The Non-Redundant Contributions of Marr's Three Levels of Analysis for Explaining Information Processing Mechanisms

Abstract

Are all three of Marr's levels needed? Should they be kept distinct? We argue for the distinct contributions and methodologies of each level of analysis. It is important to maintain them because they provide three different perspectives required to understand mechanisms, especially information processing mechanisms. The computational perspective provides understanding of how a mechanism functions in broader environments that determine the computations it needs to perform (and may fail to perform). The representation and algorithmic perspective offer an understanding of how information about the environment is encoded within the mechanism and what are the patterns of organization that enable the parts of the mechanism to produce the phenomenon. The implementation perspective yields an understanding of the neural details of the mechanism and how they constrain function and algorithms. Once we adequately characterize the distinct role of each level of analysis, it is fairly straightforward to see how they relate.

1. Introduction

The term *level* is used in a wide variety of ways. One use refers to levels of organization—a whole system is at a higher level than the parts that constitutes it and the activities the whole performs are at a higher level than those of its components (Craver & Bechtel, 2007). This is not the use Marr (1982) had in mind in distinguishing the computational, representational and algorithmic, and implementational levels. These are levels of analysis, not levels of organization and so, contrary to French et al. (this issue), Marr is not addressing topics such as the emergence of higher levels. The main question raised by the various papers in this issue is whether all three types of analysis are required and, if so, how they relate. The question of how they relate, we will argue, is fairly straightforward once we (1) characterize the *distinctive* role of each type of analysis and (2) show how each type of analysis makes an important, non-redundant contribution to understanding information processing mechanisms.

As the explanatory relevance of the implementational level is generally not contested, we will not dwell on it. Within the context of a mechanistic framework of explanation, it specifies the parts (e.g., neurons) and operations (excitation or inhibition of other neurons) that are employed or recruited in the generation of the phenomenon of interest. Challenges to Marr's conception of levels focus on the representational and algorithmic and the computational levels of analysis. The idea of algorithms governing the transformations of representations is familiar from computer science but, as we will see, is not limited to the sorts of representations and algorithms used in digital computers. Rather, it allows for the identification of the organization of the mechanism that explains how representations are manipulated to generate the phenomenon. We will show how this contributes to explaining the phenomenon in a manner that is not redundant with what is explained at the implementation level. The most distinctive, and least well understood, of Marr's three types of analysis is the computational level. As we will argue, Marr was not concerned just to specify the function being computed, but to explain *why* this function needs to be computed. This requires looking outside the mechanism to the environment and to the tasks that need to be performed in that environment. We begin with the computational level, then turn to the representation and algorithm level.

2. The Computational Level

Marr's notion of computational-level theory has received a variety of interpretations (Shagrir & Bechtel, in press). Many have argued that the computational level aims at stating the cognitive phenomenon to be explained; the explanation itself is then provided at the algorithmic and implementation levels (Bermúdez, 2005; Hardcastle, this volume; Ramsey, 2007). Others have lumped together the computational and algorithmic levels, describing them as sketches that are "elliptical or incomplete mechanistic explanations" (Piccinini & Craver, 2011, p. 284) to be later filled in by full-blown mechanistic explanations. Yet others have associated the computational level with an idealized *competence* and the algorithmic and implementation levels with actual performance (Craver, 2007; Frixione, 2001; Horgan & Tienson, 1994; Polger, 2004; for a teleological variant, see Anderson, this volume). Finally, Egan (2010) associates the computational level with an explanatory formal theory, which mainly specifies the computed mathematical function (see also van Rooij, 2008).

Proponents of Bayesian optimality analysis often refer to Marr, emphasizing that their "focus is on computational-level theories, characterizing the functional capacities of human inference rather than specific psychological processes that implement those functions (Tenenbaum, Griffiths, & Kemp, 2006, p. 206). This level of analysis, they say, "is focused entirely on the nature of the problem being solved – there is no commitment concerning how the cognitive system actually attempts to solve (or approximately to solve) the problem" (Chater, Tenenbaum, & Yuille, 2006, p. 290). While this probabilistic viewpoint might fit with each of interpretations above, it seems closest to the last two interpretations. According to this viewpoint, probabilistic models of cognition provide explanatory quantitative theories of a cognitive capacity without referring to specific psychological and neural mechanisms.

We have defended a different interpretation (Shagrir, 2010; Shagrir & Bechtel, in press) that emphasizes the role of the environment in Marr's notion of computational analysis. Marr characterizes the computational type of analysis as specifying "what the device does and why" (1982, p. 22). Whereas most commentators have addressed only the *what* aspect, Marr insists it includes the *why* aspect whose aim is to demonstrate the basis of the computed function in the physical world (1977, p. 37). Marr associates this *why* aspect with what he calls *physical constraints*, which are physical facts and features in the physical *environment* of the perceiving individual (1982, p. 22-23). These are constraints in the sense that they limit the range of functions that the system could compute to perform a given visual task successfully.

What exactly are the relations between the physical constraints and the computed function? How do these constraints substantiate the basis of the computed function in the physical world? The gist of our interpretation is that Marr assumes implicitly that the visual system mirrors or preserves certain structural relations in the visual field. By *structural relations* we mean "high order" mathematical, geometrical or other formal relations. The visual system would *preserve* these relations if there were an isomorphic mapping from the visual system onto the visual field; more realistically we talk about homomorphism or partial-isomorphism and acknowledge that even these mappings involve a vast amount of approximation and idealization so that a precise morphism relation never actually obtains. Nonetheless, our claim is that a computational analysis appeals to the physical constraints in order to underscore these morphism relations, which, in turn, play both explanatory and methodological roles in theories of vision. Explanatorily they serve to demonstrate the appropriateness and adequacy of the computed function to the information-processing task (Marr, 1982, pp. 24-25). Methodologically they serve to guide discovery of the function that the visual system computes (Hildreth & Ullman, 1989).

Marr never discusses isomorphism or structural similarities explicitly. Nevertheless, we have shown that it is central to his computational analysis of edge detection and stereo vision (Shagrir, 2010; Shagrir & Bechtel, in press). Thus to take the theory of edge-detection, early visual processes compute the zero-crossings of (Laplacian) second derivative filterization of the retinal images. These mathematical relations reflect sharp changes in light reflection in the visual field that often occur along physical edges such as object boundaries (whereas the latter changes can be described in terms of extreme points of first-derivatives or zero-crossings of second derivatives of the reflection function). This physical fact ("constraint") – that sharp changes in reflection often occurs along physical edges – explains why the visual system computes derivation, and not (say) factorization or exponentiation, for the task of edge-detection. It also guides the visual theorist in discovering the mathematical function that the system computes, namely, derivation.

Here we focus briefly on another, non-visual, example—the neural integrator in the oculomotor system. This example indicates that Marr's notion of computational analysis is not confined to vision but is widely applicable in computational cognitive neuroscience.

The neural integrator converts eye-velocity inputs to eye-position-outputs, and thus enables the oculomotor system to move the eyes to the right position (Robinson, 1989; Leigh & Zee, 2006). The inputs arrive from fibers coding vestibular, saccadic or pursuit movements (figure 1); the system produces eye-position codes by computing mathematical integration over these eye-velocity encoded inputs. In cats, monkeys, and goldfish, the network that computes *horizontal* eye movements appears to be localized in two brainstem nuclei, the nucleus prepositushypoglossi (NPH) and the medial vestibular nucleus (MVN).

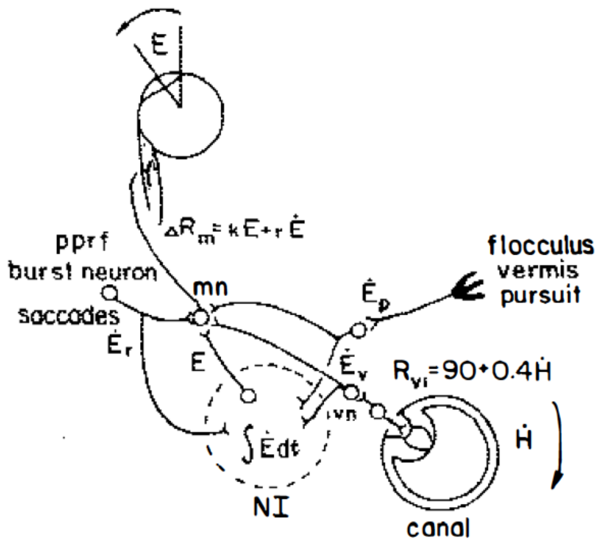

Figure 1: The neural integrator (NI) receives eye-velocity coded inputs, $\dot{E}$, and, computing integration, produces eye-velocity coded outputs, E. The hypothesis is that the integrator is common to vestibular, saccadic and pursuit movements, thus receiving vestibular ($\dot{E}_v$), saccadic ($\dot{E}_r$), and pursuit ($\dot{E}_p$), velocity coded inputs. On the right it is shown how the head velocity signals, $\dot{H}$ are converted into eye-velocity codes ($\dot{E}_v$). These codes are projected directly to the motoneurons (mn) that have to produce velocity commands, but also the neural integrator (NI) which produces position codes projected to the motoneurons for position commands (from Robinson 1989, p. 35).

Mathematical integration characterizes operations performed in two *very different* places. One is in the neural representing system, namely, the neural integrator. It performs integration on the neural inputs to generate neural commands. This is of course the reason that the system is known as *integrator*. Another and very different place, however, is in the target domain being represented, in our case the eyes. The relation between position and velocity of the eye can be described in terms of integration too! The distance between the previous and current positions of the eye is determined by integrating over its velocity with respect to time. So what we have here is an (iso-)morphism between the representing sensory-motor neural system (the integrator) and the represented target domain (the eyes and their properties). The neural integrator mirrors or preserves certain relation in the target domain, namely the distances between two successive eye positions. By computing integration, the neural function mirrors, reflects or preserves the integration relation between eye velocity and eye positions.

Let us put these findings in the context of Marr's notion of computational analysis. The *what* aspect describes the mathematical function, integration, computed by the neural integrator. The *why* aspect relates the computed function with the physical environment, namely, the eyes with their properties. The analysis invokes a physical constraint, which in our case is the relation between eye-velocity and the distance between successive eye-positions. Using this constraint, it is shown that there is a morphism mapping relation between the neural function and the target domain. This mapping relation is underscored by the fact that the two domains have a shared structure, which is mathematical integration.

As said, the morphism relation plays both explanatory and methodological roles. On the explanatory side, it serves to explain *why* computing integration is appropriate for the task of controlling eye movement. The neural network computes integration and not, say, multiplication, exponentiation, or factorization, *because* integration preserves the integration relation between eye movement and eye positions in the target domain. Factorizing numbers would not result in moving the eyes to the right place, precisely because it does not preserve relations in the target domain that are relevant to eye movements. Integration does: When you compute integration over eye-velocity encoded inputs, you mirror the integration relation between velocity and position; hence, you output representations of a new eye position. The algorithmic and implementation levels complement this explanation by specifying how this integration function is carried out in the neural system.

On the methodological side, the morphism relation is instrumental in discovering what function is computed. In our example, experimental electrophysiological results indicated that the neural system converts eye-velocity pulses into eye-position codes. Looking at the relation ("physical constraint") between the represented velocity and position, theoreticians quickly inferred that the internal relations between the representing states must be of integration. This logic of discovery assumes that the computed function is that of integration since the computed function must correspond to the velocity-position integration relation, which is already known.

How does this interpretation relate to the Bayesian approach to cognition? It is not easy to answer this question because there appear to be multiple Bayesian approaches (Jones & Love, 2011). We will make few preliminary points. It is obvious that both Marr and the Bayesians hold the conviction that theories of cognition go above and beyond mechanisms, whether algorithms or their neural implementations. And they both think that the computational analysis of the task does not focus on psychological and neural mechanisms; rather it highlights and identifies non-mechanistic elements of a cognitive phenomenon. Marr and the Bayesians are also in full agreement that this computational analysis ideally produces mathematical or formal descriptions, and that these descriptions are explanatory.

Another point of convergence between Marr and the Bayesians is that computational analysis provides some sort of an optimal solution to the problem. Marr says that computational theories state "that what is being computed is optimal in some sense" (p. 19), and he compares Chomsky's notion of competence with his computational theories (p. 28). Bayesian models aim to show how a problem can be solved in principle, which amounts to how rational agents should update their beliefs in light of new data (Griffiths, Kemp, & Tenenbaum, 2008). Despite the similarities, there is an important difference—Marr's theories aim at the characterization of the real (even if idealized) mathematical function computed by the cognitive system, whereas the Bayesians models aim at the characterization of the function that the cognitive system should compute and assumes that the actual cognitive system approximates this solution.

A further point of divergence concerns the elements that are being included in computational models. Marr definitely assumes that some elements at the computational level are representations. However, he thinks that questions about kinds of representations belong to the algorithmic level. To use his example of an adding machine, the computational level specifies that the machine computes addition; part of the specification is that the inputs and outputs represent numbers. But whether the machine uses Arabic, Roman or binary representational system is a question left to the algorithmic level (Marr, 1982, p. 20ff). Bayesians, it seems to us, put more emphasis on the kind and structure of representations involved in cognitive functions, and their models often refer to internal representational structures. For example, they show that relations between different biological species could be represented by a tree, a ring, a set of clusters, or a low dimensional space. (Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010)

What is the role of the environment in the computational analysis? Jones and Love (2011) argued that the Bayesians downplay the role of the environment in their analysis. Chater, Goodman, Griffiths, Kemp, Oaksford, and Tenenbaum (2011) responded that as a matter of fact Bayesian analysis is often based, explicitly, on assumptions about environmental structure. In this respect the Bayesian conception of computational analysis might not be very different from Marr's (as we interpreted him). But our interpretation emphasizes the role of the environment in the analysis. We have shown, first, that the environment underlies the explanatory power of the computational analysis. The explanatory role of the environment extends to the Bayesians examples as well. A directed graph is a better model of the properties generated by a causal transmission process (Tenenbaum, Griffiths, & Kemp, 2006) precisely because a directed graph preserves the formal structure of the causal process, whereas (say) a taxonomic tree does not.

The environment also plays a role in fixing the appropriate mathematical description. Chater et al. (2011) say that what is distinctive about Bayesian approach is "a top-down, or 'function-first' research strategy, as recommended by Marr (1982): from computational, to algorithmic, to implementational levels" (p. 196; see also Griffith et al. 2010). They give as an example a pocket calculator that uses input and output symbols we do not well understand. Their claim is that we are much more likely to understand what algorithms are carried out if we first realize that the system computes addition than the other way around. This claim is certainly in accord with Marr's approach. The question, however, is how do we figure out that the system computes addition and not another function. Marr's answer is that when we study cognitive systems we use cues from the environment that constrain the computed function. We see that the relation between the encoded velocity and the encoded position is that of integration to conclude that the neural system is an integrator. Marr was overly optimistic about this method, thinking that the environmental constraints are always apparent to us; however, it is often not the case (Shagrir & Bechtel, in press). In this respect, techniques developed in the Bayesian analysis might constitute an important contribution to the proposed function-first methodology (Yuille & Kersten, 2006). What we emphasize, however, is that the prospects of this methodology crucially depend, as Marr noticed, on successful deployment of cues from the environment.

3. The representation and algorithm level

Those authors who downplay the distinction between the representation and algorithm type of analysis and the implementation type of analysis (Bickle, this issue; Hardcastle, this issue) implicitly hold that the specific details of the neural implementation are all that is required for explanation. But while the specific parts and operations are certainly relevant to explaining a phenomenon, they are not the only relevant factors. As any engineer or designer knows, it is also crucially important to understand the way in which the parts and operations are organized and how this organization facilitates generation of the phenomenon. Different ways of putting the same parts together will result in different phenomena (many of them not very interesting) and the challenge for a designer or engineer is to discover an organizational design that is able to generate the phenomenon they desire. For scientists the challenge is much the same, although they are typically engaged in reverse engineering—trying to discover the organization that enabled the parts together to produce the phenomenon.

Someone critical of treating the representation and algorithm level as distinct from the implementation level might note that in any implementation a particular organization is realized. However, crucial to the endeavors of both designers and scientists is the discovery of *design principles*—patterns of organization that produce the same results across a wide range of different implementations. If this were not possible, then each specific organization would have to be analyzed on its own to determine its effects by, for example, representing the important properties of all of its parts in differential equations, identifying the correct parameters, and simulating the behavior of the whole. To explain why a given physical system behaved a particular way one could do no more than appeal to such a simulation. But designers and increasingly scientists have found success with a different strategy—identifying design principles that generate the same phenomenon when realized in mechanisms composed of different parts performing different operations as long as they maintain the required relations to each other. These relations are often presented in graphs in which nodes represent entities and edges the effects of specific nodes on others.

Identifying design principles requires abstracting from the details of a particular instantiation (Levy & Bechtel, 2013). A graph such as the one below showing a double-negative feedback motif (Figure 2) does not indicate what plays the roles of X and Y, only that each inhibits the activity of the other. To determine what will happen in a mechanism in which such a design is implemented, researchers turn to computational modeling. For example, after finding the double-negative feedback motif occurring frequently in gene regulation networks in eukaryotic cells, Tyson and his collaborators (Tyson, Chen, & Novák, 2003; Tyson & Novák, 2010) developed computational models and demonstrated that this design can, under a broad range of parameter values, generate a bi-stable switch that requires a higher level of the input to turn on but will only turn off when the input drops to a significantly lower level. This is particularly useful in forcing normally reversible processes to operate sequentially.

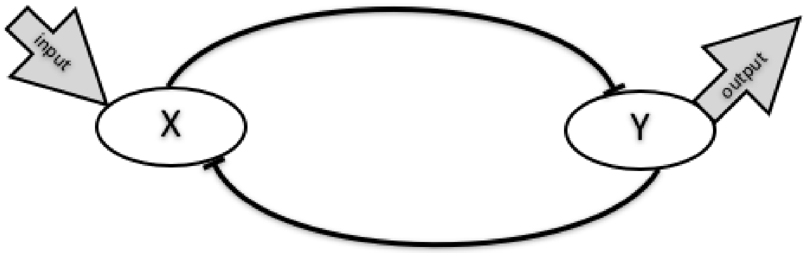

Figure 2. Double-negative feedback loop motif

The example we just offered is from molecular biology, not cognitive science or neuroscience. Accordingly, while it might seem that one can describe a motif like the double-negative feedback loop in an algorithm, it doesn't seem to be operating over representations and involved in information processing. As such, it might not seem appropriate to Marr's level of analysis. But in fact molecular biologists are increasingly employing information processing vocabulary to describe circuits like these whose parts (e.g., transcription factors) carry information that needs to be processed by the cell in determining its responses. Moreover, very similar analyses are currently being developed by Sporns and his colleagues (Sporns & Kötter, 2004; Sporns, 2010) in their attempts to understand the patterns of connection in the brain. Treating whole brain regions as units (that serve representational functions), they find the dual-dyad motif (Figure 3) occurring particularly frequently in contexts in which the apex node is a hub-region that has an especially large number of connections to other regions. Computational analysis of this motif reveals that it is especially effective in promoting synchronization of activity with zero phase-lag across long distances (Vicente, Gollo, Mirasso, Fischer, & Pipa, 2008), a function important in connecting regions representing related information and so underlying cognitive performance.

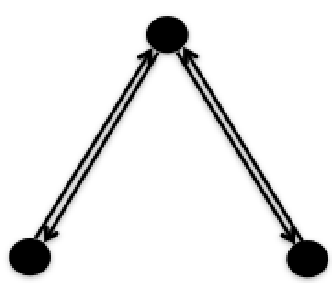

Figure 3. Dual-dyad motif

We have identified two simple motifs that have been shown to have important information processing functions either in cells or in the brain. Both cell and brain networks are enormously complex, and identifying motifs is only one tool for understanding their function. Other tools are being developed to analyze larger scale networks (see van den Heuvel & Sporns, 2011, for discussion of what is called the "rich-club" hub structure in the primate brain). What is important is that these network analyses abstractly characterize the organization in the mechanism and use computational analysis to determine the contribution to the performance of a phenomenon. These results are established without knowing what entities implement the activity of the nodes and what operations implement the edges. Moreover, we contend that such accounts are indicative of the type of analysis Marr had in mind in emphasizing the importance of the representation and algorithm level of analysis and differentiating it from the implementational level.

4. Conclusion: Relating Marr's Levels

In the two previous sections we have emphasized (1) the *distinctive* role of Marr's computational and representation and algorithmic analyses and (2) shown how both make an important contribution to understanding information processing mechanisms that is non-redundant to that of the other or of the implementation level. What remains to discuss is how these levels are related. Each offers answers to different questions that nonetheless are related to each other. This can be seen by recognizing that we are dealing with information processing mechanisms and that understanding mechanisms requires integrating different pieces of information. First, one requires information about the phenomenon being explained by the mechanism. The computation level, by focusing on the structure of the environment from which organisms need to acquire information to live their lives, specifies the phenomenon to be explained. Empirical, including experimental, investigation at this level is crucial since our intuitive understanding of the task may be wrong and if it is, so are the explanations offered for it. Second, in any situation where the operations of the individual parts are organized in a complex manner, explaining how the mechanism works requires understanding the contribution of the organization. The discovery and articulation of design principles provides accounts of what types of phenomena may be produced given those designs. As in engineering, identifying the relevant design principles and understanding what behavior they will generate under a range of implementations is different than determining what is actually performing the various roles in a specific mechanism. This is not to discount the implementation level. Since not every implementation will satisfy the conditions for the algorithm (design), it is important to investigate the actual implementation. Moreover, many features of the phenomenon result from the details of the implementation.

In addition to answering different questions about a mechanism, research at each of Marr's three levels of analysis can productively constrain research at other levels. Working from the top-down, knowing the structure of the environment and the information it makes available to the organism limits the types of information processing algorithms that can utilize that information. Likewise, having identified a mode of organization and the conditions under which it will generate a form of behavior can guide the search for the components that implement the design. Constraints also arise from the bottom up. Knowing features of the implementation can put constraints on the search for algorithms. Some algorithms might not be implementable given the components available, and alternatives must be sought. Likewise, knowing the algorithm that seems to be functioning can guide investigations into the environment and reveal different features of its structure that are relevant to the organism.

References

Bermúdez, J. L. (2005). *Philosophy of psychology : a contemporary introduction*. New York: Routledge.

Chater, N., Goodman, N., Griffiths, T. L., Kemp, C., Oaksford, M., & Tenenbaum, J. B. (2011). The imaginary fundamentalists: The unshocking truth about Bayesian cognitive science. *Behavioral and Brain Sciences, 104*, 181-204.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences, 10*, 287-291.

Craver, C. F. (2007). *Explaining the brain: Mechanisms and the mosaic unity of neuroscience*. New York: Oxford University Press.

Craver, C. F., & Bechtel, W. (2007). Top-down causation without top-down causes. *Biology and Philosophy, 22*, 547-563.

Egan, F. (2010). Computational models: a modest role for content. *Studies in History and Philosophy of Science, 41*, 253-259.

Frixione, M. (2001). Tractable Competence. *Minds and Machines, 11*, 379-397.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: exploring representations and inductive biases. *Trends in Cognitive Sciences, 14*, 357-364.

Griffiths, T. L., Kemp, C., & Tenenbaum, J. B. (2008). Bayesian models of cognition. In R. Sun (Ed.), *Cambridge Handbook of Computational Cognitive Modleing* (pp. 59-100). Cambridge: Cambridge University Press.

Hildreth, E. C., & Ullman, S. (1989). The computational study of vision. *Foundations of cognitive science* (pp. 581-630): MIT Press.

Horgan, T. E., & Tienson, J. L. (1994). A nonclassical framework for cognitive science. *Synthese, 101*, 305-345.

Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences, 34*, 169-231.

Leigh, R. J., & Zee, D. S. (2006). *The neurology of eye movements* (4th ed.). New York: Oxford University Press.

Levy, A., & Bechtel, W. (2013). Abstraction and the organization of mechanisms. *Philosophy of Science, 80*, 241-261.

Marr, D. C. (1977). Artificial Intelligence - personal view. *Artificial Intelligence, 9*, 37-48.

Marr, D. C. (1982). *Vision: A computation investigation into the human representational system and processing of visual information*. San Francisco: Freeman.

Piccinini, G., & Craver, C. (2011). Integrating psychology and neuroscience: functional analyses as mechanism sketches. *Synthese, 183*, 283-311.

Ramsey, W. (2007). *Representation reconsidered*. Cambridge, UK New York: Cambridge University Press.

Robinson, D. A. (1989). Integrating with neurons. *Annual Review of Neuroscience, 12*, 33-45.

Shagrir, O. (2010). Marr on computational-level theories. *Philosophy of Science, 77*, 477-500.

Shagrir, O., & Bechtel, W. (in press). Marr's computational-level theories and delineating phenomena. In D. M. Kaplan (Ed.), *Integrating psychology and neuroscience: Prospects and problems*. Oxford: Oxford University Press.

Sporns, O. (2010). *Networks of the brain*. Cambridge, MA: MIT Press.

Sporns, O., & Kötter, R. (2004). Motifs in brain networks. *PLoS Biol, 2*, e369.

Tenenbaum, J. B., Griffiths, T. L., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Science, 10*, 309-318.

Thomas W. Polger. (2004). Neural Machinery and Realization. *Philosophy of Science, 71*, 997-1006.

Tyson, J. J., Chen, K. C., & Novák, B. (2003). Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Current Opinion in Cell Biology, 15*, 221-231.

Tyson, J. J., & Novák, B. (2010). Functional motifs in biochemical reaction networks. *Annual Review of Physical Chemistry, 61*, 219-240.

van den Heuvel, M. P., & Sporns, O. (2011). Rich-club organization of the human connectome. *The Journal of Neuroscience, 31*, 15775-15786.

van Rooij, I. (2008). The tractable cognition thesis. *Cogn Sci, 32*, 939-984.

Vicente, R., Gollo, L. L., Mirasso, C. R., Fischer, I., & Pipa, G. (2008). Dynamical relaying can yield zero time lag neuronal synchrony despite long conduction delays. *Proceedings of the National Academy of Sciences, 105*, 17157-17162.

Yuille, A., & Kersten, D. (2006). Vision as Bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences, 10*, 301-308.