Journal of Theoretical and Computational Nanoscience, vol.1 (1) American Scientific Publisher, 2004, in print

Computational Neurogenetics

Nikola Kasabov[*] and Lubica Benuskova

Knowledge Engineering and Discovery Research Institute

School of Information Technology

Auckland University of Technology

Auckland, New Zealand

Address: AUT Technology Park, 581-585 Great South Road, Penrose, Auckland, New Zealand

Phone 64 9 917 9506

Fax 64 9 917 9501

Emails: nkasabov@aut.ac.nz; lbenusko@aut.ac.nz

WWW: http://www.kedri.info

[*] The author to whom correspondence regarding the manuscript should be directed.

Abstract

The aim of the paper is to introduce the scope and the problems of a new research area called *Computational Neurogenetics* (CNG), along with some solutions and directions for further research. CNG is concerned with the study and the development of dynamic neuronal models integrated with gene models. This area brings together knowledge from various science disciplines, such as computer and information science, neuroscience and cognitive study, genetics and molecular biology. A computational neurogenetic model is created to model a brain function or a brain disease manifestation, or to be used as a general mathematical model for solving complex scientific and engineering problems. The CNG area goes beyond modelling simple relationship between a single gene and a single neuronal function or a neuronal parameter. It is the interaction between hundreds and thousands of genes in a neuron and their relationship with the functioning of a neuronal ensemble and the brain as a whole (e.g., learning and memory, speech and vision, epilepsy, mental retardation, aging, neural stem cells, etc.). The CNG models constitute a *second generation* of neural network models that are closer to biological neural networks in their complex symbiosis of neuronal learning dynamics and molecular processes. Concrete models are presented as examples – evolving connectionist systems (ECOS) with evolutionary parameter optimisation and the CNG model of a class of spiking neural network ensembles (CNG-SNN).

Keywords: computational neurogenetics; neural networks; gene regulatory networks; brain study; adaptive learning; computational modelling.

1. Introduction

Neuroscience, along with the information and mathematical sciences, have developed a variety of theoretical and computational models to model complex brain functions [1]. Along with this development, artificial neural networks – computational models that adopt principles from the nervous system, have been developed to become powerful tools for learning from data and generalisation [2-6]. Artificial neural networks have been applied to a large amount of complex problems such as classification, prediction, diagnosis, planning, not only for brain- and molecular data modelling [7-9], but across all disciplines of science and engineering [10].

With the recent advancement of the genetic research and with the successful sequencing of the human genome and other genomes, more information is becoming available about the interaction between brain functions and genes, about genes related to brain diseases (e.g. epilepsy [11], mental retardation [12], etc.) and about gene-based treatment of them [13]. It is well accepted now that brain functions are better understood and treated if information from molecular and neuronal level is integrated as shown in Table 1.

[Table 1]

For this purpose, computational models that combine genetic and neuronal information are needed for modelling and prognosis. Such models are called here *computational neurogenetic (CNG) models*. The CNG area goes beyond the study of a simple relationship between a single gene and a single neuronal function. Rather it is concerned with the study and modelling of the interactions between hundreds and thousands of genes in a neuron and their relationship with the functioning of a neuronal ensemble and the brain as a whole. The paper presents principles of computational neurogenetic modelling, problems and some directions for further research. It is believed that integrated molecular and neuronal information processing models will be an important part of the theoretical and computational nanoscience of the future.

The material is presented as follows. In section two a brief introduction to the principles of information processing in the nervous system is given. Section 3 presents the basics of molecular processes in a cell (neuron). Section 4 presents some biological facts of neurogenetic information processing in the brain. Section 5 introduces the principles of a general CNG model. Section 6 presents basic principles of evolutionary computation (EC) and of genetic algorithms (GA) in particular along with their use for the optimisation of the parameters of a CNG model. Section 7 presents a simple CNG model of evolving connectionist systems (ECOS) [14].

2. A Brief Introduction to the Information Processes in the Brain

2.1. Organisation of the Brain

It is estimated that there are 10^{11}–10^{12} of neurons in the human brain [15]. Three quarters of neurons form a 4–6 mm thick cerebral cortex that constitutes a heavily wrinkled brain surface. Cerebral cortex is thought to be a seat of cognitive functions, like perception, imagery, thinking, etc. The cortex cooperates with evolutionary older subcortical nuclei that are located in the middle of the brain, in and around the so-called brain stem (Fig. 1).

[Figure 1]

Subcortical structures and nuclei include thalamus, basal ganglia, hypothalamus and dozens of other groups of neurons with more or less specific functions in modulating the state of the whole brain. For example, the input from all sensory organs comes to the cortex preprocessed in thalamus. All brain parts, either cortical or subcortical, are directly or indirectly heavily interconnected, thus forming a huge recurrent neural network. In the parietal-temporal-occipital association cortex, sensory and language informations are being associated. Memory and emotional informations are associated in the limbic association cortex (internal and bottom portion of hemispheres). The prefrontal association cortex takes care of all associations, evaluation, planning ahead and attention.

2.2. Synapses and Learning in the Central Nervous System

Neuro-information processing in the brain depends upon the properties of processing units – neurons. A neuron (Fig. 2) receives and sends out electric and chemical signals. The place of signal transmission is a *synapse*. In the synapse, the signal can be nonlinearly strengthened or weakened. The strength or efficacy of synaptic transmission is also called a *synaptic weight*. One neuron receives and sends out signals through 10^3 to 10^5 synapses. Dendrites (i.e. numerous bushy cell extensions) and soma constitute the input surface. Electrical signals transmitted by synapses can have a positive or negative electric sign. In the former case, we speak about *excitatory* synapses, and in the latter case about *inhibitory* synapses. Most of excitatory synapses are formed on *spines*, tiny extensions on dendrites. Spines are very important devices in relation to learning and memory.

[Figure 2]

When the sum of positive and negative contributions (signals) weighted by synaptic weights is larger than a particular value, called the excitatory threshold, a neuron fires, that is, it emits an output signal called a *spike* (Fig. 3). A spike is also called an action potential or a nerve impulse. The output frequency of a spike train (from 1 to10^2 Hz) is proportional to the overall sum of positive and negative synaptic contributions. Spikes are produced at the initial segment of an *axon* (the only neuronal output extension). Then they quickly propagate along the axon towards other neurons within a network (propagation speed is 5−100 m/s). At its distant end, an axon makes thousands of branches, each of which is ended by a synaptic terminal (bouton).

[Figure 3]

A synapse consists of a presynaptic terminal (bouton), synaptic cleft and postsynaptic membrane (Fig. 4). In the presynaptic terminal, there are dozens of vesicles filled with molecules of neurotransmitter (NT) ready to be released. When a presynaptic spike arrives into a terminal, calcium ions rush in and cause the fusion of vesicles with the presynaptic membrane. This process is also called exocytosis. Molecules of NT are released into the synaptic cleft (Fig. 4), and diffuse towards the receptors within a postsynaptic membrane. Molecules of NT form a transient bond with the molecules of receptors. This causes opening of ion channels associated with postsynaptic receptors. In the excitatory synapse, receptors are associated with sodium (Na^+) ion channels, and a positive excitatory postsynaptic potential (EPSP) is generated. In the inhibitory synapse, receptors are associated with chlorine (Cl^-) ion channels, and a negative inhibitory postsynaptic potential is generated. Eventually, NT releases its bond with receptors and diffuses back to the presynaptic membrane and out of the synaptic cleft. Special molecular transporters within a presynaptic membrane take molecules of NT back inside the terminal, where they are recycled into new vesicles. This process is called a reuptake of NT. The whole synaptic transmission lasts for about 1 milisecond. Such a synapse is called a chemical synapse, because the transmission of an electric signal is performed in a chemical way.

[Figure 4]

The postsynaptic potential (PSP), either excitatory or inhibitory, has some amplitude and duration. The amplitude and duration of PSP depend upon the number of activated receptor-ion channels and upon the time for how long they stay open. This may last for miliseconds, tens of miliseconds or hundreds of miliseconds. The duration of channel opening depends upon the number of released NT molecules and upon the type of receptors that are associated with ion channels. The amplitude of PSP also depends upon the electric input resistance for ions, which in turn depends upon the size and shape of a postsynaptic spine and dendrites, and upon the distance of synapse from soma. For instance, a short and stubby dendritic spine has a much smaller electric resistance than a long and thin spine. All these pre- and postsynaptic factors determine the weight (strength, efficacy) of a particular synapse.

It is now widely accepted that the activity of a neuron itself, influences its processing of information, and even its life, whether it survives or not.

3. Genes and Gene Networks in Living Cells and Neurons

3.1 The Main Dogma in Molecular Biology

In living systems many dynamic, adaptive, evolving processes are observed at different levels and different stages of the development that are involved in a complex interaction. At a molecular level and a cell level the DNA, the RNA and the protein molecules evolve and interact in a continuous way. The genes form dynamic *gene networks* (GN) that define the complexity of the living organism [7]. It is not just the number of the genes in a genome, but the interaction between them that makes one organism more complex than another. The confirmation that there might be only about 30,000 protein-coding genes in the human genome is one of the key results of the monumental work of the human genome project [16]. There is a mere one-third increase in gene numbers from a rather unsophisticated nematode *Caenorhabditis elegans*, with about 20,000 genes [17] to humans (and other mammals). In fact, the genomes of all mammals are so similar that it is hard to understand how they can produce such different animals. If their genes are alike, it is probably changes in when, where and how active they are that drives the differences between species.

The DNA (Deoxyribonucleic Acid) is a chemical chain, present in the nucleus of each cell of an eukaryotic organism, and it consists of ordered in a double helix pairs of small chemical molecules (bases) which are: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T), linked together by deoxyribose sugar phosphate nucleic acid backbone.

The central dogma of the molecular biology (see Fig. 5) *states that the DNA is transcribed into RNA, which is translated into proteins* [18].

[Figure 5]

The RNA (Ribonucleic Acid) has a similar structure as the DNA, but here the Thymine base (T) is substituted by the Uridine base (U). In the pre-RNA, only segments that contain genes are extracted from the DNA. Each gene consists of two types of segments – exons, that are segments translated into proteins, and introns – segments that are considered redundant and do not take part in the protein production. Removing the introns and ordering only the exon parts of the genes into a sequence is called splicing and this process results in the production of a messenger RNA (or mRNA) sequences.

mRNAs are directly translated into proteins. Each protein consists of a sequence of aminoacids, each of them defined by a base triplet, called a *codon*. From one DNA sequence there are many copies of mRNA produced, the presence of certain gene in all of them defines the level of the gene expression in the cell and can indicate what and how much of the corresponding protein will be produced in the cell.

The above description of the central dogma of the molecular biology is very much a simplified one, but that would help us to understand the rationale behind using connectionist and other information models in bioinformatics.

3.2 Genes and Gene Regulatory Networks

In a single cell, the DNA, the RNA and the protein molecules interact in a continuous way during the process of the RNA transcription from DNA (genotype), and the subsequent RNA to protein (phenotype) translation [19-23]. A single gene interacts with many other genes in this process, inhibiting, directly or indirectly, the expression of some of them, and promoting others at the same time. This interaction can be represented as a *gene regulatory network* (GRN) [7, 8, 24-29]. A major challenge in computational system biology is to create computational models of GRN from both dynamic data (e.g. gene expression data of thousands of genes over time, and also from protein data) and from static data (e.g. DNA), under different external inputs (diseases, drugs, etc.). A large amount of data on gene interactions for specific genomes, as well as on partial models, is available from public domain databases such as GenBank and PubMed (http://www.ncbi.nlm.nih.gov/), KEGG (http://www.genome.ad.jp/kegg/), Stanford Microarray Database, the European Bioinformatics Institute EBI (www.ebi.ac.uk/microarray) and many more [30]. Sophisticated information and mathematical methods are needed for the analysis, modelling and discovery of GRN from this data.

Several generic information methods for modelling and for the discovery of variable interaction networks from time course data have been proposed and used in the domain of GRN modelling. Among them are: statistical methods, that include correlation techniques, linear regression, Bayesian networks, hidden Markov models [9, 29, 31-33]; neural networks [34, 35]; evolutionary computation, and genetic algorithms in particular [30, 36, 37]; directed graphs [19, 30, 31]; Petri nets [30]; ordinary and partial differential equations [8]. There have also been specific methods developed for the purpose of cell modelling [8, 38, 39]. Detailed survey of the elements and pathways in the control of gene expression and principles of their computational modelling can be found for instance in the books [8, 24, 27, 30] and in the review papers [40, 41]. With respect to the taxonomy of GRN models, their principles and descriptions, exhaustive reviews can be found for instance in [41, 42].

4. Neurogenetic Information Processing in the Brain

4.1. General Comments

It is mainly the up-regulation of genes involved in synaptic transmission and plasticity (learning and memory), energy metabolism and growth, that have important consequences in cognitive and behavioral capacities of humans that distinguishes them from other species [133].

Complex faculties of brains, such as for example intelligence, are under the influence of both genetic and environmental factors. But what it actually means for instance for intelligence to be under the influence of genes? *What is the nature and rules of this dependence*? There is vast scientific evidence that intelligence (like other mental faculties, processes, states etc.) depend on the normal functioning of brain neural networks. It is the setup of brain neural networks that is under the influence of genes. The whole brain development since conception is guided by the complex sequence of switching on and off many different genes operating in their own complex and intricate networks – gene regulatory networks (GRN). Through the protein synthesis, genes determine the structure and connectivity of the brain including all the biochemical processes involved in information processing and mere survival of brain cells.

For the correct brain function, there is interplay between genetic and epigenetic factors, like signals from outside of neurons, either from other neurons or other cells in the brain (i.e. glia) or from somewhere else in the body (e.g. hormones). Proteins, molecules and ions acting upon neurons from outside can and do act upon the genome to influence its activity. It is how the environment exerts its influence upon the structure of brain neural networks.

What is the nature of questions we can ask?

– Can we perform a reverse engineering of brain genetic networks? That is, can we identify causal regulatory interactions between genes from time-dependent multigene expression measurements?

– Which are the ways of genes to determine the function of brain neural networks?

In the first research area on genes and CNS (central nervous system), researchers have applied GRN inference techniques to an extensive survey of gene expression in the CNS development. Detailed cluster analysis has uncovered waves of expression for about 112 genes that characterize distinct phases of the development of spinal cord and hippocampus [28, 43]. Thus the first area of research covers monitoring of the activity of many genes in parallel and use the models of GRN to help and guide the inference of regulatory connections between genes, resulting in a gene interaction diagram of gene interaction pathways. There is still a long way to go before all 6,000 genes are processed in a similar way with better techniques and better theoretical and computational models.

The second area of studies is to reveal the consequences of mutated genes upon neural and mental functions (see Table 2) [44]. It is crucial to study, both theoretically and experimentally, the consequences of mutated genes upon the activity of neural networks and upon the neurological and mental deficits that can follow. It is how the links can be established, that is which genes and which interactions between genes are responsible for which neuronal and eventually which mental function. We can see that the detailed pathogenesis of these diseases is at present unknown. This is the area where computational models can bring new insights.

[Table 2]

4.2. Examples of Genes Related to Neuronal Functions, Parameters and Diseases

Let us take as an example a set of genes that define specific neuronal functions (e.g. epileptic behaviour) – in particular - the set of genes that are presumably mutated in individuals suffering from the Childhood Absence Epilepsy (CAE). CAE is an idiopathic (i.e. arising from an unknown cause), generalised non-convulsive epilepsy. The main features are absence seizures. A typical absence is a non-convulsive epileptic seizure, characterized by a briefly (4–20 s) lasting impairment of consciousness. This may happen up to ≈200 times a day. Absence seizures occur spontaneously, i.e. they are not evoked by sensory or other stimuli [11, 45]. Absence is accompanied by a generalised, synchronous, bilateral, 2.5–4 Hz spike and slow-wave discharge (SWD) in the electroencephalogram (EEG). SWD's can start anywhere in the cortex and from there they quickly spread to the entire cortex and thalamus [46].

Table 3 lists the genes which are most probably mutated in CAE, their coded proteins, neuronal function(s) these proteins are responsible for, related parameters in the NN, and a putative alterations in these function, as well as changes in NN parameters. Putative changes in the neural function can be derived from numerous studies performed on humans, rats and mice [11, 45, 47-53]. It should be pointed out that other types of idiopathic epilepsies like the frontal lobe epilepsy (ADNFLE), temporal lobe epilepsy (TLE), juvenile myoclonic epilepsy (JME), adult myoclonic epilepsy (AME), etc. are connected to different channelopathies and receptoropathies [50, 54], i.e. to gene mutations different from CAE, so the presented table applies only to CAE.

[Table 3]

5. Computational Neurogenetic Modelling

5.1. General Comments

The above presented material and many other observations point to the significance of modelling a neuron and a neuronal ensemble at both molecular and neuronal level in order to predict the state of a neural network (or a neuronal ensemble). The process of modelling the gene interaction with the goal of brain understanding is a significant challenge to biologists, mathematicians, information and computer scientists, brain scientists, and researchers from many other areas.

In [41] a computational model of GRN of early neurogenesis in *Drosophila* is developed with the use of artificial neural networks of a Hopfield type, where each gene (gene product) is a node in a recurrent network and the values of connections express the sign and strength of their interactions. The link between the gene expression state and a neuronal function (proliferation, growth) is described with the use of the Landemauer language of describing cell ensembles. In [14] neural networks, gene networks and evolutionary systems are brought together as both brain and gene modelling techniques.

5.2. Principles of a General Computational Neurogenetic Model

Here a computational neurogenetic (CNG) model is introduced that is based on two sets of essential genes – a set of genes $\mathbf{G}_{\mathrm{gen}}$ that defines generic neuronal functions (parameters) and a set of genes $\mathbf{G}_{\mathrm{spec}}$ that defines specific neuronal functions (e.g. synaptic activity). Consider a function that describes measurement of GABA (for example GABA_A) in the synaptic cleft. The measurement of cross-membrane potential using patch-clamp procedure provides a function of time during an event that probes spiking properties of the neuron when a certain voltage is delivered to the neuron.

The two sets of genes form together a set **G** = $\{g_1, g_2, \ldots, g_n\}$ that will be used and a GRN of this set will be defined in the model. The expression level of each gene $g_j(t + \Delta t)$ is a function of the gene expression levels of the rest of the genes **G**(t). As a simple model, we will assume that this function is a linear function:

$$g_j(t+\Delta t) = w_{j,0} + w_{j,1}g_1(t) + \ldots + w_{j,n}g_n(t) \tag{1}$$

The square matrix of gene connection weights **W** represents the GRN. This model can be run for consecutive time moments. An example of GRN of four genes is given in Fig.6.

[Figure 6]

Here we use genes instead of proteins using a standard assumption that if a gene is up regulated, more protein defined by this gene will be produced in the neuronal cell, and vice versa, the less the gene is expressed the less particular protein is produced.

A set of neuronal functions (parameters) **P** = $\{p_1, \text{p}_2, \ldots, p_\text{m}\}$ from a neural network model is related to particular genes, so that each parameter p_j is a function of the expression of several (or in partial case – one) genes. For simplicity in our model we will assume that one parameter p_j depends on one gene g_k through a liner function:

$$p_j(t+\Delta t) = z_{j,0} + z_{j,k}g_k(t) \tag{2}$$

The parameter vector **Z** defines the relationship between the selected genes **G** and the characteristics of a neuron.

This CNG model can be run continuously over time in the following way:

1) Define the initial expression values of the genes **G**, **G**(t = 0) in the neuron and the matrix **W** of the GRN if that is possible.
2) Run the GRN and define the next moment state of the gene set **G**(t+Δt) using equation (1)
3) Define the values of the parameters **P** from the gene state **G** using equation 2.
4) Define the activity of neuron(s) (taking into account all external inputs to the neural network).
5) Go to step 2.
6) We assume that the same matrix **W** defines the GRN of each neuron in a NN. The activity of all neurons are defined for a time interval T thus allowing to calculate the probability distribution function PDF of the activity of neurons in the NN.

This general model is presented in the next sections as two concrete CNG models:

1) A single gene – single neuronal parameter model, where one gene parameter defines one neuronal parameter, such as learning rate or synaptic plasticity. Such model is the evolving connectionist systems (ECOS).
2) Gene network – single neuronal parameter type, where a GRN is used to define the expression values of a gene (a protein) related to certain neuronal parameter. Such model is the CNG of spiking neural networks (CNG-SNN).

First, the basics of evolutionary computation (EC) techniques are present in the next section as EC are used to optimize the gene parameters in the two models presented later in the paper.

6. Evolutionary Computation (EC) for Parameter Optimisation of CNG Models

6.1. Why EC for CNG model optimisation?

Parameter estimation is a very difficult task in inferring GRN models, mainly because of the lack of observation data relative to the number of genes involved. In this respect, evolutionary computation (EC) that are robust, global optimisation methods, become important tools to accurately infer and optimise a GRN.

EC, inspired by the Darwin theory of evolution, searches with a swarm of points based on the objective function (say, the output error) feedback of these points. It has been used for parameter estimation or optimisation in many engineering applications. Unlike classical derivative-based (like Newton) optimisation methods, EC is more robust against noise and multi-modality in the search space. In addition, EC does not require the derivative information of the objective function and is thus applicable to complex, black box problems.

These characteristics make EC highly suitable for identifying parameters of GN models for three reasons. First, the derivative information of the underlying model is usually not available. Second, data is scarce or missing at all (causing multi-modality) and noisy, requiring a robust, global optimisation algorithm that is not easily misled by sub-optima and noise. Third, qualitative inference of parameters is difficult with such small number of observations relative to the large number of genes involved.

6.2. A General Introduction to Evolutionary Computation (EC)

Evolutionary computation (EC) is concerned with population-based search and optimisation of individual systems through generations of populations [56-59]. In other words, some property (or properties) of an individual will be improved not only through an individual development but also through natural selection. Methods of EC include in principle two stages:

(a) a stage of creating a new population of individuals, and

(b) a stage of development of the individual systems, so that a system develops, evolves through interaction with the environment that is also based on the genetic material embodied in the system.

The most popular among the EC techniques are the Genetic Algorithms (GA). They are computational models for the optimisation of complex combinatorial and organisational problems with many variants, by employing analogy with Nature's evolution. Genetic algorithms were introduced for the first time in the work of John Holland [58]. They were further developed by him and other researchers [56-59].

The terms used in the GA are analogous to the terms used to explain the evolution processes. They are:

- *gene* - a basic unit, which defines a certain characteristic (property) of an individual;
- *chromosome* - a string of genes; it is used to represent an individual, or a possible solution to a problem in the solution space;

- *population* - a collection of individuals;
- *crossover (mating)* operation - sub-strings of different individuals are taken and new strings (off-springs) are produced;
- *mutation* - random change of a gene in a chromosome;
- *fitness (goodness) function* - a criterion that evaluates how good each individual is;
- *selection* - a procedure of choosing part of the population which will continue the process of searching for the best solution, while the other part of individuals "die".

```
The main steps of a GA are outlined below:
    1. Initialize a population of n individuals P
    2. REPEAT
        2a {apply a crossover operation between the individuals from P to create an off-spring set of
        individuals R}
        2b {apply a fitness function to evaluate the fitness of the individuals in R}
        2c {apply a selection criteria to select the fittest individuals from R in a new set P}
        2d {apply a mutation operator on the individuals from P}
UNTIL {an individual from P has reached a desired fitness or end of the procedure is reached}
```

When using the GA method for a complex multi-optional optimisation problem, there is no need for in-depth problem knowledge, neither a need for many data examples stored beforehand. What is needed here is merely a "fitness" or "goodness" criterion for the selection of the most promising individuals (they are partial solutions to the problem). This criterion may require a mutation as well, which is a heuristic approach of a "trial-error" type. This implies keeping (recording) the best solutions at each of the stages.

The simple genetic algorithms introduced by John Holland are characterized by:

• *simple, binary genes,* i.e. the genes take values of 0 and 1 only;

• *simple, fixed single-point crossover operation;* the crossover operation is done by choosing a point where a chromosome is divided into two parts swapped with the two parts taken from another individual (see Fig. 7);

• *fixed-length encoding*, i.e. the chromosomes had fixed length of 6 genes.

[Figure 7]

The main issues in using GA relate to the choice of genetic operations (crossover, selection, mutation). GA are comprised of a great deal of parallelism. Thus, each of the branches of the search tree for best individuals can be utilized in parallel with the others. This allows for an easy realization of the genetic algorithms on parallel architectures. GA are search heuristics for the "best" instance in the space of all possible instances. The following issues are important for any GA:

- *the encoding scheme,* i.e. how to encode the problem in terms of genetic algorithms - what variables to choose as genes, how to construct the chromosomes, etc.
- *the population size* - how many possible solutions should be kept for further development;
- *the crossover operations* - how to combine old individuals and produce new, more prospective ones;
- *the selection criteria;*
- *the mutation operator* - when and how to apply mutation.

Other EC techniques are:

- Evolutionary strategies. These techniques use only one chromosome and a mutation operation, along with a fitness criterion, to navigate in the solution (chromosomal) space.

- Evolutionary programming. These are EC techniques applied to the automated creation and optimisation of sequence of commands (operators) that constitute a program (or an algorithm) to solve a given problem [57].

The theory of GA and the other EC techniques includes different methods for selection of individuals from a population, different crossover techniques, different mutation techniques.

Selection is based on fitness. A common approach is a proportional fitness. Roulette wheel selection gives chances to individuals according to their fitness evaluation. Other selection techniques include tournament selection (every time of selection, the roulette wheel is turned twice, and the individual with the highest fitness is selected), rank ordering, and so on [60]. An important feature of the selection procedure is that fitter individuals are more likely to be selected. The selection procedure can involve also keeping the best individuals from the previous generation. This operation is called elitism.

After the best individuals are selected from a population, a crossover operation is applied between these individuals. Different single or multiple-point crossover operations can be used. Then the selected individuals undergo mutation.

Mutation can be performed in the following ways:

- For a binary string, just randomly “flip” a bit

- For a more complex structure, randomly select a site, delete the structure associated with this site, and randomly create a new sub-structure

Some EC methods just use mutation (no crossover, e.g. evolutionary strategies). Normally, however, mutation is used to search in a “local search space”, by allowing small changes in the genotype (and therefore hopefully in the phenotype). In the field of NNs, optimal values of parameters (weights, architecture, etc.) can be sought not only through learning but also through evolution, that is the process of selection and crossover of the best individual neural networks. This process can be combined with individual learning to lead to the Baldwin effect [61]. The genotypes after Baldwinian learning remain unchanged, however learning can influence indirectly the selection process by altering the fitness of individuals, thus eventually evolution is affected [62].

7. A CNG Model of a Single Gene – Single Neuronal Parameter Type: Evolving Connectionist Systems (ECOS)

7.1. Evolving Connectionist Systems

Evolving connectionist systems (ECOS) are multi-modular connectionist architectures that facilitate modelling of evolving processes and knowledge discovery [14].

An evolving connectionist system is a neural network or a collection of such networks that operate continuously in time and adapt their structure and functionality through a continuous interaction with the environment and with other systems. An example of a simple ECOS as evolving fuzzy neural network (EFuNN) is given in fig.8a [14] .

The adaptation is defined through:

(i) a set of parameters ("genes") that are subject to change during the system operation;

(ii) an incoming continuous flow of information with unknown distribution;

(iii) a goal (rationale) criteria (also subject to modification) that is applied to optimize the performance of the system over time.

The ECOS set of parameters *P* can be regarded as a chromosome of "genes" and both developmental learning and evolutionary computation can be applied for the system's optimization as explained in the next sub-section and in fig.8b.

[Figure 8a,b]

ECOS have the following characteristics[14]:

(1) They evolve in an open space;

(2) They learn in on-line, pattern mode, incremental learning, and possibly through one pass of the incoming data through the system;

(3) They learn in a life-long learning mode;

(4) They learn both as individual systems and as evolutionary population systems;

(5) They use constructive learning and have evolving structures;

(6) They learn and partition the problem space locally, thus allowing for a fast adaptation and tracing the evolving processes over time;

(7) They facilitate different types of knowledge, mostly a combination of memory-based, statistical and symbolic knowledge (e.g. Zadeh-Mamdani fuzzy rules - EFuNN – [63]; Takagi-Sugeno fuzzy rules (DENFIS - [64]), and type-2 fuzzy rules [65]).

Table 4 is a generalised algorithm of the functioning of an ECOS [14].

[Table 4]

7.2. ECOS Parameter (`Gene`) Optimisation with a GA

EC, and the genetic algorithms (GA) in particular, include in principal a stage of development of each individual system, so that a system develops, evolves through interaction with the environment that is also based on the genetic material embodied in the system. Both individual system development (life-long learning) and evolutionary development (evolution over generations) are part of an integrated procedure that is concerned with the system's on-line evolving[14].

ECOS have efficient learning algorithms that allow for on-line, life-long capturing of associations between input and output clusters, rules, statistics. But there are several problems as well. One of them is the large number of parameters and modes of operation that need to be properly selected and optimised for a particular application. These parameters may include thresholds for the creation of new nodes, thresholds for the removal of nodes, parameters for the aggregation of nodes, etc.

A method [14] is presented here to use EC for finding the optimum set of parameters (`genes`) of ECOS for a given task. The connection weights of the ECOS are not optimised through EC as they are modified through the ECOS learning procedures and they capture meaningful information and knowledge. A schematic diagram of the method is given in fig. 8b.

As new data is presented to the network, the system strives to find a better set of parameters, allowing the ECOS to better adapt and better solve the problem in current time interval. A balance must be found between improving the solution for the new data, while not loosing the performance gained on the old data. In principle catastrophic forgetting can occur in neural networks. However, an ECOS with appropriate parameters should be more resilient to this undesirable effect as ECOS use local learning and flexible structure representation. In addition to that, a population of ECOS is used and the solution is being produced either by the best ECOS for the time, or by the collective output from all ECOS weighted through their fitness evaluation at the moment.

An initial period of adaptation and training is followed by a period of evolutionary parameter optimisation across many ECOS in a population over consequent generations. The performance of the ECOS networks is measured by a desired fitness function, usually defined as a function of the error over a window of stored data and the increase in size of ECOS. The initial parameters of ECOS are either set to sensible defaults, or may be initialised randomly. The subsequent evolutionary process is iterative. It may begin after a predetermined number of examples have been presented to the baseline network population and further generations can be produced in a life-long mode.

8. Multiple Genes – Single Neuronal Parameter Type Model: A CNG Model of a Spiking Neural Network (CNG-SNN)

8.1. Principles of Spiking Neurons and Spiking Neural Networks

Spiking model of a neuron – element of the spiking neural network (SNN) can be for instance inspired by the spike response model (SRM) of a neuron [66, 67]. Neuron *i* receives input spikes from presynaptic neurons $i \in \Gamma_i$, where Γ_i is a pool of all neurons presynaptic to neuron *i* (see Fig. 9). The state of neuron *i* is described by the state variable $u_i(t)$ that can be interpreted as a total postsynaptic potential (PSP) at the membrane of soma. When $u_i(t)$ reaches the firing threshold $\vartheta_i(t)$, neuron *i* fires, i.e. emits a spike (see Fig. 10). The moment of $\vartheta_i(t)$ crossing defines a firing time t_i of an output spike. The value of the state variable $u_i(t)$ is the sum of all postsynaptic potentials, i.e.:

$$u_i(t) = \sum_{j \in \Gamma_i} \sum_{t_j \in F_j} J_{ij} \varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax}) \qquad (3)$$

The weight of synaptic connection from neuron *j* to neuron *i* is denoted by J_{ij}. It takes positive (negative) values for excitatory (inhibitory) connections, respectively. Depending on the sign of J_{ij}, a presynaptic spike generated at time t_j increases (or decreases) $u_i(t)$ by an amount $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax})$. Δ_{ij}^{ax} is an axonal delay between neurons *i* and *j* which increases with Euclidean distance between neurons.

[Figure 9]

[Figure 10]

The positive kernel $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax}) = \varepsilon_{ij}(s)$ expresses an individual postsynaptic potential (PSP) evoked by a presynaptic neuron *j* on neuron *i*. A double exponential formula can be used

$$\varepsilon_{ij}^{synapse}(s) = A^{synapse}\left(\exp\left(-\frac{s}{\tau_{decay}^{synapse}}\right) - \exp\left(-\frac{s}{\tau_{rise}^{synapse}}\right)\right) \qquad (4)$$

where $\tau_{decay/rise}^{synapse}$ are time constants of the rise and fall of an individual PSP, *A* is the PSP's amplitude, and *synapse = fast_excitation, fast_inhibition, slow_excitation, and slow_inhibition,* respectively. These types of PSPs are based on neurobiological data [47, 53].

Immediately after firing an output spike at *t_i*, neuron's firing threshold $\vartheta_i(t)$, increases *m* times and then returns to its initial value ϑ_0 in an exponential fashion:

$$\vartheta_i(t-t_i) = m \times \vartheta_0 \exp\left(-\frac{t-t_i}{\tau^{\tau}_{decay}}\right) \tag{5}$$

where $\vartheta^{\vartheta}_{decay}$ is the time constant of the threshold decay. In such a way, absolute and relative refractory periods are modelled.

External inputs from the input layer are added at each time step, thus incorporating the background noise and/or the background oscillations. Each external input has its own weight $J_{ik}^{ext\_input}$ and $\varepsilon_k(t)$, such that

$$u_i^{ext\_inpu}(t) = J_{ik}^{ext\_input}\,\varepsilon_{ik}(t) \tag{6}$$

It is optional to add some degree of Gaussian noise to the right hand side of the equation above to obtain a stochastic neuron model instead of a deterministic one.

Fig. 11 illustrates the basic architecture of a spiking neural network (SNN). Spiking neurons within the network can be either excitatory or inhibitory. There can be as many as about 10-20% of inhibitory neurons positioned randomly on the rectangular grid of *N* neurons. Lateral connections between neurons have weights that decrease in value with distance from neuron *i* for instance according to a Gaussian formula while the connections between neurons themselves can be established at random.

[Figure 11]

Fig. 12 illustrates a record of activity of the introduced SNN. It is useful to keep record of spiking activities of all neurons individually and in total, as well as the record of the total membrane potential that is in fact proportional to EEG [68]. Various analytical tools are developed for instance for evaluation of degree of synchrony between neurons [67], and for evaluation of frequency spectra, like the fast Fourier transform, and others [67, 69]. Presented SNN belongs among the simplest ones. There are much more detailed models of spiking neurons than SRM or the so-called integrate-and-fire (I&F) model neurons. These more detailed models include for instance the various ion receptor and channel kinetics [47, 70], and also multicompartmental neuron models where the effect of spatial not only temporal summation of PSPs on the neuron input surface is taken into account [71, 72].

[Figure 12]

8.2. EC for the Parameter Optimisation of the CNG-SNN

We assume that the functioning of a neural network is evaluated as its probability distribution function (PDF) of neural activity, and as an integrated membrane potential value which is equivalent to an EEG, thus making possible to observe and model different brain functions such as epilepsy, alpha-, beta-, and gamma states, learning, memorizing, sleeping, etc. Usually EEG data from real brain observations is available to test these models. In general, the functioning of a neural network can be expressed and evaluated in other terms, for instance metabolic, or other activity terms.

The task for us is to define a set of parameter values for **W** and **Z** (see section 5.2) so that the SNN has a desired PDF of neural activity denoted here as PDF^ and a desired EEG denoted as EEG^. To solve this problem with the use of the above described model, we can apply the methods of evolutionary computation (EC) and in particular – a genetic algorithm (GA) method [30, 73, 74] with a fitness function of PDF=PDF^ or/and EEG=EEG^ within a margin of tolerance.

In the GA implementation here, two chromosomes of parameters will be used – **W** and **Z,** so that for every generation (a set of values) of these chromosomes, the SNN is run for the time period of T and the PDF is evaluated. Then it is compared with the desired PDF^ and/or EEG^. If the fitness function is not satisfied, the process continuous with modified values for the parameters **W** and **Z** according to the selected GA strategy.

9. Conclusions and Future Development

Nanotechnology has a tremendous potential for the cure of brain diseases. This approach requires a deep understanding of chemical and information processes in the brain, in single neurons and in the nuclei of these neurons, and especially how these processes relate to each other. In this respect there is a need for theories and computational models to model and predict the outcome of brain abnormalities and their treatment. Computational neurogenetic modelling is a promissing approach that will be further developed and applied. A new field of nanomedicine can develop in future to deal with nanotechnology in medicine and health care [75, 76].

In between, many nanotechnical problems must be solved. For instance, brain similarly like other tissues, responds to alien substances with the healing process. For instance, various neural probes (usually composed of silicon) become encapsulated with glial scar tissue what can impede with normal neuron function. From a nanomaterial point of view, nanophase materials can influence interaction with proteins and other molecules that take part in cell processes in many unwanted ways [77]. Another area of future studies and possible applications will be the development of nanoscale logic networks and nanochips [78, 79]. Not only this will lead to an unprecedented miniaturization of conventional computers but also to miniaturization of neurocomputers and neurochips. It is not only about implementation of classical NNs in hardware but also about the development of the so-called neuromorphic systems. Neuromorphic systems are implementations in silicon of sensory and neural systems whose architecture and design are based on neurobiology that can compete with human senses and pattern recognition systems and run in real time [80, 81]. Researchers in this area also work on developing communication between living vertebrate neurons and electronic systems [80].

Although a lot is known about the brain, issues about its functioning, representation and processing of information are still subjects of an intense research. Applicability of nanotechnology will depend not only upon the nanotechnological progress itself but also on the progress in understaning the brain, its dynamical behavior, and how normal and disturbed neural functions affect the brain dynamics. On the other hand, nanotechnology by targeted and controlled manipulation of selected molecules, might help in the search for those very crucial material phenomena in the brain that might be causally linked to consciousness and subjective experience.

References

1. M. Arbib, Ed., The Handbook of Brain Theory and Neural Networks*,* 2nd ed., MIT Press, Cambridge, MA (2003)

2. S. Amari, Proc. IEEE 78, 1143 (1990)

3. T. Kohonen, Proc. IEEE 78, 1464 (1990)

4. G. Carpenter and S. Grossberg, Pattern Recognition by Self-Organizing Neural Networks, MIT Press, Cambridge, MA (1991)

5. C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford (1995)

6. S. Amari and N. Kasabov, Eds., Brain-like Computing and Intelligent Information Systems, Springer-Verlag, Singapore (1998)

7. P. Baldi and S. Brunak, Bioinformatics - a Machine Learning Approach*,* 2nd ed, MIT Press, Cambridge, MA (2001)

8. J. Bower and H. Bolouri, Eds., Computational Modelling of Genetic and Biochemical Networks, MIT Press, Cambridge, MA (2001)

9. H. de Jong, Journal of Computational Biology 9, 67 (2002)

10. N. Kasabov, Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, MIT Press, Cambridge, MA (1996)

11. V. Crunelli and N. Leresche, Nature Rev. Neurosci. 3, 371 (2002)

12. H.-H. Ropers, M. Hoeltzenbein, V. Kalscheuer, H. Yntema, B. Hamel, J.-P. Fryns, J. Chelly, M. Partington, J. Gecz and C. Moraine, Trends in Genetics 19, 316 (2003)

13. Y. Zhang, F. Schlachetzki and W. M. Pardridge, Molecular Therapy 7, 11 (2003)

14. N. Kasabov, Evolving Connectionist Systems. Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines, Springer-Verlag, London (2003)

15. E. R. Kandel, J. H. Schwartz and T. M. Jessell, Principles of Neural Science*,* 4th ed, McGraw-Hill, New York (2000)

16. J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural and G. G. Sutton, Science 291, 1304 (2001)

17. C. ElegansSequencingConsortium, Science 282, 2012 (1998)

18. F. Crick, Nature 227, 561 (1970)

19. S. Kauffman, Journal of Theoretical Biology 44, 167 (1974)

20. J. L. deRisi, V. R. Iyer and P. O. Brown, Science 275, 680 (1997)

21. B. Sobral, in From Jay Lush to Genomics: visions for animal breeding and genetics, Edited by J. M. Dekkers, Lamont, S.J., Rothschild, M.F., Iowa State Univ. Press, Iowa (1999).

22. P. A. Pevzner, Computational Molecular Biology: An Algorithmic Approach, MIT Press, Cambridge, MA (2000)

23. J. R. Koza, W. Mydlowec, G. Lanza, J. Yu and M. A. Keane, in Proc. Pacific Symposium on Biocomputing, (2001), Vol. 6, pp. 434-445.

24. J. Collado-Vides and R. Hofestadt, Eds., Gene Regulation and Metabolism. Post-Genomic Computational Approaches, MIT Press, Cambridge, MA (2002)

25. L. Hunter, Canadian Artificial Intelligence 35, 10 (1994)

26. J. Dow, G. Lindsay and J. Morrison, Biochemistry Molecules, Cells and the Body, Addison-Wesley, Boston, MA (1995)

27. J. Collado-Vides, B. Magasanik and T. F. Smith, Eds., Integrative Approaches to Molecular Biology, MIT Press, Cambridge, MA (1996)

28. P. D'Haeseleer, X. Wen, S. Fuhrman and R. Somogyi, in Proc. Pacific Symposium on Biocomputing, Hawai (1999), Vol. 4, pp. 41-52.

29. P. D'Haeseleer, S. Liang and R. Somogyi, Bioinformatics 16, 707 (2000)

30. G. Fogel and D. Corne, Evolutionary Computation for Bioinformatics, Morgan Kaufmann, San Francisco, CA (2003)

31. T. Akutsu, S. Miyano and S. Kuhara, in Proc. Pacific Symposium on Biocomputing, Hawai (1999), Vol. 4, pp. 17-28.

32. S. Gomez, S. Lo and A. Rzhetsky, Genetics 159, 1291 (2001)

33. A. Lindlof and B. Olsson, Information Sciences 146, 103 (2002)

34. J. Vohradsky, Journal of Biological Chemistry 276, 36168 (2001)

35. J. Vohradsky, The FASEB Journal 15, 846 (2001)

36. S. Ando, E. Sakamoto and H. Iba, in Proc. 6th Joint Conference on Information Sciences, Cary, NC (2002), pp. 1249-1256.

37. Mimura and H. Iba, in Proc. 6th Joint Conference on Information Sciences, Cary, NC (2002), pp. 1243-1248.

38. K. W. Kohn and D. S. Dimitrov, in Computer Modeling and Simulation of Complex Biological Systems, Edited by S. S. Iyengar, CRC Press, Boca Raton (1998), pp. 101-123.

39. J. Vides, B. Magasanik and T. Smith, Integrated Approaches to Molecular Biology, MIT Press, Cambridge, MA (1996)

40. M. A. Savageau, Chaos 11, 142 (2001)

41. G. Marnellos and E. D. Mjolsness, in Modeling Neural Development, Edited by A. vanOoyen, MIT Press, Cambridge, MA (2003), pp. 27-48.

42. L. F. A. Wessels, E. P. vanSomeren and M. J. T. Reinders, in Proc. Pacific Symposium on Biocomputing, Hawai (2001), pp. 508-519.

43. R. Somogyi, S. Fuhrman and X. Wen, in Computational Modeling of Genetic and Biochemical Networks, Edited by J. M. Bower and H. Bolouri, MIT Press, Cambridge, MA (2001), pp. 119-157.

44. The Nervous System, in Genes and Disease, Edited, National Centre for Biotechnology Information (NCBI), http://www.ncbi.nlm.nih.gov/books/bv.fcgi?call=bv.View..ShowSection&rid=gnd.chapter.75 (2003).

45. C. Marini, L. A. Harkin, R. H. Wallace, J. C. Mulley, I. E. Scheffer and S. F. Berkovic, Brain 126, 230 (2003)

46. H. K. M. Meeren, J. P. M. Pijn, E. L. J. M. VanLuijtelaar, A. M. L. Coenen and F. H. LopesdaSilva, J. Neurosci. 22, 1480 (2002)

47. A. Destexhe, J. Neurosci. 18, 9099 (1998)

48. A. Contractor, G. T. Swanson, A. Sailer, S. O'Gorman and S. F. Heineman, J. Neurosci. 20, 8269 (2000)

49. R. A. Deisz, Neuropharmacology 38, 1755 (1999)

50. R. M. Gardiner, Epilepsy Res. 36, 91 (1999)

51. G. E. Homanics, T. DeLorey, L. L. Firestone, J. J. Quinlan and A. Handforth, Proc. Natl. Acad. Sci. USA 94, 4143 (1997)

52. S. Jones, S. Sudweeks and J. L. Yakel, Trends Neurosci. 22, 555 (1999)

53. A. V. Semyanov, Neurophysiology 34, 71 (2002)

54. O. K. Steinlein and J. L. Noebeles, Current Opinion in Genetics and Development 10, 286 (2000)

55. N. Kasabov and D. Dimitrov, in Proc. ICONIP'2002 - International Conference on Neuro-Information Processing, IEEE Press, Singapore (2002).

56. D. E. Goldberg, Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, Reading (1989)

57. J. Koza, Genetic Programming, MIT Press, Cambridge, MA (1992)

58. J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor (1975)

59. J. H. Holland, Emergence, Oxford Univ. Press, Oxford (1998)

60. D. Fogel, L. Fogel and V. Porto, Biol. Cybernetics 63, 487 (1990)

61. J. M. Baldwin, American Naturalist 30, 441 (1896)

62. V. Kvasnicka and J. Pospichal, in Advances in Soft Computing - Engineering Design and Manufacturing, Edited by R. Roy, T. Furuhashi and P. K. Chawdhry, Springer-Verlag, London (1999), pp. 481-496.

63. N. Kasabov, IEEE Trans. Systems, Man and Cybernetics - B. Cybernetics 31, 902 (2001)

64. N. Kasabov and Q. Song, IEEE Trans. on Fuzzy Systems 10, 144 (2002)

65. N. Kasabov, in Proc. 1st International IEEE Symposium on Intelligent Systems, (2002), Vol. 1, pp. 24-28.

66. W. Maass and C. M. Bishop, Eds., Pulsed Neural Networks, The MIT Press, Cambridge, MA (1999)

67. W. Gerstner and W. M. Kistler, Spiking Neuron Models, Cambridge Univ. Press, Cambridge, MA (2002)

68. J. W. Freeman, Mass action in the nervous system, Academic Press, New York (1975)

69. W. J. Freeman, M. D. Holmes, B. V. Burke and S. Vanhatalo, Clinical Neurophysiology 114, 1053 (2003)

70. P. Kudela, P. J. Franaszcuk and G. K. Bergey, Biol. Cybernetics 88, 276 (2003)

71. K.-H. Yang, P. J. Franaszczuk and G. K. Bergey, Biol. Cybernetics 89, 242 (2003)

72. J. M. Bower and D. Beeman, The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System*,* 2 ed, TELOS/Springer-Verlag, New York (1998)

73. J. H. Holland, Adaptation in natural and artificial systems,, The University of Michigan Press, Ann Arbor, MI (1975)

74. D. E. Goldberg, Genetic Algorithms in Search, Optimization and machine Learning., Reading,MA, Addison-Wesley. (1989)

75. R. A. Freitas, Nanomedicine, Volume I: Basic Capabilities, Landes Bioscience, Georgetown, TX (1999)

76. R. A. Freitas, Nanomedicine, Volume IIA: Biocompatibility, Landes Bioscience, Georgetown, TX (2003)

77. T. J. Webster, M. C. Waid, J. L. McKenzie, R. L. Price and J. U. Ejiofor, Nanotechnology 15, 48 (2004)

78. A. S. Sadek, K. Nikolic and M. Forshaw, Nanotechnology 15, 192 (2004)

79. G. Bauer, J. Hassmann, H. Walter, J. Haglmueller, C. Mayer and T. Schalkhammer, Nanotechnology 15, 1289 (2004)

80. L. S. Smith and A. Hamilton, Eds., Neuromorphic Systems: Engineering Silicon from Neurobiology*,* Progress in Neural Processing, World Scientific Publishing Co. Ltd., London (1998)

81. P. Tikovic, M. Voros and D. Durackova, Journal of Electrical Engineering 52, 68 (2001)

Table captions

Table 1. Different levels of evolving processes in the human brain that cannot be modelled by the use of existing AI methods.

Table 2. Single and multiple genes related to some brain functions and abnormalities (constructed according to [44])

Table 3. List of genes putatively mutated in CAE, their coded proteins, neuronal function these proteins are responsible for, related parameters to the activation of a neuronal ensemble (see section 8.1), and a putative alteration in this function: ↑ means increase, ↓ means decrease.

Table 4. A generalised ECOS algorithm (from [14]).

Figure captions

Figure 1. Gross anatomical and functional division of the human cerebral cortex. The same division applies for the other, in this case, the right hemisphere. A dashed curve marks the position of evolutionary older subcortical nuclei in the brainstem of the brain. Each of the depicted areas has far more subdivisions.

Figure 2. Schematic illustration of a neuron and its parts. There is a synapse at every dendritic spine. Synapses are also formed on the dendritic shafts and on the soma.

Figure 3. Electric synaptic potentials and axonal ion channels responsible for spike generation and propagation. EPSP = excitatory postsynaptic potential, IPSP = inhibitory postsynaptic potential, ϑ = excitatory threshold for an output spike generation.

Figure 4. Transmission of electric signals in a chemical synapse. NT = neurotransmitter, R = AMPA-receptor-gated ion channel for sodium, N = NMDA-receptor-gated ion channel for sodium and calcium.

Figure 5. DNA is transcribed into RNA, which is translated into proteins – the central dogma of molecular biology.

Figure 6. A hypothetical example of a GN comprised of four genes related to both generic and specific functions of a neuron. The genes are connected with arcs that represent the relationship (sign and strength) between the level of expression of this gene at time moment (*t*) and the next time moment ($t + \Delta t$).

Figure 7. Schematic illustration of the operation of single-point crossover in GA. Parenting individuals exchange parts (in this case one half) of their chromosomes to create exactly two offsprings.

Figure 8. (a) A block diagram of a simple evovling connectionist system (ECOS) – evovling fuzzy neural network (EfuNN); (b) ECOS optimised with the use of a GA

Figure 9. In response to input series of spikes from the pool of presynaptic neurons Γ_i a neuron *i* generates its own series of output spikes.

Figure 10. Spiking neuron model. (a) When the state variable $u_i(t)$ of a spiking neuron reaches the firing threshold $\vartheta_i(t)$ at time t_i, a neuron fires an output spike. However, an actual firing threshold rises after each output spike and decays back to the initial value. (b) Subthreshold temporal summation of individual postsynaptic potentials. (c) The kernel $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax})$ describes an individual PSP evoked by the presynaptic spike fired at time t_j after some axonal delay Δ_{ij}^{ax}.

Figure 11 Architecture of the spiking neural network (SNN). About as many as 10–20% of neurons are inhibitory neurons that are randomly positioned on the grid (filled circles).

Excitatory and inhibitory lateral connections decrease in strength with distance according to the Gaussian distribution. There are one-to-one feed-forward connections from the input layer.

Figure 12 Temporal evolution of a spiking neurogenetic network. From top to bottom: Changing expression of genes within a neuron; Total number of spikes generated by all neurons at each time step; Traces of each neuron spiking activities; Total sum of individual membrane potentials as a measure proportional to EEG.

Table. 1.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

1. Evolutionary development Function examples: *genome evolution, creation of new individuals and species*
2. Brain level Function examples: *cognition, speech and language, consciousness*
3. Neural network level Function examples: *sound perception, visual image processing*
4. Whole cell, neuronal level Function examples**:** *synaptic processes; cell life*
5. Molecular level Function examples*: DNA translation into RNA, RNA transcription into proteins*

Table. 2

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

Disease	**Mutations of genes identified so far**	**Location of genes on chromosomes**	**Brain abnormality**	**Symptoms**
Alzheimer disease (AD)	PS2 (AD4) PS1 (AD3) ? ?	1 14 19 21	plaques made of fragmented brain cells surrounded by amyloid-family proteins, tangles of cytoskeleton filaments	progressive inability to remember facts and events and later to recognize friends and family
Amyotrophic lateral sclerosis (ALS)	SOD1	21	progressive degeneration of motor neuron cells in the spinal cord and brain	loss of motor control which ultimately results in paralysis and death
Angelman syndrome (AS)	UBE3A	maternally derived chromosome 15 (segment	mutations in UBE3A disrupt protein break down during brain development	mental retardation, abnormal gait, speech impairment, seizures, frequent

		15q11–q13)		laughing, smiling, and excitability
Epilepsy (many forms)	Multiple	Multiple	abnormal cell firing in the brain	recurring seizures
Fragile X syndrome	FMR1	X	impaired synaptic function of glutamatergic synapses	the most common inherited form of mental retardation
Huntington disease (HD)	HD gene	4	dilatation of ventricles and atrophy of caudate nucleus	degenerative neurological disease that leads to dementia
Williams syndrome	LIM kinase and elastin coding sequences	7	?	high competence in language, music and interpersonal relations, with low IQ

Table. 3.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

MUTATED GENE	PROTEIN	NEURONAL FUNCTION AND ITS PUTATIVE ALTERATION	BRAIN ACTIVITY PARAMETERS
GRIK1	Ionotropic GluR5 (Kainate receptor 2 for glutamate)	Fast excitation ↑ or ↓ (depending on the place in the brain)	$A^{fast\_exc}, \tau^{fast\_exc}_{decay/rise}$ ↑ or ↓
GABRB3	GABA$_A$ receptor β3 subunit	Fast inhibition ↓	$A^{fast\_inh}, \tau^{fast\_inh}_{decay/rise}$ ↓
GPHN	Gephyrin	Fast inhibition ↓	$A^{fast\_inh}, \tau^{fast\_inh}_{decay/rise}$ ↓
CHRNA4	nAChR α4 subunit	Both fast and slow inhibition ↓	J^{inh}_{ij} ↓
OPRM1	μ-Opioid receptor type 1	Firing threshold ↓	$\vartheta_0, \tau^{\vartheta}_{decay}$ ↓

Table. 4.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

```
Set some preliminary parameter values for the ECOS parameters ("genes" in "chromosomes")
REPEAT {in a life-long learning mode}
IF input, or input-output data is available DO
Read input data (or input- output data pairs if such are available)
Evaluate the input-output features (variables):
(a) add new ones if necessary;
(b) select the current most appropriate ones for the task
Propagate input data through the ECOS modules and evaluate the similarity of the input data to the modules.
If there is not sufficient similarity – create new modules, or create new connections in an existing module.
Calculate the output of the system.
Calculate a feedback from the output to the system through:
A supervised mode of learning, if output error values are calculated, or
A reinforcement mode of learning – if just hints about the correctness of the output values are available, or
Report the output values if the system is in a recall mode.
Modify the structure of ECOS based on the feedback.
```

Extract and report the current knowledge learned by the ECOS, e.g. through rule extraction techniques.

```
Optimize the ECOS parameters (“genes”) and its structure based on some accumulated statistical information of the system’s performance  (e.g. using Evolutionary Computation  (EC) methods)
ELSE
Apply inner structural and functional learning and structure and parameter optimisation (e.g. sleep learning, EC methods)
UNTIL {the system is stopped, or interrupted}
```

Figure 1.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

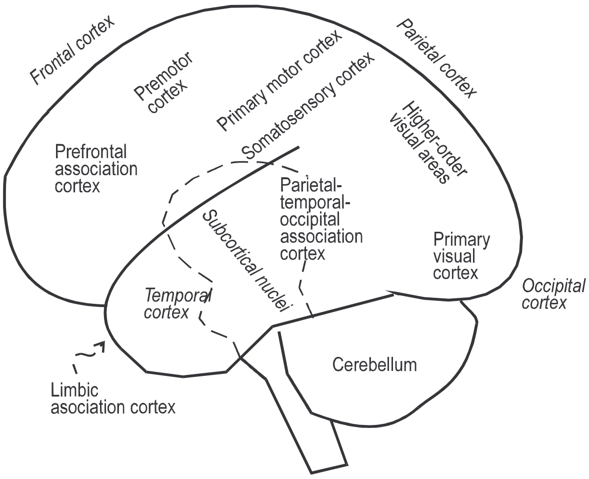


Figure 2.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

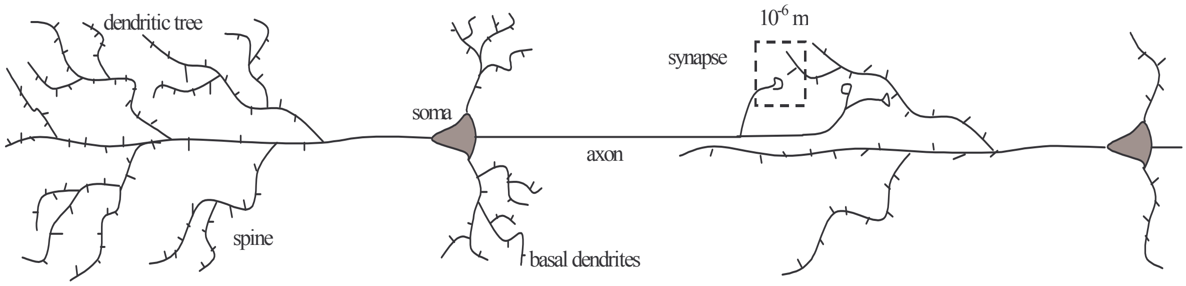


Figure 3.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

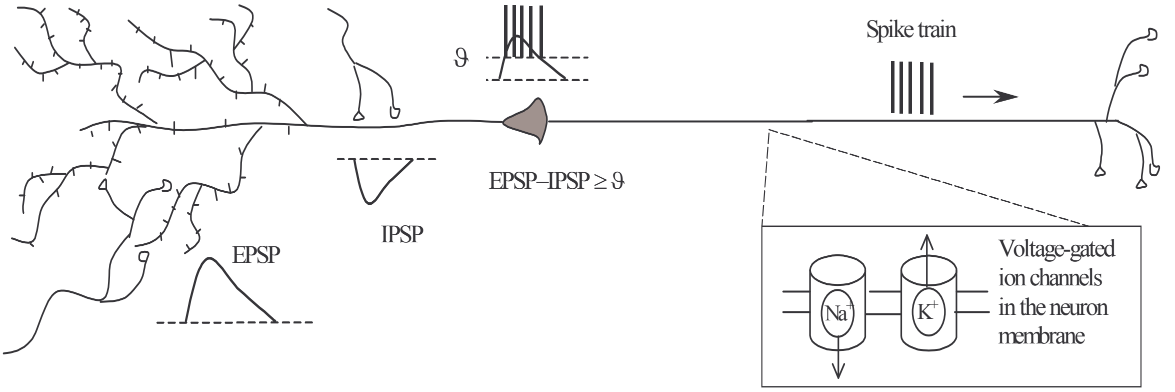


Figure 4.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

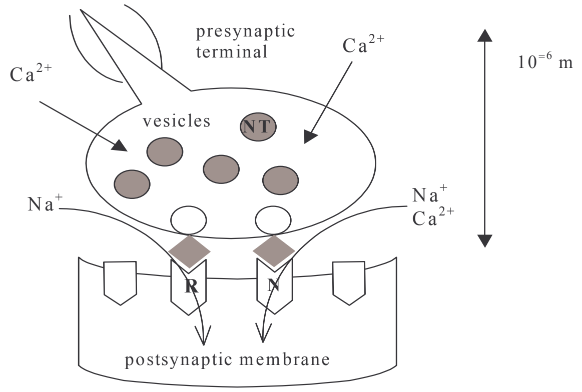


Figure 5.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

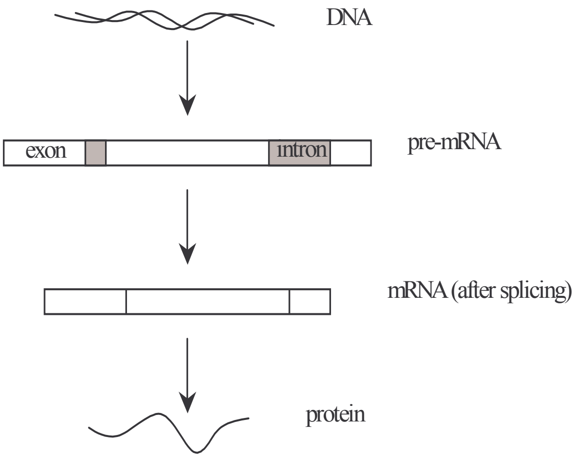


Figure 6.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

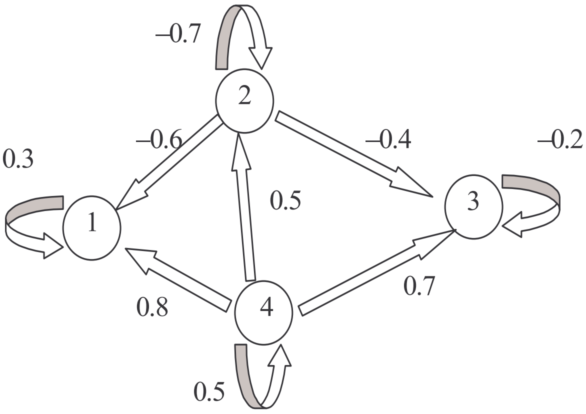


Figure 7.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

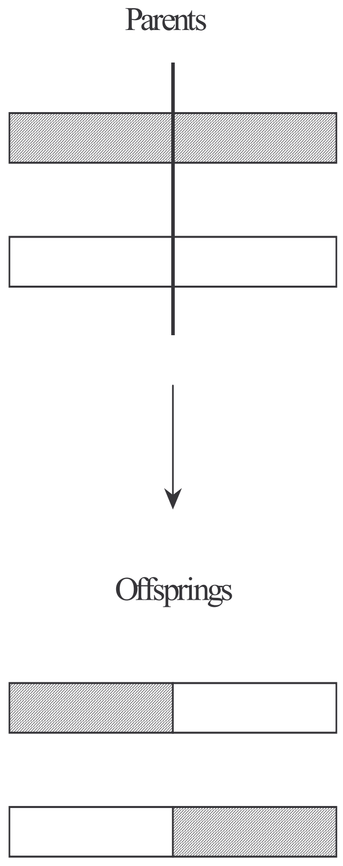
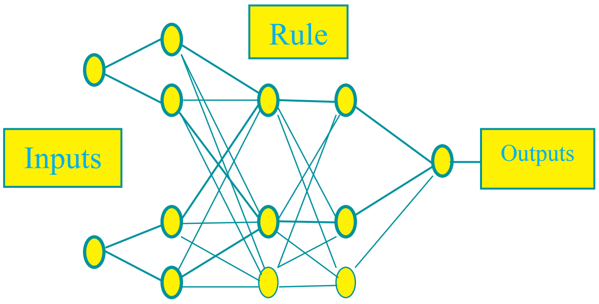


Figure 8.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

(a)



(b)

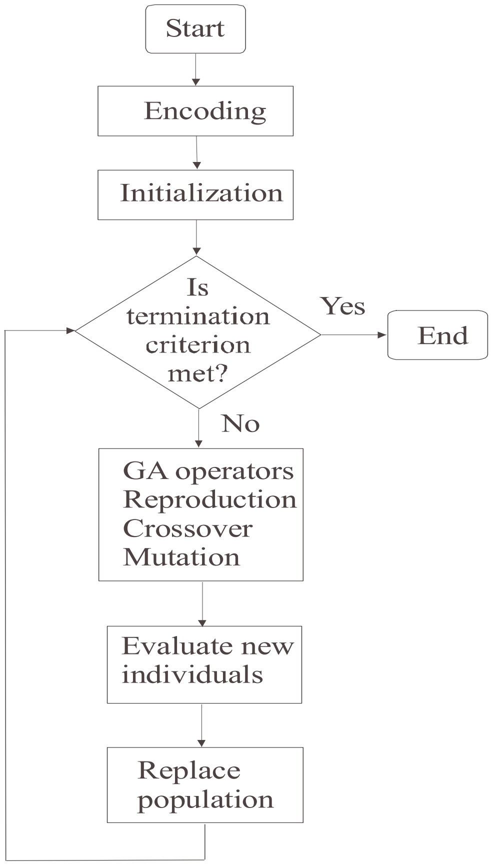


Figure 9

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

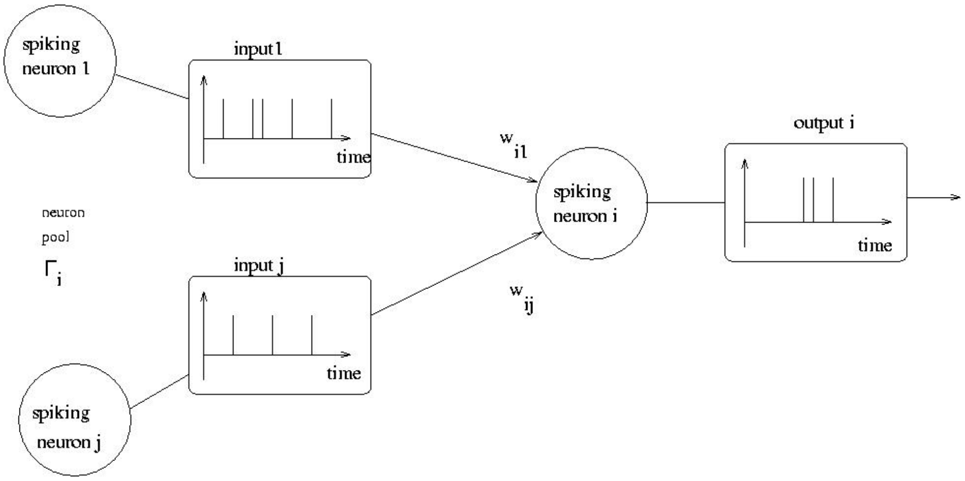


Figure 10.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

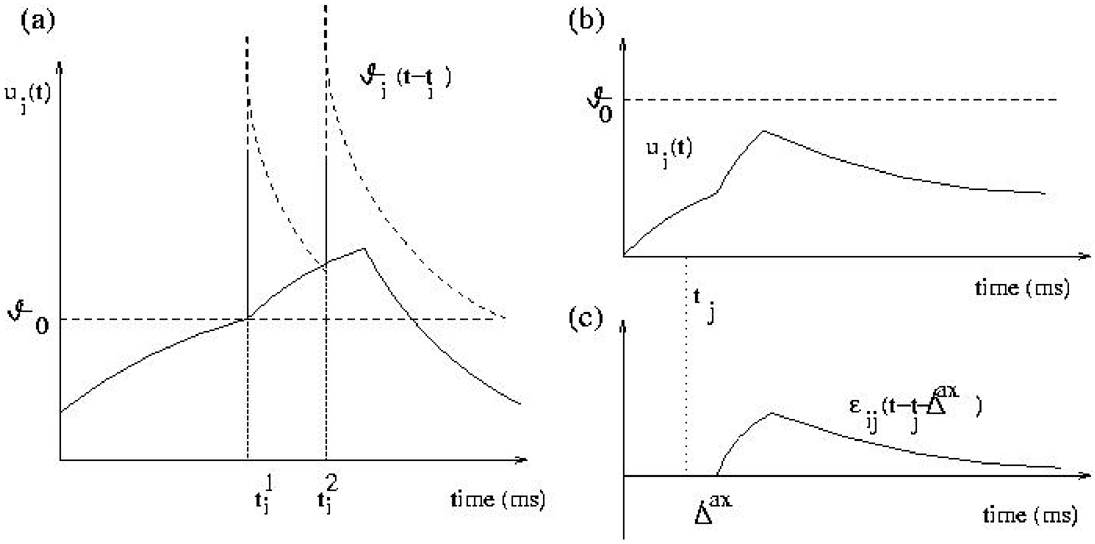


Figure 11.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics

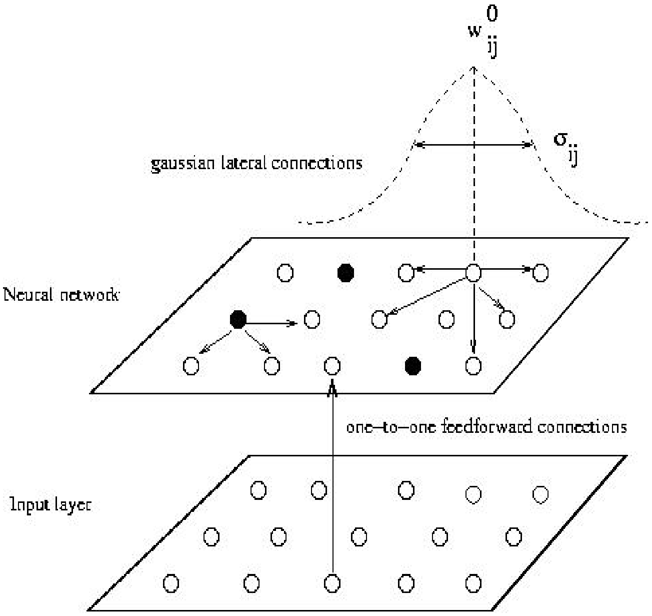
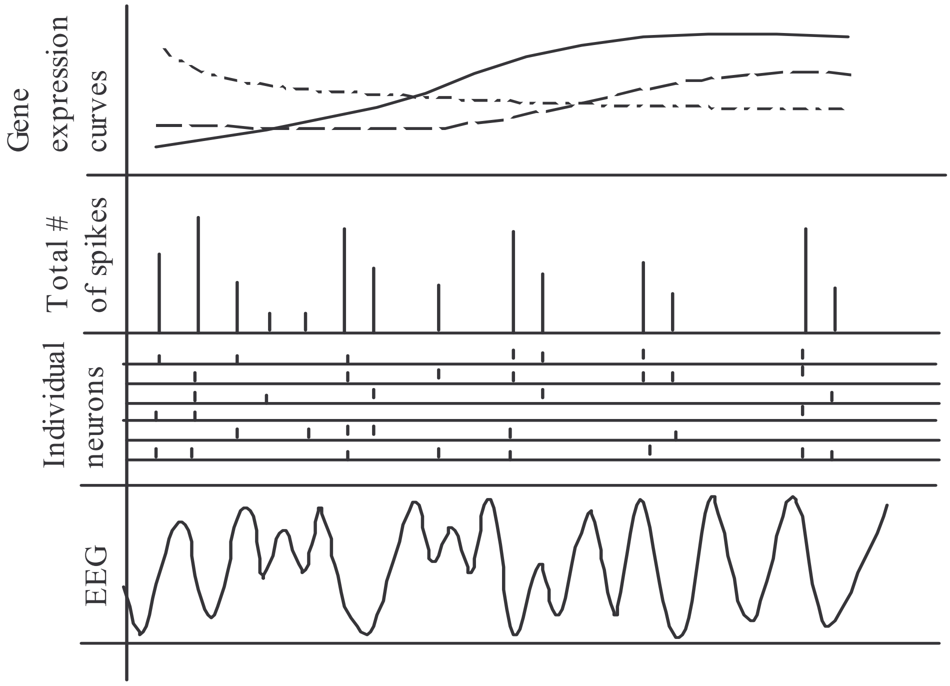

w_{ij}^0
σ_{ij}
gaussian lateral connections
Neural network
one–to–one feedforward connections
Input layer

Figure 12.

Nikola Kasabov and Lubica Benuskova

Computational Neurogenetics


Gene expression curves
Total # of spikes
Individual neurons
EEG