bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Brain computation by assemblies of neurons

Christos H. Papadimitriou* Santosh S. Vempala† Daniel Mitropolsky‡
Michael Collins§ Wolfgang Maass¶

December 8, 2019

Abstract

Assemblies are large populations of neurons believed to imprint memories, concepts, words and other cognitive information. We identify *a repertoire of operations* on assemblies. These operations correspond to properties of assemblies observed in experiments, and can be shown, analytically and through simulations, to be realizable by generic, randomly connected populations of neurons with Hebbian plasticity and inhibition. Operations on assemblies include: projection (duplicating an assembly by creating a new assembly in a downstream brain area); reciprocal projection (a variant of projection also entailing synaptic connectivity from the newly created assembly to the original one); association (increasing the overlap of two assemblies in the same brain area to reflect cooccurrence or similarity of the corresponding concepts); merge (creating a new assembly with ample synaptic connectivity to and from two existing ones); and pattern-completion (firing of an assembly, with some probability, in response to the firing of some but not all of its neurons). Our analytical results establishing the plausibility of these operations are proved in a simplified mathematical model of cortex: a finite set of brain areas each containing n excitatory neurons, with *random* connectivity that is both recurrent (within an area) and afferent (between areas). Within one area and at any time, only k of the n neurons fire — an assumption that models inhibition and serves to define both assemblies and areas — while synaptic weights are modified by Hebbian plasticity, as well as homeostasis. Importantly, all neural apparatus needed for the functionality of the assembly operations is created on the fly through the randomness of the synaptic network, the selection of the k neurons with the highest synaptic input, and Hebbian plasticity, without any special neural circuits assumed to be in place. Assemblies and their operations constitute a computational model of the brain which we call the *Assembly Calculus,* occupying a level of detail intermediate between the level of spiking neurons and synapses, and that of the whole brain. As with high-level programming languages, a computation in the Assembly Calculus (that is, a coherent sequence of assembly operations accomplishing a task) can ultimately be reduced — "compiled down" — to computation by neurons and synapses; however, it would be far more cumbersome and opaque to represent the same computation that way. The resulting computational system can be shown, under assumptions, to be in principle capable of carrying out arbitrary computations. We hypothesize that something like it may underlie higher human cognitive functions such as reasoning, planning, and language. In particular, we propose a plausible brain architecture based on assemblies for implementing the syntactic processing of language in cortex, which is consistent with recent experimental results.

*Columbia University. christos@columbia.edu
†Georgia Tech. vempala@gatech.edu
‡Columbia University. dgm2144@columbia.edu
§Columbia University. mc3354@columbia.edu
¶TU Graz. maass@igi.tugraz.at

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

1 Introduction

How does the brain beget the mind? How do molecules, cells, and synapses effect cognition, behavior, intelligence, reasoning, language? The remarkable and accelerating progress in neuroscience, both experimental and theoretical-computational, does not seem to bring us closer to an answer: the gap is formidable, and seems to necessitate the development of new conceptual frameworks. As Richard Axel recently put it [1] *"we do not have a logic for the transformation of neural activity into thought and action. I view discerning [this] logic as the most important future direction of neuroscience"*.

What kind of formal system, embodying and abstracting the realities of neural activity, would qualify as the sought "logic"?

We propose a formal computational model of the brain based on *assemblies of neurons;* we call this system the *Assembly Calculus*. In terms of detail and granularity, the Assembly Calculus occupies a position intermediate between the level of individual neurons and synapses, and the level of the whole brain models useful in cognitive science, e.g. [30, 31].

The basic elementary object of our system is the *assembly of excitatory neurons*. The idea of assemblies is, of course, not new. They were first hypothesized seven decades ago by Donald O. Hebb [20] to be densely interconnected sets of neurons whose loosely synchronized firing in a pattern is coterminous with the subject thinking of a particular concept or idea. Assembly-like formations have been sought by researchers during the decades following Hebb's prediction, see for example [2], until they were clearly identified more than a decade ago through calcium imaging [17, 5]. More recently, assemblies (sometimes called *ensembles*) and their dynamic behavior have been studied extensively in the animal brain, see for example [27].

Our calculus outfits assemblies with certain *operations* that create new assemblies and/or modify existing ones: *project, reciprocal-project, associate, merge,* and a few others. These operations reflect known properties of assemblies observed in experiments, and they can be shown, either analytically or through simulations (more often both), to result from the activity of neurons and synapses. In other words, the high-level operations of this system can be "compiled down" to the world of neurons and synapses — a fact reminiscent of the way in which high-level programming languages are translated into machine code.

Model. Our mathematical results, as well as most of our simulation results, employ a simplified and analytically tractable yet plausible model of neurons and synapses. We assume a finite number of brain areas denoted A, B, C, etc., intended to stand for an anatomically and functionally meaningful partition of the cortex (see Figure 2). Each area contains a population of n excitatory neurons[1] with *random* recurrent connections. By this we mean that each ordered pair of neurons in an area has the same small probability p of being connected by a synapse, independently of what happens to other pairs — this is a well studied framework usually referred to as *Erdős–Renyi graph* or $G_{n,p}$ [11]. In addition, for certain ordered pairs of areas, say (A, B), there are random *afferent* synaptic interconnections from A to B; that is, for every neuron in A and every neuron in B there is a chance p that they are connected by a synapse.[2]

We use this model to fathom quantitatively the creation and manipulation of assemblies of

[1] Our assumption that all areas contain the same number of neurons is one of expository, not mathematical, convenience, and provides the basic numerical parameter n of our analysis. A slightly more complex version would have parameters n_A, and k_A for each area A.

[2] Several experimental results [37, 16] suggest deviations of the synaptic connectivity of the animal brain from the uniformity of $G_{n,p}$. We discuss how such deviations affect — mostly support — our framework in the discussion section.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

neurons. Since the model is probabilistic (by virtue of the random synaptic connectivity), our analytical results postulating the effectiveness of the various operations must contain the clause "with high probability," where the event space is implicitly the underlying random graph. We assume that all cells in an assembly x belong to the same brain area, denoted $\texttt{area}(x)$.

Our model also encompasses simplified forms of *plasticity and inhibition.* We assume multiplicative Hebbian plasticity: if at a time step neuron i fires and at the next time step neuron j fires, and there is a synapse from i to j, the weight of this synapse is multiplied by $(1+\beta)$, where $\beta > 0$ is the final parameter of our model (along with n, p, and k). Larger values of the plasticity coefficient β result in the operations converging faster, and render many of our proofs simpler. Finally, we model inhibition and excitatory–inhibitory balance by postulating that neurons fire in discrete time steps, and *at any time only a fixed number k of the n excitatory neurons in any area fire;* in particular, those neurons which receive the k highest excitatory inputs.

The four basic parameters of our model are these: n (the number of excitatory neurons in an area, and the basic parameter of our model); p (the probability of recurrent and afferent synaptic connectivity); k (the maximum number of firing neurons in any area); and the plasticity coefficient β. Typical values of these parameters in our simulations are $n = 10^7, p = 10^{-3}, k = 10^4, \beta = 0.1$. We sometimes assume that k is (a small multiple of) the square root of n; this extra assumption seems compatible with experimental data, and yields certain interesting further insights.

Our model, as described so far, would result, through plasticity, in gigantic synaptic weights after a long time of operation. We further assume that synaptic weights are renormalized, at a slower time scale, so that the sum of presynaptic weights at each neuron stays relatively stable. This process of *homeostasis through renormalization* is orthogonal to the phenomena we describe here, and it interferes minimally with our arguments and simulations.

We emphasize that our model is *generic* in the sense that it is not assumed that circuits specific to various tasks are already in place. Its functionality — the needed apparatus for each task such as implementing an assembly operation — emerges from the randomness of the network and the selection of the k neurons with highest synaptic input as an almost certain consequence of certain simple events — such as the repeated firing of an assembly.

Assembly projection. How do assemblies in the association cortex come about? It has been hypothesized (see e.g. [34]) that an assembly imprinting, for example, a familiar face in a subject's medial temporal lobe (MTL) is created by the *projection* of a neuronal population, perhaps in the inferotemporal cortex (IT), encoding this face as a whole object. By *projection* of an assembly x we mean the creation of an assembly y in a downstream area that can be thought of as a "copy" of x, and such that y will henceforth fire every time x fires.

How is the new assembly y formed in a downstream area B by the repeated firing of x in area A? The process was vividly predicted in the discussion section of [14], for the case in which A is the olfactory bulb and B the piriform cortex. Once x has fired once, synaptic connectivity from area A to area B excites many neurons in area B. Inhibition will soon limit the firing in area B to a smaller set of neurons, let us call it y_1, consisting in our framework of k neurons (see Figure 1A). Next, the simultaneous firing of x and y_1 creates a stronger excitation in area B (one extra reason for this is *plasticity*, which has already strengthened the connections from x to y_1), and as a result a new set of k neurons from area B will be selected to fire, call it y_2. One would expect y_2 to overlap substantially with y_1 — this overlap can be calculated in our mathematical model to be roughly 50% for a broad range of parameters. If x continues firing, a sequence $\{y_t\}$ of sets of neurons of size k in area B will be created. For a large range of parameters and for high enough plasticity, this process can be proved to converge exponentially fast, with high probability, to create

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

an assembly y, the result of the projection.

We denote the projection process described above as $\texttt{project}(x, B, y)$. Assembly projection has been demonstrated both analytically [24, 29] and through detailed simulations [33, 26]; simulation results in our model, as well as improved analysis, are presented in Section 2, Figure 1.B4. Once the $\texttt{project}(x, B, y)$ operation has taken place, we write $B = \texttt{area}(y)$ and $x = \texttt{parent}(y)$.

Dense intraconnection of assemblies. Hebb [20] hypothesized that assemblies are *densely intraconnected* — that is, the chance that two neurons have a synaptic connection is significantly larger when they belong to the same assembly than when they do not — and our analysis and simulations verify this hypothesis (see Figure 1C). From the point of view of computation this is rather surprising, because the problem of finding a dense subgraph of a certain size in a sparse graph is a known difficult problem in computer science [12], and thus the very existence of an assembly may seem surprising. How can the brain deal with this difficult computational problem? The creation of an assembly through projection as outlined in the previous paragraphs provides an explanation: Since the elements of assembly y were selected to have strong synaptic input from the union of x and y, one intuitively expects the synaptic recurrent connectivity of y to be higher than random. In addition, the weights of these synapses should be higher than average because of plasticity.

The RP&C primitive. It can be said that assembly operations, as described here, are powered exclusively by two forces known to be crucial for life more generally: *randomness* and *selection* (in addition to plasticity). No special-purpose neural circuits are required to be in place; all that is needed is *random synaptic connectivity* between, and recurrently within, areas; and selection, through inhibition in each area, of the k out of n cells currently receiving highest synaptic input. All assembly computation described here consists of applications of this operator, which we call *random projection and cap (RC&P).* We believe that RP&C is an important primitive of neural computation, and computational learning more generally, and can be shown to have a number of potentially useful properties. For example, we establish analytically and through simulations that RP&C preserves overlap of assemblies remarkably well (as first noticed empirically in [7]). Finally, we used RP&C in experiments as the nonlinearity in each layer of a deep neural network, in the place of the sigmoid or the ReLU, and we found that it seems to perform at a level competitive with these.

Association and pattern completion. In a recent experiment [21], electrocorticography (eCoG) recordings of human subjects revealed that a neuron in a subject's MTL consistently responding to the image of a particular familiar place — such as the Pyramids — starts to also respond to the image of a particular familiar person — say, the subject's sibling — once a combined image has been shown of this person in that place. A compelling parsimonious explanation of this and many similar results is that two assemblies imprinting two different entities adapt to the cooccurrence, or other observed affinity, of the entities they imprint by increasing their overlap, with cells from each migrating to the other while other cells leave the assemblies to maintain its size to k; we say that the two assemblies are *associating* with one another. The association of two assemblies x and y *in the same brain area* is denoted $\texttt{associate}(x, y)$, with the common area $\texttt{area}(x) = \texttt{area}(y)$ implicit. We can show analytically and through simulations that the simultaneous sustained firing of the two parents of x and y does effect such increase in overlap, while similar results have been obtained by simulations of networks of spiking neurons through STDP [33].

Assemblies are large and in many ways random sets of neurons, and as a result any two of

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

them, if in the same area, may overlap a little by chance. If the assembly size k is about the square root of n, as we often assume in our simulations and quantitative analysis, this random overlap should be at most a very small number of neurons. In contrast, overlap resulting from the operation $\texttt{associate}(x, y)$ is quite substantial: the results of [21] suggest an overlap between *associated* assemblies in the MTL of about $8 - 10\%$ of the size of an assembly. The association between assemblies evokes a conception of a brain area as the arena of complex association patterns between the area's assemblies; for a discussion of certain mathematical aspects of this view see [3].

One important and well studied phenomenon involving assemblies is *pattern completion:* the firing of the whole assembly x in response to the firing of a small subset of its cells [27]; presumably, such completion happens with a certain a priori probability depending on the particular subset firing. In our experiments pattern completion happens in a rather striking way, with small parts of the assembly being able to complete very accurately the whole assembly (see Figure 1.F1 and F2).

We believe that association and pattern completion open up fascinating possibilities for a genre of probabilistic computation through assemblies, a research direction which should be further pursued.

Merge. The most sophisticated and complex operation in the repertoire of the Assembly Calculus is *merge.* Denoted $\texttt{merge}(x, y, A, z)$, this operation starts with two assemblies x and y, in different brain areas, and creates a new assembly z, in a third area A, such that there is ample synaptic connectivity from x and y to z, and also vice versa, from z to both x and y.

Linguists had long predicted that the human brain is capable of combining, in a particularly strong sense, two separate entities to create a new entity representing this specific combination [4, 19], and that this ability is *recursive* in that the combined entity can in turn be combined with others. This is a crucial step in the creation of the hierarchies (trees of entities) that seem necessary for the syntactic processing of language, but also for hierarchical thinking more generally (e.g., deduction, discourse, planning, story-telling, etc.). Recent fMRI experiments [40] have indicated that, indeed, the completion of phrases and sentences (the completion of auditory stimuli such as "hill top" and "ship sunk") activates parts of Broca's area — in particular, the pars opercularis BA 44 for phrases, and the pars triangularis BA 45 for sentences. In contrast, unstructured word sequences such as "hill ship" do not seem to activate Broca's area. Recall that Broca's area has long been believed to be implicated in the syntactic processing of language.

A parsimonious explanation of these findings is that phrases and sentences are represented by assemblies in Broca's area that are the results of the `merge` of assemblies representing their constituents (that is, assemblies for words such as "ship" and "sunk"); presumably the word assemblies reside in Wernicke's area implicated in word selection in language. As these hierarchies need to be traversed both in the upward and in the downward direction (e.g., in the processes of language parsing and language generation, respectively), it is natural to assume that merge must have *two-way connections* between the new assembly and the constituent assemblies.

Our algorithm for implementing `merge`, explained in Section 2, is by far the most complex in this work, as it involves the coordination of *five* different brain areas with ample reciprocal connectivity between them, and requires stronger plasticity than other operations; our simulation results are reported in Section 2, see also Figure 1.G1 –G2.

Finally, a simpler operation with similar yet milder complexity is $\texttt{reciprocal.project}(x, A, y)$: It is an extension of $\texttt{project}(x, A, y)$, with the additional functionality that the resulting y has strong backward synaptic connectivity to x. Reciprocal projection has been hypothesized to be instrumental for implementing *variable binding* in the Brain — such as designating "cats" as the subject of the sentence "cats chase mice," see [26]. The plausibility of `reciprocal.project` has

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

been experimentally verified through detailed simulations of networks of spiking neurons with STDP [26], as well as in our simplified model.

Readout and control operations. The purpose of the Assembly Calculus is to provide a formal system within which high level cognitive behavior can be expressed. Ultimately, we want to be able to write meaningful *programs* — in fact, parallel programs — in this system, for example containing segments such as:

`if read`(A) `is null then project`(x, A, y).

With this goal in mind, we next introduce certain additional low-level *control operations,* sensing and affecting the state of the system.

First, a simple *read* operation. In an influential review paper on assemblies, Buzsáki [5] proposes that, for assemblies to be functionally useful, readout mechanisms must exist that sense the current state of the assembly system and trigger appropriate further action. Accordingly, the Assembly Calculus contains an operation `read`(A) that identifies the assembly which has just fired in area A, and returns `null` othewise.

Finally, the Assembly Calculus contains two simple control operations. We assume that an assembly x in an area A can be explicitly caused to fire by the operation `fire`(x). That is to say, at the time an assembly x is created, a mechanism is set in place for igniting it; in view of the phenomenon of pattern completion discussed above, in which firing a tiny fragment of an assembly leads to the whole assembly firing, this does not seem implausible. We also assume that by default the excitatory cells in an area A are inhibited, unless explicitly disinhibited by the operation `disinhibit`(A) for a limited period of time, whose end is marked by the operation `inhibit`(A); the plausibility of the disinhibition–inhibition primitives is argued in [25] in terms of VIP cells [18].

Example: For a simple example of a program in the assembly calculus, the command `project`(x, B, y) where `area`$(x) = A$, is equivalent to the following:

```
disinhibit(B)
repeat T times:  fire(x)
inhibit(B)
```

Simulations show that, with typical parameters, a stable assembly is formed after about $T = 10$ steps. An alternative version of this program, relying on the function `read`, is the following:

```
disinhibit(B)
repeat fire(x) until read(B) is not null
inhibit(B)
```

For another simple example, the command `associate`(x, y) whose effect is to increase the overlap of two assemblies in the same area A is tantamount to this:

```
disinhibit(A)
repeat T times:  do in parallel {fire(parent((x)), fire(parent(y))}
inhibit(A)
```

A far more elaborate program in the Assembly Calculus, for a proposed implementation of the creation of *syntax trees* in the generation of natural language, is described in the discussion section.

Computational power. It can be proved that the Assembly Calculus as defined above is capable of implementing, under appropriate assumptions, *arbitrary computations on* $O(\frac{n}{k})$ *bits of memory.*

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Given the intended values of n and k, this is a rather remarkable computational capability, suggesting — in view of the well established equivalence between parallel computation and space-bounded computation [28] — that any parallel computation with several hundred parallel steps, and on several hundred registers, can in principle be carried out by the Assembly Calculus.

The assembly hypothesis. The Assembly Calculus is a formal system with a repertoire of rather sophisticated operations, where each of these operations can be ultimately reduced to the firing of randomly connected populations of excitatory neurons with inhibition and Hebbian plasticity. The ensuing computational power of the Assembly Calculus may embolden one to hypothesize that such a computational system — or rather a neural system far less structured and precise, which however can be usefully abstracted this way — underlies advanced cognitive functions, especially in the human brain, such as reasoning, planning, and language.

Related work. Assemblies of excitatory neurons are, of course, not new: They have been hypothesized [20, 2], identified in vivo [17, 5], studied experimentally [27] and discussed extensively [10] over the past decades — even, occasionally, in connection to computation, see [35]. However, we are not aware of previous work in which assemblies, with a suite of operations, are proposed as the basis of a computational system intended to explain cognitive phenomena.

Assemblies and their operations as treated here bear a certain resemblance to the research tradition of *hyperdimensional computing,* see [32, 22, 23], systems of high-dimensional sparse vectors equipped with algebraic operations, typically component-wise addition, component-wise multiplication, and permutation of dimensions. Indeed, an assembly can be thought as a high-dimensional vector, namely the characteristic vector of its support; but this is where the similarity ends. While assembly operations as introduced here are meant to model and predict cognitive function in the Brain, hyperdimensional computing is a machine learning framework — inspired of course by the Brain, like many other such frameworks — and used successfully, for example, for learning semantics in natural language [23]. In sparse vector systems there is no underlying synaptic connectivity between the dimensions, or partition of the dimensions into Brain areas. Finally, the operations of the Assembly calculus (project, associate, merge) are very different in nature, style, detail, and intent from the operations in sparse vector systems (add, multiply, permute).

Assembly computation is closer in spirit to Valiant's model of neuroids and items [38], which was an important source of inspiration for this work. One difference is that whereas in Valiant's model the neuron (called neuroid there) has considerable local computational power — for example, to set its pre-synaptic weights to arbitrary values —, in our formalism computational power comes from a minimalistic model of inhibition and plasticity; both models assume random connectivity. Another important difference is that, in contrast to an item in Valiant's theory, an assembly is densely intraconnected, while the mechanism for its creation is described explicitly.

2 Results and Methods

Projection. The operation $\texttt{project}(x, B, y)$ entails activating repeatedly assembly x while B is disinhibited. Such repeated activation creates in the disinhibited area B a sequence of sets of k cells, let us call them $y_1, y_2, ..., y_t, ...$ The mathematical details are quite involved, but the intuition is the following: Cells in B can be thought of as competing for synaptic input. At the first step, only x provides synaptic input, and thus y_1 consists of the k cells in B which happen to have the highest sum of synaptic weights originating in x — note that these weights are subsequently increased by a factor of $(1 + \beta)$ due to plasticity. At the second step, neurons both in x and y_1 spike, and as

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

a result a new set y_2 of "winners" from among cells of B is selected; for typical parameters, y_2 overlaps heavily with y_1. This continues as long as x keeps firing, with certain cells in y_t replaced by either "new winners" — cells that never participated in a $y_{t'}$ with $t' < t$ — or by "old winners," cells that did participate in some $y_{t'}$ with $t' < t$. We say that the process has *converged* when when there are no new winners. Upon further firing of x, y_t may evolve further slowly, or cycle periodically, with past winners coming in and out of y_t; in fact, this mode of assembly firing (cells of the assembly alternating in firing) is very much in accordance with how assemblies have been observed to fire in Ca+ imaging experiments in mice, see for example [6]. It is theoretically possible that a new winner cell may come up after convergence; but it can be proved that this is a highly unlikely event, and we have never noticed it in our simulations. The number of steps required for convergence depends on the parameters, but most crucially on the plasticity coefficient β; this dependence is fathomed analytically and through simulations in Figure 1.B4).

Density. It was postulated by Hebb [20] that assemblies are *densely interconnected* — presumably such density was thought to cause their synchronized firing. Since assemblies are created by projection, increased density is intuitively expected: cells in y_t are selected to be highly connected to y_{t-1}, which is increasingly closer to y_t as t increases. It is also observed in our simulations (see Figure 1C) and predicted analytically in our model.

Association and pattern completion. Our simulations (Figure 1E), as well as analytical results, show that the overlap of two assemblies x, y in the same area increases substantially in response of simultaneous firing of the two parent assemblies $\texttt{parent}(x)$ and $\texttt{parent}(y)$ (assumed to be in two different areas). The amount of post-association overlap observed in our simulations is compatible with the estimates in the literature [36, 8], and increases with the extent of co-occurrence (the number of consecutive simultaneous activations of the two parents.

Since association between assemblies is thought to capture affinity of various sorts, the question arises: If two associated assemblies x and y are both projected in another area, will the size of their overlap be preserved? Our results, both analytical and through simulations, strongly suggest that assembly overlap is indeed conserved reasonably well under projection (see Section 2 and Figure 1,). This is important, because it means that the signal of affinity between two assemblies is not lost when the two assemblies are further projected in a new brain area.

Pattern completion involves the firing, *with some probability,* of an assembly x following the firing of a few of its neurons [27, 6]. Simulations in our model (Figure 1.F2) show that, indeed, if the creation of an assembly is completed with a sufficient number of activations of its parent, then by firing fewer than half of the neurons of the assembly will result in most, or essentially all, of the assembly firing.

Reciprocal projection and Merge. *Reciprocal projection*, denoted $\texttt{reciprocal.project}(x, B, y)$, is a more elaborate version of projection. It involves $\texttt{parent}(x)$ firing, which causes x in area A to fire, which in turn causes, as with ordinary projection, a set y_1 in B to fire. The difference is that now there is synaptic connectivity from area B to area A (in addition to connectivity from A to B), which causes in the next step x to move slightly to a new assembly x_1, while y_1 has become y_2. This continuing interaction between the x_t's and the $y_{t\pm1}$'s eventually converges, albeit slower than with ordinary projection, and under conditions of ampler synaptic connectivity and plasticity. The resulting assembly y has strong synaptic connectivity both to and from x (instead of only from x to y, as is the case with ordinary projection). That reciprocal projection works as described above has been shown both analytically and through simulations in our model, as well as in simulations in a more realistic neural model with explicit inhibitory neurons and STDP in [25].

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

The operation $\texttt{merge}(x, y, A, z)$ is essentially a double reciprocal projection. It involves the simultaneous repeated firing, in different areas, of the parents of both x and y, which causes the simultaneous repeated firing, also in two different areas B and C, of x and y. In the presence of enabled afferent two-way connectivity between A and B, and also between A and C, this initiates a process whereby a new assembly z is eventually formed in area A, which through its firing modifies the original assemblies x and y. In the resulting assemblies there is strong two-way synaptic connectivity between x and z, as well as between y and z. Analytical and simulation results are similar to those for reciprocal projection (see Figure 1.G1).

Simulations. We gain insights into the workings of our model, and validate our analytical results, through simulations. In a typical simulation task, we need to simulate a number of discrete time steps in two or three areas, in a random graph with $\sim n^2p$ nodes (where the notation $\sim f$ means "a small constant multiple of f). Creating this graph requires $\sim n^2p$ computation. Next, simulating each firing step entails selecting, in each area, the k out of n cells that receive that largest synaptic input. This takes $\sim knp$ computation per step. Since the number of steps needed for convergence are typically much smaller than the ratio $\frac{n}{k}$, the n^2p computation for the creation of the graph dominates the computational requirements of the whole simulation (recall that $n >> k$).

In our simulator we employ a maneuver which reduces this requirement to $\sim knp$, enabling simulations of the required scale on a laptop. The trick is this: We do not generate all $\sim n^2p$ edges of the random graph a priori, but generate them "on demand" as the simulation proceeds. Once we know which cells fired at the previous step, we generate the k cells in the area of interest that receive the most synaptic input, as well as their incoming synapses from the firing cells, by sampling from the tail of the appropriate Bernoulli distribution; we then compare with previously generated cells in the same area to select the k cells that will fire next.

3 Discussion

We have defined a formal system intended to model the computations underlying cognitive functions. Its elementary object is an *assembly,* a set of excitatory neurons all belonging to the same brain area, and capable of near-simultaneous firing. The operations of this system enable an assembly to create a "copy" of itself in a new brain area through projection; two assemblies to increase their overlap in order to reflect observed co-occurrence in the world or other similarity; and, furthermore, two assemblies to create a new assembly with ample connectivity to and from the two original ones. These operations do not rely on pre-existing neural circuits; instead, their apparatus is generated on-the-fly, with high probability, through plasticity and randomness of synaptic connectivity. The resulting formal system, equipped with certain simple control operations, is fraught with considerable computational power. Central to our work is the speculation that something akin to this formal system may underly cognitive functions, especially in the human brain.

What is an assembly, exactly? The precise definition is a conundrum. In our narrative so far, assembly is simply a set of k excitatory neurons in a brain area capable of synchronous firing. According to Hebb's prediction [20] as well as current observations [27] and neurorealistic simulations [33], the neurons of an assembly are not all activated simultaneously, but instead fire in a"pattern" with different neurons firing at different times[3]. In our formalism, and in our analytical proofs and our experiments, the assumption of discrete time steps suppresses to a large extent this sequential behavior. But even in our model, once the projection $\texttt{project}(x, B, y)$ has stabilized (no new neurons in area B fire that had never fired before), a small number of neurons often keep

[3]In musical metaphor, an assembly is thought to be not an octave but a cord sequence.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

coming in and out of the spiking set y_t. One possible principled definition of an assembly is this: an assembly in an area B is a sparse distribution on the neurons of area B, whose support (the set of neurons with non-zero probability) is not much larger than k.

One novel aspect of our assembly operations is the crucial role of plasticity. While plasticity is well studied as an important aspect of the way the brain works and learning happens, we are not aware of many models in which plasticity takes place at the sub-second time scale, as it is hypothesized to do in assembly operations. Our use of assemblies as the basis of a computational system also departs from the usual discourse on assemblies, typically considered, implicitly, as fairly stable representations. In contrast, here we conjecture that assemblies can be formed and modified by the human brain at 4 Hz, the frequency of syllables in language (see the discussion on language below).

Our results and simulations assume uniformly random synaptic connectivity; however, experimental measurements [37, 16] suggest a departure from uniformity (but not from randomness). Our analytical results can be extended routinely to non-uniform random synaptic connectivity of this kind. In fact, our conclusions regarding the density and stability of assemblies can be further strengthened in such regimes. For example, suppose that, conditioned on the existence of two directed edges (a, b) and (a, c), the presence of edge (b, c) is much more likely, as concluded in [37]. Depending on the precise values of the parameters n, k, p, this would likely trigger a "birthday paradox" phenomenon (existence of two assembly cells b, c with synaptic connections from the same cell a of the parent assembly) that would further enhance the synaptic density, and hence the stability, of assemblies.

The basic operations of the Assembly Calculus as presented here — projection, association, reciprocal projection, and merge — correspond to neural population events which (1) are plausible, in the sense that they can be reproduced in simulations and predicted by mathematical analysis, and (2) provide parsimonious explanations of experimental results (for the merge and reciprocal project operations see the discussion of language below). In contrast, the read and control operations — read, fire, disinhibit —, however simple and elementary, lack in such justification, and were added for the purpose of rendering the Assembly Calculus a programmable computational system. Replacing them with more biologically plausible control operations leading to the same effect would be very interesting.

Assemblies and Language. We hypothesized that assemblies and their operations may be involved in mediating higher cognitive functions in humans. In ending, we speculate below on how assemblies may be implicated in *language*.

Linguists have long argued that the language faculty is unique to humans, and that the human brain's capability for syntax (i.e., for mastering the rules that govern the structure of sentences in a language) lies at the core of this ability [19, 4]. In particular, the linguistic primitive *Merge* has been proposed as a minimalistic basis of syntax. Merge combines two syntactic units to form one new syntactic unit; recursive applications of Merge can create a *syntax tree,* which captures the structure of a sentence (see Figure 3(a)). If this theory is correct, then which are the brain correlates of syntax and Merge? Recent experiments provide some clues — see [15] for an extremely comprehensive and informative recent treatment of the subject.

- Two different brain areas of the left superior temporal gyrus (which contains Wernicke's area, known since the 19th century to be implicated in the use of words in language) seem to respond to the subject *vs* the object of a presented sentence [13];
- The completion of phrases and sentences presented is marked by activity in Broca's area in the left hemisphere [39], known to be implicated in syntax;

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

- A sequence of sentences consisting of four monosyllabic words (such as "bad cats eat fish") presented at the rate of four words per second (the natural frequency of articulated syllables) elicit a pattern of brain responses, with energy peaks at one, two, and four Hz, consistent with the hypothesis that syntax trees for the sentences are being constructed during the presentation [9].

If one accepts the hypothesis that indeed something akin to syntax trees is constructed in our brain during the parsing — and presumably also during the generation — of sentences, one must next ask: how is this accomplished? According to [15] Chapter 4, these and a plethora of other experimental results point to a *functional cortical network* for lexical and syntactic processing, involving the MTL, Broca's areas BA 44 an BA 45, and Wernicke's area in the superior temporal gyrus, as well as axon fibers connecting these four areas, all in the left hemisphere. Syntactic processing of language seems to entail a complex sequence of activations of these areas and transmission through these fibers. How is this orchestrated sequence of events carried out? Does it involve the generation and processing, on-the-fly, of representations of the constituents of language (words, phrases, sentences)? Angela Friederici [15] proposes in page 134 that "for language there are what I call *mirror neural ensembles* through which two distinct brain regions communicate with each other." Could it be that assemblies and their operations play this role?

We propose a dynamic brain architecture for the generation of a simple sentence, powered by assemblies and the operations `reciprocal.project` and `merge`, and consistent with the experimental results and their interpretations outlined above (see Figure 3). In particular, we propose that the construction of the syntax tree of a simple sentence being generated by a subject can be implemented by the following program of the Assembly Calculus:

`do in parallel:` `find-verb`(Im, MTL, x), `find-subj`(Im, MTL, y), `find-obj`(Im, MTL, z);
`do in parallel:` `reciprocal.project`(x, WVb, x'),
  `reciprocal.project`(y, WSubj, y'), `reciprocal.project`(x, WObj, z');
`merge`$(x', z', \mathrm{Broca44}, p)$;
`merge`$(y', p, \mathrm{Broca45}, s)$

The generation of a sentence such as *"the boy kicks the ball"* starts with a desire by the subject to assemble — and possibly articulate — a particular fact. The raw fact to be put together is denoted here by Im — an image, sensed or mental. In the first line, the subject's lexicon, presumably a collection of tens of thousands of assemblies in the medial temporal lobe (MTL), is searched in order to identify the verb (the action in the fact relayed in Im); the subject (the agent of the fact); and the object (the patient of the fact). We assume that the corresponding brain functions `find-verb` etc. are already in place. For the example *"the boy kicks the ball"*, the three assemblies x, y, and z are identified, encoding the words *"kick"*, *"boy"*, and *"ball"*, respectively. In the second line, these three assemblies are projected to the three sub-areas of Wernicke's area specializing in the verb, subject, and object of sentences, respectively. In fact, instead of ordinary projection, the `reciprocal.project` operation is used, as first proposed in [25], for reasons that will become clear soon. Next, an assembly p is formed in the pars opercularis of Broca's area (BA 44) representing the verb phrase *"kicks the ball"* through the `merge` operation applied to the two assemblies x' and z' encoding the constituent words of the phrase in Wernicke's area. Finally, an assembly s corresponding to the whole sentence is formed in the pars triangularis of Broca's area (BA 45) via the `merge` of assemblies p and y', completing the construction of the syntax tree of the whole sentence.

The Assembly Calculus program above accounts only for the first phase of sentence production, during which the syntax tree of the sentence is constructed. Next the sentence formed may be

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

articulated, and we can speculate on how this process is carried out: Assembly s is activated, and this eventually causes the assemblies x', y', z' — the leaves of the syntax tree — to fire. The activation of these three assemblies is done in the order specific to the speaker's language learned at infancy (for example, in English *subject-verb-object*); note that the first phase of building the syntax tree was largely language independent. Eventually, the lexicon assemblies will be activated in the correct order (this activation was the purpose of utilizing the `reciprocal.project` operation), and these in turn will activate the appropriate motor functions which will ultimately translate the word sequence into sounds.

The above narrative is only about the building of the basic syntactic structure — the "scaffold" — of extremely simple sentences, and does not account for many other facets: how the three `find`-tasks in the first line are implemented; how the noun phrases are adorned by determiners such as "the" and how the verb is modified to reflect person or tense ("kicks" or "kicked"); and the important inverse processes of *parsing and comprehension,* among a myriad of other aspects of language that remain a mystery.

Acknowledgments. The authors are grateful to Larry Abbott for his insightful ideas and comments concerning this work. We also wish to thank several colleagues for helpful discussions and feedback, including Richard Axel, Bob Berwick, Costis Daskalakis, Boris Harizanov, Pentti Kanerva, Robert Legenstein, Chris Manning, Bruno Olshausen, Tommy Poggio, Les Valiant, and Rafa Yuste.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

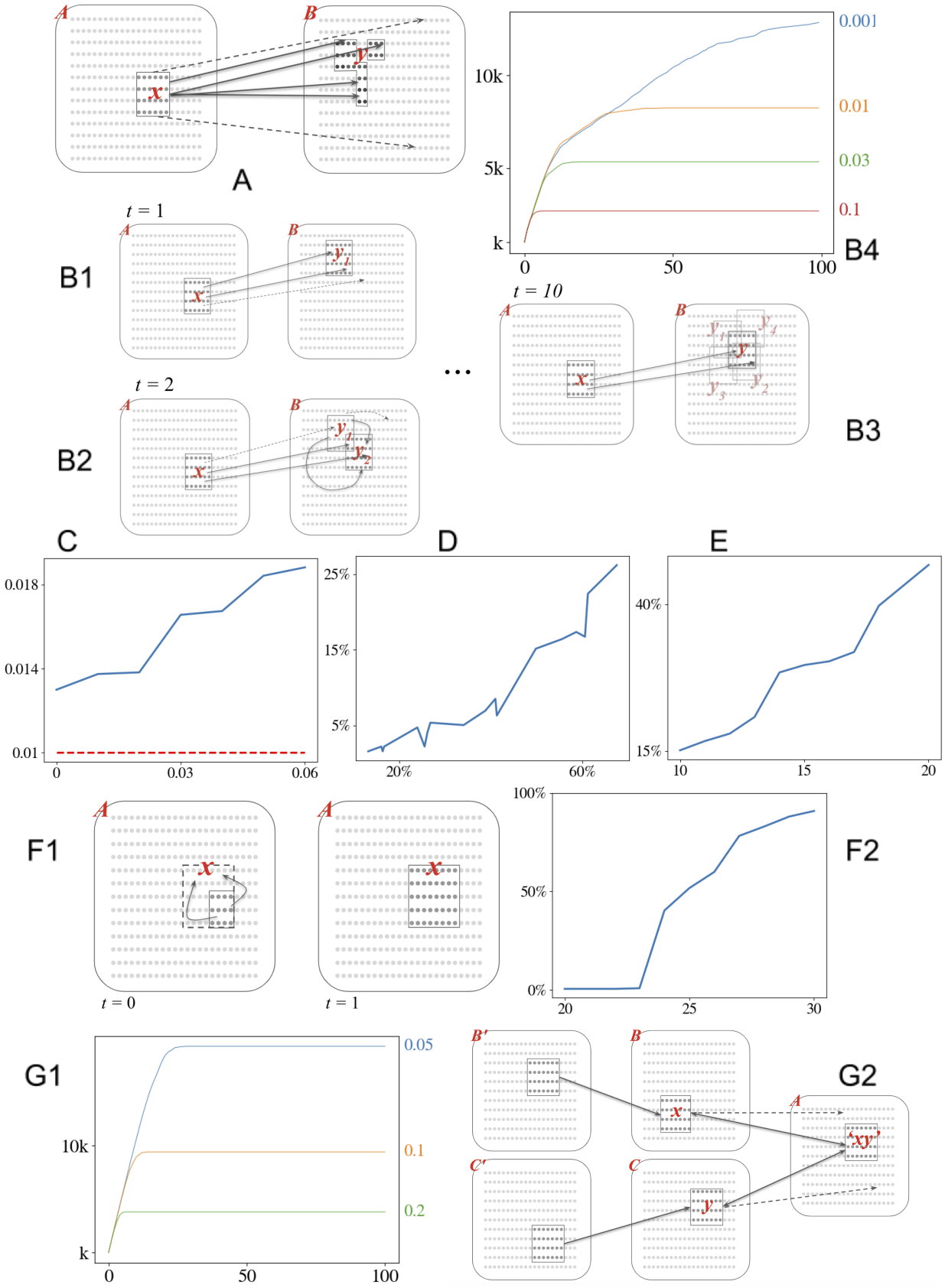

Figure 1

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Figure 1: Assembly operations. A RP&C: If an assembly x fires in area A, and there is afferent connectivity to area B, the ensuing synaptic input will cause a set of k neurons in area B, call it y, to fire. The set y contains the k neurons in area B that receive the highest synaptic input from assembly x. This is an important primitive called *random projection and cap* (RP&C). **B1 – B3:** If assembly x fires again, afferent synaptic input from area A will be combined with recurrent synaptic input from y_1 in area B to cause a new set of k neurons in area B, y_2, to fire. Continued firing of x will create a sequence of sets of firing neurons in B: $y_1, y_2, y_3, \ldots$ In the presence of Hebbian plasticity, this sequence converges exponentially fast to a set of neurons, an assembly y which we call the *projection of x to B*. If subsequently x fires, y will follow suit. **B4 Exponential convergence of assembly projection:** The horizontal axis is the number of times assembly x fires; the vertical axis is the total number of neurons in area B that fired in the process. Different color lines represent different values of the plasticity coefficient β; for higher levels of β, convergence is faster. **C**: Synaptic density within the resulting assembly increases with higher values of β (blue line), and is always higher than the baseline random synaptic connectivity p (dotted red line). **D Preservation of overlap:** Assembly overlap is preserved under RC&P. The x axis represents the overlap in two assemblies that are then projected to a new area once (RC&P); the y axis represents the percentage overlap of the two resulting assemblies in B. **E Association:** If two assemblies in two different areas have been independently projected in a third area to form assemblies x and y, and subsequently the two parent assemblies fire simultaneously, then each of x, y will respond by having some of its neurons migrate to the other assembly; this is called *association* of x and y. Such overlap due to association may reach $8 - 10\%$ [21]. The figure shows results from simulation: first, two separate, stable assemblies are created in two areas A and B. Then, the assemblies in A and B are projected simultaneously into C for x time steps, represented by the x axis; the y axis shows the resulting overlap in the projected assemblies of A and B after the simultaneous firing. **F1 Pattern completion:** The firing of a few cells of an assembly results in the firing of the whole assembly. **F2:** An assembly is created in an area A by repeated firing of its parent; the number of times the parent fires is depicted in the horizontal axis. Next, 40% of neurons of the assembly are selected at random and fire for a number of steps: the y axis shows the overlap of the resulting assembly with the original assembly. With more reinforcement of the original assembly, the subset recovers nearly all of the original assembly. **G1 Merge:** The most advanced, and demanding in its implementation, operation of the Assembly Calculus involves five brain areas: areas B and C contain the assemblies x, y to be merged, while areas B' and C' contain the parents of these assemblies. The simultaneous spiking of the parents induces the firing of x and y, which then initiates a process whereby two-way afferent computation between B and A, as well as C and A, results in the adjustment of both x and y and the creation of an assembly z in A having strong synaptic connections *to and from* both x and y. **G2:** Speed of convergence depends critically on β.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

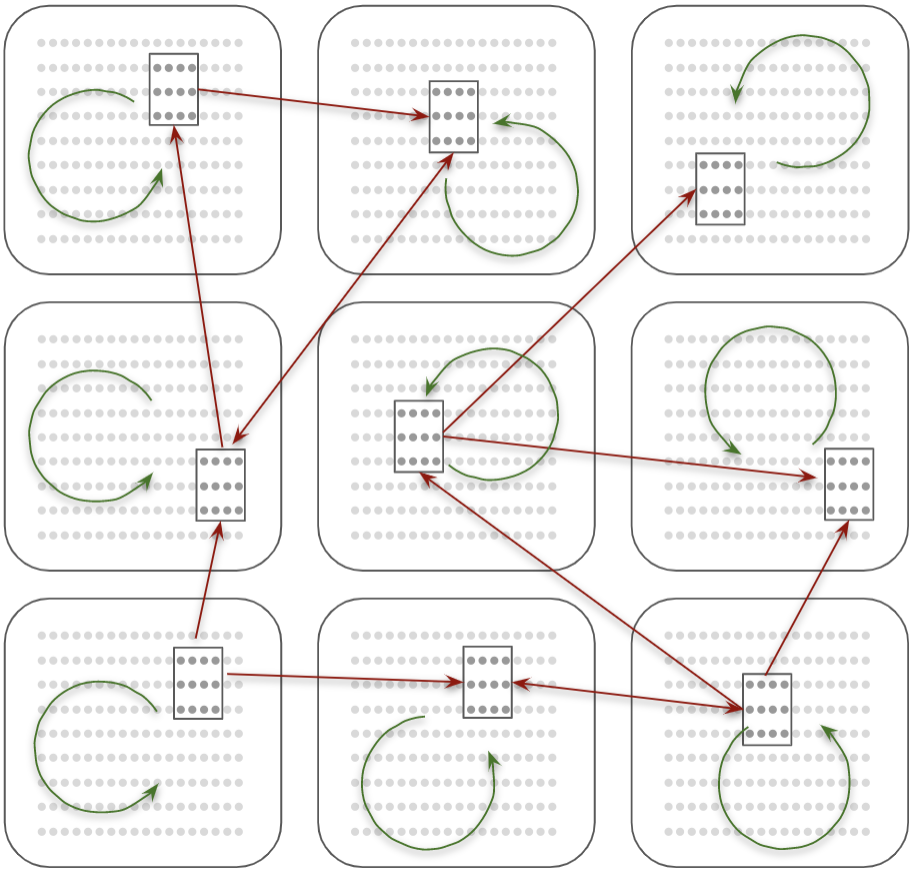

Figure 2

Figure 2: A mathematical model of the brain. Our analytical results, as well as most of our simulations, use a novel formal model of the brain, intended to capture cognitive phenomena in the association cortex. Our model encompasses a finite number of *brain areas* denoted $A, B, \ldots$ each containing n excitatory neurons. We assume that any pair of neurons in each area have a recurrent synaptic connection independently with probability p; and that for certain pairs of areas (A, B) there are also afferent connection from any neuron in A to any in B, also independently and with probability p. We assume that firing of neurons proceeds in discrete steps, and at each step k of the neurons in each area fire: namely, those k neurons which have the largest sum of presynaptic inputs from neurons that fired in the previous step. Synaptic weights are modified in a Hebbian fashion, in that synapses that fired (i.e., the presynaptic neuron fired and in the next step the postsynaptic neuron did) have their weight multiplied by $(1 + \beta)$, where β is a plasticity constant. In our simulations we typically use $n = 10^7$, $k = 10^4$, $p = 0.001$, and $\beta = 0.1$.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

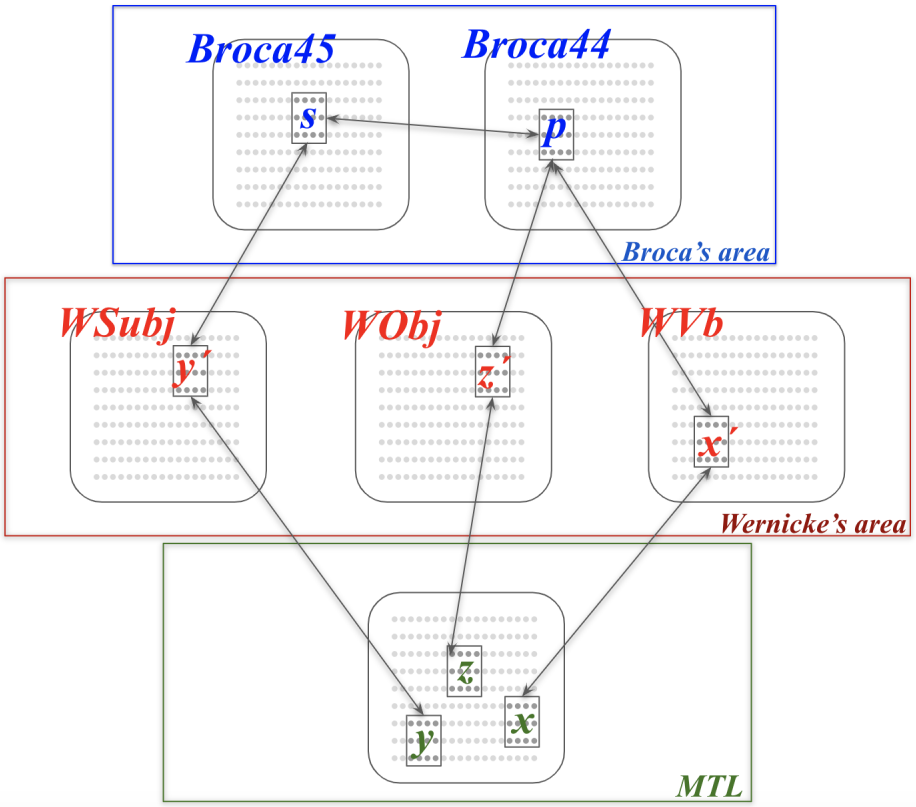

Figure 3

Figure 3: A proposed cortical architecture for syntax in language (see also [15]). In order to generate a simple sentence such as *"the boy kicks the ball,"* the subject must first *find* in the left MTL the representations of the verb, subject and object of the sentence. Next, these three assemblies are reciprocal-projected to the corresponding subareas of Wernicke's area. Next the verb and the object are merged in BA 44 of Broca's area to form a representation p of the verb phrase *"kicks the ball"*, and finally p is merged with the subject to create a representation s of the whole sentence in BA 45. This concludes the phase of building of the sentence's *syntax tree.* Now in order to articulate the sentence, s fires and this results in the firing of the constituents of the tree, until eventually the three words in the MTL fire *in the correct order* for the subject's language — an order learned at infancy (in English, subject-verb-object). This latter activation of the three words mobilizes the corresponding motor functions resulting in the production of sound.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

References

[1] Q & A. Richard Axel. *Neuron*, 99:1110–1112, 2018.

[2] Moshe Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.

[3] Nima Anari, Constantinos Daskalakis, Wolfgang Maass, Christos H. Papadimitriou, Amin Saberi, and Santosh S. Vempala. Smoothed analysis of discrete tensor decomposition and assemblies of neurons. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 10880–10890, 2018.

[4] Robert C Berwick and Noam Chomsky. *Why only us: Language and evolution*. MIT Press, 2016.

[5] G Buzsaki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3), 2010.

[6] Luis Carrillo-Reid, Shuting Han, Weijian Yang, Alejandro Akrouh, and Rafael Yuste. Triggering visually-guided behavior by holographic activation of pattern completion neurons in cortical ensembles. *Biorxiv*, page 394999, 2018.

[7] Sanjoy Dasgupta, Charles F. Stevens, and Saket Navlakha. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796, 2017.

[8] Emanuela De Falco, Matias J Ison, Itzhak Fried, and Rodrigo Quian Quiroga. Long-term coding of personal and universal associations underlying the memory web in the human brain. *Nature Communications*, 7:13408, 2016.

[9] Nai Ding, Lucia Melloni, Hang Zhang, Xing Tian, and David Poeppel. Cortical tracking of hierarchical linguistic structures in connected speech. *Nature neuroscience*, 19(1):158, 2016.

[10] Howard Eichenbaum. Barlow versus hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? *Neuroscience letters*, 680:88–93, 2018.

[11] Paul Erdős and Alfred Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hungary. Acad. Sci.*, 5:17–61, 1960.

[12] Uriel Feige, David Peleg, and Guy Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[13] S. M. Frankland and J. D. Greene. An architecture for encoding sentence meaning in left mid-superior temporal cortex. *Proceedings of the National Academy of Sciences*, 112(37):11732–11737, 2015.

[14] Kevin M Franks, Marco J Russo, Dara L Sosulski, Abigail A Mulligan, Steven A Siegelbaum, and Richard Axel. Recurrent circuitry dynamically shapes the activation of piriform cortex. *Neuron*, 72(1):49–56, 2011.

[15] Angela D Friederici. *Language comprehension: A biological perspective*. Springer Science & Business Media, 2012.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[16] S Guzman, A J Schlö gl, M Frotscher, and P Jonas. Synaptic mechanisms of pattern completion in the hippocampal ca3 network. *Science*, 353(6304):1117–1123, September 2016.

[17] Kenneth D Harris. Neural signatures of cell assembly organization. *Nature Reviews Neuroscience*, 6(5):399, 2005.

[18] Kenneth D Harris and Gordon MG Shepherd. The neocortical circuit: themes and variations. *Nature neuroscience*, 18(2):170, 2015.

[19] Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. The faculty of language: What is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.

[20] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, 1949.

[21] Matias J Ison, Rodrigo Quian Quiroga, and Itzhak Fried. Rapid encoding of new memories by individual neurons in the human brain. *Neuron*, 87(1):220–230, 2015.

[22] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159, 2009.

[23] Jussi Karlgren and Pentti Kanerva. High-dimensional distributed semantic spaces for utterances. *Natural Language Engineering*, 25(4):503–517, 2019.

[24] R. Legenstein, W. Maass, C. H. Papadimitriou, and S. S. Vempala. Long-term memory and the densest k-subgraph problem. In *Proc. of 9th Innovations in Theoretical Computer Science (ITCS) conference, Cambridge, USA, Jan 11-14. 2018*, 2018.

[25] Robert Legenstein, Michael G Müller, Christos H Papadimitriou, Santosh Vempala, and Wolfgang Maass. Assembly projections support the assignment of thematic roles to concepts in networks of spiking neurons. *arXiv preprint arXiv:1611.03698*, 2016.

[26] Robert Legenstein, Christos H Papadimitriou, Santosh Vempala, and Wolfgang Maass. Assembly pointers for variable binding in networks of spiking neurons. *arXiv preprint arXiv:1611.03698*, 2016.

[27] Jae-eun Kang Miller, Inbal Ayzenshtat, Luis Carrillo-Reid, and Rafael Yuste. Visual stimuli recruit intrinsically generated cortical ensembles. *Proceedings of the National Academy of Sciences*, 111(38):E4053–E4061, 2014.

[28] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[29] Christos H. Papadimitriou and Santosh S. Vempala. Random projection in the brain and computation with assemblies of neurons. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 57:1–57:19, 2019.

[30] Steven T Piantadosi, Joshua B Tenenbaum, and Noah D Goodman. Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition*, 123(2):199–217, 2012.

bioRxiv preprint doi: https://doi.org/10.1101/869156; this version posted December 8, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[31] Steven T Piantadosi, Joshua B Tenenbaum, and Noah D Goodman. The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological review*, 123(4):392, 2016.

[32] Tony Plate. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *IJCAI*, pages 30–35, 1991.

[33] C. Pokorny, M. J. Ison, A. Rao, R. Legenstein, C. Papadimitriou, and W. Maass. Associations between memory traces emerge in a generic neural circuit model through STDP. *bioRxiv:188938*, 2017.

[34] Rodrigo Quian Quiroga. Neuronal codes for visual perception and memory. *Neuropsychologia*, 83:227–241, 2016.

[35] João Ranhel. Neural assembly computing. *IEEE transactions on neural networks and learning systems*, 23(6):916–927, 2012.

[36] Hernan G Rey, Emanuela De Falco, Matias J Ison, Antonio Valentin, Gonzalo Alarcon, Richard Selway, Mark P Richardson, and Rodrigo Quian Quiroga. Encoding of long-term associations through neural unitization in the human medial temporal lobe. *Nature communications*, 9(1):4372, 2018.

[37] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii. Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biology*, 3(3):e68, 2005.

[38] Leslie G. Valiant. *Circuits of the mind*. Oxford University Press, 1994.

[39] Emiliano Zaccarella and Angela D. Friederici. Merge in the human brain: A sub-region based functional investigation in the left pars opercularis. *Frontiers in Psychology*, 6:1818, 2015.

[40] Emiliano Zaccarella, Lars Meyer, Michiru Makuuchi, and Angela D Friederici. Building by syntax: the neural basis of minimal linguistic structures. *Cerebral Cortex*, 27(1):411–421, 2017.