



HAL
open science

DNA, differential methods and the Böhm-out technique (written in 2008)

Giuseppe Longo, Pierre-Emmanuel Tendero

► **To cite this version:**

Giuseppe Longo, Pierre-Emmanuel Tendero. DNA, differential methods and the Böhm-out technique (written in 2008). 2021. hal-03319640

HAL Id: hal-03319640

<https://ens.hal.science/hal-03319640>

Preprint submitted on 16 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DNA, DIFFERENTIAL METHODS AND THE BÖHM-OUT TECHNIQUE

GIUSEPPE LONGO AND PIERRE-EMMANUEL TENDERO

Laboratoire d'Informatique, CNRS - École Normale Supérieure, Paris

e-mail address: giuseppe.longo@ens.fr or giuseppe.longo@ens.psl.eu ; h

<http://www.di.ens.fr/users/longo/>

CREA, Polytechnique

ABSTRACT. The “DNA is a program” metaphor is still widely used in Molecular Biology. There are good historical reasons for the use of such a metaphor or theoretical model. Yet we argue that both the metaphor and the model are essentially inadequate *also from the point of view of Physics and Computer Science*. Relevant work has already been done, in Biology, criticizing the programming paradigm. We will refer to empirical evidence and theoretical writings in Biology, although our arguments will be mostly based on a comparison with the use of differential methods (in Molecular Biology: a mutation or alike is observed or induced and its phenotypic consequences are observed) as applied in Computer Science and in Physics, where this fundamental tool for empirical investigation originated and acquired a well-justified status. In particular, as we will argue, the programming paradigm is not theoretically sound as a *causal* (as in Physics) or *deductive* (as in Programming) *framework* for relating the genome to the phenotype, in contrast to the physicalist and computational grounds that this paradigm claims to propose.

1. INTRODUCTION: MODERN LOGIC AND THEORY OF COMPUTING

Let's try to summarize a long history in a few lines. Computer Science, in particular the modern and rigorous notion of “program” for our arithmetic machines, emerged directly, in the '30s, from Mathematical Logic. The project of defining (and proving the limitations of) *mechanic arithmetic procedures* for mathematical proofs guided the work of Gödel, Church, Turing and others.

In Turing's Machine lies the logical core of the notion of program: *sequence-checking* and *sequence-replacement*. That is, of checking whether two sequences of numbers (or of 0s and 1s) are identical; according to their identity or not, copy or change one or more digits in the sequence (it really looks like a - parody - of a genome). But the digital environment must be *exact* (and *absolute*, at least in the sequential machine, we will go back to this). And the Machine is *Cartesian*, as Turing introduces a novel *dualism*: the program (the software, the “soul”) is totally independent from the hardware, the (physical) “body”. Moreover, and this is crucial, perfect *iteration* is at the core of computing: primitive

2000 ACM Subject Classification: Biology and Mathematics, 92-01.

Key words and phrases: Molecular Biology, the differential method in Physics and in Logic.

Dedicated to Henk Barendregt in his 60th birthday. A preliminary French version will appear in “Evolution des concepts fondateurs de la biologie du XXIe siècle”, DeBoeck, Paris, 2008.

recursion, the mathematical essence of it, is iteration plus updating a register. Moreover, by the software/hardware distinction and by identical iterability, one has the *portability* of software, another form of iteration: without it, Computing as a science (and Microsoft as a business) would not exist. In particular, you may take the soul of a computer and transfer an identical copy of it to the body of another, with a identical or just compatible operating system, and iterate it at will.

Finally, and most importantly, Turing's Discrete State Machine is *Laplacian*. Turing observes this twice [1], [2] and contraposes its *predictable determinism* to the unpredictability of what he calls deterministic "continuous systems", subject to the "exponential drift" as are the morphogenetic systems he models [3]. We now call these deterministic systems "sensitive dynamics to border conditions"¹. It is the exact nature of the databases that allows iteration and predictability² (iteration per se, as a time translation symmetry, is a form of predictability, see sect. 4.1). Of course, computer programs may also be *practically* unpredictable, as they may be *very* long and complicated, he observes; but unpredictability in non-linear systems is a key *theoretical* property (and a richness of these systems), not a practical issue.

Now, the use of notions coming from such a robust theory, such as Arithmetic Computability, either as technical model or as metaphor, is far from neutral: we will develop here the consequences, from a physicalist and computational viewpoint. Notice that the use of a metaphor is no less committing than a technical reference: a metaphor carries an entire conceptual world, a structure of thought, a guide for conjectures and experiments.

2. NETWORKS OF CONCURRENT AND DISTRIBUTED PROCESSES

Today's Computer Science is witnessing a major change in computer hardware which forces, beyond expectations, a change in programming paradigms. The Church Thesis (the claim that all logical-computational systems compute the same class of functions) is becoming inadequate or false if one considers distributed and concurrent computers. That is, different formal descriptions of concurrent computing may yield different computational powers, as far as processes are concerned ([4]). Thus, the myth of the absolute notion of computation is fading away, while the enrichment of this very notion may broaden its applicability (to life phenomena, for example).

The crucial issue with Concurrency resides in the synchronization of possibly asynchronous processes that are distributed in space and may "concur" while computing: they may share databases, each other's ongoing processes.... In short, through networks which are possibly distributed over the Earth's surface and over physical space-time, which we better describe using continuous mathematics, concurrent computation has stepped in, thus enriching the purely logical, stepwise sequential computing. Thus, Discrete State Machines and their programming are changing in nature. Is this enough to consider these novel paradigms for computing to be broad enough as to include the genome's and the proteome's dynamics?

¹Laplacian means that determination *implies* predictability (Laplace's conjecture). Deterministic chaotic (non-linear) systems are unpredictable, by the sensitivity and "mixing" - intervals are shuffled - properties: fluctuations below observability may cause major and unpredictable changes in the process.

²Also Shroedinger, who uses the word "program" for chromosomes, is aware of its Laplacian implication: given a complete knowledge of the code, "the all-penetrating mind, once conceived by Laplace" would achieve complete prediction ([34]).

3. DNA

Our first argument will focus on the fact that, despite the reiterated historical assertions of genetics, the existence of a direct causal relationship between genes and characters cannot be solidly established on the sole basis of experimental data, inasmuch as they present themselves *a priori* as results of *differential* experiments. After a reflection on the *principles* that justify the differential method in Physics, we will argue that programming theory and methods are unsuitable *in principle* for providing a conceptual framework capable of unifying under the notion of gene the structural and functional properties of DNA. We will therefore add arguments from Physics and Computing to the many writings in Biology criticizing the programming paradigm (see ([5], [6]) and others, quoted below).

3.1. Differential Methods in Genetics. The dominant empirical methodology in molecular genetics consists in the introduction or observation of genotypical modifications and the observation of eventual phenotypical variations ([7]). The *loci* identified by means of Morganian hybridization protocols were related to characters on the basis of a *causal* interpretation, *scarcely argued on the physiological level*³. The “causes” were given as a set of *empirical correlations* established between, on the one hand, the observable distribution of recombinations of pairs of characters amongst the lineage of a given parental generation and, on the other hand, the frequency of *crossing-overs* at the level of the individual’s germinal cells. Since the 60s, with the structural definition of genes in relationship to protein synthesis, the question of the relationship between genotype and phenotype is no longer posed at the level of observable characters as such, but at the level of the cellular metabolism, thus indicating for molecular biology a very clear regression, in terms of “domain of explication” in comparison to classical genetics. Strictly speaking, even the level of the organism’s metabolism is not attained. At most, only that of the *cellular* metabolism is attained, or, strictly speaking, that of the potential enzymatic role of proteins in biochemical cycles⁴.

However, it is well known that modifications, for instance, in the disposition of homeotic genes induce *teratogenic* effects on the development of drosophila embryos. But there lies, precisely, something of a problem to our logician and computer scientist eyes: this teratological, or simply *differential* aspect, specific to the methodology of genetics, does not seem to lend itself to an analysis in *causal terms* as is done in Physics, nor to an analysis in *deductive* terms, like in programming.

3.2. Differential Methods in Physics. In what way is it possible to interpret empirical correlations of this type by means of causal vocabulary? How does such theorization relate to the usual practices of Physics, for which differential methods constitute a common experimental protocol? Or to those of Computer Science, where the notion of “program” has its own theoretical framework and practical applications? What makes the discourse of genetics “causally incomplete” is simply the fact that it is not in a position to establish the formal existence of a relationship of *direct causation* between genotype and phenotype on the sole basis of currently available teratogenic experiments. The reference to a pre-theoretical

³And it could not have been otherwise given the time’s means of molecular investigation.

⁴These cycles, moreover, are not limited to simple chemical reactions between proteins but involve, sometimes crucially, various chemical species, molecular or not, of which it is rather difficult to assert that their physicochemical properties are dependent of genes (think of the case of ions).

(even pre-scientific) belief, such as the evocation of the metaphor of the computer program, cannot per se produce a systematic *theoretical determination*, as we argue next.

In Physics, the “differential method” consists in the experimental practice of deriving certain causal relationships by setting all of a system’s parameters except, in general, for one that is modified within a certain restricted domain of values. This can lead to set a relationship (generally, an *equation*) between parameters and observables and, thus, *propose causal relationships*.

Let’s consider, for instance, the case of the equation of the perfect gas in a state of equilibrium, $pV = KT$. This equation expresses causal links between pressure, volume and temperature and it may be derived by varying p or V and observing the changes in T , say. However, an infinite number of polynomials may interpolate any finite set of data: how can one then justify this specific equational dependency? First, a mathematical criterion steps in: write the simplest polynomial (is DNA the shortest program by any chance? most molecular biologists claim that, in eukaryotes, it contains from 90 to 95 per cent of “junk”...). Second, in this case (but it is always the case) 50 years after its historical formulation, this equation (law) was *mathematically deduced* in an *a priori* fashion within a more general theoretical framework: Statistical Physics. This theory proposed a framework of systematic *theoretical determination*. In short, *geodesic principles* (or of “least action”), which govern the trajectories of the particles of a gas, were transferred to the thermodynamic limit (an integral) and enabled to deduce the laws of Thermodynamics. This fully justifies the empirical differential analysis (in an *a priori* way from a deductive point of view, even if it is generally subsequent from an historical point of view). Moreover, the formal derivation produces *only one* solution out of many possible ones; thus, the recourse to experiments limits itself to the determination of the constant K or to the eventual falsification of the *whole* theory, without permitting the trick of only changing the equation. Within well constructed theoretical frameworks, counterexamples destroy theories.

The differential method, as it is applied in Physics, supposes no correlations, or at best, only weak ones (then finely analyzed) between the contextual parameters and the parameter being modified in an experiment or observation. This assumption would not seem legitimate for the analysis of extragenomic or epigenetic contexts of ontogenesis, because the aspect of biological systems, which constitutes the greatest challenge for current physicalist approaches, consists in the existence of this particular “causal field”, specific to the living cell, where (almost) everything is correlated to (almost) everything else. In this “coherence structure”, the effects of “resonance” seem infinitely (and we use this word in a mathematical sense) more complex than in any physical dynamics ([8]), even non-linear⁵.

A further critique of the differential method, not as an empirical practice, beautifully mastered in Molecular Biology, but as a direct ground for theorizing, is suggested by the

⁵This does not exclude, of course, that there exists very local cases (such as those, for instance, of certain rare genetic diseases) for which the causal relationship between genotype and phenotype may be considered as direct (and even Laplacian). However, a mutation again, a difference thus, causes a pathology (a difference from normal). Moreover, it is not a given that these particular cases constitute the general model on the basis of which all possible cases can be analyzed. And even if these specific cases were not rare, as they are, it could actually be the exact opposite of a paradigm: in Physics (...Physics, once more... but don’t molecular biologists want to be physicalist ?), we can recall that the Aristotelian principle, grounded on large empirical evidence, according to which all mobiles immobilize from the moment one ceases to apply motion to them, is precisely opposite to the Galilean inertia principle, which is alone to be theoretically relevant.

case of phenocopy, known since ([9]). Development biologists are able to create “phenotypical clones” by simply modifying, in various manners, the experimental conditions of the development (see ([10]) or ([11]) for recent references). So, from a differential standpoint, the pressure, the chemical composition of the environment or the local intensity of the electromagnetic field, as parameters of an extragenic context, *are* as much causes of the development of the embryo as is the genome, if we remain at a *purely empirical* level. One can, for instance, induce teratogenic effects similar to those induced by the displacement of a homeotic gene, by modifying the pressure at a certain moment during embryogenesis. We do not see any reason *in principle* why to exclude at once the possibility that certain modifications in the disposition of homeotic genes would in fact *cause* a change in the embryo’s reactivity to pressure.

3.3. The status of the concept of gene in the metaphor of genetic program. The notion of genetic program poses further theoretical problems of interpretation inasmuch as, in practice, it seems to be employed only very locally, so that *it is unable to make intelligible the physiological role of genes as a whole*, that is, in no way does it play a role even vaguely comparable to the geodetic principles in Physics.

We first recall that originally, the notion of genetic program was developed within the perspective of the works of Monod, Lwoff and Jacob concerning the regulation of the expression of the lactose operon in *Escherichia Coli* ([12]). In the same period, Ernst Mayr independently introduced a similar idea, in reference to the issue of causality in biology ([13]). Yet early hints in this direction were given in ([14], [15]), for instance, concerning the role of genes in the process of embryonic development; but they believed that the role of genes was not *preponderant*, as closely dependent on cytoplasmic regulations.

These regulation mechanisms, isolated in the case of the bacterial genome, constitute in fact a legion in the case of eukaryotes. Moreover, the metabolic processes within cells are subject to controls of the same type over the whole course of the individual’s lifetime.

3.3.1. Structural genes vs. regulation genes. The initial model of the lactose operon introduces an ambivalent distinction between the notion of *structural genes*, whose sole function would be to enable the synthesis of the molecules necessary to the cellular metabolism, and that of *regulation genes*, which can be of different types according to their mode of intervention upon the regulation of the expression of structural genes. Now it is precisely from this point of view that genes rather difficultly lend themselves to an interpretation in terms of genetic information and programming. First, the portions of DNA to which they correspond are sometimes not associated to the synthesis of any particular protein. Fundamentally, they serve only to favour those bonds of molecular complexes which induce epigenetic modifications of the genome’s structure, which enables the opening or closing of certain adjacent frameworks of interpretation. Second, as for regulation genes and the enzymatic activity of proteins, there visibly exist *relationships between these two entities which in no way involve mediation by means of a code*. This problem is not exclusively a theoretical issue relating to the sole definition of genes in molecular terms since it also leads to formulate very specific questions on a practical level. As a matter of fact, one needs to number the portions of DNA susceptible of behaving like genes in the genomes of “decrypted” species ([16], [17], [18]). In addition, part of the regulation mechanisms of the expression of genomes do not rest, strictly speaking, upon relationships which are analyzable in terms of code in the deterministic sense of information theory, but rather upon

dynamic physicochemical interactions, such as those studied by *statistical* chemistry. This leaves open the possibility for alternate interpretations of the physiological role of the gene *stemming from the same corpus of experimental data*.

One can thus oppose the Laplacian determinism which the notion of program *imposes a priori* to the relationship between genes and proteins (be it a relationship of translation or of regulation), and this because *this type* of determinism is *fundamentally incompatible with the probabilistic character of the predictions of statistical chemistry*⁶, ([19], [20]). And we see no physically admissible reason according to which one could suppose that the molecules that intervene in the reactions specific to vital phenomena would be excluded from this type of restriction for chemical analysis. It seems moreover that the great majority of molecular factors isolated to this day are never specific, but rather *ubiquitous and totipotent*. To account for this phenomenon, many still non-isolated co-factors are summoned to explain the *noted absence of specificity with regard to the studied proteins*. In fact, the concept of stereospecificity is highly problematic for the interpretation of physicochemical phenomena specific to living organisms. It signifies that the molecules which intervene in the regulation of genetic expression are endowed with *specific bonding properties* on particular substrates, *excluding all others*. Now, the great plasticity of macromolecules is well known to chemical kinetics since it is not uncommon for them to have several enthalpic isomers between which oscillations continually take place, under the effect of thermal agitation *as long as no constraint is applied to them*.

There is therefore not, *a priori*, any fundamental reason justifying the fact that the apparent non-specificity of proteins in living organisms would be an object of surprise and give way to interminable empirical and/or theoretical enquiries given that the hypothesis of stereospecificity is not a *sine qua non* condition of the validity of the analyses of statistical chemistry. Stochastic behaviours within structural stability may also explain genetic processes, even if they are largely incompatible with the notion of genetic program (unless one is talking of a very different notion of “program” , a “randomized” one ?).

4. THE MACHINE AND DNA.

But then, is it still possible to believe that DNA is a program in the Computer Science sense of the term? Many claim that it is “just” *a metaphor*, not a “mathematical model” , to be used as a suggestion for intelligibility. But when we project, upon natural phenomena, a human construction (historically, the alphabet and clocks, digital computers more recently), it is necessary to think of the constitutive principles of these constructions, rich in human practices and internal logic, which impact their meaning and their possible role for intelligibility. As we said, metaphors are far from neutral: they project an understanding and even a Philosophy of Nature, sometimes even more so than a technical model. Typically, in this case, a Laplacian culture of determination, from genes to phenotype, is being largely broadcasted.

⁶These predictions are, indeed, probabilistic inasmuch as they concern the *global* behaviour of populations of molecules and not the individual behaviour of each of these molecules, which remains *submitted to the perturbing influence of thermal agitation*.

4.1. The rules and their context. In the first two sections, we have highlighted that software portability is essential to applied Computing and is a consequence of its perfect theoretical (but also practical) *iterability*. This property is a *local* one, in a given context (starting exactly on specific, local digits, it enables the perfect iteration of a computation), in contrast to the *global* nature of the *structural stability* of living beings and processes. Locally, life is always different: it is in the local description of the phenotype that one best appreciates variability, as core component of the (relative) global stability of phylogenesis and ontogenesis.

The central issue with a genome-phenotype vs. program-semantics comparison, or with intelligibility by metaphors, consists in the analysis of the various contexts of expression of the discrete-state machine or of the genome. Because, prior to any expression, there is something very important that is common to these two structures of the intellect and of the world: the essentially discrete nature of the notational, conceptual, or chemical support. Discrete in the following mathematical sense: discrete topology *naturally* organizes these structures (in the informal but rigorous geometrical sense). But *it is the operational context which generates the meaning* (the computed function) *of a program or, as for a genome, the structures of living matter* (up to the phenotype).

In Programming Theory (and practice), it is the discrete data type and environment, including those of the *compiler*, that allows predictability and iteration. Even in concurrent networks, of which the (difficult) mathematics is being developed, prediction and iteration work. As we already observed, physical space and time pose difficult problems to the synchronization of asynchronous concurrent processes ([4]). Yet, we all know that if we open a distant Web page a thousand times, we will obtain, except in rare cases, always the same page, identically, independently of the spatiotemporal access. Programming, be it sequential or concurrent, is intended for that: the rule *must* dominate, without exception. And it must be iterable. *Finally, randomness does not exist in programming*: pseudo-random generators are small programs which generate sequences of 0s and 1s, *perfectly iterable under the same initial conditions*, but which provide good distributions of probability in the sequence spaces (they *seem* random). Just as it is for Laplace, chance is perfectly foreign to “determination by a rule”⁷. Now, since Poincaré, it has been understood that a deterministic (classical) process is random when, *iterated in the same border conditions, in terms of physical measure, it does not undergo the same evolution*. This does not exist in sequential programming theory. In concurrent systems, it is rare and it is due to the physical (or human) context and it is avoided at all costs (and successfully so).

What about this *programmed insensitivity to context*, specific to the digital machine, in the case of the genome⁸? We do find in the sequences of nucleotides elements having a discrete structure, these letters reminiscent of a four letter alphabet. But *the role held by the context of expression in the analysis of data and the evolution of processes changes radically*. First, in ontogenesis, the non-linear network of interactions, of which the genome is part, is a complex blend of (global) stability and high sensitivity, as stressed by many(21, 22). Second,

⁷This Laplacian split between normative law and randomness may be found identically in Monod and Jacobs approach to necessity and randomness.

⁸Breeders claim that with mammals, 40% of fertilizations fail: what a flawed program the genetic one is! But these failures, in general, are not “errors”, “exceptions”... they are part of the phylogenetic/ontogenetic game of variability and selection, of which the theory should take care. In contrast to this, the notion of “exception” in programming refers to deterministic (and predictable) bifurcations, devoid of physical *criticality* (with its associated relevant role of *fluctuations*). The notion is therefore still of a Laplacian type laws with foreseen exceptions.

even from a strictly physicalist, but modern, viewpoint, it would be necessary to abandon the classical conception of the laws of Nature: a *normative* conception, derived, since Aristotle up to Newton, from social and religious practices (civil and penal law, divine law) which pervade Logic and Computing. As a matter of fact, *this is not this way that Physics analyses phenomena today*. Relativity and quantum mechanics proposed new conceptual frameworks. The trajectory of a planet *is not* a course obeying Newton's laws/equations, *but a geodesic within an adequate Riemannian manifold*. The evolution of a quantum system can be understood in Hamiltonian terms and in terms of symmetries, in suitable abstract spaces.

That the programming of a digital machine follows old paradigms is, on one hand, *intrinsic to its essentially artificial nature*. It is precisely the normativity which makes the strength of programming: the computer must follow the rule. We are God to the machine, we dictate the rules, even in networks. They are the result of an "intelligent design" .

4.2. Degeneracy. The discrete structure of DNA has a rather singular property for computability and even for Physics: *degeneracy*. This notion was introduced in ([23]) with regard to the functioning of the brain, but it has its origins in the works of Edelman concerning the immune system. Shortly, degeneracy applies if non-isomorphic structures can produce a same (or similar) functionality or, conversely, if a given structure can be found to underlie several (different) functionalities. This concept differs from *redundancy* as given in information theory (a theory of communication ([24])) and in Computer Science, where identical copies of a given structure may replace the failing original part. In biological degeneracy, a small fluctuation in the context, even one of physical origin, can then generate in time an expression, a function for instance, which may be either similar or quite different. This causal analysis of the variability of living matter is compatible, but considerably enriches analyses based on the notion of mutation. The so-called genetic redundancy, an apparent informational "surplus ([25])" , is better expressed as structural degeneracy as the cybernetic analogy is most lax and does not enable to introduce *all the rigor* (and rigidity) *of the concept of redundancy in information theory*, while, at the same time, it *misses the richness of the biological notion of degeneracy*⁹.

To conclude, genes can no longer be associated without ambiguity to a succession of nucleotides encoding a protein, let alone in a Laplacian-programming sense. And, even if this association was conceivable by a more modern notion of physical determination (non-linear continuous dynamics, for example) the succession of amino-acids within the given protein would not suffice to determine the function associated to it, *unless referring to the constraints which will be applied at the cytoplasmic level*. And, in fact, it does appear

⁹Many examples show that the concept of degeneracy for biology means much more than a simple informational "surplus" or "deficit". The "knock-out" experiments demonstrate that sometimes the suppression of entire sections of DNA can have absolutely no observable "consequence" for the development of individuals, at least within some reasonable limits. In contrast, there also exist cases where the modification of a sole nucleotide "conduces" to important phenotypical changes. And, doubtlessly, cases must exist where these two phenomena are observable *for a same portion of DNA*. It appears to us, in this respect, that a rigorous conceptual distinction, not only between redundancy and degeneracy, but also within the latter notion itself, between structural degeneracy and functional degeneracy, could provide a clearer grasp of several phenomena. For example, the remarkable presence in eukaryotic organisms of a quantifiable *functional deficit* (cf., for example, the successive and different assessments provided over the course of the sequencing of the human genome) jointly to an apparent *structural surplus* (the genes are fragmented, repeated, can substitute one another).

today that in order to produce a minimally coherent interpretation of the genome, it may be impossible to do without a reference to the epigenetic and extragenic contexts, with their immense biological complexity, and within which the genome is expressed ([26], [27], [28]). And then causality needs to be broadened to different *and* interacting levels of organisation, each with its own causal correlations, far away from Laplacian determination (and, actually, even badly understood correlations with regard to more modern physical theories ([29], [30])).

4.2.1. *The differential method in Programming.* In our critique of the differential method in molecular biology, it has been question of a theoretical incompleteness with regard to the inference of “causal laws” . Specifically, we have demonstrated the fact, important to us, from a purely logical point of view, that simple empirical correlations between certain modifications of DNA and the differences observable in the phenotypes of individuals do not immediately and/or evidently provide a “law” which causally correlates the various structures of DNA to their “consequences” deemed to be normal. It is necessary to emphasize here, once more, the specificity of the differential method in Physics, from the moment that it is question of formally establishing any causal dependency between variables. Indeed, beyond the analyses of the correlation/decorrelation of variables which we have already addressed, physicists generally need to endow themselves with a theoretical framework of interpretation, essential to the construction of scientific objectivity.

Firstly, the finite number of experiments and, further on, the discrete character of the data does not enable to propose a *unique* equation otherwise than by means of a certain “conceptual optimality” : in other words, we are looking for the equation (the polynomial, the curve) having the “simplest” mathematical form. It is obvious that the validity of this criterion *is not an absolute* but *its effectiveness* is at the center of the methodological strength of Physics, since Copernic, Kepler and Galileo (and more explicitly, Lagrange and Hamilton). Secondly, the differential method in Physics finds, in a large measure, its mathematical justification in the variational methods of differential calculus: small (infinitesimal) perturbations enable the reconstruction of geodesics. In both cases, we can see, the mathematical analysis presupposes a continuous framework smooth spatiotemporal (even conceptual) “surfaces” where these criteria of optimality have a strictly physico-mathematical meaning and coherence.

Now, this framework is precisely lacking in biology’s differential analyses, probably because of the huge difficulty of sorting out, due to their physical singularity, the “interlinked” causal relationships characterizing living matter, which appear to us to be actually stemming from Physics’ most dynamic theoretical frameworks. And in no way does the “model” (the metaphor) of program provide such a framework by way of its causally Laplacian computational roots. We have already recalled, in fact, that the latter find their origin in a theory of computability (sequential or, for the little that exists, of concurrency) of which the consequences for causality regimes have been analyzed elsewhere ([30]) and are very removed from the smoothness of the physico-mathematical theories which are implicitly referred to in biology, by proposing a differential analysis largely unfit to Computer Science, as argued further below. In short, the “flaw” in terms of explication which we have noted in genetics does not stem from differential analysis itself, a technique which has given us, throughout the XXth century, information and results of great interest, but it must, in our opinion, be attributed to the sole attempts to frame it within more or less naive notions of “genetic programming” or “information”.

As already mentioned, in Physics strong and explicit principles (symmetries, geodetics) justify the theoretical proposals of which the differential experiments are the counterpart. In spite of the presence of physical constraints in life phenomena, living processes do not seem to rely *only* on similar principles (symmetries and geodetics), which of course participate to it (think of phyllotaxis); so the differential methods can hardly rely *only* on similar theoretical background. The point now is that the Computer Science theory and practice cannot help at this regards: there is no way to analyse the behaviour of a program by forcing small changes and... see what happens. Let's be more precise.

First, a modification in the order of the instructions of a program does not necessarily modify the result of a calculus despite that there are strong reasons, namely within the paradigmatic framework of Turing machines, to speak of a relationship of causality with regard to the relationship between programs and calculi (see §. 2). And, following that, it is clear that in logic and Computer Science, the analysis of the *semantics* of deduction or of programs (the “meaning” of the theorem or the “function” that is computed) would not make do with a simple differential, or perturbation method, to establish a relationship of causality between deductions or programs and deduced meaning or calculated functions.

A programmer may be willing to experiment “local mutations” , by changing here and there one or more instructions or part of them, while looking for a variant of a known program. But in no way one would use such a practice to understand the “meaning” of (the computed function by) a given well-formed program: the rare, but possible practice above leads to no theory. The nowadays rich semantic theories of programs are based on completely different mathematics. The major streams are called Denotational Semantics ([31]) and it is largely based on categorical meaning of logical theories ([32]), or the Abstract Interpretation and a few more.

There nevertheless exists a type of analysis, in logic and programming theory, which may resemble a differential method: the “Böhm-out” technique ([33]). The theorem at the basis of this technique enables to demonstrate that a *difference* between two programs, as terms of Church's lambda-calculus (in their so-called Böhm-trees, at a finite level) induces a difference in their semantics, *given* very specific operational semantics or certain mathematical domains of interpretation (or, at least and more technically, it demonstrates that two terms in normal form, which are syntactically different even for a very small “mutation” - do not define or compute the same function, *in any already given semantic environment*, i.e. in any mathematical meaning of programs). But this result, which may be considered of the “differential” type, *does not* provide the semantics of a program (or of a lambda-term), its “phenotype” , if we may say so. It may instead help to characterizes syntactically the terms having *the same semantics*, within an already given operational semantics or mathematical interpretation ([33], [34]).

More so than in Physics, therefore, a coherent and sound, conceptual interpretative framework must *first* be given (with its metrics or topology, its perfectly clear mathematical interpretation ... a sort of framework for signification-determination comparable to that of Statistical Physics for thermodynamics, to recall our previous analysis), and *then* the analysis of semantic equality (or difference or inclusion) between programs is refined by means of a differential method. Once more, the analysis of variations contributes to make more explicit the generated meanings or even the processes/functions/“phenotypes” , once a direct and sound interpretation *has been given*; in short, that which signifies/calculates a program very generally. Thus, *the notion of genetic program does not enable to account for the relationship between genes and characters by means of a sole differential analysis,*

as surely no technical reference can be made to a similar semantic analysis in Programming Theory. This Theory first requires a determination of causal/deductive relationships in the *direct* sense (for “normal” genes or programs, before “mutations” or Böhm-out). And this is precisely what we miss in Genetics.

5. CONCLUSION: ON CAUSAL STRUCTURES AND FINALISM.

What still pushes so many biologists, of the highest scientific level, towards this myth of genetic program or information, completely determining the phenotype (“coding for structure”, whatever this may mean)? It is clear that the discrete structure of DNA, so well described by sequences of letters somewhat reminiscent of the alphabetic encoding of a (formal) *language*, suggests such a metaphor. However, that does not suffice to justify the reference to the notion of *program*, not of “*information for*”. The need, so strong in biology, to find a justification (or an allegedly reductionist explanation) to *finalism* (that “lover which we try to hide, but which we cannot do without”) does transpire here all the more clearly. By means of a play of language around the notion of program, one indeed surreptitiously slides towards a conception of living matter which is permeated with finalism: the notion of program, a secular notion, without needing God, contains *an end* within its lines of code that are so nicely embodied in our modern machines. The computer program is *made for*, it has an objective, but it is, or can be, materially encoded.

But let’s try to approach the issue of finality from a physicalist standpoint. Consider a boulder forming a slope endowed with some irregularities and upon which rain gradually forms a cavity, by the repeated action over time of friction and small turbulences. It may happen that this first cavity will, in turn, generate a new turbulence which will *cause* a second cavity to appear, a “child” of sorts of the first one. But then, would it really make sense to say that the first notch is a *program*, written by physical evolution, in order to generate a second one? Or further, that it contains *information* about the form of future notches with their apparent structural specificity? With this example, we clearly see that it is *always* possible, from a simply descriptive point of view, to “manufacture” an apparent finality by playing the rhetoric game of programming or by employing the notion of information, so laden with meaning within Aristotelian culture¹⁰. But does it make the phenomena thus described *physically* intelligible? Modern Physics has chosen a different approach to explain this kind of phenomena: the geodesic principles having bypassed finalism for over a century ([30]). The point, of course, is not to conform to physical paradigms, but to be at least physically compatible, mostly when claiming to be physicalist. And this, in order to develop a Genetic Theory, by taking into account the entangled causal relations proper to the physical singularity of life.

REFERENCES

- [1] Turing A. “Computing Machines and Intelligence”, **Mind**, LIX, 1950
- [2] Longo G. “Laplace, Turing and the “imitation game” impossible geometry: randomness, determinism and programs in Turing’s test”. In Epstein, R., Roberts, G., and Beber, G. (Eds.). **The Turing Test Sourcebook**. Kluwer , Dordrecht, 2007.
- [3] Turing A. M. “The Chemical Basis of Morphogenesis” **Philo. Trans. Royal Soc.**, B237, 37-72, 1952.

¹⁰The concerned wordplay *is* Aristotelian, in the precise sense that it refers to the idea that genes would “in-form” the proteins, or “put them into form”, as the Philosopher would say.

- [4] Aceto L., Longo G., Victor B. (eds.) The difference between Sequential and Concurrent Computations. Special issue, **Mathematical Structures in Computer Science**, Cambridge UP, n. 4-5, 2003.
- [5] Wilkins A. **Genetic Analysis of Animal Development**, John Wiley and Sons, 1986.
- [6] Fox-Keller E. **The Century of the Gene**, Harvard UP, 2000.
- [7] de Vries H., **Espèces et variétés, leur naissance par mutations**, trad. L. Blaringhem, Alcan, 1909.
- [8] Goodwin B., "What are the Causes of Morphogenesis ?", **BioEssays**, 3, 32-36, 1985.
- [9] Goldschmidt R., **Physiological Genetics**, McGraw-Hill, 1938.
- [10] Lewontine R., "The Confusion over Cloning", **New York Review of Book**, October 1997.
- [11] Stewart J. **La vie existe-t-elle?** Vuibert, Paris, 2004.
- [12] Jacob F., Monod J., "Gènes de structure et gènes de régulation dans la biosynthèse des protéines", **Comptes rendus de l'Académie des Sciences de Paris**, 349, 1282-1284, 1959.
- [13] Mayr E., "Cause and Effect in Biology", **Science**, 24, 1-14, 1959.
- [14] Morgan T. H., **The Theory of the Gene** (1926), Yale UP, 1928.
- [15] Goldschmidt R., **Physiological Genetics**, McGraw-Hill, 1938.
- [16] Bernot A., **Analyse de génomes, transcriptomes et protéines**, Dunod, Paris, 2001.
- [17] Boffelli D., Nobrega M. et Rubin E., Comparative genomics at the vertebrates extremes, **Nature Reviews Genetics**, 5, 456-465, 2004.
- [18] Bartel D., "MicroRNAs : Genomics, Biogenesis, Mechanism and Function", **Cell**, 116, 281-297, 2004.
- [19] Creager A., Gaudillière J.-P., "Meanings in Search of Experiments and Vice-versa : the invention of allosteric regulation in Paris and Berkeley", 1889-1968, **Historical Studies in the Physical and Biological Sciences**, Tome 27, 1-89, 1996.
- [20] Kupiec J.-J., Sonigo P., **Ni Dieu, ni gène**, Seuil, 2000.
- [21] Nijhout F., "Metaphors and the Role of Genes in Development", **BioEssays**, 12, 441-446, 1990.
- [22] Nijhout F., "The Nature of Robustness in Development", **BioEssays**, 24, 553-563, 2002.
- [23] Edelman G., Tononi G. **A Universe of Consciousness. How Matter Becomes Imagination**, Basic Books, 2000.
- [24] Shannon C. E. **A Mathematical Theory of Communications**, 1948.
- [25] Thomas J. H., "Thinking about Genetic Redundancy", **Trends in Genetics**, 9, 395-399, 1993.
- [26] Allis C., Jenuwein T., "Translating the Histone Code", **Science**, 293, 1074-1080, 2001.
- [27] Boguski M. et Hieter P., "Functional Genomics : it's all about how you read it", **Science**, Tome 278, 601-602, 1997.
- [28] Turner B., "Cellular Memory and the Histone Code", **Cell**, 111, 285-291, 2002.
- [29] Rosen R. **Life itself**, Columbia UP, 1991.
- [30] Bailly F., Longo G. **Mathématiques et sciences de la nature. La singularité physique du vivant**. Hermann, Paris, 2006
- [31] Amadio R., Curien P.-L. **Domains and lambda-calculi**, Birkhuaser, 1998.
- [32] Asperti A., Longo G. **Categories, Types and Structures**, M.I.T. Press, 1991.
- [33] Barendregt H. **The lambda-calculus: its syntax, its semantics**, North-Holland, rev. edit., 1984.
- [34] Barendregt H., Longo G. "Equality of lambda terms in the model Tw," in **Combinatory Logic, Lambda-calculus and Formalism**, (Seldin, Hindley eds.) Academic Press, London, 303-337, 1980.
- [35] Shroedinger E., **What is life?**, 1944.