Cell Learning Theory 2 : An Invitation

> "But as artificers do not work with perfect accuracy, it comes to pass that mechanics is so distinguished from geometry that what is perfectly accurate is called geometrical; what is less so, is called mechanical. However the errors are not in the art but the artificers."
>
> — Newton, Principia.

If we are interested in the fundamental nature of computation, we must reckon with the fact that Nature was computing for a billion years before Church, Gödel and Turing. But what kind of computation is this, and does understanding this "natural computation" lead us any deeper into the theory of computation?

In this seminar we explore some of the features that distinguish natural computation from other models of computation that are more familiar in computer science, e.g. Turing Machines. This is a very broad topic, so in order to avoid empty generalities we focus on Gene Regulatory Networks (GRNs) as our main example, but this by no means the only example of computation in living systems (the example of neurons performing computations as a collective in the brain is of course familiar, if you're able to read this).

The central mathematical object in theoretical computer science is the program. One way of distinguishing natural computation is to discuss how "natural programs", e.g. the algorithms embodied by GRNs, differ from "normal" programs. We focus on three differences:

1. Natural programs form a space
2. Natural programs are learned, not constructed.
3. Natural programs are singular

} Natural programs ≠ Turing Machines

Background on GRNs

We assume familiarity with the terms DNA, mRNA, RNA, protein, transcription (of DNA into mRNA by RNA polymerase), translation (of mRNA into protein by Ribosomes) and Transcription Factors (TFs) (proteins, themselves typically gene products, which regulate genes by promoting or suppressing their transcription into mRNA).

Def^n A Gene Regulatory Network is a collection of genes that interact with each other to control cell function, including development, differentiation and responding to environmental cues [N]. They

- regulate thousands of genes to be expressed in specific spatial and temporal patterns [DL].

- regulatory modules contained in the genome receive multiple inputs and "process them in ways that can be mathematically represented as combinations of logic functions (e.g. "and" functions, "switch" functions, "or" functions). At the system level, a gene regulatory network consists of assemblages of these information-processing units; thus it is essentially a network of analogue computational devices" [DL].

Example 1 "if lactose then enzyme" The bacterium E. Coli has three genes that code for enzymes that enable it to split and metabolise lactose. Under normal conditions (i.e. lactose is not present) transcription of these genes is repressed by a repressor protein (itself a gene product) which binds to a region of DNA near the three genes. Lactose can bind to this repressor and remove it, allowing transcription to proceed:

1. Natural Programs form a Space

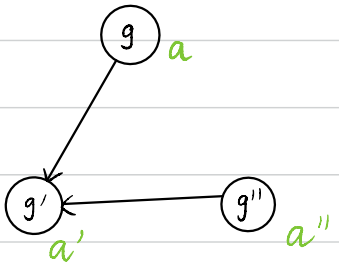
The interior of a cell is, in part, a statistical mechanical system in which large numbers of interacting molecules move thermally and interactions are governed by probabilities and concentrations as in chemistry. In particular gene transcription is stochastic (but as RNA and proteins can be actively affecting biological function at levels as low as a few copies per cell, one should use this framing carefully, [Ka]) although probabilities approach 0 or 1 at low or high concentrations of regulating molecules.

- "Stochastic gene expression is thought to be the consequence of the random activation and inactivation of transcription due to successive cycles of binding and release of transcription factors" [Ni, K].

From [BuLB], some early ideas on "programs in cells"

> However, it is important to point out that some of the ideas mentioned earlier are not entirely new, having their roots in the 1970s. In the seventies of the 20th century, researchers such as Liberman (1972, 1979) established an analogy between the cells of living beings and computers. In this analogy the computer is not a von Neumann machine or the popular idea of a computer, but a physical system in the tradition of Turing machines with which to study information processing, memory, and other features present in the cells of organisms. Liberman and
>
> Minina (1996) suggested that while a program is executed in a computer, in the brain, for example, the program code is written at the molecular level in DNA/RNA, with neurons playing the role of molecular computers. In this way living beings endowed with a brain are able to predict the changes and future state of the environment around them. According to this approach, the substitution of a base in DNA, transcription, etc. are interpreted as possible transformations of the molecular text in DNA. In Liberman's view, and within the previous theoretical framework, in living beings the cell would be under the control of a stochastic molecular machine programmed in the DNA. Such molecular machine is made up of enzymes, which would perform operations on the DNA itself as well as on the RNA and proteins. In

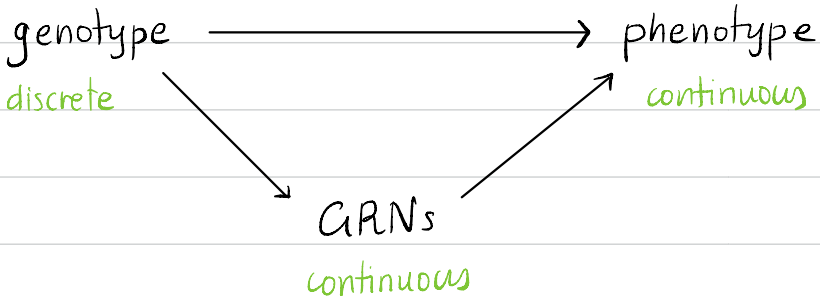
It is reasonable to think of the GRN as just such a "stochastic molecular machine". One possible abstraction is to think of a GRN as a graphical network where the nodes are genes and an edge $g_1 \longrightarrow g_2$ represents the regulation of g_2 by the product of g_1 (e.g. a TF). Since many genes are regulated by multiple upstream TFs, the logical structure of this network may be complex, and we can view concentrations of gene products as a function assigning to each gene a real number.



$$\underline{a}(t) = \begin{pmatrix} a(t) \\ a'(t) \\ a''(t) \end{pmatrix}$$ "state"

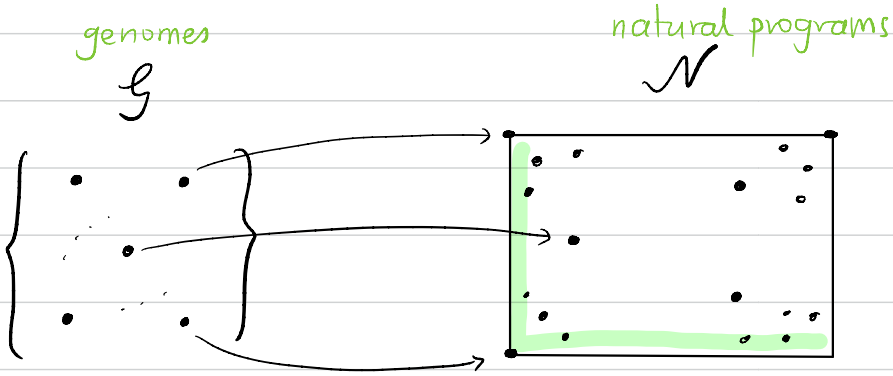
The time evolution of these gene expression vectors is sometimes described in Bayesian terms, and sometimes by differential equations, but the point relevant here is that the map from genotype to phenotype goes via a continuously parametrised set of possible GRNs. One way to think of the expression vector $\underline{a}(t)$ is as the tape of a (naive probabilistic) Universal Turing Machine, some part of which is the "working tape" specifying working memory for the program, and some of which is the "code tape" which determines the program being run.

Note Every cell in the human body has the same genome, so cell-types are distinguished (during development) by the level of expression of genes: different cell-types run different sets of "programs". But there is continuous variation even within cell-types [Ad].



We define <u>natural programs</u> to be Liberman's stochastic molecular machines, e.g. GRNs, which are continuously parametrised and thus form <u>spaces</u> (of some kind).

This however raises a natural question: how can DNA, which in an individual cell is <u>discrete</u> (at the population level there is a distribution but that is not the point here) code for a natural program which is <u>continuous</u>? Ask a biologist for the correct answer, but here's a guess: one can view the genome as encoding a discrete set of <u>points</u> in the space $\mathcal{N}$ of natural programs (see [Ad], which seems consistent with this idea)



<u>Remark</u> Natural programs exist in many forms besides GRNs, we use this as an illustrative example of significance in biology.

2. Natural Programs are Learned not Constructed

In Nature computation and learning are both ubiquitous and closely related: learning (say to respond to a stimulus) tends to presume some form of computation, and whether by evolution or a process within a single organism's lifespan, the "programs" executed by cells (or larger systems) are the product of gradual variation and optimisation, not reason

It is worth distinguishing several different forms of learning in biology:

- Evolution (i.e. learning at the level of distributions over genomes)
- Brains/nervous systems (e.g. "standard" learning by synaptic plasticity)
- Other tissues/organs (e.g. the immune system)
- Cells (somewhat controversial)
- ?

We have clarified in the previous section what we mean by one example of "natural programs", namely GRNs, and this leads us now to examine the role of learning in this context. Clearly evolution has shaped GRNs, but by acting on the genomes $\mathcal{G}$ rather than directly on $\mathcal{N}$. Does learning happen within cells, within their lifetime? This is more less controversial depending on how one precisely defines learning. In one standard definition

Def^n 1 [Ba] Learning is structured updating of system properties based on processing of new information.

By this definition E. Coli responding to the presence of lactose (new information) by promoting the transcription of a set of enzymes via removing a repressor (structured updating) is unambiguously a form of learning. But this is more like "response" and perhaps not what we mean by learning.

Def^ 2 [G] Learning refers to any persistent and adaptive modification of an organism's behaviour as a function of its experience.

This is better and excludes the lactose example because the change in E.Coli's state is not permanent. We refer the reader to SLT Seminar 1 for our own definition. For the moment we will simply grant that learning in the sense of [G] takes place in single cells and direct the reader to loc.cit. for a full discussion and examples.

Conclusion

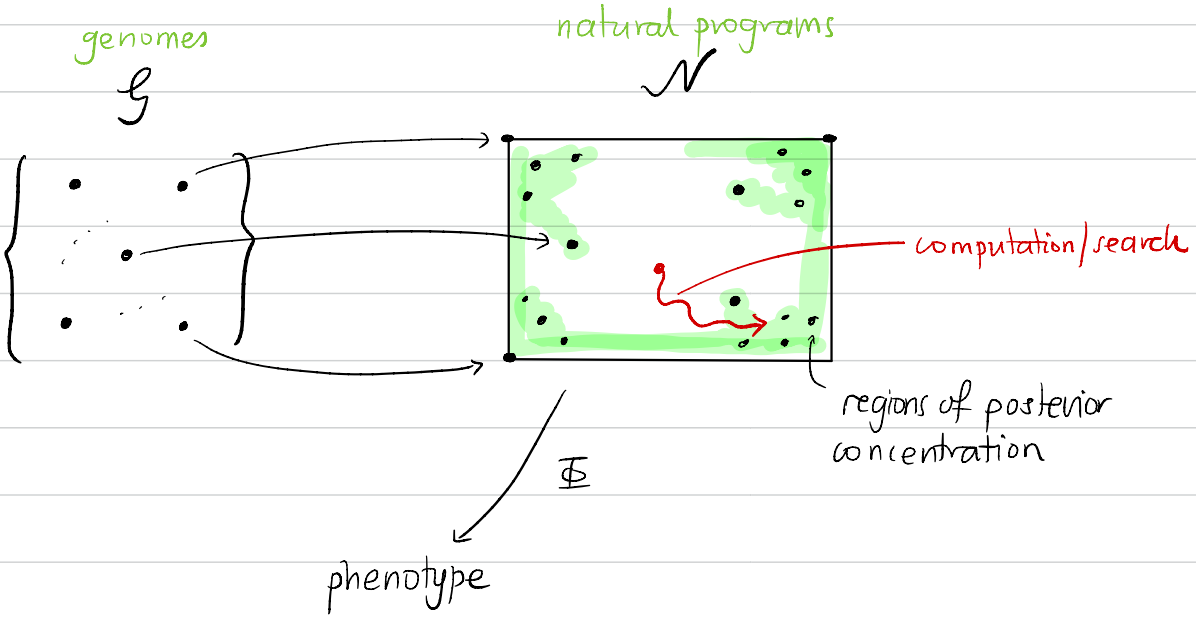
Single cells continue to surprise us. Robert Hooke, peering through his microscope in the 17th century, first likened cells to the small rooms (cellula) inhabited by monks. Fast forward to the 21st century, and it is now banal for cell biologists to think of the cell as a miniature computer, capable of sophisticated information processing (**Bray, 2009**). Among their many capabilities, it is now appreciated that cells have memory, possibly in the form of a 'histone code' (**Jenuwein and Allis, 2001**; **Turner, 2002**), though a precise computational understanding of this code has remained elusive. Whatever the memory code may be, its implications for neuroscience are far-reaching: we may finally be poised to link cellular memory codes with cognitive information processing. In this context, the studies by Gelber and others of learning in *Paramecia* become freighted with significance. They

From a mathematical point of view the form of learning in cells is probably more similar to Reinforcement Learning (RL) than regression, and this is one reason to view statistical learning theory with a changing true distribution as important. Ignoring the details for now, let us suppose that $w \in \mathcal{N}$ determines the phenotype, and that given some experience/measurements/samples D_n from the environment (e.g. pairs (x, b) consisting of physical locations and bacteria concentrations in Gelber's experiments with Paramecia [G]) of size n, it makes sense to consider the Bayesian posterior

$$p(w \mid D_n) = \frac{p(D_n \mid w)\,\varphi(w)}{p(D_n)}$$

Then we expect the natural programs executed in cells to be affected by <u>two forms of learning</u> (at least) namely evolution and "live" learning, and for this influence to be mediated

in both cases by something like the Bayesian posterior on $\mathcal{N}$ (with evolution acting on the prior $\mathcal{S}$ and model class and experience acting via D_n).



Here is the point: taking experience (D_n) and updating/modifying system properties/behaviour ($\omega \in \mathcal{N}$) requires <u>computation</u>, which can be realised for example as the cost of accurately sampling the Bayesian posterior. In this situation <u>learning</u> is the process of "burning in" and then sampling from some such sampling method. These are terms that we use to describe optimisation or sampling in computers, but it seems reasonable to conjecture that similar processes have evolved naturally.

This leads us to a fundamental point: since $\mathcal{N}$ is a <u>continuous space</u> and the map Φ from $\mathcal{N}$ to the phenotype is <u>many-to-one</u> (see the next section), the Bayesian posterior on $\mathcal{N}$ is <u>singular</u> (in the sense that the Fisher information metric is degenerate) and this means that the <u>learning process in cells</u> (if it exists, and consists of learning in continuously parametrised systems like GRNs) <u>is strongly affected by these singularities</u> (as we know from Singular Learning Theory).

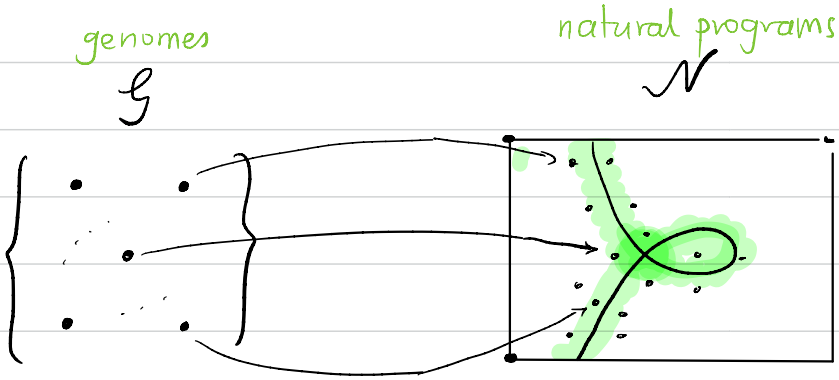
3. Natural Programs are Singular

There are a "surprising" number of degeneracies in the maps from genome to GRN and from GRN to phenotype. To name just two

- Codon degeneracy: multiple three base-pair codons specify the same amino acid (e.g. GAA, GAG code for glutamic acid) and often "nearby" codons have similar function (e.g. NUN tends to code for hydrophobic amino acids, where N = * = any nucleotide) as a kind of "fault-tolerance". This is a discrete degeneracy in the map $\mathcal{G} \longrightarrow \mathcal{N}$ from genome to GRNs.

- Multiplicative interactions: many genes are regulated by multiple transcription factors, so that it is possible for the same level of gene expression to be maintained while decreasing the concentration of one TF and increasing (in some proportion) the concentration of the other. In the algebraic expression for the expression this appears as a multiplicative interaction xy. Complex networks of regulation will display many such degeneracies, for reasons well-understood in [Wa, W].

Since natural programs are learned, not constructed, and the learning process cannot distinguish points $w, w' \in \mathcal{N}$ that determine the same behaviour, natural programs should not properly be understood as points $w \in \mathcal{N}$ but rather regions or phases $W \subseteq \mathcal{N}$ of the Bayesian posterior governing the learning problem. It's tempting to speculate on a relation between such phases and "specialisations" in cell types [Ad].



References

[A] H. Akhlaghpour "A theory of natural universal computation through RNA" arXiv: 2008.08814, Journal of Theoretical Biology 2022.

[PM] S.D. Pope and R. Medzhitov "Emerging principles of gene expression programs and their regulation" Molecular Cell 71, 2018.

[DL] E. Davidson, M. Levine, "Gene Regulatory Networks", PNAS Vol. 102 2005.

[N] https://www.nature.com/subjects/gene-regulatory-networks

[BGLB] A.G. Becerra, M. Gutiérrez, R. L-Beltra, "Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network" BioSystems 2022.

[K] M.S. Ko "Induction mechanism of a single gene molecule: stochastic or deterministic?" Bioessays, 1992.

[Ni] H.F. Nijhout "Stochastic Gene Expression: Dominance, Thresholds and Boundaries"

[Ad] M. Adler, Y.K. Kohanim, A. Tendler, A. Mayo and U. Alon "Continuum of gene-expression profiles provides spatial division of labor within a differentiated cell type" Cell Systems 2019.

[ID] S. Istrail, E.H. Davidson "Logic functions of the genomic cis-regulatory code" PNAS 2005.

[Ko] D. Kotliar et al "Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq" eLife 2019.

[Ka] Y. Kaznessis "Models for synthetic biology" BMC Systems Biology 2007.

[G] S. Gershman, P. Balbi, C. Gallistel, J. Gunawardena "Reconsidering the evidence for learning in single cells" eLife 2021.

[Ba] A. Barron et al "Embracing multiple definitions of learning" Cell 2015.

[W] T. Waring "Geometric perspectives on program synthesis and semantics" MSc thesis 2021.

[Wa] S. Watanabe "Algebraic geometry and statistical learning theory" 2009.