Preface

Unconventional computing is chock full of theoretical stuff. There are a just a handful of experimental laboratory prototypes. They are outstanding but difficult for non-experts to play with. We show how to make a universal biological computer at home nearly for free. All you need is a slime mould, oat flakes and a camera. The rest is up to your creativity.

Plasmodium of the acellular slime mould *Polycephalum polycephalum* is our computing substrate. The plasmodium is a single monstrously large cell. The plasmodium behaves as a nonlinear medium, an excitable soft matter, encapsulated in an elastic and growing membrane.

In Chap. 1 we establish links between reaction–diffusion chemical computers and plasmodium of *P. polycephalum*. We also provide an insight on history of Physarum computing.

Where to get plasmodium? How to feed it? What substrates to use? Answers are given in Chap. 1.5.

Shortest path and plane tessellation are classical problems of computer science. In Chaps. 2.9 and 3.3 we show how to solve a maze and approximate a Voronoi diagram with plasmodium.

We aim not to clutter the book with theory. Thus, we provide just one simple yet illustrative model of Physarum behavior. We simulate propagating plasmodium fronts with a two-variable Oregonator. In Chap. 4.3 we use the Oregonator to uncover mechanics of spanning tree construction by Physarum.

When the plasmodium is placed on a substrate populated with sources of nutrients, it spans the sources with its protoplasmic network. The plasmodium optimizes the network to efficiently transport protoplasm with nutrients. How exactly does the protoplasmic network unfold during the plasmodium's foraging behavior? What types of proximity graphs are approximated by the network? How does the plasmodium increase the reliability and through capacity of the network? Does it construct a minimal spanning tree first and then add additional protoplasmic tubes? We answer these questions in Chap. 5.5.

The plasmodium propagates as a traveling localization — a compact wave fragment of protoplasm — on a non-nutrient substrate. The plasmodium localization travels in its originally pre-determined direction for a substantial period of time even when no gradient of chemo-attractants is present. In Chap. 6.7.4 we utilize this property of Physarum localizations to design a two-input two-output Boolean logic gate. A first gate produces a disjunction of inputs on one output and a conjunction of inputs on another output. A second gate produces an unchanged input variable on one of the outputs. On the other output the gate brings out a conjunction of one input with the negation of another input. We cascade the logical gates into a one-bit half-adder and simulate its functionality.

A Kolmogorov–Uspensky machine and a Turing machine are birth cohorts. The Turing machine was crowned by theoreticians. The Kolmogorov–Uspensky machine became a prototype of modern computer architectures. In Chap. 7.7, we show how to imitate a Kolmogorov–Uspensky machine in plasmodium. A Physarum machine is an experimental implementation, modification and extension of a Kolmogorov–Uspensky machine in plasmodium of *P. polycephalum*. Data are represented by sources of nutrients and memory structure by protoplasmic tubes connecting the sources. By experimentally implementing the Kolmogorov–Uspensky machine in Physarum, we prove that plasmodium of *P. polycephalum* is a general-purpose computer.

Programming and reconfiguration of Physarum machines, and signal routing in Physarum circuits, are developed in Chaps. 8.4–10.10. We show how to program Physarum machines with attracting fields generated by sources of nutrients (Chap. 8.4). We demonstrate how to route signals in Physarum machines with domains of light (Chap. 9.4) and diffusive fields of repellents (Chap. 10.10).

Being encapsulated in an elastic membrane, the plasmodium can be capable of not only computing over spatially distributed datasets but also physically manipulating elements of the datasets. If a sensible, controllable and, ideally, programmable movement of the plasmodium was achieved, we would get experimental implementations of amorphous robotic devices.

Chapters 11.6–13.9 lay the experimental background for future plasmodium-based robotic devices. There we provide experimental evidence that plasmodium can sensibly manipulate tiny lightweight objects floating on a water surface (Chap. 11.6), act as an on-board 'motor' by propelling floaters with its oscillating pseudopodia (Chap. 12.3) and transport, mix and transform substances (Chap. 13.9).

Chapter 14.5 pushes the limits of our imagination a bit further. We allow the plasmodium to become a road-planning engineer. The structure of the protoplasmic networks, developed by the plasmodium, allows the plasmodium to optimize transfer of nutrients between remote parts of its body, to distributively sense its environment and to make a decentralized decision about further routes of migration. We consider the 10 most populated urban areas in the United Kingdom and study what would be an optimal layout of transport links between these urban areas from the 'plasmodium's point of view'.

A pocket manifest on Physarum machines is given in the epilogue.

Reader beware

> "...the girl I described, she's as real as the wind
> It's true I saw her today
> The other details are inventions
> Because I prefer her that way."
>
> *Bret McKenzie and Jemaine Clement (2009)*[1]

Unconventional computing is an art of interpretation. Physarum does not compute. It obeys physical, chemical and biological laws. We translate its behavior to the language of computation. Reader beware. Our experiments are unbiased. Inferences might be slanted.

Andrew Adamatzky

[1]Bret McKenzie and Jemaine Clement. Rambling Through the Avenues of Time. Flight of the Conchords. October 2009.