Project Summary

Intellectual Merit

The proposed work addresses fundamental questions regarding the computational capabilities of the central nervous systems of living organisms. The associated questions are fundamental to an integrative understanding of structure-function physiology and co-evolution in all organisms with nervous systems and the evolutionary predecessors that gave rise to them. With respect to nervous system organization, this project will deepen previous explorations into the interaction of developmental and environmental constraints on the architecture and complementary computational capacity of networks of living neurons. On the theoretical side, focusing on flexible abstract models of computation, rather than those specific to digital electronics, will enable the investigation of potentially novel computational potential implemented via architectures that differ from digital electronics in their apparent ability to take advantage of synergy between stochastic and deterministic properties. On the experimental side, cell cultured neurons will be developmentally selected for their capacity to robustly perform computations supporting various logical frameworks and the associated network architectures will be reverse engineered for further study. This investigation may, therefore, in addition to contributing to fundamental biological understanding, identify computational principles unique to natural neural networks. This project will embody the value of linking theoretical, computational and experimental methods of inquiry. This is necessary to address questions regarding co-evolution and co-development between underlying biological organization and computational tasks such structures are capable of performing in a variety of environmental conditions.

Broader impacts

Virtual interaction among all involved, including the public, will be fostered by an open online Wiki that will be used to organize and collaborate on this project and, more generally, support the movement for Open Notebook Science. All computer code will be made open source in accordance with the MIT license and the codebase history will be available to the public free and in real-time on github. The Wiki will contain tutorials on important concepts relating to the development of this project that will be accessible to motivated high-school students and undergraduates who may be considering interdisciplinary training for future careers in science. The foundations of this research and results obtained throughout its development will be incorporated directly into graduate courses at both the U.S.-based and Israel-based PIs respective institutions.

The Bergman and Moses labs both combine theory and experiment. They do so in different ways and exposure to each of these models will be crucial for the education of students and postdoctoral fellows as they move on to make important decisions in their careers. In the Bergman lab, three Ph.D. students will be involved in developing the theoretical aspect of this project in close interaction with the PI. In the Moses lab two students and a postdoctoral fellow will be involved. This culturally diverse and interdisciplinary team of experienced and developing scientists will enable the success of the research program. Extensive interaction among those involved in performing the research will be fostered by frequent video conferences and an exchange program that we plan to engage in at crucial theory-experiment integration stages throughout the project. This international collaboration will provide a foundation for future collaborations among the PIs, students and postdoctoral fellows involved.

Collaborative Research: Theory of computation in cell cultured neural networks

Contact information for the Israeli PI

Prof. Elisha Moses, Weizmann Institute of Science, +917-8-934-3139, elisha.moses@weizmann.ac.il

IOS Program Suitability

BIO/IOS Neural Systems NSF 11-572. We believe our proposed work is most appropriate for the IOS neural systems section as it deals with fundamental questions relating to the computational capabilities of the central nervous systems of living organisms. The associated questions we intend to address are fundamental to an integrative understanding of structure-function physiology and co-evolution in all organisms with nervous systems and the evolutionary predecessors that gave rise to them. With respect to nervous system *organization*, we intend to deepen previous explorations into the interaction of developmental and environmental constraints on the architecture and complementary computational capacity of networks of living neurons. Focusing on flexible abstract models of computation, rather than those specific to digital electronics, will enable us to investigate potentially novel computational potential implemented via architectures that differ from digital electronics in their apparent ability to take advantage of synergy between stochastic and deterministic properties. We may therefore, in addition to contributing to fundamental biological understanding, identify computational principles unique to natural neural networks. It is essential in this project to link theoretical, computational and experimental methods of inquiry. This is necessary to address questions we ask regarding co-evolution and co-development between underlying biological organization and computational tasks such structures are capable of performing in a variety of environmental conditions. We have assembled a culturally diverse and interdisciplinary team of experienced and developing scientists to enable the success of the research program we propose.

Research description

Motivating questions

Can we take steps toward understanding the fundamental organizing principles of neuronal architecture incorporating developmental, genetic, molecular, cellular and evolutionary perspectives that enable neuronal systems at the organism-environment interface to perform astoundingly complex computations that go beyond what digital computers are capable of in order to simultaneously improve our understanding of biological systems and deepen our perspective on and understanding of the very meaning of computation? The primary goal of this proposal is to dissect this broad question into questions that significantly push outward, without drastically overstepping, the boundaries of what is theoretically and experimentally feasible. One question regarding the structure and function of nervous systems is what features of their biological context can be abstracted and supplemented in the context of cell-cultured neurons that nevertheless enable retaining salient functional aspects of such natural neural networks. In particular, once one takes hippocampal neurons out of the rat brain and cultures them on a dish, they generally lose their individuality. Living neuronal cultures are characterized by all-or-none network bursts, where practically all the neurons fire simultaneously, with varying degrees of synchrony. While in the brain they can compute the trajectory of the animal in a maze, but as an ensemble grown out of it they seem unable to perform new computations, and can only carry very few bits of information [1]. How is it then that individual neurons are organized in the brain as a splendid computer, but in the dish their computational repertoire seems limited? Obviously, their environment and growth conditions have changed, but can we isolate those elements that are necessary, and perhaps sufficient, to support computation by a living neuronal network? How do such attempts strain or alter our very definition of *computation*? In an effort to tackle such questions, we have embarked on an investigation of the computational abilities of living neuronal networks, and propose here to expand this investigation both theoretically and experimentally.

On the conceptual side, a number of deep questions arise. What are the structural properties at the level of networks of neurons, modules of networks of neurons, and perhaps higher order forms of organization necessary to support the capacity for abstraction, which is generally considered to be fundamental to computation [2] and may likewise be fundamental to cognitive architecture, development, and function [3]? Given that we have shown boolean logic devices capable of being implemented *in vitro* using neurons [4], is it possible to biologically engineer analogous devices capable of performing computations that have natural embeddings within first or higher-order logic?

We plan to experimentally develop a number of novel *logical devices*, as well as use a number of new technologies that we have acquired for the construction of such biologically constituted computational devices. Notable among these are the optogenetic toolbox that has recently become available. Our access to this toolbox includes light induced neuronal excitation using channelrhodopsins (courtesy of the Deisseroth lab at Stanford and the Yizhar lab at Weizmann) as well as genetically encoded fluorescent labels for optical imaging of neuronal excitation (courtesy of the Cohen lab at Harvard).

Evolutionary forces have shaped the architecture, connectivity as well as the input that neurons grow with inside the brain, all of which are involved in its apparently emergent computational capability. Biological computation in general is set apart from that of electronic computers by the fact that the *hardware* (e.g. the cell or the organism) co-evolved with the *software* (DNA or the brain respectively). As a result, our *software* is naturally co-optimized for the associated *hardware*, while in the computer world that is not necessarily, if ever, the case. This insight leads us to propose experiments that will connect the neuronal network with the *real* world, and allow the structure and function to co-evolve in a range of different environmental conditions.

Natural **models of computation in living neural networks**

Since the development of the electronic digital computer, which makes use of the digital abstraction [5] from analog electrical circuits there has been a close heuristic association between boolean logic and computation. However, developments in formal logic over the past century have been largely motivated by its applications to computation and the theory of programming languages that go beyond boolean logic to support additional forms of abstraction. The capacity to support abstraction mirroring various systems of formal logic is the primary way in which languages are compared [2]. Electronic devices that are capable of supporting such forms of abstraction provide a concrete physical instantiation of the ideas inherent to the logical systems they are designed to faithfully implement.

Neuronal logic devices (NLDs) represent an alternative physical modality to digital electronics for the purpose of performing computation. However, rather than attempting to directly parallel the history of the development of digital electronics via the digital abstraction, high-level descriptions of computation, such as the λ-calculus, serve as a specification of computation that is agnostic to the physical modality of implementation. If the specification that a neuronal computation device should be capable of implementing the λ-calculus, which is a Turing-complete formal system, a natural first step toward this broad goal is to investigate simple neuronal systems that are capable of performing well-defined computations that require a capacity for first- or some higher-order logic.

One challenge to be addressed experimentally is whether such a system can be implemented reliably using central nervous system (CNS) neurons, which are individually susceptible to stochastic fluctuations and are hence considered to be *unreliable components*. We have shown previously that complex devices such as a diode, an oscillator and an AND-gate can be engineered using CNS neurons grown on particular geometric configurations [4]. While a complicated configuration of NAND gates could, in principle, enable the construction of a universal Turing machine, this may be difficult to engineer with previously developed methods. Alternative geometric configurations and measurement patterns coupled with developmental selection will enable the identification of configurations of neurons capable of performing more difficult computations.

To develop higher order computational devices, we will turn to the experimental techniques of microfluidics and optogenetics [6,7]. It is now possible to monitor with optical means the electrical activity of the network without any collateral damage caused by the fluorescent dye. This is done by genetic incorporation of a fluorescent voltage-indicating protein into the neuron. It is furthermore possible to excite a region of the network optically by activating photosensitive channels that are genetically embedded as well. On top of this, the propagation velocity of a signal inside a one-dimensional neuronal network of the type we are using is constant, and can be reliably predicted. Thus it becomes possible to identify activity in one part of a device, and then excite another region co-incidentally with the arrival of the signal into that area. Different time delays, with the activation before, during or after the arrival of the synaptic input will allow the creation of several neuronal learning scenarios, and the comparison to learning in organisms with brains.

A higher-order function to be implemented as a higher-order NLD

The λ-calculus notation is helpful in order to define any higher order function. This notation may be implicitly familiar to users of imperative programming languages as *anonymous functions*. We provide an informal set of examples necessary to clarify our work. Complete details can be found in [8]. We can define a standard binary boolean function like the AND—which is written using the "$\wedge$" symbol—function as $\lambda x.\lambda y.(\wedge\ x\ y)$. Such a lambda expression can apparently be applied to any inputs; however, the inputs to such a function are not necessarily restricted to booleans unless we infer that the standard logical operator $\wedge$ only accepts boolean arguments. Note that we have used the prefix or Polish notation for the $\wedge$ operator, which, in the perhaps more common infix notation, is written with its arguments flanking the operator as $x \wedge y$ to mean "x and y". We can provide explicit type annotations for the bound variables x and y, which indicate the types of the arguments $\lambda x : bool.\lambda y : bool.(\wedge\ x\ y)$. We can now also provide a type annotation for this expression as a whole

$$[\lambda x : bool.\lambda y : bool.(\wedge\ x\ y)] : [bool \rightarrow bool \rightarrow bool] \tag{1}$$

Depending upon conventions with respect to currying[1], one could read the type annotation as "the function that takes two arguments each of type *bool* and returns a values of type *bool*" or "the function that takes an argument of type *bool* and returns a function that takes an argument of type *bool* and returns a value of type *bool*". The first formulation may be easier to read, but the second is standard as a result of the design of functional programming languages.

Now we can imagine that if we wish to abstract from the particular binary boolean function implied by the $\wedge$ operator, we need to introduce a functional variable, whose type will be explicitly denoted for concreteness despite the fact that it could be inferred, for which this operator can be substituted. Doing this results in the following second order function

$$[\lambda f : (bool \to bool \to bool).\lambda x : bool.\lambda y : bool.(f\ x\ y)] \\ : [(bool \to bool \to bool) \to bool \to bool \to bool] \quad (2)$$

This function is intended to be read, in the less verbose uncurried form, as "the function that takes as its first argument, a function that takes two boolean values as arguments and returns a boolean value, and, as its second and third arguments, two boolean values, and returns a boolean value". It is perhaps remarkable that this simple abstraction is now capable of implementing any of the 16 possible boolean functions, provided that the proper binary boolean operator is submitted as the first argument to this function. The statement of this function in terms of λ-calculus is agnostic to any physical implementation capable of realizing extensionally equivalent behavior.

To make this more concrete, we can very simply implement the above function in any programming language. An implementation in the programming language OCaml appears in listing 1.

```
let hobf = fun (bf : ('a 'a bool)) (i1 : bool) (i2 : bool)
         -> bf i1 i2
```

Listing 1: a higher order boolean function

What is required in order to evaluate this function are corresponding implementations of boolean functions to be substituted either for f in the lambda calculus notation or for bf in terms of the corresponding OCaml implementation. For example we can implement the XOR function using pattern matching to define a truth table as shown in listing 2.

```
let xor p1 p2 = match (p1, p2)
with (false, false) -> false
   | (false,  true) -> true
   | ( true, false) -> true
   | ( true,  true) -> false
```

Listing 2: implementation of an XOR boolean operator

Other binary boolean functions are implemented in an analogous fashion. In order to evaluate hobf we then simply provide the name of a binary boolean function and two boolean values. For example $hobf\ xor\ 0\ 0 = 0$ and $hobf\ xor\ 0\ 1 = 1$.

[1] *currying* refers to the way in which a function taking multiple variables simultaneously as arguments and returning some value can be transformed into a sequence of functions that return functions as values such that when all the functions are composed they return the same value as the original multi-variable function.

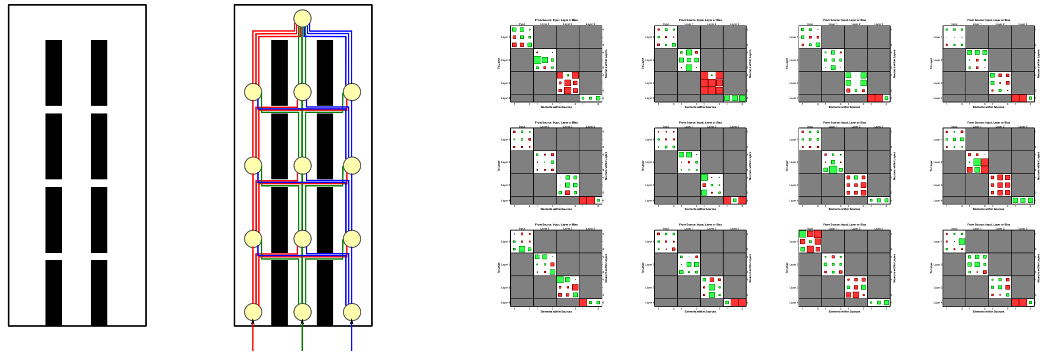
Assessment of abstraction potential in NLDs

An important consideration is to state precisely some criterion for determining that a particular NLD has achieved potential for an explicit form of abstraction such as is indicated in the relationship between expressions (1) and (2). In abstracting the binary boolean operator $\wedge$ to the functional variable f, which is the fundamental transformation enabling the derivation of (2) from (1), we imply that any physical implementation must take at least three rather than two inputs and the first of these must specify a particular binary boolean operator to apply to the latter two boolean input values. This roughly means that any system that at least partially implements the lambda expression or function specified in Equation (2) and Listing 1 respectively , must be capable of interpreting the concept of *selection from a set* whose size is determined by the subset of the 16 binary boolean operators that is already implemented in a lower-level form. Indeed another representation of Equation (2) as a partial or total set function could be written as $hobf : Hex \times Bool \times Bool \rightarrow Bool$ where we interpret $Bool$ and Hex as two and sixteen element sets respectively. This point of view makes clear that the capacity for selection from two element sets is already apparent in the binary boolean operator written as a set function $\wedge : Bool \rightarrow Bool$. The difference between these is that the system must implement the typing constraints necessary to distinguish a set that takes on sixteen possible values from one that takes on two. For example, in the application of the function $hobf$ the following evaluate as expected given the definition: $hobf\ \wedge\ 1\ 0 = 0$ or $hobf\ \wedge\ 1\ 1 = 1$. However, what is to be expected given inputs such as: $hobf\ 1\ \wedge\ 0$ or $hobf\ 1\ 1\ \wedge$? In these cases an output type intuitively associated to *Error* is required to indicate that the realization of a system implementing Equation (2) has indeed correctly implemented the necessary typing constraints to claim that the function has been realized. In the case of neuronal logic devices, this alternative output should differ in a measurable way from those associated to 1 and 0.

ANN simulation of abstract computation in NLDs

In order to facilitate the formation of a simple higher-order boolean function capable of switching between the AND and OR gate ($\wedge/\vee$) functionality as a result of the growth of patterned cultures of neurons, we designed a potential geometric constraint shown in Figure 1a to use in simulations of neural network development. We implemented one potential impact upon network connectivity in terms of an artificial neural network (ANN) model. Figure 1b shows a sampling of neural networks represented in terms of their respective Hinton diagrams[2] for the weights of 3-input 3-layer 1-output feedforward ANNs trained on data representing the $\wedge/\vee$ gate switching task and selected for near perfect performance on this task. From Figure 1b we see that it is not obvious if there are any general architectural features of ANNs that perform well in the $\wedge/\vee$ gate switching task by simply looking at representations of the network weight-bias matrices. Figure 2 shows a heatmap where neural networks are clustered according to their respective weights in order to determine the degree to which networks capable of performing the $\wedge/\vee$ gate switching task may be clustered into a reduced number of strategies despite the apparent diversity demonstrated in Figure 1b. We see from Figure 2 that, while there is considerable diversity in the 50 high-performing networks derived in our simulations, there are also relative similarities between neural

[2] Hinton diagrams are simply a method, analogous to a heat map, of visualizing numerical values from a matrix that are commonly used to display matrices of weights that comprise an ANN in machine learning. The area occupied by a given square in the diagram is proportional to the magnitude of a value in the weight matrix and the color (red for positive and green for negative in this case) indicates the sign of the weight.



(a) (b)

Figure 1: (a) Theoretical geometric constraint facilitating the formation of the ∧/∨ gate switch. The geometric constraint is shown on the left and the artificial neural network wiring diagram representing potential connections whose strength is modulated upon training of the network is shown on the right. (b) Hinton diagrams of feedforward neural network weights for neural networks trained on the and/or gate switching task. The only purpose of this panel is to demonstrate that there is no obvious pattern that distinguishes networks that all perform well in the ∧/∨ gate switching task. Clustering the weight-bias matrices that determine the neural networks under investagation is shown to reveal such patterns in Figure 2.

network designs that are all approximately equivalent in terms of their capacity to perform ∧/∨ gate switching. In particular these cluster into approximately three types that can be distinguished from one another by inspecting their weight-bias matrices, but that nevertheless perform the same function. The impact of this result on our experimental design is to account for the fact that even in natural neural networks, there is likely a degenerate relationship between structure and function, even if additional developmental or energetic constraints may favor a particular structure in an environmental context that requires performance of multiple different tasks exemplified simply by the ∧/∨ gate switching we have described to support clear exposition of the complicated issue of degeneracy in structure-function relationships as they pertain to networks of living and artificial neurons.

Experimental design

Engineering *flexible* living neuronal networks that will switch between several possible logical functions according to the input presented to them is a significant experimental challenge. Of course, designs that are possible and relevant theoretically such as the pattern of Figure 1a, pose several novel challenges in the experimental implementation of the network. In particular, the connectivity between different parts of the culture has to be precisely tailored to the specifications of the theoretical outlay. The first aspect of this is that the connections must, at times, cross each other without intersecting. The second is that the connections are all given a particular directionality, and the third is that a hierarchy of connections is assumed, with their importance changing in space.

The experimental approach to these difficulties is to use three novel techniques that have been introduced into the field recently, and include tapering geometric guidance, microfluidic 3D overlay

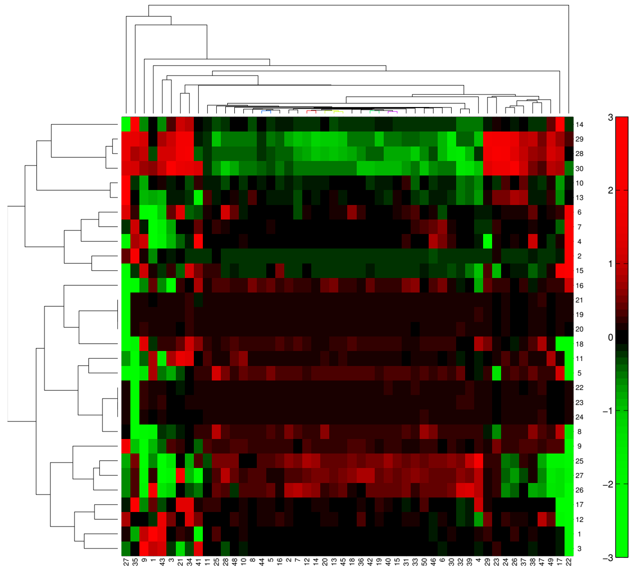
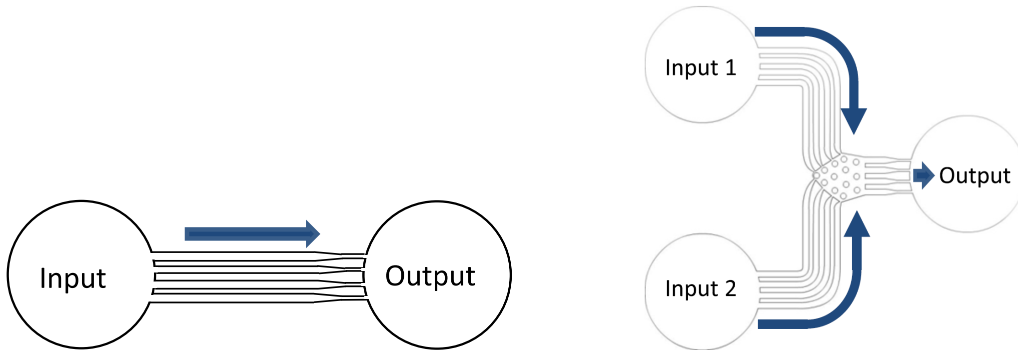


Figure 2: Theoretical geometric constraint facilitating the formation of the $\wedge/\vee$ gate switch. A heatmap that is clustered along rows representing individual weights at particular positions within the ANN and along columns according to the ANN from which the weights derive showing the relationship between low error strategies that emerge in training 50 independent ANNs on the $\wedge/\vee$ gate switching task.

and temporal patterning by sequential seeding of cultures at different times into different areas of the pattern.

Our solution to this variety of challenges will be based on the microfluidic environment that we have been developing in the laboratory for the growth of neuronal cultures. We have solved the aspect of directionality in previous work [4] by geometric patterning that funnels the growth of axons in one direction. The addition of the microfluidic construction is very useful for axonal guidance [9–11], only the axon can traverse a long thin channel. In addition, the tapering of the channel towards the *Output* section creates a high density that does not allow axons to traverse in the other direction. This is commonly used in conjunction with *temporal patterning*, where the *Input* section is seeded a day or two before the *Output* section. Neurons then develop first in the *Input* section and connections are made from the *Input* to the *Output*, with the axons traversing in one direction rather than the other. Figure 3a shows the basic structure in which a limited number of axons are allowed to propagate from the *Input* to the *Output* area. The tapering of the channels toward the end at which the *Output* is located has been shown [9] to eliminate the possibility of axonal crossing in the opposite direction. Figure 3b shows how two such constructs are united to create a simple AND gate. The array of columns at the entrance to the *Output* area serve to guide the axons and allow some variability and control over the amount of interaction that the axons undergo before entering the *Output*. In this way a transition can be precisely controlled between homogeneously and inhomogeneously distributed axons from *Inputs* 1 and 2 among the entrance channels into the *Output*.

More complex patterns, similar in spirit to the one shown in Figure 1a, necessitate a number of



(a) (b)

Figure 3: Schematic representation of tapered guidance microfluidic NLDs. (a) The basic *Threshold* element where the tapered guidance of axons from *Input* to *Output* is demonstrated. A limited number of axons cross into the *Output*, causing only amplitudes that are larger than a given threshold amplitude to be transferred into the *Output*. In (b) the principle of combining two of the devices into an AND gate is shown. The columns in the mixing region of the axons, before entering the *Input* region, allow for a controlled interaction and combination of axons from the two *Input* regions into the *Output* region thereby providing an experimental implementation of one of the ANN nodes in Figure 1a that can be replicated and connected to realize natural neural networks in terms of NLDs with arbitrary topologies.

crossover connections, where axons pass next to or across each other yet do not touch. These will be achieved by patterning an additional layer of microfluidic above the initial pattern, in which the axonal connections (for example from *Input* 1) can pass over the axons extending from another area (for example from *Input* 2). Although 3D overlay has not been used to date in neuronal cultures and is technically more demanding, the principles underlying its use and the technology [12] itself are similar to the standard 2D patterning that has been used in our laboratory. An innovative aspect of the theoretical design in Figure 4a, which has not been emphasized to date in the construction of NLDs, is the existence of multiple regions that serve as alternate *Output* areas. In particular this answers the need, raised in the theoretical part of this proposal, to differentiate the various *Input* patterns by the ability of the NLD to process them. Be it due to their temporal sequence or their spatial form, there will typically be a large set of *Inputs* that a given NLD is unable to process. In such cases we would like the NLD to classify the *Input* as unsuitable for processing, and produce a differentiated *Output* that can be interpreted as indicating an *Error*. We would thus expect a large number of input configurations to flow into this *Nonsense Input* type. The capacity to make type distinctions of *meaningless* from *meaningful* information is perhaps a more salient feature of computation than the capacity to match particular *Input/Output* pairs.

In addition, we expect to utilize *temporal patterning*, an important addition to our existing *spatial patterning* effort, to a greater degree and to rely on this novel approach more in the construction of NLDs. By controlling the sequence of seeding of neurons inside the NLD, we have already been able, in preliminary experiments, to get behavior that is very different from regular, same-time NLDs. This approach relies on the fact that neurons taken from embryonic rats or mice are promiscuous in their connections to other neurons from a different animal, a property that does not exist in the living animal where the immune system is expected to play an important role in allowing such connectivity (though this is overcome in our lab by the use of siblings from the same litter).

As in any successful experimental design effort, we expect a continuous back-and-forth process between the lab and the theory, with ideas for novel implementations of NLDs stemming from fundamental theoretical considerations being tested out in the lab. This will serve as an immediate reality check (within the limitations of experimental time scales, of course) for the conceptual ideas. Both the ability to construct such NLDs and the actual function in practice versus that predicted will be monitored in this way. We expect this type of collaboration of theory and experiment to be extremely conducive to obtaining efficient NLDs on one and deep insight into fundamental issues of biological computation on the other. Addressing the need to integrate theory and experiment in rapid feedback cycles is indeed one of the primary factors necessitating this particular international collaboration.

Higher-Order NLDs

A natural extension of the simple NLDs we have built is the addition of complexity to the network structures. Looking at our current technology, the basic one dimensional structure is designed as a feed-forward device, where the flow of information is linear and directed. Activity originates at so-called *Burst Initiation Zones* [13], and propagates from there along the linear configuration of the network. In the simplest current experimental one dimensional construction, axons propagate both forward and backward and thus resulting connections are in both directions. This implies that some recurrent connectivity is already a part of the simple linear structures that we have built. However, once we realize that the neuron that has fired has a delay time before it can fire again, we realize that the role of loops is small [1]. In particular, almost all feedback activity is eliminated in the *Diode* construction [3], which allows axons to go with strong preference in one direction. Using this engineering approach, we have built an oscillator with one closed circuit [4], which relies on the *Diode* construct and inherently disallows feedback activation. This drives the system into the basin of attraction of a limit cycle, and we have produced a periodic oscillation in this manner.

An additional level of complexity is that of networks incorporating loops, thereby allowing feedback as well as feed-forward activation, both processes that are of importance in the brain [14]. In general, for our experimental network, the fundamental object is the relation between two nodes, or a connection of two neurons. The simplest structure that describes such a network theoretically is the directed acyclic (tree-like) graph, which ignores the possibility of any connections looping back to create a closed circuit. Since a tree-like structure is more tractable analytically, many theoretical approaches utilize it. A deviation from the tree structure involves allowing for cycles or loops.

In a random graph the probability for closing a small, three node triangular loop is small [15], but in real world networks [16] the appearance of loops is the standard rather than a rarity. Loops in the neurobiological context imply the existence of recurrent connections, which are assumed to be highly represented in the brain, linking brain levels that interact both directly and inversely. This is presumably crucial for several forms of computation, in particular models combining both top-bottom and bottom-up processes in the brain.

The approach we are proposing is based upon the relation of structure and function in the computational ensemble or organism. Loops are the dominant and characteristic sub-structure and carry most of the functionality in many complex networks [17]. To incorporate loops that allow transfer

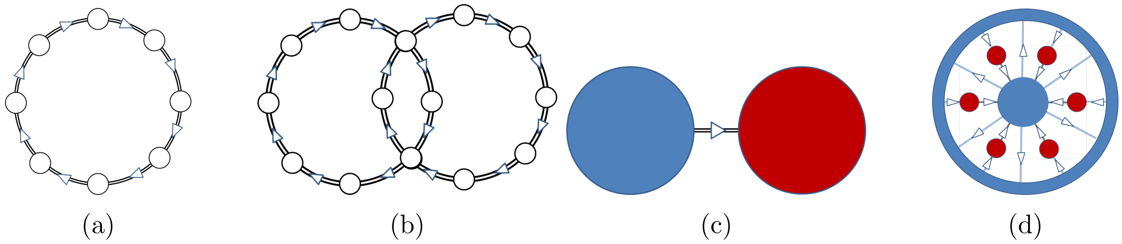


Figure 4: (a) Schematic of an 8-length loop device in which several sub-networks interact in a uni-directional, ordered mode. Neurons are seeded on the circular areas. After an initial growth period in which axons connect to neurons within the circle, axons will begin to propagate in the direction dictated by the *Diode* structure. This enforces a directional connection between the sub-networks. (b) The simple doubling and reconnection of an 8-length loop device creates several new loops of differing lengths. The dynamics is hard to predict *a priori*, but will present an excellent testing ground for theoretical approaches and models of computation. (c) Schematic diagram of a uni-directional link between two types of neuronal populations. For example, in this particular configuration, the blue culture could be comprised of sensory neurons taken from the peripheral nervous system, while the red culture indicates neurons taken from the central nervous system. (d) Schematic diagram of a loop incorporating the two types of neuronal populations. Here the time sequence is controlled by the *Diode* links between the cultures, which can serve as significant delay lines. The relative size of the different areas must be appropriate for the types of neurons employed and the functionality to be explored in the developmental selection process.

of information both back and forth, we plan to introduce two experimental configurations, both of which depend on constructing the network from a few subsets of neurons that interact.

The first construct is based on our ability to separate different parts of the network in space, and to control the direction of connectivity between them. It has been shown that by patterning a network so that different areas are active separately, a non-trivial array of interesting oscillatory activity can be observed [18]. The simplest experimental design is one in which several patches are situated on a circle comprised of a thin line along *Diode* structures. Neurons are seeded on the patches, and their axons first interact within the patch, and then propagate out, in the one *forward-biased* direction that the *Diode* allows. This is exemplified in the simple schematic of Figure 4a, which is reminiscent of a neuronal construction [19] that has been shown to support recurrent and persistent neuronal activity.

It is straightforward to see that this kind of loop structure can be generalized, and that the ensuing dynamics will display a wide variety of fascinating nonlinear behavior. The configuration shown in Figure 4b demonstrates that just by repeating the same 8-length loop structure and reconnecting, three new deterministic loops are created, of lengths 4, 8 and 8, along with a longer loop consisting of 12 nodes. The different time scales associated with propagation in each of the different loop lengths significantly complicates the dynamics.

The second experimental configuration we intend to employ is that of combining two different neuronal populations that will interact with each other. This will be an exploratory aspect, and we will have to experiment with a variety of different neuronal types. The idea is simply depicted in Figure 4c, where one type of neuronal network is connected to a second type.

This basic configuration assumes that one type of neuronal culture is more typically an output provider, for example the sensory neurons from the peripheral nervous system (PeNS) in the body that typically send their output into the central nervous system (CNS). Loop closure in such a system can be accomplished with a number of interesting patterns, for example the configuration of Figure 4d. Note that the thin lines linking the different nodes provide a delay line, so that there is a temporal sequence superimposed on the spatial pattern.

Several technical challenges arise in this project. First, the seeding of different neuronal types implies that microfluidic devices are the system of choice. Feeding channels can convey the neurons into the respective areas for either neuron along separate channels that can reside in a feeding layer that is separated vertically from the culturing layer. Second, the choice of neuronal types is a matter for continual exploration. While the harvesting of peripheral neurons is simple, their interaction with the CNS neurons is far from predictable, and the results from such a co-culture are not known. We also plan to separate different types of populations using genetic parking and flow cytometry using fluorescence-activated cell sorting (FACS). While this is clearly feasible, the separation procedure will necessarily introduce a delay on the order of an hour between harvesting and seeding of cells. We will measure the impact of this process on the survival of the neuronal types we employ.

The introduction of several different neuronal networks made of different neuronal types that communicate, interact and feedback onto each other addresses a deep problem of improving understanding of the self-assembly of computational devices in terms of living neuronal networks. The failure of a cultured neuronal network to produce a valid computation is associated with the fact that it is grown out of context, out of its natural surroundings. We have shown that geometrical constraints can coax some of the networks connections to go in preferred directions, restoring a modicum of computation [4]. However, in fact the neurons are growing out of their natural context, and the input that they are exposed to during the growth process is not the one that naturally leads them to become capable of performing the computations that are intuitively associated to cognitive and other neuronal systems. We could say that there is a *software* growing inside the network, but that this software does not have the needed input from its hardware to guide it towards a *meaningful program*.

From NLDs to the CNS and principles of cognition

The plan to co-culture different sets of neurons along with the regular hippocampal ones is thus the natural solution and a first step in creating a *hardware-software* linkage in a neuronal culture that better approximates natural neuronal systems while remaining experimentally tractable. The first choice is that of sensory neurons, since those are the subset of neurons that communicate between the body of the organism and its brain. In this way we hope to recreate an information channel that exists in the developing brain, allowing neurons to attain functional connectivity according to the inputs it receives.

The general theoretical relevance of loops to the construction of computation will lead us to address the issue of self-reference in computational systems, a problem that is both deep and stimulating [20]. From the point of view that identifies the capacity for abstraction with computation, the investigation of neuronal logic devices capable of such function provides a framework in which various hypotheses from cognitive science could begin to be evaluated at the level of well-defined neural circuits. By attempting to isolate minimal implementation criteria, this approach may serve to complement, enable simpler explanations of, or identify paradoxical results derived from studies that treat whole brains and their associated sensory apparatus as their object of study [21, 22].

Role and expertise of the PIs

Both Profs. Aviv Bergman and Elisha Moses have significant experience working in collaborative teams of experimentalists and theorists. The PIs intend this project to involve strong interaction between experiment and theory. Prof. Bergman will lead the theoretical component of this project. Prof. Bergman's expertise spans a number of theoretical areas including artificial neural networks, dynamical systems, mathematical evolutionary biology, and systems biology. The Bergman lab will develop the computational platform and analytical tools to predict neuronal architecture-function relationships, which in turn will help guide the construction of environmental conditions to guide the development of natural neural networks capable of performing fundamental computational tasks. Prof. Moses will lead the experimental component of the project, heading a laboratory that focuses on the growth and measurement of neuronal activity in networks grown from CNS neurons from rat and mouse brain. The lab has pioneered the design of complex neuronal logical devices, has generated several new experimental paradigms combining nonlinear dynamics and statistical physics with biological physics, and has participated in the formation of novel theoretical models for engineered neuronal networks.

Educational involvement

In the Bergman lab, three Ph.D. students (Mr. Cameron Smith, Mr. Daniel Biro and Mr. Ximo Pechaun) will be involved in developing the theoretical aspect of this project in close interaction with the PI. In the Moses lab two students (1 Ph.D., Ms. Shani Stern and 1 M.Sc., to be hired) and a postdoctoral fellow (Dr. Yaron Penn) will be involved. Students and postdoctoral fellows in the Bergman and Moses labs will interact on a weekly basis via video conference. More extensive interaction will be fostered by an exchange program that we plan to engage in at crucial theory-experiment integration stages throughout the project. Virtual interaction among all involved, including the public, will be fostered by an open online Wiki that will be used to organize and collaborate on this project and, more generally, support the movement for Open Notebook Science. All computer code will be made open source in accordance with the MIT license and the codebase history will be available to the public free and in real-time on github. Despite the fact that both labs already combine theory and experiment, they do so in very different ways and exposure to each of these models for combining theory and experiment will be crucial for these students as they move on to make very important decisions in their careers with respect to postdoctoral training and ultimately building labs of their own.

References

[1] Ofer Feinerman and Elisha Moses. Transport of information along unidimensional layered networks of dissociated hippocampal neurons and implications for rate coding. *The Journal of neuroscience*, 26(17):4526–34, April 2006. 2, 9

[2] Harold Abelson and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs - 2nd Edition (MIT Electrical Engineering and Computer Science)*. The MIT Press, 1996. 2

[3] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to Grow a Mind: Statistics, Structure, and Abstraction. *Science*, 331(6022):1279–1285, March 2011. 2

[4] Ofer Feinerman, Assaf Rotem, and Elisha Moses. Reliable neuronal logic devices from patterned hippocampal cultures. *Nature Physics*, 4(12):967–973, October 2008. 2, 3, 7, 9, 11

[5] Stephen A. Ward and Robert H. Halstead. *Computation Structures*. The MIT Press, 1989. 2

[6] Ofer Yizhar, Lief E Fenno, Thomas J Davidson, Murtaza Mogri, and Karl Deisseroth. Optogenetics in neural systems. *Neuron*, 71(1):9–34, July 2011. 3

[7] Joel M Kralj, Adam D Douglass, Daniel R Hochbaum, Dougal Maclaurin, and Adam E Cohen. Optical recording of action potentials in mammalian neurons using a microbial rhodopsin. *Nature methods*, 9(1):90–5, January 2012. 3

[8] H.P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*. North Holland, 1985. 3

[9] Jean-Michel Peyrin, Bérangère Deleglise, Laure Saias, Maéva Vignes, Paul Gougis, Sebastien Magnifico, Sandrine Betuing, Mathéa Pietri, Jocelyne Caboche, Peter Vanhoutte, Jean-Louis Viovy, and Bernard Brugg. Axon diodes for the reconstruction of oriented neuronal networks in microfluidic chambers. *Lab on a chip*, 11(21):3663–73, November 2011. 7

[10] Jeong Won Park, Behrad Vahidi, Anne M Taylor, Seog Woo Rhee, and Noo Li Jeon. Microfluidic culture platform for neuroscience research. *Nature protocols*, 1(4):2128–36, January 2006. 7

[11] Anne M Taylor, Mathew Blurton-Jones, Seog Woo Rhee, David H Cribbs, Carl W Cotman, and Noo Li Jeon. A microfluidic culture platform for CNS axonal injury, regeneration and transport. *Nature methods*, 2(8):599–605, August 2005. 7

[12] Assaf Rotem, Adam R Abate, Andrew S Utada, Volkert Van Steijn, and David a Weitz. Drop formation in non-planar microfluidic devices. *Lab on a chip*, 12(21):4263–8, October 2012. 8

[13] Ofer Feinerman, Menahem Segal, and Elisha Moses. Identification and dynamics of spontaneous burst initiation zones in unidimensional neuronal cultures. *Journal of neurophysiology*, 97(4):2937–48, April 2007. 9

[14] Ehud Ahissar and David Kleinfeld. Closed-loop neuronal computations: focus on vibrissa somatosensation in rat. *Cerebral cortex (New York, N.Y. : 1991)*, 13(1):53–62, January 2003. 9

[15] a Vázquez, R Dobrin, D Sergi, J-P Eckmann, Z N Oltvai, and a L Barabási. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(52):17940–5, December 2004. 9

[16] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the World Wide Web. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9):5825–9, April 2002. 9

[17] Jean-Pierre Eckmann, Elisha Moses, and Danilo Sergi. Entropy of dialogues creates coherent structures in e-mail traffic. *Proceedings of the National Academy of Sciences of the United States of America*, 101(40):14333–7, October 2004. 9

[18] Mark Shein Idelson, Eshel Ben-Jacob, and Yael Hanein. Innate synchronous oscillations in freely-organized small neuronal circuits. *PloS one*, 5(12):e14443, January 2010. 10

[19] Ashwin Vishwanathan, Guo-Qiang Bi, and Henry C Zeringue. Ring-shaped neuronal networks: a platform to study persistent activity. *Lab on a chip*, 11(6):1081–8, March 2011. 10

[20] David Levary, Jean-Pierre Eckmann, Elisha Moses, and Tsvi Tlusty. Loops and Self-Reference in the Construction of Dictionaries. *Physical Review X*, 2(3):1–10, September 2012. 11

[21] James L McClelland, Matthew M Botvinick, David C Noelle, David C Plaut, Timothy T Rogers, Mark S Seidenberg, and Linda B Smith. Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends in cognitive sciences*, 14(8):348–56, August 2010. 12

[22] Thomas L Griffiths, Nick Chater, Charles Kemp, Amy Perfors, and Joshua B Tenenbaum. Probabilistic models of cognition: exploring representations and inductive biases. *Trends in cognitive sciences*, 14(8):357–64, August 2010. 12