

DNA Tandem Repeats as Iterable Objects to Count Cell Divisions: A Computational Model

Marco Franco Giulio Regolini 

MFR Bioengineering, Milano, Italy

Email: m.regolini@libero.it

How to cite this paper: Regolini, M.F.G. (2024) DNA Tandem Repeats as Iterable Objects to Count Cell Divisions: A Computational Model. *Advances in Bioscience and Biotechnology*, 15, 207-234. <https://doi.org/10.4236/abb.2024.154015>

Received: January 18, 2024

Accepted: April 7, 2024

Published: April 10, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Cell lineages of nematodes are completely known: the adult male of *Caenorhabditis elegans* contains 1031 somatic cells, the hermaphrodite 959, not one more, not one less; cell divisions are strictly deterministic (as in the great majority of invertebrates) but so far nothing is known about the mechanism used by cells to count precise numbers of divisions. In vertebrates, each species has its invariable deterministic numbers of somites, vertebrae, fingers, and teeth: counting the number of iterations is a widespread process in living beings; nonetheless, it remains an unanswered question and a great challenge in cell biology. This paper introduces a computational model to investigate the possible role of satellite DNA in counting cell divisions, showing how cells may operate under Boolean logic algebra. Satellite DNA, made up of repeated monomers and subject to high epigenetic methylation rates, is very similar to iterable sequences used in programming: just like in the “iteration protocol” of algorithms, the epigenetic machinery may run over linear tandem repeats (that hold cell-fate data), read and orderly mark one monomer per cell-cycle (cytosine methylation), keep track and transmit marks to descendant cells, sending information to cell-cycle regulators.

Keywords

Satellite DNA, Tandem-Repeats, Epigenetics

1. Introduction

Scott F. Gilbert, in his *Developmental Biology*, wrote: “*How do our cells know when to stop dividing? If each cell in our face were to undergo just one more cell division, we would be considered horribly malformed. If each cell in our arms underwent just one more round of cell division, we could tie our shoelaces without bending over. Our arms are generally the same size on both sides of the*

body. How is cell division so tightly regulated?” [1]

The lineage of the roundworm *Caenorhabditis elegans* is absolutely invariable, autonomous, and genetically established without exceptions [2]: its midgut progenitor (E blastomere), isolated and cultured *in vitro* in the absence of morphogens, undergoes five (incomplete) division rounds as in the whole embryo [3]; an embryonic intestine of 20 cells is generated, not 32 (2^5) because only 4 cells (unerringly the same) carry out the fifth division, while the remaining 12 stop dividing after four rounds. Similarly, in the embryo of the fruit fly *Drosophila melanogaster*, the early 13 divisions are precisely counted [4] without active morphogens, that will work later [5] or other molecular clocks (early divisions occur in a syncytium without cell-cell communication involving transmembrane signaling; maternal morphogenetic mRNAs have different concentration gradients along the developmental syncytium). Abouchar *et al.* [6] measured the precision of left–right and inter-individual fly wing vein patterns: “*wing vein patterns are specified with identical spatial precision and are reproducible to within a single-cell width. The early fly embryo operates at a similar degree of reproducibility, suggesting that the overall spatial precision of morphogenesis in Drosophila performs at the single-cell level. Could development be operating at the physical limit of what a biological system can achieve?*” Interestingly, the genome of *Caenorhabditis elegans* [7] contains large amounts of well-conserved Tandem Repeats (TRs) and 21% of the *Drosophila* genome is made of TRs [8]. “*How is cell division so tightly regulated?*” No tricks and no magic: an intrinsic and autonomous molecular mechanism capable of counting cell divisions at the physical limit of single-cell level must exist: precise, independent from external signalings (morphogens), and unaffected by environmental noise.

1.1. Satellite DNA and DNA Methylation

Satellite DNA (satDNA), so named because DNA centrifugation over a density gradient forms a “satellite” band, consists of huge arrays of TRs, *i.e.* sequences of non-coding DNA, each one made up of the repetition of its characteristic monomer, sometimes not perfectly identical [9] [10] [11]; many TR sequences are evolutionarily well conserved [12] indicating they play a role in executing biological tasks. TRs have a relevant part in morphogenesis in invertebrates and vertebrates: in butterflies, wing patterns show a basic plan controlled by non-coding DNA to realize the diversity of wings of different species: TRs involved in wing patterning [13] have been conserved over millions of years allowing wing patterns to evolve fast [14]; in dogs, different TRs, found in genes that control morphological variations, cause a rapid breed evolution [15]: anatomical differences appear strictly linked to numbers of cell divisions and TRs mutations [16]. SatDNA is a strong accelerator of evolution because of its surprising self-remodeling rate: monomer similarity [9] [10] [11] causes frequent DNA-polymerase slippages [17]; homologous recombinations [18] of tandem repeated monomers constitute a major impediment for the accurate DNA repair of nucleotide loss in

double-strand breaks: TRs polymorphism and rate of errors during DNA replication is 10^5 times higher than single-point-mutations [16]. During evolution, DNA-polymerase slippages, non-disjunctions, recombinations, unequal crossing-over, rolling-circle replications, inversions, and multiple transpositions have generated diverse satDNA families, highly polymorphic and variously assembled in clusters of different textures and patterns [19] [20].

SatDNA is actively methylated [21] and plays an important role in epigenetics [22] [23] [24]; epigenetic tools comprise specialized enzymes [25] able to recognize the particular 3D structure of their target sequences: “writers” [26] catalyze biochemical modifications (e.g. DNA methyltransferases: methylation on the 5th position of the pyrimidine ring of cytosine, $C \rightarrow 5mC$), “readers” [27] recognize and interpret the significance of such modifications and recruit a set of methyl CpG-binding proteins and “erasers” [28] remove methylations [29] [30] [31]. Hypomethylated TR arrays have been found in different human diseases, cancer, and psychiatric disorders [32].

1.2. Boolean Algebra

Boolean algebra differs from elementary algebra in that its only values are “true” and “false” not numbers as in elementary arithmetic and algebra. Boolean algebra does not execute arithmetic operations such as the addition of numbers but uses logical operators (e.g. conjunction): Boolean algebra takes advantage of logical operations, as elementary algebra uses numerical operations; in cell biology, Boolean operations are realized by the 3-dimensional (3D) structural shape-changing of a protein (e.g. a receptor) after binding with its ligand (binding may assume the only values of “true” or “false” and so the correlative 3D structural change). SatDNA is made up of sequences of similar monomers as pearl necklaces: it resembles arithmetic tools like abaci, where it is possible to count without using numbers. Following these fundamentals, iterations on linear substrates from a start point to a stop (*i.e.* counting a fixed number of repetitions) are common: DNA replication (strictly time-bound to DNA methylation: see further), DNA transcription, mRNA translation, or dyne in movements on microtubules provide some clear examples.

2. Materials and Methods

The program “Lineages” takes as *INPUT* the following *in silico* 3’ →5’ DNA sequence (named “*sat_DNA*”):

**ATTCCAACGGCTTAATTCCCACGGCTTAATTCCGACGGCTTAATTC
CCACGGCTTAATTCCGACGGCTTAATTCCCACGGCTTAATTCCCACGG
CTTAATTCCTACGGCTTA**

This sequence matches with many eukaryote genomes (fish, arthropods) sharing up to 78% similarities (Fasta: $2.9E-37$): in the worm *Alitta virens* it matches with five different loci of chromosome 4, six loci of chromosome 7, and eleven loci of chromosome 9. The “*sat_DNA*” sequence is made up of 8 quasi-palindromic

monomers, each one of 14 nucleotides, almost (but not completely) identical, with palindromic terminal consensus motives (3'ATTC...CTTA 5'), necessary as recognition sites for epigenetic "readers" [33].

The program returns as *OUTPUT* cell lineages, shown step by step, division after division, obtainable through simulated possible evolutionary events that occurred on "*sat_DNA*" (see below).

Symbols in the program "*Lineages*":

"M" = 5-methyl-cytosine (5mC);

"H" = 5-hydroxymethyl-cytosine (5HmC)

Python is a high-level language, very popular among software developers: its "*for...in*" and "*while*" loops perform repetitive tasks, traverse and scan *iterable objects* like *lists* and *strings*, one item at a time, simulating processive cellular functions executed step by step on linear templates (replication, transcription, translation); also "*if/elif/else*", "*break*", "*continue*", "*and*", "*or*", "*not*", "*pop()*" simulate biological functions: they reproduce (*in silico*) biological (*in vivo*) pathways that may result "*True*" or "*False*" (e.g. ligand/receptor matching, concentration level reaching); on the contrary, "*classes, matrices, multidimensional arrays, recursive functions*", indispensable in *programming*, are too different from biological mechanisms: for these reasons, only the aforementioned "bio-compatible" statements have been used in the program "*Lineages*": the code is not "elegant", but reproduces correctly biological operative modes.

3. The Program "*Lineages*": Purpose

The program "*Lineages*" shows that three common biochemical pathways: *i) step-by-step progression*, *ii) cytosine methylation*, *iii) epigenetic marks* can work on TRs of satDNA to count the number of cell divisions. During the process of replication, the epigenetic machine copies (*i.e.* replicates on the still unmethylated new strand) cytosine methylations, and, at the same time, looks for marks (Cor 5mC as the 4th nucleotide of each monomer, C or T in the 5th position of each monomer: read further). Changes introduced into the "*satDNA*" sequence by epigenetic methylations have been computed to simulate how epigenetic "writers" and "readers" could manage the count of cell divisions.

The standard monomer (3'→5') of "*sat_DNA*"

A	T	T	C	C	A	A	C	G	G	C	T	T	A
1	2	3	4	5	6	7	8	9	10	11	12	13	14

has a consensus motif ACGGCTTA (nucleotides 7th → 14th) recognized by epigenetic enzymes and non-coding RNAs [33] [34] and three remarkable nucleotides (the 4th, 5th and 6th). In eukaryotes, DNA-methyltransferases are very sensitive, precise, and nucleotide specific: with the cooperation of non-coding RNAs [35] they can recognize 3, 2, or even only 1 nucleotide [36] [37]. In the above monomer, the 4th, 5th and 6th nucleotides have each their epigenetic peculiar task:

- the **4th nucleotide** (cytosine) is a mark used in the program as a "flag-point":

when a monomer is read (checked to know cell fate information) it is “flagged” and silenced (turned OFF) [38] [39] by methylation: $C \rightarrow 5mC$, “ C ” \rightarrow “ M ” in the code

- the **5th nucleotide** of each monomer indicates the division fate of the cell: thymine “ T ” states the entrance in quiescence *i.e.* no more divisions [40] while cytosine “ C ” states the entry into a new division cycle.

In the first monomer, the **5th nucleotide** may be different: “ T ” states the entrance in quiescence (G0-phase), “ C ” indicates a new symmetric division into two “twin sister” cells, “ H ”, hydroxymethylated cytosine (5HmC) more stable and less subject to spontaneous deamination (Hahn *et al.*, 2020) indicates the asymmetric division of a stem cell [41] [42] that produces an identical self-renewed stem cell and one “unique daughter” differentiating cell.

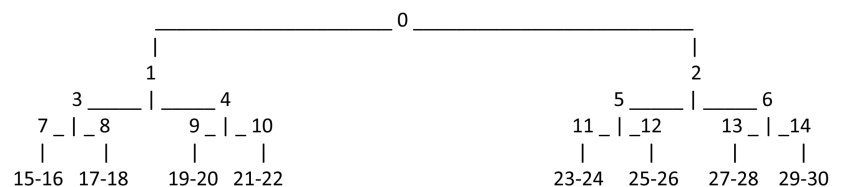
- the **6th nucleotide** of the **first** monomer of the “*sat_DNA*” sequence, adenine, is the **START** signal.

- the **6th nucleotide** of the **last** monomer, thymine, is the **STOP** signal. In the other monomers, the 6th nucleotide may be guanine or cytosine.

As output the program shows the lineages achievable when the user of the program operates a possible evolutionary “ C -to- T ” mutation by changing (overwriting) a “ C ” with a “ T ” in the 5th position of one or more monomers before running the program (**Figure 1**).

“ C -to- T ” transition [44] is a consequence of spontaneous cytosine deaminations, whose rate is estimated to be 100 to 500 cytosines per cell per day [45]; cytosine deamination is a strong modulator of genomic potential: it produces uracil, that, if not immediately repaired and substituted, will be replaced by thymine after two rounds of replication, resulting in a “ C -to- T ” mutation.

(A)



(B)

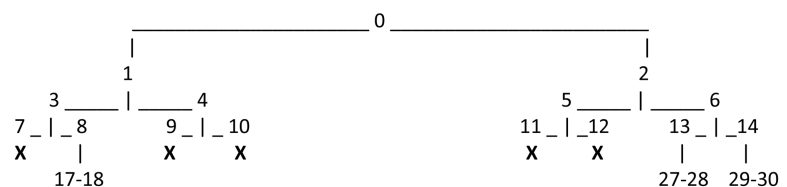


Figure 1. (A) The lineage that results without any C/T change. (B) In the 3rd monomer of “*sat_DNA*” the cytosine in the 5th position has been replaced by a thymine. the resulting lineage is shown. [Warning: 2D printed lineages do not respect the real 3D geometrical distribution of cells [43]: like genealogical trees and pedigrees, cell lineages trace progenitors and offspring of each cell in two dimensions (2D)].

The substitution of “C” with “T”, artificially operated by the user of the program in the 5th position of any monomer, simulates casual evolutionary events that realize different cell lineages: the cell fate coded by the 5th nucleotide changes from “keep dividing” (“C”) to “stop dividing” (“T”): natural selection conserves the better suitable cell lineages.

3.1. Running the Program “Lineages”

Epigenomic molecular mechanisms involved in DNA methylation work on linear nucleotide sequences in a fashion like the “*iteration protocol*” of algorithms. The computational model introduced here aims to show how these epigenetic mechanisms can count cell division rounds and keep track of their progressive movements on satDNA sequences: they can proceed linearly on TRs, read one monomer at a time, and leave a flag (cytosine-methylation) as a mark-record for descendant cells, sharing the same Boolean logic algebra that governs algorithms [46] [47] [48] [49].

The program “*Lineages*” simulates a possible cellular upstream preliminary check-point for entrance in mitosis: DNA replication (S-phase) generates two copies of the genome that must be converted back to symmetrically methylated DNA (reproducing the methylation pattern of mother cell DNA before the next S-phase, to avoid the loss of previous marks). Methylation maintenance, which occurs during chromatin re-assembly, is a fundamental epigenetic process for ensuring that methylations are copied and transmitted to daughter cells: methyltransferase enzymes are recruited to DNA methylation sites where they recognize specific epigenetic modifications on DNA strands and histones [49] [50]; the “*while loop*” at code line 208 (after the comment “*# RUNNING THE PROGRAM*”) simulates the work of epigenetic enzymes [51] that first recognize the start point on satDNA sequences [32] [36] [37] [52] [53] and then traverse the repetitive sequence:

i) readers and writers [25] run (iterate) the sequence, step by step, linearly, one monomer at a time, to check the state of each monomer (**4th nucleotide**)

ii) **4th nucleotide** = “M” means: “*already read*” (worked and silenced by methylation, 5mC: please, go on and check the next monomer)

iii) **4th nucleotide** = “C” means: “*not yet read*” (not methylated because not yet checked: please, stop here and see for the fate of the present cell stated by the 5th nucleotide).

iv) The “*sat_DNA*” sequence is read sequentially by the epigenetic “readers” until the first “*not yet read*” monomer (not switched-OFF, *i.e.* **4th nucleotide** = “C”) is found: at this point, the processive machine stops, the monomer is transcribed into a non-coding RNA and its information about cell-fate (**5th nucleotide** = “C” or “T”) is carried to cell-cycle regulators; a new “flag”, methylation of cytosine in 4th position (“C” → “M” in the code) is left by the “writer” enzyme (*de novo* methylation [24]); so, the current monomer changes its state to “*already read*” and updates the “bookmark-flags” (epigenetic cellular memory for

the offspring). After reading and sending fate information to cell-cycle regulators, the current round of epigenetic work stops, and the epigenetic complex disengages from DNA.

To simulate the asymmetric process of methylation maintenance (that distinguishes the fate of sister cells, see further) in the code it is supposed that one cell, named “*first cell*”, inherits the DNA containing the maternal filament with the old-methylated “*sat_DNA*” sequence, whereas the other daughter cell “*second cell*” receives the identical filament but newly synthesized and newly methylated; in the program, it is supposed that the process of methylation maintenance (see further) in the “*second cell*” methylates one more monomer to generate asymmetry between the two sister cells: so, the position of flags (5mC) differ for one monomer between the two daughter-sister cells (in the “*second cell*” the flag, 5mC, is shifted one monomer forward): this way, sister cells start reading “*sat_DNA*” from different positions, and find their private fate on different monomers.

3.2. How to Use the Program

The program “*Lineages*” may be easily executed by “not-pythonist” users: Google Colaboratory, commonly known as Google Colab, is a free cloud-based Python environment that provides a platform for writing and executing Python codes through one’s own browser without installing (the only requirement is a free Google account). For a short guide to Google Colab visit Dataquest: <https://www.dataquest.io/blog/getting-started-with-google-colab-for-deep-learning/>:

A) access to Colab B) once you’re on the Google Colab interface, click on File >> New notebook to create a new notebook C) on the left-hand side, click on the folder icon to open the file browser D) copy the Word version of “*Lineages*” attached in Supplementary Material E) paste it (Ctrl + V) on the line marked by a little black circle with a white triangular arrow inside F) press “Runtime” to run the program.

3.3. General Instructions

1) Copy the program “*Lineages*” attached in Supplementary Material and paste it into a Python environment.

2) Run the program: in a few seconds the complete lineage will appear on the shell: going back on the screen it is possible to follow each division round, cell by cell, and read the epigenetic modification (methylations) on the “*sat_DNA*” sequence.

3) Cancel (remove) “*C*” in the 5th position in one or more monomers and insert (write) “*T*” in the same position: the relative lineages will appear. (The last two monomers, useful for the next generation, do not affect the lineage because the program stops after four division rounds, *i.e.* after generating four sets of offspring: 30 cells is the maximum population).

4) Cancel “C” in the 5th position in the first monomer and write “H” to show the division of a stem (mother) cell in one (daughter) differentiating cell and one identical stem cell.

Instructions are also reported at the beginning of the program.

4. Results

“Lineages” is a computational model that wants to investigate the basic principle and steps of a possible biological pathway to count cell divisions and understand what are the actual molecular mechanisms.

In algorithms, iterators traverse data containers and access their elements one at a time, returning data and keeping track of the visited items; “Lineages” follows this same paradigm: it tests an *in silico* DNA repeated sequence (“sat-DNA”) as a molecular “iterable object” capable of counting cell divisions and returns the resulting lineages. “Lineages” tries to ascertain if TRs can follow the rules of Boolean algebra, necessary to count deterministic numbers of cell divisions without using numbers. “Lineages” operates with the same linear processivity as DNA replication, simulating epigenetic reactions (DNA methylation) whose patterns and paradigms are known to serve also as a template for histone reassembly during S phase [54].

SatDNA structure and composition suggest that it could effectively be used to count: made up of a sequence of similar monomers as a “pearl necklace”, its resemblance with arithmetic tools is very attractive and appealing: abaci and ancient religious objects used to count prayers, such as rosaries and misbahahs, all comprised of sequences of similar “monomers”, allow counting without numbers: if the user is equipped with many different tools of different lengths, each corresponding to some particular task, once the proper sequence has been chosen, the user enumerates its objects, one by one, and executes a precise amount of iterations without numbers. Indeed, satDNA comprises many sequences of different lengths, each distinguishable through its characteristic monomer. SatDNA is the only molecular tool that possesses a structure compatible with counting deterministic numbers of iterations without numbers. Indeed, the hypothesis that tandem repeats might work as genomic counters came up years ago [55]: telomere shortening, a much debated and controversial subject (telomeres, the ends of linear chromosomes, are made up of TRs) had been suspected to work as a “replicometer” [56]; telomere shortening occurs during cell division and has been associated with the replicative capacity of cells, in the sense that its shortening could limit the remaining number of divisions, causing cell senescence [57].

The computational model “Lineages” tests a different method to count cell divisions: no losses of material (telomere shortening is a consequence of rough nucleotide depletions), rather precise, intrinsic, and autonomous molecular mechanisms capable of running over TRs, using, as said, the same Boolean logic-algebra that governs algorithms [46] [47] [48] [49]. To count cell divisions, ordered natural numbers are not necessary for cells: different lengths of satDNA se-

quences [58] [59], selected during evolution to satisfy the iterative needs of the species, are enough to count different amounts of iterations. Big numbers of cell divisions may be counted on long CpG and non-CpG islands or on TR sequences that are re-counted several times: in the model organism *Danio rerio* (zebrafish), methylations of the “TGCT” monomer are inherited from maternal and paternal gametes, erased in mid-blastula transition, and *de novo* re-established in gastrulation in all embryonic layers [60]. The process of DNA methylation is coupled with replication and drives the movements of the replication fork (see: Ryba *et al.*: “The Temporal Order of DNA Replication Shaped by Mammalian DNA Methyltransferases” [61]). Like a metronome, cell divisions scan growth and cell fate [62]; rather than a simple metronome that controls cell divisions, a “differential metronome” must be considered: during mitosis, mother cell DNA is copied and both sister cells inherit an identical copy of DNA, nonetheless one of the two sister cells can stop dividing while the other goes on; no labels or addresses are necessary to identify cells (*in silico* as *in vivo*): each cell is distinguished by the inherited DNA, then, in sister cells, DNA must show some “non-genetic” difference due to the epigenetic (asymmetric) pathway of DNA methylation and methylation maintenance, very active on TRs [63].

The Question of Asymmetry

Epigenetic machinery, if on one hand can count the numbers of cell divisions, on the other cannot provide differential information for the (different) fate of somatic sister cells: somatic cell lineages are not symmetrical in that, as just said, one of the two cells can stop dividing, whereas her sister continues to divide: how can daughter cells, that inherit identical copies of DNA, find different instructions for their future development? Epigenetic mechanisms are the actors of symmetry breaking because they can change DNA expression without affecting DNA sequence [42]. Because of sister cell DNA identity, differences must necessarily originate from an asymmetry in the process of DNA methylation maintenance that drives sister cells to “read instructions” in different DNA positions. DNA methylation maintenance is a complex mechanism (see: Ming *et al.*: “Mitotic inheritance of DNA methylation: more than just copy and paste” [64]): during DNA replication, cytosine methylations are maintained by a plethora of enzymes (see: Jones and Liang “Rethinking how DNA methylation patterns are maintained” [95]); methylation maintenance is coupled with replication and occurs concomitantly [25] [64]-[72]; at the replication fork UHRF1 (an epigenetic reader with high affinity for 5mC) recruits DNA-methyltransferases to equalize methylations on both strands [73] [74]; this process is asymmetric on TRs: on the leading filament DNA Pol δ , with the cooperation of helicase, primase and a primer, meets monomers in opposite direction than DNA Pol ϵ on the lagging filament: a sequence of ten monomers (ideally numbered 3’_1, 2, 3...9, 10_5’ on the old-template-filament) meets DNA Pol δ from 1 to 10 (*i.e.* each monomer is met and copied from head to tail), but DNA Pol ϵ , with the

cooperation of ligase, primase and several RNA primers, synthesizes the new monomers in Okazaki fragments as 321, 654, 987 (*i.e.* each monomer is met and copied from tail to head). The role played by H3K9 histone confirms a differential action on the different (leading and lagging) filaments [71]; asymmetries in DNA methylation maintenance have been described in stem cells, where DNA methylation may be asymmetrically inherited by the two daughters [42]; timing differences during methylation maintenance have been found between the leading and lagging strands [64] [71] [72].

In the program “*Lineages*” a possible symmetry breaking mechanism has been simulated as follows: at the end of replication and methylation maintenance, two DNA molecules arise, one containing the old inherited, already methylated, filament and one with the newly synthesized and newly methylated strand: to carry different information to the offspring, these two molecules should contain a difference in the methylation pattern; the cell that inherits the “old” maternal filament receives all the previously methylated cytosines, while the other cell receives the newly synthesized strand (joined Okasaki fragments) that contains newly methylated cytosines; only this filament has been undergone to the process of methylation maintenance: in the program it is supposed that in its “*sat_DNA*” one more monomer is methylated and silenced to generate a difference between sister cells (see: Khristich and Mirkin “On the wrong DNA track: Molecular mechanisms of repeat-mediated genome instability, 2020 [18]); each cell finds its future fate (“*C*” or “*T*” in 5th position) on a different monomer. This way, a biological mechanism is outlined, autonomous, automatic, and unsupervised, able to provide, from identical DNA sequences, different fates for sister cells and asymmetric lineages.

5. Discussion

Satellite DNA can no longer be considered “junk” (useless) DNA: its role is not restricted to maintaining chromosome structure or heterochromatin establishment, in many species satDNA is present within euchromatin [75] and many TRs are actively transcribed into non-coding RNAs; in *C. elegans* different regions of satDNA are differently transcribed in different tissues [7], unveiling tissue-specific roles of TRs. CpG dinucleotides frequently occur in DNA regions called “CpG islands”: there are about 29,000 CpG islands in human euchromatin, and 50,000 if CpG islands in TRs are also counted [76]. Methylated cytosines, in CpGs and non-CpGs, are everywhere in genomes: repetitive sequences, enhancers, promoters, and gene bodies [77]. Cytosine methylation, in CpG dinucleotides but also non-CpG sites (CpA, CpT, and CpC), plays a crucial role as an epigenetic mark in animals, silencing or activating genes depending on tissues [77].

As already said, the role of satDNA in morphogenesis has been studied in invertebrates and vertebrates: the small gene diversity between chimpanzees and humans cannot explain the three times brain expansion and the twofold increase in neuron number in human cerebral cortex [78]: the real genomic disparity is in

TRs [79]. A small difference in the genome of *Homo sapiens* relative to Neanderthals, a single-point-mutation in the transketolase-like 1 gene (*TKTL1*) [80] leads to an impressive increase of basal radial glial cell divisions, that, in turn, boost the output of upper layer projection neurons: *TKTL1* is involved in DNA methylation [81] [82] particularly in micro-satellite methylation [83]. SatDNA diversity correlates with relevant anatomical differences (see: Fondon and Garner “Molecular origins of rapid and continuous morphological evolution”, 2004 [16]; Myers “Tandem Repeats and Morphological Variation”, 2007 [84]); the abundant variety of TRs of different lengths can sustain interindividual morphological diversity: in the retina of the model organism *Danio rerio*, neurogenic progenitors produce two daughter cells with different fates, one deterministic and one probabilistic; interference with the deterministic branch of the lineage affects lineage progression, in contrast, the probabilistic branch has a large range of fates [85]; lineage flexibility fits well with the large number of TRs of different lengths.

Many finding supports the hypothesis that TRs, together with DNA methylation, play a role in cell divisions and in early embryo cleavage (where counting cell division rounds is crucial): TRs and epigenetic enzymes control Polycomb [86] [87]: Polycomb group, a family of chromatin remodeling protein involved in epigenetic pathways, is known to downregulate *CyclinA* in *Drosophila*, plants and vertebrates [38] [88] [89] [90] [91] [92]; a non-coding Alu (ashort interspersed repetitive element) controls the expression of cell cycle genes in human fibroblasts [93]: its overexpression promotes the transition from G1 to S phase; long-non-coding-RNAs, associated with satellite repeats [94], modulate methylation in eukaryotes [96]: permanent exit from the cell cycle is associated with epigenetic methylations [97]. Immediately after fertilization, in mammalian embryos a strong wave of *de novo* methylation sustains and supports zygote cleavage [24]; mammalian embryonic stem cells have large amounts of not methylated CpG dinucleotides in their satDNA: many CpG cytosines adenines in roundworms [98]) are actively methylated and demethylated [32].

6. Conclusion

“Lineages” is an efficient program to simulate processive cellular functions that, executed step by step on linear templates, enumerate precise and invariable numbers of cell divisions: TRs are run over by accurate epigenetic complexes involved in DNA methylation showing that TRs together with the epigenetic machine may be the molecular basis cells use to count deterministic numbers of division rounds. “Lineages” helps to mimic evolutionary processes that have established convenient pedigrees of cell divisions, approved by natural selection as the most suitable and adapted for each species.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Gilbert, S.F. (2000) *Developmental Biology: The Questions of Developmental Biology*. 6th Edition, Sinauer Associates, Sunderland.
- [2] Cole, A.G., *et al.* (2023) Gene Regulatory Patterning Codes in Early Cell Fate Specification of the *C. elegans* Embryo. *eLife*, **12**, RP87099. <https://doi.org/10.7554/eLife.87099>
- [3] Robertson, S.M., *et al.* (2014) Uncoupling Different Characteristics of the *C. elegans* E Lineage from Differentiation of Intestinal Markers. *PLOS ONE*, **9**, e106309. <https://doi.org/10.1371/journal.pone.0106309>
- [4] Ahmad, K. and Henikoff, S. (2022) The *Drosophila* Embryo as a *Tabula rasa* for the Epigenome. *Faculty Reviews*, **11**, Article 40. <https://doi.org/10.12703/r/11-40>
- [5] Shvartsman, S.Y., *et al.* (2008) Dynamics of Maternal Morphogen Gradients in *Drosophila*. *Current Opinion in Genetics & Development*, **18**, 342-347. <https://doi.org/10.1016/j.gde.2008.06.002>
- [6] Abouchar, L., *et al.* (2014) Fly Wing Vein Patterns Have Spatial Reproducibility of a Single Cell. *Journal of the Royal Society Interface*, **11**, Article 20140443. <https://doi.org/10.1098/rsif.2014.0443>
- [7] Subirana, J.A., *et al.* (2023) Features of Satellite DNA Transcription in Different Tissues of *Caenorhabditis elegans*. *International Journal of Molecular Sciences*, **24**, Article 2970. <https://doi.org/10.3390/ijms24032970>
- [8] Jagannathan, M., *et al.* (2017) Comparative Analysis of Satellite DNA in the *Drosophila melanogaster* Species Complex. *G3 Genes|Genomes|Genetics*, **7**, 693-704.
- [9] Gemayel, R., *et al.* (2012) Beyond Junk-Variable Tandem Repeats as Facilitators of the Rapid Evolution of Regulatory and Coding Sequences. *Genes*, **3**, 461-480. <https://doi.org/10.3390/genes3030461>
- [10] Louzada, S., *et al.* (2020) Decoding the Role of Satellite DNA in Genome Architecture and Plasticity—An Evolutionary and Clinical Affair. *Genes*, **11**, Article 72. <https://doi.org/10.3390/genes11010072>
- [11] Thakur, J., *et al.* (2021) Sequence, Chromatin and Evolution of Satellite DNA. *International Journal of Molecular Sciences*, **22**, Article 4309. <https://doi.org/10.3390/ijms22094309>
- [12] Zeni Dos Santos, R., *et al.* (2021) A Long-Term Conserved Satellite DNA that Remains Unexpanded in Several Genomes of Characiformes Fish Is Actively Transcribed. *Genome Biology and Evolution*, **13**, Article evab002. <https://doi.org/10.1093/gbe/evab002>
- [13] Papa, R., *et al.* (2008) Highly Conserved Gene Order and Numerous Novel Repetitive Elements in Genomic Regions Linked to Wing Pattern Variation in *Heliconius* Butterflies. *BMC Genomics*, **9**, Article No. 345. <https://doi.org/10.1186/1471-2164-9-345>
- [14] Mazo-Vargas, A., *et al.* (2022) Deep Cis-Regulatory Homology of the Butterfly Wing Pattern Ground Plan. *Science*, **378**, 304-308. <https://doi.org/10.1126/science.abi9407>
- [15] Sears, K.E., *et al.* (2007) The Correlated Evolution of Runx2 Tandem Repeats, Transcriptional Activity, and Facial Length in Carnivora. *Evolution & Development*, **9**, 555-565. <https://doi.org/10.1111/j.1525-142X.2007.00196.x>
- [16] Fondon, J.W. and Garner, H.R. (2004) Molecular Origins of Rapid and Continuous Morphological Evolution. *Proceedings of the National Academy of Sciences*, **101**, 18058-18063. <https://doi.org/10.1073/pnas.0408118101>

- [17] Castillo-Lizardo, M., *et al.* (2014) Replication Slippage of the Thermophilic DNA Polymerases B and D from the Euryarchaeota *Pyrococcus abyssi*. *Frontiers in Microbiology*, **5**, Article 403. <https://doi.org/10.3389/fmicb.2014.00403>
- [18] Khristich, A.N. and Mirkin, S.M. (2020) On the Wrong DNA Track: Molecular Mechanisms of Repeat-Mediated Genome Instability. *Journal of Biological Chemistry*, **295**, 4134-4170. <https://doi.org/10.1074/jbc.REV119.007678>
- [19] Cacheux, L., *et al.* (2016) Diversity and Distribution of Alpha Satellite DNA in the Genome of an Old World Monkey: *Cercopithecus solatus*. *BMC Genomics*, **17**, Article No. 916. <https://doi.org/10.1186/s12864-016-3246-5>
- [20] Ruiz-Ruano, F.J., *et al.* (2016) High-Throughput Analysis of the Satellitome Illuminates Satellite DNA Evolution. *Scientific Reports*, **6**, Article No. 28333. <https://doi.org/10.1038/srep28333>
- [21] Martin-Trujillo, A., *et al.* (2023) Genome-Wide Evaluation of the Effect of Short Tandem Repeat Variation on Local DNA Methylation. *Genome Research*, **33**, 184-196. <https://doi.org/10.1101/gr.277057.122>
- [22] Probst, A.V., *et al.* (2009) Epigenetic Inheritance during the Cell Cycle. *Nature Reviews Molecular Cell Biology*, **10**, 192-206. <https://doi.org/10.1038/nrm2640>
- [23] Su, J., *et al.* (2012) Genome-Wide Dynamic Changes of DNA Methylation of Repetitive Elements in Human Embryonic Stem Cells and Fetal Fibroblasts. *Genomics*, **99**, 10-17. <https://doi.org/10.1016/j.ygeno.2011.10.004>
- [24] Lacal, I. and Ventura, R. (2018) Epigenetic Inheritance: Concepts, Mechanisms and Perspectives. *Frontiers in Molecular Neuroscience*, **11**, Article 292. <https://doi.org/10.3389/fnmol.2018.00292>
- [25] Biswas, S. and Rao, C.M. (2018) Epigenetic Tools (The Writers, the Readers and the Erasers) and Their Implications in Cancer Therapy. *European Journal of Pharmacology*, **837**, 8-24. <https://doi.org/10.1016/j.ejphar.2018.08.021>
- [26] Nicholson, T.B., *et al.* (2015) Writers, Readers, and Erasers of Epigenetic Marks. In: Gray, S.G., Ed., *Epigenetic Cancer Therapy*, Academic Press, Cambridge, 31-66. <https://doi.org/10.1016/B978-0-12-800206-3.00003-3>
- [27] Yang, A.Y., *et al.* (2016) Natural Compound-Derived Epigenetic Regulators Targeting Epigenetic Readers, Writers and Erasers. *Current Topics in Medicinal Chemistry*, **16**, 697-713. <https://doi.org/10.2174/1568026615666150826114359>
- [28] Guo, F., *et al.* (2014) Active and Passive Demethylation of Male and Female Pronuclear DNA in the Mammalian Zygote. *Cell Stem Cell*, **15**, 447-459. <https://doi.org/10.1016/j.stem.2014.08.003>
- [29] Esteve-Puig, R., *et al.* (2020) Writers, Readers and Erasers of RNA Modifications in Cancer. *Cancer Letters*, **474**, 127-137. <https://doi.org/10.1016/j.canlet.2020.01.021>
- [30] Damiani, E., *et al.* (2020) Targeting Epigenetic “Readers” with Natural Compounds for Cancer Interception. *Journal of Cancer Prevention*, **25**, 189-203. <https://doi.org/10.15430/JCP.2020.25.4.189>
- [31] Shyamala, N., *et al.* (2022) In Silico Identification of Single Nucleotide Variations at CpG Sites Regulating CpG Island Existence and Size. *Scientific Reports*, **12**, Article No. 3574. <https://doi.org/10.1038/s41598-022-05198-8>
- [32] Pappalardo, X.G. and Barra, V. (2021) Losing DNA Methylation at Repetitive Elements and Breaking Bad. *Epigenetics & Chromatin*, **14**, Article No. 25. <https://doi.org/10.1186/s13072-021-00400-z>
- [33] Gao, L., *et al.* (2020) Comprehensive Structure-Function Characterization of DNMT3B and DNMT3A Reveals Distinctive *de novo* DNA Methylation Mecha-

- nisms. *Nature Communications*, **11**, Article No. 3355. <https://doi.org/10.1038/s41467-020-17109-4>
- [34] Wei, J.-W., *et al.* (2017) Non-Coding RNAs as Regulators in Epigenetics (Review). *Oncology Reports*, **37**, 3-9. <https://doi.org/10.3892/or.2016.5236>
- [35] Mehraj, H., *et al.* (2021) Genome-Wide Analysis of Long Noncoding RNAs, 24-nt siRNAs, DNA Methylation and H3K27me3 Marks in *Brassica rapa* and Related Species. *PLOS ONE*, **16**, e0242530. <https://doi.org/10.1101/2020.11.06.370940>
- [36] Clark, T.A., *et al.* (2012) Characterization of DNA Methyltransferase Specificities Using Single-Molecule, Real-Time DNA Sequencing. *Nucleic Acids Research*, **40**, e29. <https://doi.org/10.1093/nar/gkr1146>
- [37] Furuta, Y. and Kobayashi, I. (2012) Mobility of DNA Sequence Recognition Domains in DNA Methyltransferases Suggests Epigenetics-Driven Adaptive Evolution. *Mobile Genetic Elements*, **2**, 292-296. <https://doi.org/10.4161/mge.23371>
- [38] Stern, A., *et al.* (1986) Opacity Genes in *Neisseria Gonorrhoeae*: Control of Phase and Antigenic Variation. *Cell*, **47**, 61-71. [https://doi.org/10.1016/0092-8674\(86\)90366-1](https://doi.org/10.1016/0092-8674(86)90366-1)
- [39] Ma, M. (2020) DNA Methylation Silences Exogenous Gene Expression in Transgenic Birch Progeny. *Frontiers in Plant Science*, **11**, Article 523748. <https://doi.org/10.3389/fpls.2020.523748>
- [40] Martinez, A.M., *et al.* (2006) Polycomb Group-Dependent *Cyclin* a Repression in *Drosophila*. *Genes & Development*, **20**, 501-513. <https://doi.org/10.1101/gad.357106>
- [41] Elabd, C., *et al.* (2013) DNA Methyltransferase-3-Dependent Nonrandom Template Segregation in Differentiating Embryonic Stem Cells. *Journal of Cell Biology*, **203**, 73-85. <https://doi.org/10.1083/jcb.201307110>
- [42] Zion, E. and Chen, X. (2021) Breaking Symmetry: The Asymmetries in Epigenetic Inheritance. *The Biochemist*, **43**, 14-19. https://doi.org/10.1042/bio_2020_110
- [43] Giurumescu, C.A. and Chisholm, A.D. (2011) Cell Identification and Cell Lineage Analysis. *Methods in Cell Biology*, **106**, 325-341. <https://doi.org/10.1016/B978-0-12-544172-8.00012-8>
- [44] Nabel, C.S., *et al.* (2012) The Curious Chemical Biology of Cytosine: Deamination, Methylation, and Oxidation as Modulators of Genomic Potential. *ACS Chemical Biology*, **7**, 20-30. <https://doi.org/10.1021/cb2002895>
- [45] Alsøe, L., *et al.* (2017) Uracil Accumulation and Mutagenesis Dominated by Cytosine Deamination in CpG Dinucleotides in Mice Lacking UNG and SMUG1. *Scientific Reports*, **7**, Article No. 7199. <https://doi.org/10.1038/s41598-017-07314-5>
- [46] Helikar, T., *et al.* (2011) Boolean Modeling of Biochemical Networks. *The Open Bioinformatics Journal*, **5**, 16-25. <https://doi.org/10.2174/1875036201105010016>
- [47] Chen, H., *et al.* (2016) Boolean Models of Biological Processes Explain Cascade-Like Behavior. *Scientific Reports*, **6**, Article No. 20067. <https://doi.org/10.1038/srep20067>
- [48] Boniolo, G., *et al.* (2021) Molecular Biology Meets Logic: Context-Sensitiveness in Focus. *Foundations of Science*, **28**, 307-325. <https://doi.org/10.1007/s10699-021-09789-y>
- [49] Chiang, C.-Y., *et al.* (2022) Deciphering Chemical Logic of Fungal Natural Product Biosynthesis through Heterologous Expression and Genome Mining. *Natural Product Reports*, **40**, 89-127. <https://doi.org/10.1039/D2NP00050D>
- [50] Nishiyama, A., *et al.* (2020) Two Distinct Modes of DNMT1 Recruitment Ensure Stable Maintenance of DNA Methylation. *Nature Communications*, **11**, Article No. 1222. <https://doi.org/10.1038/s41467-020-15006-4>

- [51] Ishiyama, S., *et al.* (2017) Structure of the Dnmt1 Reader Module Complexed with a Unique Two-Mono-Ubiquitin Mark on Histone H3 Reveals the Basis for DNA Methylation Maintenance. *Molecular Cell*, **68**, 350-360. <https://doi.org/10.1016/j.molcel.2017.09.037>
- [52] Patel, D.J. (2016) A Structural Perspective on Readout of Epigenetic Histone and DNA Methylation Marks. *Cold Spring Harbor Perspectives in Biology*, **8**, Article a018754. <https://doi.org/10.1101/cshperspect.a018754>
- [53] Andrews, F.H., *et al.* (2016) Insights into Newly Discovered Marks and Readers of Epigenetic Information. *Nature Chemical Biology*, **12**, 662-668. <https://doi.org/10.1038/nchembio.2149>
- [54] Cedar, H. and Bergman, Y. (2009) Linking DNA Methylation and Histone Modification: Patterns and Paradigms. *Nature Reviews Genetics*, **10**, 295-304. <https://doi.org/10.1038/nrg2540>
- [55] Cooke, H.J. and Smith, B.A. (1986) Variability at the Telomeres of the Human X/Y Pseudoautosomal Region. *Cold Spring Harbor Symposia on Quantitative Biology*, **51**, 213-219. <https://doi.org/10.1101/SQB.1986.051.01.026>
- [56] Suram, A. and Herbig, U. (2014) The Replicometer Is Broken: Telomeres Activate Cellular Senescence in Response to Genotoxic Stresses. *Aging Cell*, **13**, 780-786. <https://doi.org/10.1111/accel.12246>
- [57] Zotta Mota, A.P., *et al.* (2024) Unzipped Genome Assemblies of Polyploid Root-Knot Nematodes Reveal Unusual and Clade-Specific Telomeric Repeats. *Nature Communications*, **15**, Article No. 773. <https://doi.org/10.1038/s41467-024-44914-y>
- [58] Fotsing, S.F., *et al.* (2019) The Impact of Short Tandem Repeat Variation on Gene Expression. *Nature Genetics*, **51**, 1652-1659. <https://doi.org/10.1038/s41588-019-0521-9>
- [59] Reinar, W.B., *et al.* (2021) Length Variation in Short Tandem Repeats Affects Gene Expression in Natural Populations of Arabidopsis Thaliana. *The Plant Cell*, **33**, 2221-2234. <https://doi.org/10.1093/plcell/koab107>
- [60] Ross, S.E., *et al.* (2020) Developmental Remodelling of Non-CG Methylation at Satellite DNA Repeats. *Nucleic Acids Research*, **48**, 12675-12688. <https://doi.org/10.1093/nar/gkaa1135>
- [61] Ryba, T., *et al.* (2021) The Temporal Order of DNA Replication Shaped by Mammalian DNA Methyltransferases. *Cells*, **10**, Article 266. <https://doi.org/10.3390/cells10020266>
- [62] Fischer, A.H., *et al.* (2014) A Cleavage Clock Regulates Features of Lineage-Specific Differentiation in the Development of a Basal Branching Metazoan, the Ctenophore *Mnemiopsis leidyi*. *EvoDevo*, **5**, Article No. 4. <https://doi.org/10.1186/2041-9139-5-4>
- [63] Choi, S.H., *et al.* (2009) Changes in DNA Methylation of Tandem DNA Repeats Are Different from Interspersed Repeats in Cancer. *International Journal of Cancer*, **125**, 723-729. <https://doi.org/10.1002/ijc.24384>
- [64] Ming, X., *et al.* (2021) Mitotic Inheritance of DNA Methylation: More than Just Copy and Paste. *Journal of Genetics and Genomics*, **48**, 1-13. <https://doi.org/10.1016/j.jgg.2021.01.006>
- [65] Feng, S., *et al.* (2010) Epigenetic Reprogramming in Plant and Animal Development. *Science*, **330**, 622-627. <https://doi.org/10.1126/science.1190614>
- [66] Felsenfeld, G. (2014) A Brief History of Epigenetics. *Cold Spring Harbor Perspectives in Biology*, **6**, Article a018200. <https://doi.org/10.1101/cshperspect.a018200>

- [67] Rothbart, S.B. and Strahl, B.D. (2014) Interpreting the Language of Histone and DNA Modifications. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, **1839**, 627-643. <https://doi.org/10.1016/j.bbagr.2014.03.001>
- [68] Deshmukh, S., *et al.* (2018) Levels of DNA Cytosine Methylation in the *Drosophila* genome. *PeerJ*, **6**, Article e5119. <https://doi.org/10.7717/peerj.5119>
- [69] Wang, S.Y., *et al.* (2019) The Demethylase NMAD-1 Regulates DNA Replication and Repair in the *Caenorhabditis elegans* Germline. *PLOS GENETIC*, **15**, e1008252. <https://doi.org/10.1371/journal.pgen.1008252>
- [70] Hahm, J.Y., *et al.* (2020) Acetylation of UHRF1 Regulates Hemi-Methylated DNA Binding and Maintenance of Genome-Wide DNA Methylation. *Cell Reports*, **32**, Article 107958. <https://doi.org/10.1016/j.celrep.2020.107958>
- [71] Ming, X., *et al.* (2020) Kinetics and Mechanisms of Mitotic Inheritance of DNA Methylation and Their Roles in Aging-Associated Methylome Deterioration. *Cell Research*, **30**, 980-996. <https://doi.org/10.1038/s41422-020-0359-9>
- [72] Petryk, N., *et al.* (2021) Staying True to Yourself: Mechanisms of DNA Methylation Maintenance in Mammals. *Nucleic Acids Research*, **49**, 3020-3032. <https://doi.org/10.1093/nar/gkaa1154>
- [73] Liu, X., *et al.* (2013) UHRF1 Targets DNMT1 for DNA Methylation through Cooperative Binding of Hemi-Methylated DNA and Methylated H3K9. *Nature Communications*, **4**, Article No. 1563. <https://doi.org/10.1038/ncomms2562>
- [74] Hahn, M.A., *et al.* (2020) 5-Hydroxymethylcytosine: A Stable or Transient DNA Modification? *Genomics*, **104**, 314-323. <https://doi.org/10.1016/j.ygeno.2014.08.015>
- [75] Feliciello, I., *et al.* (2021) Satellite DNA-Mediated Gene Expression Regulation: Physiological and Evolutionary Implication. In: Ugarković, D., Eds., *Satellite DNAs in Physiology and Evolution. Progress in Molecular and Subcellular Biology*, Vol. 60, Springer, Cham, 145-167. https://doi.org/10.1007/978-3-030-74889-0_6
- [76] Lander, E.S., *et al.* (2001) Initial Sequencing and Analysis of the Human Genome. *Nature*, **409**, 860-921.
- [77] Jang, H.S., *et al.* (2017) CpG and Non-CpG Methylation in Epigenetic Gene Regulation and Brain Function. *Genes*, **8**, Article 148. <https://doi.org/10.3390/genes8060148>
- [78] Ma, S., *et al.* (2022) Molecular and Cellular Evolution of the Primate Dorsolateral Prefrontal Cortex. *Science*, **377**, Article eabo7257.
- [79] Ahmad, S.F., *et al.* (2020) Dark Matter of Primate Genomes: Satellite DNA Repeats and Their Evolutionary Dynamics. *Cells*, **9**, Article 2714. <https://doi.org/10.3390/cells9122714>
- [80] Pinson, A., *et al.* (2022) Scaling Brain Neurogenesis across Evolution. *Science*, **377**, 1155-1156. <https://doi.org/10.1126/science.ade4388>
- [81] Sun, W., *et al.* (2010) TKTL1 Is Activated by Promoter Hypomethylation and Contributes to Head and Neck Squamous Cell Carcinoma Carcinogenesis through Increased Aerobic Glycolysis and HIF1 α Stabilization. *Clinical Cancer Research*, **16**, 857-866. <https://doi.org/10.1158/1078-0432.CCR-09-2604>
- [82] Li, M., *et al.* (2016) Filtrating Colorectal Cancer Associated Genes by Integrated Analysis of Global DNA Methylation and Hydroxymethylation in Cancer and Normal Tissues. *Scientific Reports*, **6**, Article 31826. <https://doi.org/10.1038/srep31826>
- [83] Saghafinia, S., *et al.* (2018) Pan-Cancer Landscape of Aberrant DNA Methylation across Human Tumors. *Cell Reports*, **25**, 1066-1080. <https://doi.org/10.1016/j.celrep.2018.09.082>

- [84] Myers, P.Z. (2007) Tandem Repeats and Morphological Variation. *Nature Education*, **1**, 1.
- [85] Nerli, E., *et al.* (2023) Deterministic and Probabilistic Fate Decisions Co-Exist in a Single Retinal Lineage. *The EMBO Journal*, **42**, e112657. <https://doi.org/10.15252/embj.2022112657>
- [86] Casa, V. and Gabellini, D. (2012) A Repetitive Elements Perspective in Polycomb Epigenetics. *Frontiers in Genetics*, **3**, Article 199. <https://doi.org/10.3389/fgene.2012.00199>
- [87] Walter, M., *et al.* (2016) An Epigenetic Switch Ensures Transposon Repression upon Dynamic Loss of DNA Methylation in Embryonic Stem Cells. *eLife*, **5**, e11418. <https://doi.org/10.7554/eLife.11418>
- [88] Jacqueline, J.L., *et al.* (2002) Polycomb Repression: From Cellular Memory to Cellular Proliferation and Cancer. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, **1602**, 151-161. [https://doi.org/10.1016/S0304-419X\(02\)00052-5](https://doi.org/10.1016/S0304-419X(02)00052-5)
- [89] Pirrotta, V. (2009) Polycomb Repression under the Skin. *Cell*, **136**, 992-994. <https://doi.org/10.1016/j.cell.2009.03.004>
- [90] De Lucas, M., *et al.* (2016) Transcriptional Regulation of Arabidopsis Polycomb Repressive Complex 2 Coordinates Cell-Type Proliferation and Differentiation. *The Plant Cell*, **28**, 2616-2631. <https://doi.org/10.1105/tpc.15.00744>
- [91] Ma, K.H., Duong, P., Moran, J.J., Junaidi, N. and Svaren, J. (2018) Polycomb Repression Regulates Schwann Cell Proliferation and Axon Regeneration after Nerve Injury. *Glia*, **66**, 2487-2502. <https://doi.org/10.1002/glia.23500>
- [92] Flora, P., *et al.* (2021) Polycomb Repressive Complex(ES) and Their Role in Adult Stem Cells. *Genes*, **12**, Article 1485. <https://doi.org/10.3390/genes12101485>
- [93] Cantarella, S., *et al.* (2019) Alu RNA Modulates the Expression of Cell Cycle Genes in Human Fibroblasts. *International Journal of Molecular Sciences*, **20**, Article 3315. <https://doi.org/10.3390/ijms20133315>
- [94] Lee, H., *et al.* (2019) Long Noncoding RNAs and Repetitive Elements: Junk or Intimate Evolutionary Partners? *Trends in Genetics*, **35**, 892-902. <https://doi.org/10.1016/j.tig.2019.09.006>
- [95] Jones, P.A. and Liang, G. (2009) Rethinking How DNA Methylation Patterns Are Maintained. *Nature Reviews Genetics*, **10**, 805-811. <https://doi.org/10.1038/nrg2651>
- [96] Statello, L., *et al.* (2021) Gene Regulation by Long Non-Coding RNAs and Its Biological Functions. *Nature Reviews Molecular Cell Biology*, **22**, 96-118. <https://doi.org/10.1038/s41580-020-00315-9>
- [97] Ma, Y. and Buttitta, L. (2017) Chromatin Organization Changes during the Establishment and Maintenance of the Postmitotic State. *Epigenetics & Chromatin*, **10**, Article No. 53. <https://doi.org/10.1186/s13072-017-0159-8>
- [98] Greer, E.L., *et al.* (2015) DNA Methylation on N⁶-Adenine in *C. elegans*. *Cell*, **161**, 868-878. <https://doi.org/10.1016/j.cell.2015.04.005>

Appendix

```
# LINEAGES
#
# GENERAL INSTRUCTIONS:
#
# 1) Change "C" with "T" in 5th position of one or more sublists/monomers of 'sat_DNA' to silence one or more
monomers:
#     run the program and, division after division, look, step by step, at the printed cell-lineage.
#
# 2) "H" in 5th position of the first sublist of 'sat_DNA' starts "asymm_mitosis".
```

```
from copy import deepcopy
import sys
```

```
sat_DNA = [['A','T','T','C','C','A','A','C','G','G','C','T','T','A'],['A','T','T','C','C','C','A','C','G','G','C','T','T','A'],\
           ['A','T','T','C','C','G','A','C','G','G','C','T','T','A'],['A','T','T','C','C','C','A','C','G','G','C','T','T','A'],\
           ['A','T','T','C','C','G','A','C','G','G','C','T','T','A'],['A','T','T','C','C','C','A','C','G','G','C','T','T','A'],\
           ['A','T','T','C','C','C','A','C','G','G','C','T','T','A'],['A','T','T','C','C','T','A','C','G','G','C','T','T','A']]
```

```
cells = {0:sat_DNA}    # dictionary of cells (founder cell has "key" = '0') and their methylated satDNA ("value")
```

```
arisen_cells = [0]    # list of generated cells
```

```
existing_cells = [0]  # after division, the mother cell is replaced by its two daughters; [n]
                    # (a number between squared brackets)is the number (label) of a cell in G0
```

```
dividing_cells = [0] # list of cells allowed to divide
```

```
not_div_cells = []   # list of cells banned to divide
```

```
Stop = 0             # setting Stop at '1' avoids infinite loops
```

```
pos = 0              # 'pos' is the label (key) of each cell in 'cells' dictionary
```

```
num_mit = 1
```

```
# FUNCTION "SYMMETRICAL MITOSIS"
```

```
#Division of a somatic (mother) cell in two (daughter) sister cells
```

```

def symm_mitosis(cell_num):
    global num_mit
    global Stop
    global existing_cells
    global dividing_cells
    global arisen_cells
    global pos
    pos = 2*cell_num          # from cell n° 'x', cells n° '2x+1' and '2x+2' arise
    if Stop == 1:
        return
    elif sat_DNA[0][3] == 'C' and sat_DNA[0][4] == 'H' and sat_DNA[0][5] == 'A':
        asymm_mitosis(cell_num) # "H" in 5th position of the first sublist/monomer of 'sat_DNA' indicates
"asymm_mitosis"
    else:
        cells[pos+1] = []      # list for deepcopying DNA
        cells[pos+2] = []
        arisen_cells.append(pos+1)
        arisen_cells.append(pos+2)
        print()
        print('Mitosis N°',num_mit)
        print()
        print('from division of cell N°', cell_num, 'cells N°', pos+1,\
            'and',pos+2,'arise')

        # First cell

        cells[pos+1] = deepcopy(cells[cell_num]) # satDNA is copied and associated as 'value' of this cell in
'cells' dictionary
        i = 0                                     # 'first cell' starts reading from the first sublist
(monomer)of 'sat_DNA'
        while i<len(cells[pos+1]):                # step by step satDNA is traversed
            if cells[pos+1][i][3] != 'M' and cells[pos+1][i][3] != 'C': # check for not recognized se-
quences
                Stop = 1
                break
            if cells[pos+1][i][4] != 'C' and cells[pos+1][i][4] != 'T': # check for not recognized sequences
                Stop = 1
                break
            if cells[pos+1][i][3] == 'M':          # this monomer has been silenced, check the next
                i+=1                                # processive mechanism activation
                continue
            elif cells[pos+1][i][3] == 'C'and cells[pos+1][i][4] == 'C' and cells[pos+1][i][5] == 'T': # this is
the last monomer

```

```

        dividing_cells.append(pos+1)
        cells[pos+1][i][3] = 'M'          # methylate this monomer and enter G0
        Stop = 1
        break
    elif cells[pos+1][i][3] != 'M' and cells[pos+1][i][4] == 'C':
        cells[pos+1][i][3] = 'M'          # methylate this monomer and divide
        dividing_cells.append(pos+1)
        break
    elif cells[pos+1][i][3] != 'M' and cells[pos+1][i][4] == 'T': # methylate this monomer and enter
G0
        cells[pos+1][i][3] = 'M'
        not_div_cells.append(pos+1)
        break
    i+=1

# Second cell

cells[pos+2] = deepcopy(cells[pos+1]) # at the replication fork the second cell DNA is bookmarked:
                                         # so, in the program, second cell DNA = first cell DNA
i = 1                                     # 'second cell' starts reading from the second sublist of
'sat_DNA'
while i<len(cells[pos+2]):
    if cells[pos+2][i][3] != 'M' and cells[pos+2][i][3] != 'C':
        Stop = 1
        break
    if cells[pos+2][i][4] != 'C' and cells[pos+2][i][4] != 'T':
        Stop = 1
        break
    if cells[pos+2][i][3] == 'M':
        i+=1                               # processive mechanism activation
        continue
    if cells[pos+2][i][3] == 'C' and cells[pos+2][i][4] == 'C' and cells[pos+2][i][5] == 'T': # this is the
last monomer
        not_div_cells.append(pos+2)
        cells[pos+2][i][3] = 'M'          # methylate this monomer and enter G
        Stop = 1
        break
    elif cells[pos+2][i][3] != 'M' and cells[pos+2][i][4] == 'C': # methylate this monomer and
divide
        cells[pos+2][i][3] = 'M'

        dividing_cells.append(pos+2)
        break

```

```

elif cells[pos+2][i][3] != 'M' and cells[pos+2][i][4] == 'T':      # methylate this monomer and
enter G0
    not_div_cells.append(pos+2)
    break
i+=1

num_mit +=1
print('cell N°',cell_num,'satDNA:\n',cells[cell_num]) # 'sat_DNA' and its metilations are printed
print()
print('cell N°',pos+1,'satDNA:\n',cells[pos+1]) # " " " "
print()
print('cell N°',pos+2,'satDNA:\n',cells[pos+2]) # " " " "
print()
print()
existing_cells = dividing_cells + not_div_cells
existing_cells.sort()
for q in range(len(existing_cells)):
    if existing_cells[q] in not_div_cells:
        existing_cells[q] = [existing_cells[q]]
print('existing_cells', existing_cells)
print('dividing_cells',dividing_cells )
print('not_div_cells', not_div_cells)
print()
if dividing_cells == []:
    Stop = 1

# FUNCTION ASYMMETRICAL MITOSIS (Stem cells)
# Division of a stem (mother) cell in: one (daughter) differentiating cell and one identical stem cell

def asymm_mitosis(cell_num):
    global num_mit
    global Stop
    global arisen_cells
    global existing_cells
    global not_div_cells
    global dividing_cells
    global pos
    if Stop == 1:
        return
    pos = pos+1
    cells[pos] = []
    print()

```

```
print()
print()
print('Mitosis N°',num_mit)
print()
print('Division of cell N°', cell_num)
print()
print('from division of cell N°', cell_num, 'cell N° 0 remains and the new cell N°', pos,' arises')
print()

# 'First cell' is an invariant "immortal" stem cell that maintains its original sat_DNA

if cell_num not in dividing_cells:
    dividing_cells.append(cell_num)
num_mit +=1

#Second cell

cells[pos] = deepcopy(cells[cell_num]) # 'second cell' copies the sat_DNA from stem cell
cells[pos][0][4] = 'C' # 'second cell' erases hydroxymethylation in the first sublist
dividing_cells.append(pos) # 'second cell' will divide
arisen_cells.append(pos)

print('cell N° 0 satDNA:\n',cells[0])
print()
print('cell N°',pos,'satDNA:\n',cells[pos])
print()
print()
if pos == 30:
    existing_cells = dividing_cells + not_div_cells
    existing_cells.sort()
    print('existing_cells', existing_cells[:31])
    print('dividing_cells',dividing_cells )
    print('not_div_cells', not_div_cells)
    print()
    Stop = 1
    return

asymm_mitosis(0)
```

```
# RUNNING THE PROGRAM
```

```
while pos < 30:
```

```

if Stop == 0 and dividing_cells[0]<15:
    a = dividing_cells.pop(0)
    symm_mitosis(a)
else:
    break
if sat_DNA[0][4] == 'H':
    continue

```

LINEAGE GRAPHICS

```

print('cell lineage:')
print()
for i in range(0,1):
    print(14*' ','_'*22,'0','_'*23)
for i in range(1,2):

    if i in arisen_cells:
        print(' '*13,i,' '*47,end=")
    else:
        print(' '*60,end =")
for i in range(2,3):
    if i in arisen_cells:
        print(i)
    else:
        print(' '*20)
for i in range(3,4):
    if i in arisen_cells:
        print(' '*5,i,'_'*5,'|','_'*5,end=")
    else:
        print(' '*18,end=")
for i in range(4,5):
    if i in arisen_cells:
        print(",i,' '*25,end=")
    else:
        print(' '*31,end=")
for i in range(5,6):
    if i in arisen_cells:
        print(' '*5,i,'_'*5,'|','_'*5,end=")
    else:
        print(' '*19,end=")
for i in range(6,7):
    if i in arisen_cells:
        print(",i,' '*4)
    else:

```

```
        print("")
for i in range(7,8):
    if i in arisen_cells:
        print(' ',i,' ','|','_'*1,end=")
    else:
        print(' '*5,end=")
for i in range(8,9):
    if i in arisen_cells:
        print(" ",i,' '*4,end=")
    else:
        print(' '*11,end=")
for i in range(9,10):
    if i in arisen_cells:
        print(' ',i,' ','|','_'*1,end=")
    else:
        print(",end=")
for i in range(10,11):
    if i in arisen_cells:
        print(" ",i,' '*19,end=")
    else:
        print(' '*32,end=")
for i in range(11,12):
    if i in arisen_cells:
        print(' ',i,' ','|','_'*1,end=")
    else:
        print(' '*6,end=")
for i in range(12,13):
    if i in arisen_cells:
        print(i,' '*3,end=")
    else:
        print(' '*10,end=")
for i in range(13,14):
    if i in arisen_cells:
        print(' ',i,' ','|','_'*1,end=")
    else:
        print(",end=")
for i in range(14,15):
    if i in arisen_cells:
        print(i)
    else:
        print(' ')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 in arisen_cells and 23 in arisen_cells and 25 in arisen_cells and\
```



```

27 in arisen_cells and 29 in arisen_cells:
    print(' ','|','*3','|','*5','|','*6','|','*21','|','*5','|',\
          '*5','|','*5','|')
    print('15-16','|','17-18','|','19-20','|','21-22','|','*18','23-24',\
          '|','25-26','|','27-28','|','29-30')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 in arisen_cells and 23 not in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|','*5','|','*6','|')
    print('15-16','|','17-18','|','19-20','|','21-22')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 in arisen_cells and 23 in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|','*5','|','*6','|','*21','|')
    print('15-16','|','17-18','|','19-20','|','21-22','|','*18','23-24')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 in arisen_cells and 23 in arisen_cells and 25 in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|','*5','|','*6','|','*21','|','*5','|')
    print('15-16','|','17-18','|','19-20','|','21-22','|','*18','23-24',\
          '|','25-26')
if 15 in arisen_cells and 17 not in arisen_cells and 19 not in arisen_cells and\
21 not in arisen_cells and 23 not in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|')
    print('15-16')
if 15 in arisen_cells and 17 in arisen_cells and 19 not in arisen_cells and\
21 not in arisen_cells and not 23 in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|')
    print('15-16','|','17-18')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 not in arisen_cells and 23 not in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|','*5','|')
    print('15-16','|','17-18','|','19-20')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 not in arisen_cells and 23 in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|','*5','|','*30','|')
    print('15-16','|','17-18','|','19-20','|','*26','23-24')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not in\
arisen_cells and 21 not in arisen_cells and 23 in arisen_cells \

```

```
and 25 in arisen_cells and 27 in arisen_cells and 29 in arisen_cells:
print(' *48,|', '*5,|', '*5,|', '*5,|')
print(' *46,23-24', ',25-26', ',27-28', ',29-30')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 in arisen_cells \
and 25 in arisen_cells and 27 in arisen_cells and 29 not in arisen_cells:
print(' *50,|', '*5,|', '*5,|')
print(' *48,23-24', ',25-26', ',27-28')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 in arisen_cells \
and 25 in arisen_cells and 27 not in arisen_cells and 29 not in arisen_cells:
print(' *50,|', '*5,|')
print(' *48,23-24', ',25-26')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 not in arisen_cells \
and 25 not in arisen_cells and 27 in arisen_cells and 29 \
not in arisen_cells:
print(' *66,|')
print(' *64,27-28')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 not in arisen_cells \
and 25 not in arisen_cells and 27 not in arisen_cells and 29 \
in arisen_cells:
print(' *74,|')
print(' *72,29-30')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 not in arisen_cells \
and 25 not in arisen_cells and 27 in arisen_cells and 29 \
in arisen_cells:
print(' *66,|', '*5,|')
print(' *64,27-28', ',29-30')
if 15 not in arisen_cells and 17 in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 not in arisen_cells \
and 25 not in arisen_cells and 27 in arisen_cells and 29 \
in arisen_cells:
print(' *9,|', '*54,|', '*5,|')
print(' *7,17-18', '*50,27-28', ',29-30')
if 15 not in arisen_cells and 17 in arisen_cells and 19 not in \
arisen_cells and 21 not in arisen_cells and 23 not in arisen_cells \
and 25 not in arisen_cells and 27 in arisen_cells and 29 \
not in arisen_cells:
print(' *9,|', '*54,|')
print(' *7,17-18', '*50,27-28')
```

```
if 15 not in arisen_cells and 17 in arisen_cells and 19 not \
    in arisen_cells and 21 not in arisen_cells and 23 not in \
    arisen_cells and 25 not in arisen_cells and 27 not in arisen_cells \
    and 29 not in arisen_cells:
    print(' *9,|')
    print(' *7,'17-18')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not \
    in arisen_cells and 21 not in arisen_cells and 23 in \
    arisen_cells and 25 not in arisen_cells and 27 not in arisen_cells \
    and 29 not in arisen_cells:
    print(' *48,|')
    print(' *46,'23-24')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not \
    in arisen_cells and 21 not in arisen_cells and 23 not in \
    arisen_cells and 25 in arisen_cells and 27 not in arisen_cells \
    and 29 not in arisen_cells:
    print(' *57,|')
    print(' *55,'25-26')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 not \
    in arisen_cells and 21 not in arisen_cells and 23 in \
    arisen_cells and 25 not in arisen_cells and 27 not in arisen_cells \
    and 29 in arisen_cells:
    print(' *48,|', ' *20,|')
    print(' *46,'23-24', ' *16,'29-30')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 \
    not in arisen_cells and 21 in arisen_cells and 23 not in \
    arisen_cells and 25 not in arisen_cells and 27 not in arisen_cells \
    and 29 not in arisen_cells:
    print(' *25,|')
    print(' *23,'21-22')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 in \
    arisen_cells and 21 in arisen_cells and 23 not in arisen_cells \
    and 25 not in arisen_cells and 27 not in arisen_cells and 29 not in arisen_cells:
    print(' *17,|', ' *6,|')
    print(' *15,'19-20', ' *2,'21-22')
if 15 not in arisen_cells and 17 not in arisen_cells and 19 in \
    arisen_cells and 21 not in arisen_cells and 23 not in arisen_cells \
    and 25 not in arisen_cells and 27 not in arisen_cells and 29 not in arisen_cells:
    print(' *17,|')
    print(' *15,'19-20')
if 15 not in arisen_cells and 17 in arisen_cells and 19 not in arisen_cells and \
    21 not in arisen_cells and 23 not in arisen_cells and 25 not in arisen_cells and \
    27 not in arisen_cells and 29 in arisen_cells:
```

```
print(' ','*3','|','*62,|')
print(' *5,','17-18 ','*57,29-30')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 in arisen_cells and 23 in arisen_cells and 25 in arisen_cells and\
27 in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*3','|','*5,|','*6,|','*21,|','*5,|\
        '*5,|')
    print('15-16','17-18','19-20','21-22','*18,23-24,\
        ','25-26','27-28')
if 15 in arisen_cells and 17 in arisen_cells and 19 in arisen_cells and\
21 not in arisen_cells and 23 in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 in arisen_cells:
    print(' ','|','*3','|','*5,|','*31,|','*19,\
        '|')
    print('15-16','17-18','19-20','*25,23-24,\
        '*15,29-30')
if 15 in arisen_cells and 17 not in arisen_cells and 19 not in arisen_cells and\
21 in arisen_cells and 23 not in arisen_cells and 25 in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*21,|','*21,|')
    print('15-16 ','*14,21-22 ','*16,\
        '25-26')
if 15 in arisen_cells and 17 not in arisen_cells and 19 not in arisen_cells and\
21 in arisen_cells and 23 not in arisen_cells and 25 not in arisen_cells and\
27 not in arisen_cells and 29 not in arisen_cells:
    print(' ','|','*22,|')
    print('15-16 ','*17,21-22')
```