STING occurs almost exclusively in bacteria lacking other signalling pathways that also involve 3′,3′ c-di-GMP, thereby avoiding this potential conflict.

Human STING defends against viruses by relying on the expression of antiviral genes, such as those encoding interferon proteins, which have been identified only in vertebrates. However, bacterial CBASS systems instead fight viral infection by either arresting bacterial cell growth or inducing cell death to prevent further phage spread[10].

Morehouse *et al.* report that bacterial STING most commonly exists as a STING–TIR fusion protein that has a STING domain connected to a TIR domain, which is involved in plant and animal defence responses. The TIR domain is best known for its role in protein–protein interactions in mammalian defence pathways that are part of the innate immune response, which provides a broad defence against pathogens. Some TIR domains in plants and animals also have enzymatic activity[13,14] that degrades the molecule NAD^+, which is essential for cellular metabolism.

The authors showed that the presence of 3′,3′ c-di-GMP was sufficient to cause a bacterial STING–TIR fusion protein to assemble into long filaments that rapidly degraded NAD^+. This NAD^+ destruction halted cell growth in the bacterium *Escherichia coli*. Mutation in the CDN binding site blocked the toxicity of the system in *E. coli*, suggesting that 3′,3′ c-di-GMP controls filament formation and NAD^+ destruction mediated by the TIR domain.

STING–TIR fusion proteins are not limited to bacteria. Using a bioinformatics approach, Morehouse *et al.* identified such proteins in some invertebrates, including the Pacific oyster (*Crassostrea gigas*). Structural analysis of a STING–TIR fusion protein from *C. gigas* revealed that it binds tightly to 2′,3′-cGAMP, which is the CDN that most potently binds to and activates mammalian STING. Notably, in 2′,3′-cGAMP, the phosphodiester bonds between the nucleotides guanosine monophosphate (GMP) and adenosine monophosphate (AMP) have an asymmetric pattern of linkages (a 2′–5′ linkage between the 2′-OH group of GMP and the 5′-phosphate group of AMP, and a 3′–5′ linkage between the 3′-OH of AMP and 5′-phosphate of GMP). This arrangement is found in many multicellular animals, but not in bacteria, and suggests that the dominant ligand for STING changed after it was acquired by our animal ancestors. The reason for this change is unclear.

One unresolved question is how the bacterial cGAS–STING pathway is activated by phage infection. Morehouse *et al.* showed that, like many bacterial cGAS-like proteins[5,15], purified CdnE protein is constitutively active *in vitro*. Therefore, it is possible that the active protein is normally inhibited and is released from inhibition only on phage infection. An example of this type of system is the cGAS-like enzyme DncV in the bacterium *Vibrio cholerae*, which is inhibited by metabolites (folate-like molecules) that are presumably depleted during phage infection[16]. More research will be needed to determine whether this is how CdnE and other cGAS-like proteins are regulated, or if other regulatory mechanisms exist.

As we learn more about the diverse and complex defence systems in bacteria, it might be tempting to consider these immune systems as mirroring those of vertebrates. For example, the CRISPR–Cas system used by organisms such as bacteria can form what is akin to an immunological memory to fight specific phage reinfection. This shows echoes of our own adaptive immune systems, which can remember and respond to specific pathogens. Likewise, the bacterial CBASS systems, including the cGAS–STING pathway, provide broad protection against phage invasion, much as our own innate immune systems do. Interestingly, whereas the CRISPR–Cas system is absent in humans, and specialized immune cells called T and B cells instead do the job, the cGAS–STING pathway and its antiviral defence function are preserved from bacteria to humans.

Justin Jenson and **Zhijian J. Chen** are in the Department of Molecular Biology, University of Texas Southwestern Medical Center, Dallas, Texas 75390, USA. **Z.J.C.** is also at the Center for Inflammation Research and the Howard Hughes Medical Institute, University of Texas Southwestern Medical Center.
e-mails: justin.jenson@utsouthwestern.edu; zhijian.chen@utsouthwestern.edu

1. Margolis, S. R., Wilson, S. C. & Vance, R. E. *Trends Immunol.* **38**, 733–743 (2017).
2. Ablasser, A. & Chen, Z. J. *Science* **363**, eaat8657 (2019).
3. Burroughs, A. M., Zhang, D., Schäffer, D. E., Iyer, L. M. & Aravind, L. *Nucleic Acids Res.* **43**, 10633–10654 (2015).
4. Cohen, D. *et al. Nature* **574**, 691–695 (2019).
5. Whiteley, A. T. *et al. Nature* **567**, 194–199 (2019).
6. Morehouse, B. R. *et al. Nature* **586**, 429–433 (2020).
7. Sun, L., Wu, J., Du, F., Chen, X. & Chen, Z. J. *Science* **339**, 786–791 (2013).
8. Ishikawa, H. & Barber, G. N. *Nature* **455**, 674–678 (2008).
9. Wu, J. *et al. Science* **339**, 826–830 (2013).
10. Millman, A., Melamed, S., Amitai, G. & Sorek, R. *Nature Microbiol.* https://doi.org/10.1038/s41564-020-0777-y (2020).
11. Ross, P. *et al. Nature* **325**, 279–281 (1987).
12. Römling. U., Galperin, M. Y. & Gomelsky, M. *Microbiol. Mol. Biol. Rev.* **77**, 1–52 (2013).
13. Essuman, K. *et al. Neuron* **93**, 1334–1343 (2017).
14. Horsefield, S. *et al. Science* **365**, 793–799 (2019).
15. Davies, B. W., Bogard, R. W., Young, T. S. & Mekalanos, J. J. *Cell* **149**, 358–370 (2012).
16. Zhu, D. *et al. Mol. Cell* **55**, 931–937 (2014).

This article was published online on 28 September 2020.

Computer science

Brain-inspired computing becomes complete

Oliver Rhodes

Hardware modelled on the brain could revolutionize computing, but implementing algorithms on such systems is a challenge. A proposed conceptual framework could simplify implementation, accelerating research in this field. **See p.378**

The next generation of high-performance, low-power computer systems might be inspired by the brain. However, as designers move away from conventional computer technology towards brain-inspired (neuromorphic) systems, they must also move away from the established formal hierarchy that underpins conventional machines — that is, the abstract framework that broadly defines how software is processed by a digital computer and converted into operations that run on the machine's hardware. This hierarchy has helped enable the rapid growth in computer performance. On page 378, Zhang *et al.*[1] define a new hierarchy that formalizes the requirements of algorithms and their implementation on a range of neuromorphic systems, thereby laying the foundations for a structured approach to research in which algorithms and hardware for brain-inspired computers can be designed separately.

The performance of conventional digital computers has improved over the past 50 years in accordance with Moore's law, which states that technical advances will enable integrated circuits (microchips) to double their resources approximately every 18–24 months. But although such advances enable ever-more-powerful hardware, they also create challenges for system architects looking to optimize the performance of algorithms executed on these constantly changing devices.

An important feature of conventional computer design that has allowed the best performance to be obtained from new devices (chips, memory, and so on) has been the absence of a tight coupling between software and hardware development. By setting minimum requirements for hardware, it has become feasible to transform a software program written in a high-level language into the precisely equivalent instruction sequence needed for any machine, a process known as compilation (Fig. 1). Computers that support the use of instructions representing fundamental computational operations in this compilation process are said to be Turing complete. Software code is therefore generally written just once, and can then be compiled and executed on multiple Turing-complete processor architectures to produce equivalent results.

However, it is widely acknowledged that the era of progress characterized by Moore's law is coming to an end: rates of advance in digital-computer power seem to be slowing. Moreover, digital computing can be highly energy-consuming, prompting a search for alternatives. Scientists have long been fascinated by the computational abilities of the brain, which is not only incredibly energy efficient, but also boasts unique information-processing performance as a result of its architecture of neurons and synapses. This has inspired the field of neuromorphic computing, an area of research that uses the architecture of neural networks in the brain as the basis for next-generation computers[2].

The focus of neuromorphic computing is typically on spiking neural networks — systems of interconnected artificial neurons in which each neuron exhibits a short 'spike' of activity when its level of activation reaches a threshold value[3]. Such systems are more similar to biological neural networks than are the artificial neural networks commonly used in modern deep-learning applications. Neuromorphic hardware has been produced in a range of formats, both digital and analog. However, most systems share common design principles, such as co-location of the memory and processor[2].

A challenge for researchers developing applications of neuromorphic hardware is that a formal hierarchy such as Turing completeness does not currently exist. Instead, each new chip architecture requires a custom software toolchain — a set of programming tools — that defines algorithms and executes them by mapping them onto the unique hardware. This makes it difficult to compare the performance of different neuromorphic systems executing the same algorithm, and requires researchers to understand all aspects of the algorithm and hardware to obtain the potentially brain-like performance.

Zhang *et al.* now present a breakthrough solution to this problem by proposing a concept that they call neuromorphic completeness — which, in a nod to Turing completeness, aims to decouple algorithm and hardware development. In a relaxation of the hierarchy for conventional computers, the authors propose that a brain-inspired system is neuromorphic complete if it can execute a given set of fundamental operations with a prescribed level of accuracy (Fig. 1). This is a deviation from Turing completeness, in which a system can be defined as complete only if it provides an exact and equivalent result for a given set of fundamental operations.

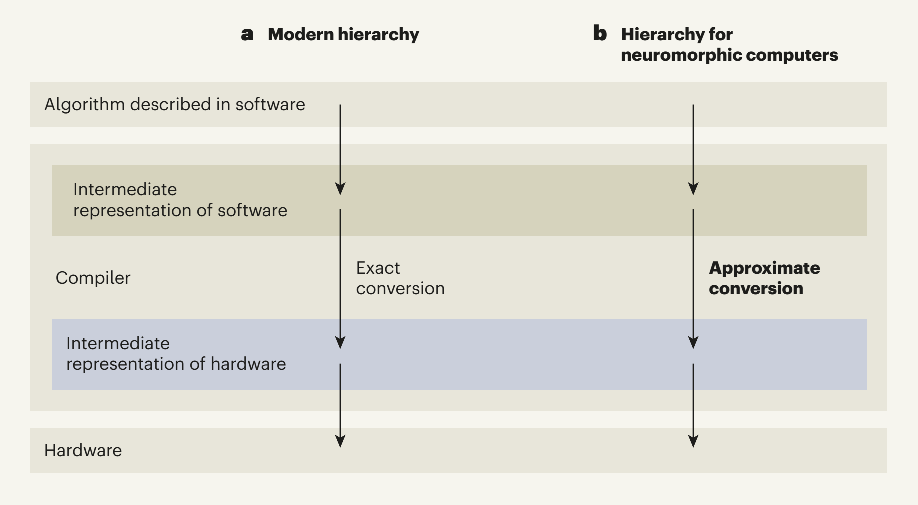
a Modern hierarchy
b Hierarchy for neuromorphic computers
Algorithm described in software
Intermediate representation of software
Compiler
Exact conversion
Approximate conversion
Intermediate representation of hardware
Hardware

Figure 1 | Hierarchies for implementing algorithms on computer hardware. a, A computer hierarchy broadly defines how software is processed by modern digital computers. Algorithms written in a high-level computer language are broken down into fundamental computing operations to produce an intermediate representation of the software. These operations are converted into an exactly equivalent intermediate representation of hardware — a set of instructions that is then run on the hardware. Software can thus be developed separately from hardware. However, no similar hierarchy had been defined for neuromorphic computers (those that use networks of artificial neurons as the basis of their computations). **b**, Zhang *et al.*[1] now propose a similar hierarchy for neuromorphic computers, in which the intermediate representation of the hardware is only an approximation of the intermediate representation of the software — overcoming the difficulty of producing exact representations in neuromorphic systems. This hierarchy will allow hardware and software for neuromorphic computers to be developed separately, rather than being co-developed for each application, as they are now.

Fundamental operations in the proposed neuromorphic-complete framework include two known as the weighted-sum operation and the element-wise rectified linear operation, which enable hardware systems to support both spiking and non-spiking artificial neural networks. The authors demonstrate how their hierarchy for brain-inspired computing provides a mechanism for converting a given algorithm into a form suitable for a range of neuromorphic-complete devices.

A welcome feature of the new hierarchy is that a continuum of completeness is proposed — different levels of algorithm performance can be accepted, depending on the accuracy with which a neuromorphic system can execute the fundamental operations. This continuum of completeness means that the new hierarchy can be implemented using the whole range of available analog and digital neuromorphic systems, including those that sacrifice accuracy for execution speed or energy efficiency.

The continuum of completeness also allows different implementations of an algorithm to be run on the same hardware — for example, to explore how algorithm accuracy can be traded off against chip size to reduce power consumption. Zhang *et al.* demonstrate this aspect of their approach in the execution of algorithms for three tasks ('driving' an unmanned bicycle, simulating the movement of flocks of birds, and performing a linear algebra analysis called QR decomposition). Each task was executed using three typical neuromorphic-complete hardware platforms: the authors' own neuromorphic chip[4]; a graphics-processing unit (GPU) used in conventional computers; and a platform, based on devices called memristors, that accelerates the execution of neural networks.

The proposed hierarchy is a welcome step for the field, because it enables comparison of different hardware platforms implementing equivalent versions of the same algorithm, and comparison of different algorithms implemented on the same hardware. These are both crucial tasks for effective benchmarking of neuromorphic architectures. The inclusion of conventional Turing-complete hardware (the GPU) in their proof-of-principle experiments is also extremely valuable, because this demonstrates that the hierarchy could potentially be used to prove the superiority of neuromorphic devices over mainstream systems for certain applications.

Another substantial benefit of the proposed hierarchy is its potential to split algorithm and hardware development into independent research streams. The scale and complexity of algorithms will need to increase over time if the benefits of the underlying neuromorphic architectures are to be obtained, and so this split will help researchers to focus on specific aspects of research problems, rather than trying to find entire end-to-end solutions. This is likely to result in better understanding of the problems, and feed into the design of higher-performing neuromorphic systems in the future.

There is still much to be done to unite the work carried out by the many industrial and academic research groups in the field of neuromorphic computing. Zhang and colleagues' proposed hierarchy is a useful step in this direction. It remains to be seen whether actual brains — biological 'hardware' — are themselves neuromorphic complete, but the authors' approach nevertheless brings us closer to the great gains that could be made using brain-inspired hardware.

Oliver Rhodes is in the Department of Computer Science, University of Manchester, Manchester M13 9PL, UK.
e-mail: oliver.rhodes@manchester.ac.uk

1. Zhang, Y. *et al. Nature* **586**, 378–384 (2020).
2. Roy, K., Jaiswal, A. & Panda, P. *Nature* **575**, 607–617 (2019).
3. Gerstner, W., Kistler, W. M., Naud, R. & Paninski, L. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition* (Cambridge Univ. Press, 2014).
4. Pei, J. *et al. Nature* **572**, 106–111 (2019).

Neuroscience

Brain's immune cells put the brakes on neurons

Thomas Pfeiffer & David Attwell

Microglia are the brain's immune cells. A previously unknown role for microglia has now been uncovered: providing negative feedback to active neurons, to help the brain process information. **See p.417**

Neural circuits in the brain rely on neuronal excitation (a positive change in the electrical potential across the cell membrane), combined with delayed inhibition (Fig. 1). Inhibition is crucial for keeping neuronal activity in the optimal range for encoding information, minimizing the brain's energy use and computing useful neuronal outputs. It has conventionally been thought that inhibition is mediated by a neuronal subtype called interneurons that release neurotransmitter molecules (such as the amino acid GABA) to make the membrane potential of the downstream neuron more negative — although neurotransmitter release from non-neuronal cells called astrocytes can also contribute[1]. On page 417, Badimon *et al.*[2] extend this repertoire of inhibitory influences to include microglia, the resident immune cells of the brain. The authors' work raises fascinating questions about the role of microglia in information processing.

Badimon and colleagues took advantage of the fact that blocking activation of the growth-factor receptor protein CSF1R in mice leads to a lack of microglia[3]. The authors found that if they gave neurostimulants to animals that lacked microglia, the drugs produced long-lasting epileptic seizures, indicative of hyperactive neuronal excitation. Seizures were not observed in wild-type animals receiving the same drugs, indicating that microglia normally exert a brake on neuronal activity. This result echoes and extends two previous studies[4,5]. Microglial processes are attracted to the cell bodies (housing the nucleus) of active neurons by the release of ATP molecules. There, the processes decrease neuronal activity, both normally[4] and in pathological[5] conditions.

Whereas these previous studies focused on cell bodies, Badimon and colleagues focused on the synaptic junctions between neurons, which also release ATP to attract microglial processes. The microglial enzyme CD39 converts ATP into ADP (and then into AMP); ADP activates $P2Y_{12}$ receptor proteins found only on microglia (go.nature.com/3iuewxa and go.nature.com/33hwjft; Fig. 2). Blocking $P2Y_{12}$ receptors has been shown to inhibit the attraction of microglia to cell bodies and synapses[5], and Badimon *et al.* found that such a block also reduces neuronal inhibition by microglia in response to neurostimulants.

How might microglia–neuron interactions inhibit the electrical activity of neurons? The authors found that deleting microglia decreased extracellular levels of the molecule adenosine (ADO). Pharmacologically blocking CD39 or the downstream enzyme CD73 (which converts AMP into ADO; Fig. 2) also lowered ADO levels. Furthermore, blocking the activity of CD39 increased the susceptibility of mice to seizures in response to neurostimulants. Together, these observations implicate ADO as the microglia-derived factor that dampens neuronal activity.

It is well known that ADO lowers neuronal excitability[6]. Indeed, the reason that coffee makes us more alert is that caffeine blocks ADO's inhibitory effects. ADO lowers excitability by acting on what are called A1 receptors, which (by lowering the concentration of the intracellular messenger molecule cyclic AMP) decrease the release of the excitatory neurotransmitter glutamate, and reduce its effects on the downstream neuron that receives the neurotransmitter. A1 receptors also activate potassium ion channels in neuronal membranes to keep their membrane potential negative (and so keep the neurons unexcited). Thus, Badimon *et al.* have uncovered a previously unknown feedback loop for neuronal regulation mediated by microglia, which, when attracted to active synapses, generate ADO to inhibit excessive neuronal activity (Fig. 2).

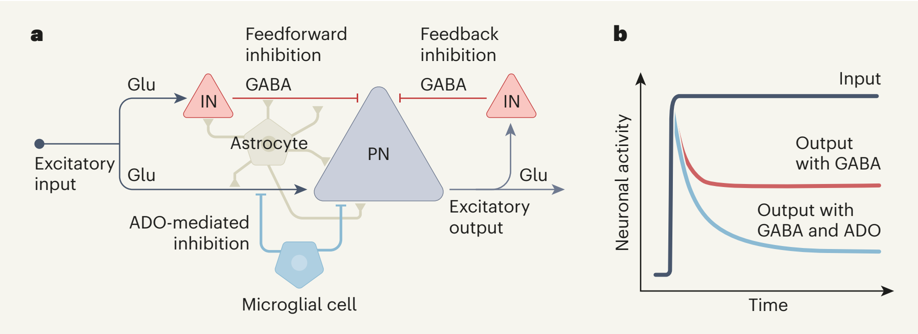

Figure 1 | Inhibition of active neurons by microglial cells. a, A generic neuronal circuit, centred on a principal neuron (PN). The PN and an excitatory input to the circuit both release the excitatory neurotransmitter molecule glutamate (Glu). Interneurons (IN) release the inhibitory neurotransmitter GABA. Neurotransmitters derived from cells called astrocytes fine-tune the neuronal circuits (these signals are not shown). The circuit is also inhibited by the molecule adenosine (ADO), which Badimon *et al.*[2] show is generated, in part, by microglial cells. **b,** When the input to the circuit is increased, GABA-mediated inhibition decreases the output on a rapid timescale. Microglia-derived ADO adds a slower component to the inhibition.