Why Build a Virtual Brain?
Large-scale Neural Simulations as Test-bed for Artificial Computing Systems

Matteo Colombo (M.Colombo@UvT.nl)
Tilburg Center for Logic, General Ethics and Philosophy of Science, Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

Abstract

Despite the impressive amount of financial resources invested in carrying out large-scale brain simulations, it is controversial what the payoffs are of pursuing this project. The present paper argues that in some cases, from designing, building, and running a large-scale neural simulation, scientists acquire useful knowledge about the computational performance of the simulating system, rather than about the neurobiological system represented in the simulation. What this means, why it is not a trivial lesson, and how it advances the literature on the epistemology of computer simulation are the three preoccupations addressed by the paper.

Keywords: Large-scale neural simulations; epistemology of computer simulation; target-directed modeling; neuromorphic technologies

Introduction

In the last twenty years or so, several research groups have been working on large-scale brain simulations. In the face of the impressive amount of financial resources invested in such projects, it is controversial what the payoffs are of carrying out large-scale brain simulations. The present paper explores this issue, asking: Currently, what do scientists learn from designing, building, and running large-scale neural simulations? One plausible answer is that at least for *some* such simulations scientists learn about the computational performance of the simulating system.

Plausible as it sounds, the significance of this answer should not be downplayed, for at least two reasons. First, most work in the epistemology of computer simulation overlooks or downplays the computational and material aspects of computer simulation. But learning about the computational performance of a machine is far from trivial. Second, the kinds of neural simulations examined in this paper involve an interesting set of practices that have not been adequately discussed in the epistemology of modelling and computer simulation.

In particular, these simulations have two *kinds* of targets: one target is a real neural system, which is represented in the simulation; the other target is the computing system itself, which is not represented in the simulation but studied both directly and through complicated inferences. If this is correct, then two interesting conclusions follow. (1) When scientific models and computer simulation are employed to gain new knowledge, it is not always knowledge about their represented target systems that is sought. For some neural simulations, the real neural system that one tries to represent is not the system about which one wants to learn. (2) Some neural simulations imitate some features of a real neural system (i.e., their representational target) not in order to serve as surrogates that are investigated to gain new knowledge about the brain. Rather, these neural simulations imitate some features of a real neural system in order to gain useful knowledge about the simulating system itself.

While claim (1) concerns the type of knowledge one may want or hope to acquire with computer simulation, claim (2) concerns one possible representational function of computer simulation.

Large-scale Neural Simulations: Aims and Prospects

For many *large-scale neural simulations*, a simulating system implements some algorithm that finds solutions to mathematical equations that are believed to describe the dynamics and pattern of connectivity of a large number (e.g., over a million) of neurons and synapses (for reviews Brette et al. 2007; de Garis et al. 2010; Goertzel et al. 2010; Eliasmith & Trujillo 2014).

A large-scale neural simulation is a type of computer simulation. Computer simulation can be characterised broadly as “a comprehensive method for studying systems,” which “includes choosing a model; finding a way of implementing that model in a form that can be run on a computer; calculating the output of the algorithm; and visualizing and studying the resultant data” (Winsberg 2013). Accordingly, some real-world system should be picked as the representational target of the computer simulation; some mathematical equations should be chosen, which are believed to model (some aspect of) the behavior of the target system; and an appropriate simulating system, consisting of both hardware and software components, should be used to implement the mathematical model.

In line with much of the philosophical literature, where models and simulations are understood as serving as representations of some system about which one wants or hopes to gain knowledge (e.g., Humphreys 2004; Parker 2009; Grüne-Yanoff & Weirich 2010; Weisberg 2013), Winsberg (2013) claims that the entire process constituting computer simulation is “used to make inferences about the target system that one tries to model.”

The claim also coheres with the stated aims of many large-scale neural simulations. For example, the *Blue Brain Project* set out to “simulate brains of mammals with a high level of biological accuracy and, ultimately, to study the steps involved in the emergence of biological intelligence” (Markram 2006, 153). The objective of carrying out certain large-scale neural simulations is to understand why and how many different ion channels, receptors, neurons, and synaptic pathways in the brain contribute to different brain functions and to emergent, intelligent behavior (158). The aim of Izhikevich & Edelman's (2008) simulation of a million spiking thalamo-cortical neurons and half a billion synapses was analogous. They explained that "[o]ne way to deepen our understanding of how synaptic and neuronal processes interact to produce the collective behavior of the brain is to develop large-scale, anatomically detailed models of the mammalian brain" (3597). Similarly, the objective of Eliasmith and colleagues' (2012) 2.5 million neuron simulation was to understand why and how the robust and rapid flexibility of biological systems can be generated from a unified set of neural mechanisms.

Despite significant differences, the aim shared by these projects is to use large-scale neural simulations to understanding of how and why brains' multi-scale, complex organization generates different brain functions and emergent cognitive phenomena. This aim may be reached. Yet, it is far from uncontroversial that, currently, a large-scale neural simulation is a fruitful approach to addressing questions about why and how neurons and synapses' dynamics generate different brain functions and cognitive phenomena (Mainen & Pouget 2014).

Commenting on this approach, Carandini (2012) argues that, currently, "putting all of the subcellular details (most of which we don't even know) into a simulation of a vast circuit is not likely to shed light on the underlying computations" (509). If the underlying neural computations are not understood, there is little hope to learn how and why neural circuits generate different brain functions and cognitive phenomena. In a similar vein, Sporns (2012) points out that the success of projects like Markram's *Blue Brain* "depends on knowledge about the organization of neurons and molecules into complex networks whose function underpins system dynamics" (168). Such knowledge is currently sparse and not easily incorporable into large-scale neural simulations. So, it is doubtful that, currently, carrying out large-scale neural simulations is a fruitful approach to learn about the neurobiological systems represented in the simulation.

Brains and Computational Performance

More plausible is that, currently, from at least *some* large-scale neural simulations, scientists gain knowledge about the computational performance of the simulating system itself, rather than about the neural system that the simulation represents.

Simulating systems are computing systems comprising both software and hardware components. They include a computational architecture and a set of algorithms formulated as computer programs that can be executed on concrete computing machines. The computational performance of the simulating system depends on a complex combination of properties of its architecture, of the algorithms it uses, the programs it executes, and of the materials and technological devices of which it is made.

Three dimensions on which computational performance can be assessed are: the time it takes for the computing system to carry out a given task, the maximum number of tasks that can be completed by the system in a given time interval, and the electrical power it takes for the system to carry out a task.

The total time required for a computing system to complete a task is called *execution time*. One way to measure the execution time of a program is in terms of clock period, which is the time length (in nanoseconds) of a cycle of the clock built into the system that determines when events take place in the hardware. The clock rate (in hertz) is the inverse of the clock period. Increasing computational performance for a given program requires decreasing its execution time, which may be tackled as an engineering problem—viz. as the problem of reducing the clock period—or as a computational problem—viz. as the problem of designing a more efficient computational architecture or more efficient algorithms and programs.

The number of tasks that can be completed per unit time by a computing system is called *throughput*. If we focus on the communication channels of a computing system, then the maximum throughput of a channel is often called *bandwidth* (measured in bits of data/second). The amount of time it takes for a communication channel to become unoccupied so that it can allow for data transfer is called *latency*. The available bandwidth of a communication channel is a limited resource, and should be used sparingly. The greater the bandwidth capacity, or the lower the latency of the communication channels, the more likely it is that the system displays better computational performance. The throughput, bandwidth, and latency of a computing system are a complex function of the physical medium being used for communications, the system's wiring architecture and the type of code used for programming.

The microprocessors of computing systems dissipate heat. Heat must be removed from a computing system; else, its hardware components will overheat. Conserving power and avoiding overheating, while improving computational performance, have led computer scientists and engineers to explore novel architectures, hardware technologies, software solutions and programming languages for highly-efficient computing systems.

There are two reasons why carrying out a computer simulation of a large number of neurons and synapses can yield non-trivial knowledge of the computational performance of the simulating system. First reason: brains can be understood as computational systems, which can be used to set a real biological benchmark for artificial computing systems' performance. Second reason: *scalability*, which indicates how efficient an application is when using increasing numbers of parallel processing units or amount of computational resources.

If the brain is a computing system, then it displays high performance in the face of low power consumption and small size. On average, the human brain weighs around 1.3 to 1.5 Kg, is constituted by about 100 billion neurons and around 100 trillion synapses, and its volume is about 1,400 ml. For carrying out its computations, it consumes energy at a rate of about 20 watts. Brains' computational architecture and style of computing are very different from those of modern artificial computing systems. Modern artificial computing systems possess von Neumann architecture and have stored programs, which are typically implemented in digital, serial, synchronous, centralized and fast microcircuits. By contrast, biological brains possess a non-von Neumann, multiscale, network architecture; they have distributed computational units, which carry out mixed-mode analog-digital, parallel, asynchronous, slow, noisy, computations (Montague 2007; Piccinini & Bahar 2013).

Available information about general computational features of biological brains can provide one basis for benchmarking the performance of artificial computational systems along some dimension of interest like power consumption or scalability. Comparing the computational performance of the simulating system in a large-scale neural simulation to that of its neurobiological target along some dimension of interest allows scientists to learn about why and how certain features of the simulating system (e.g., its network architecture, its materials) impact its performance relative to that dimension.

What about scalability? Although it is problematic to precisely define 'scalability,' the term is generally used in computer science to denote the capacity of a multiprocessor parallel computing system to accommodate a growing number of processing units or to carry out a growing volume of work gracefully (Hill 1990). Scalability is a desirable feature of a computing system because it allows for hardware or software components to be added in the system without outgrowing it. Two more specific notions, helpful to assess the performance of a large-scale simulation, are those of *strong scaling* and *weak scaling*, which denote respectively the capacity of a system to reduce execution time for solving a fixed-size problem by adding processors, and the capacity to keep execution time constant by adding processors so as to accommodate additional workload. Assessing strong scaling is particularly relevant to learning about why some program takes a long time to run (something that is CPU-bound). Assessing weak scaling is particularly relevant to learning why some program takes a lot of memory to run (something that is memory-bound).

Lack of scalability in large-scale neural simulation can indicate that the architecture of the simulating system cannot effectively solve problems of a certain size that biological brains can solve quickly. It can indicate that adding more simulated neurons and synapses to the simulating system is not an efficient strategy to execute a certain program more quickly, as the communication costs would increase as a function of the number of processors added to the system. It can also indicate that the power consumption required by a system that grows larger is too costly. So, by taxing an artificial computing system by simulating millions of neurons and synapses, scientists can learn about trade-offs between memory, computation, and communication in a certain computational architecture.

Brains, simulations, and neuromorphic devices

Learning about the computational performance of a computing system can be important for developing *neuromorphic technologies*. Neuromorphic technologies are devices for information processing and data analysis that aim to approximate the computational architecture and style of computing of biological brains. Such technologies include vision systems, auditory processors, multi-sensor integrators, autonomous robots, and tools for handling and analysing large amount of data (Indiveri & Horiuchi 2011).

SyNAPSE (Systems of Neuromorphic Adaptive Plastic Scalable Electronics) is an on-going research program funded by the U.S. Defense Advanced Research Projects Agency (DARPA). "The vision for the SyNAPSE program is to develop electronic neuromorphic machine technology that scales to biological levels" (DARPA BAA08-28). This research program aims to develop electronic technology with similar computational performance to the mammalian brain in terms of size, speed, and energy consumption.

Under the SyNAPSE program, Preissl and colleagues (2012) carried out a computer simulation of a very large neural circuit with the ultimate goal of exploring how closely one can "approximate the function, power, volume and real-time performance of the brain within the limits of modern technology" (10). The representational target system of their simulation was a network comprising 65 billion neurons and 16 trillion synapses, which imitated the largest known wiring diagram in the macaque monkey's brain. This biological target was modelled as a network of neurosynaptic cores containing digital integrate-leak-and-fire neurons.

The simulating system involved a 16-rack Blue Gene/Q supercomputer of 16,384 to 262,144 CPUs and 256 TB of main memory, and *Compass*, a multi-threaded, massively-parallel software, which enabled the simulation of billions of neurosynaptic cores operating in a parallel, distributed, and semi-synchronous fashion.

The modelling choices of Preissl and colleagues were congenial to the pursuit of an engineering goal. The neurons, synapses, and axons in their simulation were modelled as event-driven (asynchronous), digital, integrate-leak-and-fire circuits. The leaky integrate-and-fire model is one of the simplest models of spiking neurons. Given its lack of biophysical detail, the range of phenomena that this model can address is limited. Nonetheless, the model is analytically solvable and relatively easy to implement in a computer simulation. For many integrate-and-fire neurons models, the model fits nicely with an event-driven simulation, whereby all operations in the simulation are driven by neural spike events, which is generally well suited to decrease computational time and minimize memory load. The inter-core pattern of connections embodied in *Compass* imitated the macaque's neural wiring. The relationship between the model-network and its neurobiological target was not isomorphic; it was a *similarity* relation, which is generally sufficient to allow scientists to learn from computer simulation, especially when, like in this case, some relevant aspects and degrees of similarity are specified based on the question at hand, available background knowledge and the larger scientific context (Teller, 2001; Giere, 2004; Weisberg, 2013).

Implementing the macaque's wiring diagram "challenges the communication and computational capabilities of *Compass* in a manner consistent with supporting brain-like networks" (11). The performance of the simulating system could then be compared with that of the real neurobiological system represented in the computer simulation. A quantitative characterization of the deviations between the real neural system and the simulating system allowed scientists to identify which features of architectural and communication-design contributed to computational efficiency.

Preissl and colleagues' computer simulation could be used as a test-bed for learning about the performance of hardware and software components of a simulating system put under serious computational stress. Simulating a neural network at that scale poses major challenges for computation, memory, and communication, even with current supercomputers. If we consider N neurons, whose average firing rate is H, and whose average number of synapses is S, and we take account of all spike transmissions, then a real-time simulation of 1 second of biological time should process N x H x S spike transmissions. This minimal number of operations set a benchmark to assess the computational performance of a neural simulation (Brette et al. 2007, 350-1).

Preissl et al.'s (2012) simulation yielded two main results. First, as the average spiking rate of neurons was 8.1 Hz, the simulation was 388x slower than real time. Second, simulating the pattern of structural connectivity of the macaque's brain, the simulating system displayed near-perfect weak and strong scaling. While acquiring this type of information does not obviously yield novel insight about phenomena produced by biological brains, it is relevant to the development of more efficient artificial computing systems. As Preissl and colleagues put it: "*Compass* is a harbinger of an emerging use of today's modern supercomputers for midwifing the next generation of application-specific processors that are increasingly proliferating to satisfy a world that is hungering for increased performance and lower power while facing the projected end of CMOS scaling and increasing obstacles in pushing clock rates ever higher" (11).

Representing and Learning with Large-scale Neural Simulations

Two claims are widely shared in the literature about the epistemology of computer simulation and scientific modelling (Frigg & Hartmann 2012). First, in target-directed modelling, when scientific models and computer simulations are used to acquire new knowledge, it is knowledge about their *represented targets* that is ultimately sought (Weisberg 2013, Ch. 5). Second, computer simulations imitate some features of their represented target just to serve as surrogates that are investigated to gain new knowledge about it (Swoyer 1991). That is, the representational relation that holds between computer simulations and their represented targets allow scientists to perform inferences just from the simulation to its represented target.

These two claims should be rectified in the light of computer simulations like Preissl and colleagues'. For some large-scale neural simulations, computer simulations have two *kinds* of targets about which one may want to gain new knowledge. One kind of target is a real neural system, which is represented in the simulation; the other kind of target is the computing system itself, which is not represented in the simulation, but studied either directly, or through complicated inferences. Depending on the goal of the scientists designing and running the computer simulation, these inferences may or may not be based on the assumption that the simulating system bears some representational relation with its neural target.

Generally, computer simulations can instruct scientists about some aspect of reality even if it is not assumed that the mathematical model implemented in the simulation has counterparts in the world about which scientists want or hope to learn. In these cases, the aspects of reality about which scientists hope to gain novel information are some of the computational features of the simulating system, rather than some of the features of the real system represented in the computer simulation. Assuming that the simulating system bears some representational relation with a neurobiological target is not necessary to gain this information. In fact, benchmarking software exists that can be used to assess the relative performance of artificial computing systems' hardware or programs.

However, assuming that a simulating system does bear some representational relation with its neural target allows scientists to study performance discrepancies between the simulating system and the neurobiological system, which can function as useful benchmark along some dimension of interest. By characterising such discrepancies, constraints on computational efficiency can then be identified, which is particularly useful when the goal is to acquire knowledge useful for designing neuromorphic innovations.

The claim that computer simulation can instruct scientists about kinds of target systems that are different from those represented in the simulation resonates with Humphreys' (2009), Parker's (2009), and Winsberg's (2010) emphasis on the specifically computational and material features of computer simulations. Commenting on the philosophical novelty of computational science, writes Humphreys: a "novel feature of computational science is that it forces us to make a distinction between what is applicable in practice and what is applicable only in principle… Ignoring implementation constraints can lead to inadvisable remarks [e.g. about the epistemology of computer simulations]" (2009, 623).

Learning about a simulating system's computational performance is one way to learn about "what is applicable in practice and what is applicable only in principle" with respect to the engineering of novel computing technologies. If some computer simulations are intended to yield new knowledge only about the computing system used in the simulation, then scientific models and simulations need not be vehicles to learn about their represented targets.

Sometimes, scientists do not translate the results of a computer simulation into knowledge about the represented target. Since these simulating systems are computing systems, they instantiate a set of computational, measurable properties. Running a large-scale neural simulation can yield measurements of these properties, which provide information about the computational performance of the system, given some benchmark. Knowing about the computational performance of the system along some dimension of interest can ground the practical design of neuromorphic computing devices.

Examining the relationship between computer simulations and traditional experiments, Parker (2009) stresses "the importance of… understanding computer experiments as, first and foremost, experiments on real material systems. The experimental system in a computer experiment is the programmed digital computer—a physical system made of wire, plastic, etc… In a computer simulation study, scientists learn first and foremost about the behavior of the programmed computer" (488-9).

Learning about the behavior of a programmed computer is far from being trivial or unimportant, as Preissl and colleagues' (2012) work illustrates. *Compass* incorporated "several innovations in communication, computation, and memory" based on available knowledge of some aspects of the function, power and volume of organic brains (10). *Compass* was found to have near-perfect weak and strong scaling when a model was run of the neural dynamics of a large circuit of the macaque's brain. By themselves, these types of results do not yield novel information about some set of computational properties instantiated by biological brains; and, given the aims of Preissl et al.' simulation, they were not translated into knowledge about the represented target system. Instead, the specific importance of these results lies in their offering the basis for developing a novel, efficient, computational architecture that can support a host of neuromorphic applications (Modha et al. 2011).

Having stressed the importance of recognizing that "in a computer simulation study scientists learn first and foremost about the behavior of the programmed computer," Parker (2009) claims that: "from that behavior, taking various features of it to represent features of some target system, they hope to infer something of interest about the target system" (489). This widely-held claim should be qualified in two ways, however.

First, Preissl and colleagues' (2012) study shows that, from the behavior of a computing system that simulates the dynamics of a large-scale neural network, scientists need not draw any inference about the neural system represented in the simulation. Second, assuming that the simulating system does bear some representational relation with a set of computational properties instantiated by some biological neural network allows scientists to characterise the performance discrepancies between neurobiological network and artificial simulating system. The characterisation of this discrepancy can be valuable for some scientific or engineering aim.

The brain is a kind of computing machine. If the brain is a computing machine, then there is a set of properties possessed by both biological brains and artificial computing systems such that specific instantiations of these properties determine the computational performance that the computing machine—biological or otherwise—can reach. From available information, biological brains instantiate determinate properties such that the computational performance they can reach is significantly higher than the performance of the best current artificial super-computers. If these properties are known, and if some information is available about how they determine the performance of biological brains, then scientists may justifiably assume that in some large-scale neural simulation the simulating system imitates some features of the brain relevant to instantiate those computational properties.

Unlike scale models such the scale model of a bridge or of a car, which are typically down-sized or enlarged copies of their target systems, Preissl et al.'s (2012) large-scale neural simulation imitated some features of the brain not in order to serve as a surrogate that is investigated to draw conclusions on the represented neurobiological target. Rather, the assumed representational relation between the simulation and the biological brain justified scientists to draw inferences about how closely the function, power, volume and real-time performance of the brain can be approximated within the limits of current technology. The neural scale and pattern of connectivity embodied in *Compass* challenged its communication, memory and computational capabilities. In the face of these challenges, the simulating system performance could be compared to that of a biological brain along some dimensions of interest like neural spiking rates, latency and bandwidth. For example, running on the IBM Blue Gene/Q supercomputer, *Compass* was found to be 388x slower than real-time performance of the brain, which is useful to characterise its computational performance.

So, in some cases, large-scale neural simulations imitate the brain not in order to serve as a surrogate investigated in its stead. The brain is imitated because it offers a biological benchmark against which the simulating system's design and performance can be assessed. Information about how certain properties determine the computational performance of biological brains can then be used not only to try and instantiate those properties in the design of artificial systems, but also to characterise the discrepancy between the brain's and the simulating system's performance. This characterisation might provide insight into what types of constraints and what determinate properties an artificial computing system need to instantiate for carrying out some task of interest more efficiently.

Conclusions

For *some* large-scale neural simulation, what is learned concerns the computational performance of the simulating system itself. Learning about the computational performance of a computing machine is far from trivial, and can afford knowledge useful for several engineering purposes. Once this role is recognized of some large-scale neural simulations, some widely held beliefs about the epistemology of computer simulations and modelling are in need of qualification. First, computer simulation can involve more than one kind of target system, about which one wants or hopes to acquire new knowledge. Second, when scientific models and computer simulations are employed to gain new knowledge, it is not always knowledge about their represented target systems that is sought. Third, assuming that some large-scale neural simulations imitate some features of their target neurobiological system allows scientists to characterize the performance discrepancies between biological brains and artificial computers, which may help identify constraints on computational efficiency for the design of neuromorphic technologies.

Acknowledgments

Thank you to Lasha Abzianidze, Luigi Acerbi, Julia Bursten and Liz Irvine for their comments on previous drafts. The work on this project was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the priority program “New Frameworks of Rationality” ([SPP 1516])

References

Brette R., *et al.* (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of Computational Neuroscience*, 23, 349-398.

Carandini, M. (2012). From circuits to behavior: a bridge too far? *Nature Neuroscience*, 15, 507-509.

de Garis, H., Shuo, C., Goertzel, B., & Ruiting, L. (2010). A world survey of artificial brain projects, Part I: Large-scale brain simulations. *Neurocomputing*, 74, 3-29.

Modha, D.S., *et al*. (2011). Cognitive Computing. *Communications of the ACM*, 54(8), 62-71.

Eliasmith, C. & Trujillo, O. (2014). The use and abuse of large-scale brain models. *Current Opinion in Neurobiology*, 25, 1-6.

Eliasmith, C., *et al.* (2012). A Large-Scale Model of the Functioning Brain. *Science*, 338, 1202-1205.

Frigg, R. & Hartmann, S. (2012). Models in Science *The Stanford Encyclopedia of Philosophy*, EN Zalta (ed), URL=<http://plato.stanford.edu/archives/fall2012/entries/models-science/>.

Giere, R. (2004). How Models Are Used to Represent Reality. *Philosophy of Science*, 71, S742-752.

Goertzel, B., Ruiting, L., Itamar, A., de Garis, H., Chen S. (2010). A world survey of artificial brain projects, Part ii. Biologically inspired cognitive architectures. *Neurocomputing*, 74, 30-49.

Grüne-Yanoff, T. & Weirich, P. (2010). Philosophy of Simulation. *Simulation and Gaming: An Interdisciplinary Journal*, 41(1), 1-31.

Hill, M.D. (1990). What is scalability? *ACM SIGARCH Computer Architecture News*, 18(4):18-21.

Humphreys, P. (2004). *Extending ourselves: Computational science, empiricism, and scientific method*, New York: Oxford University Press.

Indiveri G, Horiuchi TK (2011). Frontiers in neuromorphic engineering. *Frontiers in Neuroscience*, 5, 10.3389/fnins.2011.00118.

Mainen, Z. F., & Pouget, A. (2014). European Commission: Put brain project back on course. *Nature*, *511*, 534.

Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, 7, 153-160.

Modha, D.S., R. Ananthanarayanan, S.K. Esser, A. Ndirango, A.J. Sherbondy, R. Singh (2011). Cognitive computing. *Communications of the ACM*, 54 (8), 62-71.

Montague, P.R. (2007). *Your Brain is Almost Perfect*. New York: Plume.

Parker, W. (2009). Does Matter Really Matter? Computer Simulations, Experiments and Materiality, *Synthese*, 169(3), 483–96.

Piccinini, G., & Bahar, S. (2013). Neural Computation and the Computational Theory of Cognition. *Cognitive Science*, 34, 453-488.

Preissl R, *et al.* (2012). Compass: A scalable simulator for an architecture for cognitive computing. *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (SC 2012), ed Hollingsworth JK (IEEE Computer Society Press Los Alamitos, CA), 1–11.

Sporns, O. (2012). *Discovering the Human Connectome*. MIT Press, Cambridge, MA.

Swoyer, C. (1991). Structural Representation and Surrogative Reasoning. *Synthese* 87: 449–508.

Teller, P. (2001). Twilight of the Perfect Model. *Erkenntnis*, 55, 393–415.

Weisberg, M. (2013). *Simulation and similarity: using models to understand the world*. Oxford University Press.

Winsberg, E. (2013). Computer Simulations in Science, *The Stanford Encyclopedia of Philosophy*, EN Zalta (ed.), URL=http://plato.stanford.edu/archives/sum2013/entries/simulations-science.

Winsberg, E. (2010). *Science in the age of computer simulations*. Chicago: University of Chicago Press.