes its extinction, but not quite. Thereafter, the predator population declines and the cycle repeats.

- It might now seem obvious that oscillations would be predicted by the theory. However, the step of putting the theory into equations (model) was required in order to reach this understanding.

How do we fix the model parameter values?

- Step 1: Fix the known parameter values and make educated guesses for the remaining unknown parameter values.

- Step 2: Use the model to simulate experiments and/or observations, producing model data.

- Step 3: Compare the model data with experimental data.

- Step 4: Adjust one or more parameter values and repeat from Step 2 until the simulated data sufficiently matches experimental data.
  - We simply try different values of parameters and see which ones yield the best fit with the data
  - It is recommended to change only one parameter value at a time, in order to distinguish, which parameter affects the results and how.

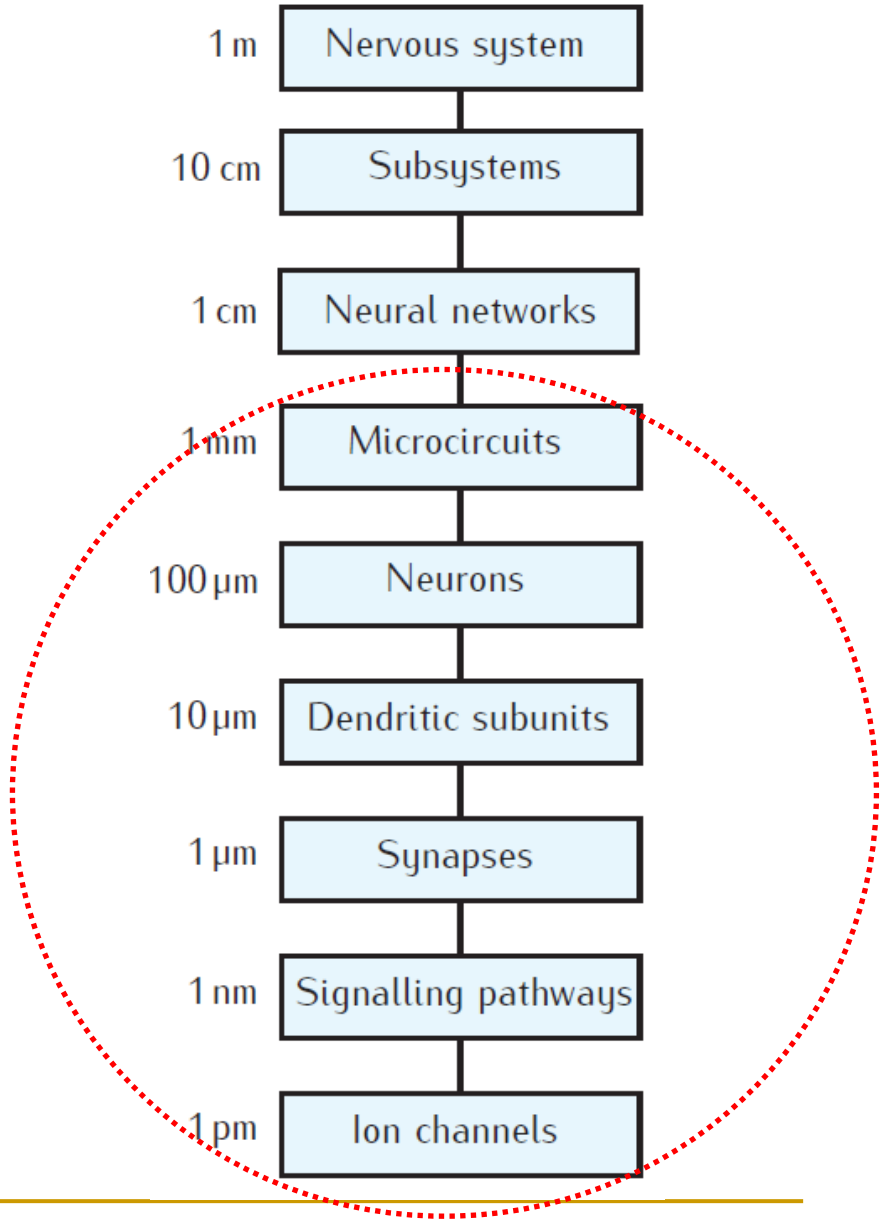
If we have several alternative models

- The aim is to explain a phenomenon with the fewest assumptions and select the simplest model to predict data.
  - **Simplicity and elegance in problem-solving**.

- Occam's Razor: when faced with competing hypotheses or explanations, the one that requires the fewest assumptions, entities, or complexities should be preferred.
  - This principle of parsimony is attributed to 14th-century English theologian William of Ockham.

Computational Neuroscience

- To understand the nervous system requires an understanding at many different levels.
- At each of these levels there are detailed computational models for how the elements at that level function and interact.
- In this course, it'll be neurons, networks of neurons, synapses and ions involved in signalling pathways inside neurons.



Simulation environment NEURON

- A free open-source software developed at Duke and Yale by Michael Hines and Ted Carnevale available at https://neuron.yale.edu/neuron/
- Used world-wide to build and simulate neurons with realistic dendritic trees and biological networks of hundreds of neurons. More than a thousand of scientific publications reported work that was done with NEURON.
- NEURON is actively developed and supported, with new standard releases each year, supplemented by bug fixes as needed. Available for number of operating systems (i.e., Linux, Windows and MacOS).
- ModelDB website provides a free location for storing and retrieving computational neuroscience models for sharing.

Simulation environment GENESIS

- GENESIS (the GEneral NEural SImulation System) is freely available at http://www.genesis-sim.org/GENESIS/
- GENESIS was originally developed in the laboratory of Dr. James M. Bower at Caltech.
- GENESIS was the first broad scale modeling system in computational biology and is based on Unix / Linux.
- But wasn't open source so it lost the race with NEURON.
- Now it's open source too.

Plan

- Next lecture, I will start to teach you how to run simulations in NEURON.
- We will also build a soma with NEURON that implements the Hodgkin-Huxley model.
- After the next lecture you will have your first code in NEURON.
- During the semester, we will make this code more complex, thus simulating more complex neural behaviour and we will also build a small network of neurons.