Journal of Uncertain Systems
Vol.9, No.2, pp.113-123, 2015
Online at: www.jus.org.uk

Algorithmic Information Theory may Explain the Pathogenic Number of DNA Repeats in Myotonic Dystrophy Type 1 (and in Similar Diseases)

Misha Koshelev[1],*, Luc Longpré[2]

[1] *1330 Old Spanish Trail #5303, Houston, TX 77054-1835, USA*

[2] *Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA*

Received 1 January 2014; Revised 1 February 2014

Abstract

Myotonic Dystrophy Type 1 (DM1), the most common form of adult-onset muscular dystrophy, is caused by an abnormal number of repeats of the trinucleotide sequence CTG outside of the protein-coding part of DNA – more than fifty repeats can cause disease. Several other diseases are caused by similar repeats in non-coding regions; in most of these diseases, up to approximately fifty repeats is normal, while pathogenic cases usually have > 50 repetitions. The fact that this level of approximately fifty repeats can be seen in many diseases, with different repeating sequences, indicates that there may be a fundamental explanation for this threshold.

In this paper, we conjecture that such an explanation may come from Algorithmic Information Theory. Crudely speaking, this threshold can be viewed as a measure of the redundancy in healthy DNA; this indirect estimate of the measure of redundancy is in good accordance with a more direct estimate – based on the fact that approximately 1/50 of DNA is directly functionally useful (e.g., protein-coding).

©2015 World Academic Press, UK. All rights reserved.

Keywords: Kolmogorov complexity, DNA repeats, myotonic distrophy

1 Introduction

Medical problem. The most common form of adult-onset muscular dystrophy is Myotonic Dystrophy Type 1 (DM1); see, e.g., [10, 11, 13, 27, 28, 33, 45] and references therein. This disease has a well-established genetic origin: it is caused by an abnormal number of repeats of a trinucleotide sequence Cytosine - Thymine - Guanine (CTG) in a 3' untranslated region, a non-protein coding part of DNA. Specifically, in DM1, this repetition occurs inside the *DMPK* gene (myotonic dystrophy protein kinase). This gene codes for a protein which is important for the function of skeletal muscle.

The *DMPK* gene of a healthy person usually has between 5 and 37 CTG repeats, while people affected by DM1 usually have more than fifty. The severity of the disease and the age of the onset increase when the number of repeats increases:

- patients with a smaller number of repeats exhibit a milder form of the disease, and their symptoms start at a later age;
- on the other hand, patients with a larger number of repeats exhibit a more severe form of the disease, with symptoms starting at an earlier age.

Several other diseases (Table 1) are caused by similar repeats outside of the protein-coding regions of the respective genes; see, e.g., [15, 29, 33, 35, 36] and references therein. For example, an abnormal number of repeats of the same CTG sequence in a non-protein coding section of a different gene *ATXN8OS* causes Spinocerebellar Ataxia Type 8 (SCA8), a neurodegenerative disease. Healthy individuals have 7 - 34 repeats in this genes, while people affected by the disease have one hundred or more repeats.

*Corresponding author.
Emails: misha680@gmail.com (M. Koshelev), longpre@utep.edu (L. Longpré).

Several other diseases are caused by excessive repeats of different trinucleotide sequences in non-protein coding parts of DNA. For example:

- Excessive repeats of the CAG sequence in *PPP2R2B* gene cause Spinocerebellar Ataxia Type 12 (SCA12). Here, 7 - 28 repeats are normal, while 66-78 are pathogenic.
- Repeats of GAA cause Friedreich's Ataxia (FRDA): 7 - 34 repeats are normal, while 100 and more are pathogenic.
- Excessive repeats of CGG in the *FMR1* gene (on the X chromosome) cause Fragile-X Syndrome: 5 - 44 repeats are normal, while 200 and more are pathogenic [14]. Fifty five to 200 CGG repeats in the same location lead to a disease known as Fragile X-associated Tremor/Ataxia Syndrome.

Table 1: Diseases caused by repetitions of three nucleotides outside protein-coding regions

Disease	**Abbreviation**
Myotonic Dystrophy type 1	DM1
Spinocerebellar Ataxia Type 8	SCA8
Spinocerebellar Ataxia Type 12	SCA12
Friedreich's Ataxia	FRDA
Fragile-X Syndrome	FXS
Fragile-X Associated Tremor/Ataxia Syndrome	FXTAS

Disease	**Gene**	**Sequence**	**Normal**	**Disease**
DM1	*DMPK*	CTG	$5 - 35$	> 50
SCA8	*ATXN8OS*	CTG	$15 - 34$	> 89
SCA12	*PPP2R2B*	CAG	$7 - 28$	$66 - 78$
FRDA	*FXN*	GAA	$5 - 30$	> 70
FXS	*FMR1*	CGG	$5 - 44$	> 200
FXTAS	*FMR1*	CGG	$5 - 44$	$55 - 200$

Universal threshold: an observation. By comparing the number of repeats in healthy and pathogenic individuals for different diseases, we can observe that in most diseases caused by repeats in non-protein coding regions of DNA:

- up to approximately fifty repeats is normal, while
- pathogenic cases usually have > 50 repeats.

Natural conjecture. The fact that this threshold of approximately fifty can be seen in many diseases, with different repeating sequences, indicates that there probably is a fundamental explanation for this threshold.

Comment. It is worth mentioning that repeats can occur also in protein-coding parts of genes. Such repeats can also cause diseases [29, 32]: e.g., excessive CAG repeats in different genes can produce different types of Spinocerebellar Ataxia, Huntington's disease, and several other neurological diseases. In most such diseases, the pathogenic effects start at a much lower level of repeats. For example, for Spinocerebellar Ataxia Type 6, 21 repeats can already cause disease.

What we do in this paper. In this paper, we conjecture that such an explanation may come from Algorithmic Information Theory.

Crudely speaking, we show that this threshold can be viewed as a measure of redundancy in healthy DNA. We show that this indirect estimate of the measure of redundancy is in good accordance with a more direct estimate – based on the fact that approximately 1/50 of DNA is protein-coding.

Outline. First, we explain how the biological properties of DNA are naturally related to the notions of Algorithmic Information Theory such as Kolmogorov complexity. For readers who are not very familiar with these notions, we provide basic definitions and explanations. (Readers who are familiar with these notions can skip the corresponding parts.)

After emphasizing the similarity between DNA and Kolmogorov complexity, we explain the differences – and how we can naturally modify the notion of Kolmogorov complexity so that this notion will become more appropriate for describing DNA.

This formalization naturally leads to an explanation of the above threshold.

Comment Some results from this paper were first announced in [19].

2 DNA, Algorithmic Information Theory, and Kolmogorov Complexity: a Brief Reminder

Important information aspect of DNA coding. From the information viewpoint, the most important feature of the DNA molecule is that this single molecule contains a large amount of information about the biological organism. In comparison with the detailed data about all the cells and the molecules within each cell, the DNA provides a compact, highly compressed description.

Enter Algorithmic Information Theory. Since the genetic code is a highly compressed version of the description of a biological organism, in the analysis of DNA it is natural to use techniques developed for describing such highly compressed sequences.

These techniques exist in Algorithmic Information Theory; see, e.g., [21].

Original idea: definition of randomness. Algorithmic Information Theory did not start with the analysis of compression, it started with the analysis of the notion of *randomness* – and compression came as a side effect.

Historically, this research was started by A. N. Kolmogorov who formulated the following problem:

- In traditional *probability theory*, we talk about the probabilities of different outcomes, the probabilities of different sequences of outcomes, the probabilities that a sequence satisfies a given property – but there is no formal way of classifying sequences into "random" and "non-random."
- On the other hand, in practice, we can say that some sequences are clearly not random – for example, we do not expect 100 coin tosses to generate a sequence of all heads. This is a problem which is solved in applied *statistics*: given a sequence of observations, can we check that this sequence is random?

This "gap" between probability theory and applied statistics is unusual, because most techniques of applied statistics are based on solid probability theory foundations. To close this gap between probability theory and statistics, Kolmogorov proposed to supplement probability theory with an additional well-defined notion of a random sequence. He first proposed this idea in his paper [17]. Later, with his student P. Martin-Löf, he formalized this idea into the precise definition of a random sequence. (For a more detailed description of this idea and its applications, see [21].)

Kolmogorov and Martin-Löf started by observing that, in general, if a binary sequence s is regular, e.g., has the form 111... or 010101..., then, according to common sense, this sequence cannot be random. What does it mean that a sequence is regular? It means, e.g., that we can write a short program p that generates all the elements of this sequence. For example, to generate a sequence consisting of 1,000 pairs 01, we can write a simple for-loop:

```
for (i = 0; i < 1000; i++)
  print("01");
```

Vice versa, if a short program can generate a long sequence, this means that this sequence s is regular – and thus, not random.

If a sequence is "random" in the physical sense of this word, then we do not expect a short program to be able to generate all the elements of this sequence. In other words, we expect any program that generates this sequence to not be much shorter than a program that simply prints this sequence bit by bit:

```
print("11001...");
```

This "print-verbatim" program has the same length as the original sequence (even somewhat longer, as we need a print command).

Thus, the difference between regular sequences and random sequences is that

- regular sequences s can be generated by programs which are much shorter than the length $\text{len}(s)$ of the sequence s, while
- random sequences s can only be generated by programs of approximately the same length as the length of the sequence s.

Notion of Kolmogorov complexity. To capture the above difference between "random" and "regular" ("non-random") sequences, Kolmogorov and Martin-Löf defined the *Kolmogorov complexity* $K(s)$ of a finite sequence s as the shortest length of a program that generates the sequence s.

Historical comment. Although the notion of Kolmogorov complexity is traditionally associated with the seminal works of Kolmogorov and Martin-Löf, there are other names associated with this concept that the historically acute reader should know. Specifically, Ray Solomonoff first described the concept now known as Kolmogorov Complexity in 1960 and again in 1964 [39, 40, 38]. Although his work was acknowledged by Kolmogorov [18], it became associated by the scientific community with a slightly different concept of Algorithmic Probability. An Argentine-American mathematician and computer scientist, Gregory Chaitin, also independently invented the same concept [4, 3]. For a more detailed historical discussion of Algorithmic Information Theory, please see [21].

Comment. Strictly speaking, to make the above definition precise, we need to fix a (universal) programming language, i.e., a programming language in which every algorithm can be programmed (like C or Java). However, it turns ot that all the languages are, in some reasonable sense, equivalent: namely, the differences between the values of the Kolmogorov complexity $K_L(s)$ and $K_{L'}(s)$ – corresponding to two different universal programming languages L and L' – are bounded by a constant (depending on the specific language chosen), i.e., $|K_L(s) - K_{L'}(s)| \le C_{L,L'}$ for all strings s.

Resulting definition of randomness. We can always generate a given sequence s by simply printing it bit by bit. We have already mentioned that the length of this program is equal to the length $\text{len}(s)$ of the sequence s plus a small number of bits C needed to describe the print statement. Thus, the Kolmogorov complexity $K(s)$ of an arbitrary sequence s – which is defined as the shortest length of the program that generates s – cannot exceed the length $\text{len}(s) + C$ of this bit-by-bit program: $K(s) \le \text{len}(s) + C$.

For random sequences, no significantly shorter programs are possible. Thus, for a random sequence s, its Kolmogorov complexity $K(s)$ – the length of the shortest program for generating s – cannot be much smaller than its length. This idea is the basis for the definition of the randomness of a binary sequence. Let C be an integer. A finite binary sequenced is called *C-random* if $K(s) \ge \text{len}(s) - C$.

Comment. Several similar definitions of randomness have been proposed, e.g., a more relaxed definition requiring that $K(s) \ge \text{len}(s) - \log_2(\text{len}(s)) - C$.

Relation to compressibility. The notion of Kolmogorov complexity was shown to have interesting consequences for the compressibility of different sequences.

Indeed, if a sequence s is highly compressible, this means that we can compress it into a much shorter sequence c, with $\text{len}(c) \ll \text{len}(s)$, so that a simple decoding program will be able to reconstruct s from c. By combining c and the decoding program, we thus get a short program p that generates s – a program whose length is much shorter than the length of the original sequence s: $\text{len}(p) \ll \text{len}(s)$.

According to Kolmogorov's definition of randomness, a sequence is random if it cannot be generated by a program which is much shorter than the length of this sequence. Thus, if a sequence s is compressible, this sequence is not random. So, if a sequence is random, this means that this sequence cannot be compressed.

On the other hand, if a sequence s is not random, i.e., if it can be generated by a short program p, we can then store this program p instead of the original sequence s – and, if necessary, generate the original sequence s by simply running this program p.

Comment. It should be mentioned that the shortest program p is itself random. Indeed, if we could generate p by running a much shorter program q with $\text{len}(q) \ll \text{len}(p)$, then we would be able to generate x from q as well:

- first by generating the code p, and
- then by running this code to generate x.

The fact that p is the shortest program for generating x means that q cannot be much shorter than p – i.e., that the Kolmogorov complexity $K(p)$ must be approximately equal to the length $\text{len}(p)$ of this sequence p.

By definition, sequences for which $K(p) \approx \text{len}(p)$ are called random. Thus, p is indeed a random sequence.

Back to DNA: in the first approximation, DNA is random. We started this section by observing that DNA is the result of a very drastic compression of information. Moreover, it is known that the DNA sequence itself is very difficult to compress: standard general-purpose compression algorithms, algorithms that do wonders in compressing images, music, movies, etc., practically cannot compress DNA at all [5, 7, 8]. Special compression algorithms had to be designed to enable a significant degree of compression; see, e.g., [5].

Thus, in the first reasonable approximation, we can say that the DNA sequences are incompressible (=random in the sense of Kolmogorov-Martin-Löf).

Of course, this is not exactly true: DNA sequences can be (somewhat) compressed [5], and, repeats (like the ones that cause diseases) are examples of pieces of the DNA code that can be drastically compressed. However, this approximate treatment of DNA sequences as random (incompressible) has been successfully used: e.g., it has led to the design of efficient algorithms for DNA sequencing; see, e.g., [16]. Other methods of analysis of DNA have also found that DNA is random – specifically, in the case of a recent paper in *Science*, that DNA does not exhibit much sequential structure as compared to several human languages (it has high "conditional entropy") [34].

In reality, DNA is not random, there is redundancy. As we have mentioned, in practice, DNA is not exactly random (= incompressible): there is redundancy in DNA.

This redundancy makes perfect biological sense: we do not want a single mutation to destroy an organism, so redundancy is very beneficial. For the same reason, redundancy is built into existing control and communication systems, to increase reliability, and to ensure that the disruption of a single communication tower does not incapacitate all communications.

The existing Algorithmic Information Theory approaches to DNA redundancy. The redundancy in the DNA sequences is well understood, and several papers use notions related to Kolmogorov complexity to analyze this redundancy.

Detecting functionally important regions and coding regions. From the biological viewpoint, the more functionally important the region, the more redundancy should be added to it – and thus, the more the information stored in this region can be compressed. Thus, by analyzing the complexity of different regions, we can find functionally important regions as regions where the possible degree of compression is higher – or, equivalently, where the Kolmogorov complexity is lower; see, e.g., [6, 9, 42, 44].

On the other hand, since one of the main purposes of a non-coding region is to serve as an additional redundancy tool, the Kolmogorov complexity of such regions should be smaller than the Kolmogorov complexity of similar-size coding regions [43]. Thus, by analyzing the complexity of different regions, we can also distinguish between coding and non-coding regions.

Comment. It should be mentioned that Kolmogorov complexity itself is not computable, so in practical applications, we must use estimates of Kolmogorov complexity based on existing compression algorithms.

Tools that use such estimates have indeed been proposed and successfully used in the analysis of DNA sequences; see, e.g., [31].

Applications to classification of species. The notion of Kolmogorov complexity has been used to classify the genomes of different biological organisms into evolutionary trees; see e.g., [5, 22]. This classification is based on the following natural idea:

- For two identical organisms, with identical genetic sequences s and s', the Kolmogorov complexity of the joint sequence ss' is practically equal to the complexity of each individual sequence: once we can generate the sequence s, repeating it is easy.
- On the other hand, if the organisms are drastically different, then generating s will not help us generate s'. In effect, the easiest way to generate ss' would be to generate s then s'. In this case, the Kolmogorov complexity of ss' is equal to the sum of the corresponding Kolmogorov complexities.

In general, the larger the difference $K(ss') - K(s)$, the less the organisms are related. Thus, this difference (and ratios related to this difference) can serve as a measure of distance between the sequences s and s'. This measure of distance can be successfully used to reconstruct an evolutionary tree.

Other applications of Kolmogorov complexity to the analysis of DNA sequences are described in [20].

The sequence of DNA is "almost" incompressible: in what sense? We have mentioned that the DNA sequence is almost incompressible. It would be nice to describe this "approximate incompressibility" in precise terms.

This problem is formulated in [21], where a "first-approximation" definition of such approximate incompressibility is given – as, crudely speaking, the possibility to be compressed by no more than a constant number of bits. The authors of [21] themselves emphasize that this definition is not yet fully satisfactory.

One reason why this definition is not completely sufficient is because redundancy is usually achieved by repeating the original information multiple times, rather than by adding a few control bits. In this case, we can compress not by a few bits, but a constant number of times.

What we plan to do. In this paper, we give a new, more adequate definition of this redundancy-related "approximate incompressibility," and we show how this definition enables us to explain the pathogenic threshold for the number of repeats.

3 Towards a New Definition of Redundancy-Related "Approximate" Incompressibility

Motivations. Let us assume that we want to maintain the level of redundancy r, i.e., crudely speaking, that every important piece of information should be stored in r different places.

Of course, from the biological viewpoint, this assumption is an approximation: in reality, as we have mentioned earlier, we may want to provide

- more redundancy for more important information and
- less redundancy for less important information.

However, we believe that this assumption of a uniform level of redundancy can serve as a reasonable first approximation.

Under the above assumption, the whole sequence can be compressed into a sequence that is r times shorter – but not any shorter. If some pieces of information are stored with less redundancy, then the compression ratio – the ratio $\text{len}(s)/\text{len}(c)$ of the length $\text{len}(s)$ of the original sequence s to the length $\text{len}(c)$ of the compressed sequence c – will be smaller than r.

If we take a subsequence of this sequence, then for this subsequence we may not be able to attain the same level of compression – since this subsequence may include only some of the r copies. But again, we should not expect a compression ratio higher than r.

Let us use this conclusion as a definition of approximate incompressibility. In this formal definition, we use the fact (explained in the previous section) that the shortest length $\text{len}(c)$ of the compressed sequence c is, in effect, equal to its Kolmogorov complexity : $\text{len}(c) \approx K(s)$. Thus, the compression ratio $\text{len}(s)/\text{len}(c)$

is, in effect, equal to $\text{len}(s)/K(s)$. So, the requirement that this ratio does not exceed the given constant r means that $\text{len}(s)/K(s) \le r$, i.e., equivalently, that

$$K(s) \ge \frac{1}{r} \cdot \text{len}(s).$$

To be more precise, since Kolmogorov complexity is only defined modulo an additive constant, we should add a constant C to this definition – just like a similar constant is used in the definition of Kolmogorov complexity.

To formulate the resulting definition, we will need a notation for subsequences.

Notation. Let s be a given sequence, and let $i \le j$ be such that $1 \le i \le j \le \text{len}(s)$. Then, by $s_{i,j}$, we will denote a subsequence of the sequence s that starts at the i-th symbol of s and ends at the j-th symbol of the sequence s.

Comment. In particular, the sequence s itself is its own subsequence, with $i = 1$ and $j = \text{len}(s)$: $s = s_{1,\text{len}(s)}$.

Now, we are ready to formulate the definition.

Definition. *Let $r > 1$ and $C > 0$ be real numbers. We say that a sequence s is C-*incompressible with redundancy *r if, for all $i \le j$, we have*

$$K(s_{i,j}) \ge \frac{1}{r} \cdot \text{len}(s_{i,j}) - C.$$

Comments.

- Similarly to the notion of algorithmic randomness, we can alternatively require that, e.g.,

$$K(s_{i,j}) \ge \frac{1}{r} \cdot \text{len}(s_{i,j}) - \log_2(\text{len}(s_{i,j})) - C.$$

  The main idea is that compression by a factor more than r is not possible.

- Our new definition is related to the notion of an *ε-random sequence* defined as an infinite sequence $\omega = \omega_1\omega_2 \ldots \omega_n \ldots$ for which, for almost all n, we have $K(\omega_1 \ldots \omega_n) \ge \varepsilon \cdot n - O(1)$; see, e.g., [2, 24, 25, 37, 41].

Mathematical comment. The reader may have noticed that:

- In the original definition of a random sequence, we only required that the appropriate inequality hold for the sequence s.
- In contrast, in the new definition, in addition to this requirement, we also require that this inequality holds for all subsequences of s as well.

The explanation for this difference is simple. In Kolmogorov's original definition, we can use the fact that:

- if a sequence is incompressible,
- then all its subsequences are incompressible as well.

Indeed,

- if we were able to compress a subsequence, i.e., replace this subsequence with a shorter description,
- then we would be able to replace, in the original sequence s, this subsequence with a compressed code – and thus, get a compression of the sequence as a whole (in contradiction to the incompressibility assumption).

In contrast, in compression with redundancy, the corresponding incompressibility of a sequence does not imply incompressibility of all its subsequences. As an example, let us take a long incompressible sequence s_0 followed by a sequence s_1 of 0s of the same length. The resulting sequence $s = s_0 s_1$ can be compressed at most twice, since no matter how we compress s, we still need to describe the original incompressible sequence s_0 of size $\text{len}(s)/2$ – and since it is incompressible, it requires at least $\text{len}(s)/2$ bits. So, for the sequence as a whole, the compression ratio cannot exceed 2.

On the other hand, the right half s_1 of the sequence s – consisting of all 0s – can be compressed into a very small program p, with $\text{len}(p) = K(s_1) \ll \text{len}(s_1)$, with a huge compression ratio $\text{len}(s)/K(s) \gg 1$ (much larger than 2).

Biological comment. Since redundancy is so important, why not simply use maximal number of repetitions for each piece of information? The reason is that the replication of an extra piece of DNA wastes energy, so we have to balance

- the advantage of storing this extra information vs.
- the disadvantage of wasting energy on its storage and replication.

4 Applications to DNA

What is the level of redundancy of DNA? To apply the above definition to the genetic code it is necessary to estimate the level of redundancy of DNA r.

This level of redundancy can be estimated as follows. In the human genome (and in the genomes of other organisms for which the genetic code has been sequenced), we know which regions of the DNA correspond to protein coding, RNA genes, and regulatory genes. In the human genome, these regions form approximately 2% of the entire DNA; see, e.g., [23].

As we have mentioned, these regions are almost impossible to compress [5]. So, we can conclude that the incompressible part of the DNA is about $2\% = 0.02$ of its original size – i.e., that level of redundancy is $r = 1/0.02 \approx 50$.

This level of redundancy can explain the pathogenic threshold of DNA repeats. According to the above definition,

- not only the DNA itself cannot be compressed by more than fifty times, but also
- every subsequence of the DNA cannot be compressed with a better compression ratio.

Let us apply this conclusion to the sequence of repeats. We have shown earlier in this paper that a long sequence of repeats can be compressed, in effect, into code the size of the length of the repeated sequence. Thus, the compression ratio for such sequences is approximately equal to the number of repeats.

Since we assume that the compression ratio for this subsequence of repeats cannot exceed fifty, we thus conclude that the number of repeats cannot exceed this same value. This explains why in individuals with healthy DNA, the number of repeats usually does not exceed fifty.

If this conclusion is correct, this means that all the DNA replication processes are adjusted to work well on DNA sequences in which this property – that number of repeats never exceed fifty – is satisfied. Not surprisingly, once this property is not satisfied, we may encounter pathogenic consequences – since the replication schemes are simply not designed to handle such excessive repeats, repeats which (in this number) never occur in normal genomes.

Medical comment. In this paper, we considered the effect of repeats in the non-protein coding part of the DNA. This part of the DNA, by definition, should not directly affect the sequence of synthesized proteins. Its only negative effect is when it gets erroneously involved in protein synthesis. From this viewpoint, minor changes in this part of the DNA should not lead to any diseases – as long as this part is still recognized as non-protein coding.

In contrast, changes and mutations in the protein-coding part of the DNA directly affect protein synthesis. As as result, even minor mutations in this part of the DNA can lead to drastic changes and, potentially, to diseases. This difference explains the above fact that

- relatively short trinucleotide repeats (e.g., repeats of size 20) in the protein-coding part of the DNA can cause severe diseases,
- in contrast to the case of non-protein coding repeats, where 20 repeats never cause any problems.

Biological comment. There are good reasons to believe that a large portion of the non-protein coding DNA is indeed there for the purpose of redundancy.

Indeed, on the one hand, it is known that the large part of this DNA does not seem to perform any function which is not also performed by the coding part of the DNA. Indeed, a recent well-publicized paper [30] showed that eliminating a significant part of the mouse genome does not affect the viability of mice.

Since most of these portions of DNA do not have any additional functions, this means that these DNA portions:

- either serve no useful function at all,
- or simply duplicate existing functions.

While some portions of the DNA may be indeed non-functional, there are evolutionary arguments against assuming that most non-coding DNA is not functional:

- first, as we have mentioned, reproduction of non-functional DNA wastes energy, so evolution should delete such non-functional portions;
- second, since non-functional pieces of DNA do not affect the organism's ability to survive, they keep and inherit all possible mutations; in contrast, in the functional part, mutations often decrease the survivability, so they are selected against.

As a result, evolutionary, the mutation rate in non-functional portions of the DNA must be much higher than in the functional portions. However, comparative genomics data shows that many non-coding portion have the same slower rate of evolution as the coding portions.

- This fact was directly shown in [1].
- This fact was also indirectly shown in [26], on the example of the comparison between human and mouse genomes: for many regions DNA (in particular, for several regions outside the 2% that codes for protein), the rate of change is much smaller than would have been seen according to random mutations in non-functional DNA segments.

All this shows that the non-coding portions are also functional.

5 Conclusions

Several genetic diseases, including Myotonic Dystrophy Type 1 (DM1), the most common type of muscular dystrophy in adults, are caused by an abnormal number of repeats of a trinucleotide sequence in a non-protein coding part of DNA. The more excess repeats, the more severe the disease. It turns out that there exists an empirical threshold of fifty that covers all these diseases: in each case,

- normal persons have less than fifty repeats, while
- persons with diseases usually have more than fifty repeats.

In this paper, we propose a possible explanation for this universal threshold: that DNA usually has a level of redundancy 50, and abnormal genes with a higher level of redundancy may therefore be misinterpreted during DNA replication.

This assumed level of redundancy is in perfect accordance with the DNA as a whole: it is known that, e.g., in the human genome, most functions are performed by 2% of the DNA, and that this portion of the DNA is practically incompressible. Thus, the DNA as a whole also has a level of redundancy ≈ 50.

Acknowledgments

The authors are thankful to Jöran Mielke for important references. One of the authors (MVK) is thankful to all the members of the Human Neuroimaging Lab for valuable discussions.

References

[1] Bejerano, G., Pheasant, M., Makunin, I., Stephen, S., Kent, W.J., Mattick, J.S., and D. Haussler, Ultraconserved elements in the human genome, *Science*, vol.304, pp.1321–1325, 2004.

[2] Calude, C.S., Staiger, L., and S.A. Terwijn, On partial randomness, *Annals of Pure and Applied Logic*, vol.138, pp.20–30, 2006.

[3] Chaitin, G.J., On the length of programs for computing finite binary sequences, *Journal of the ACM*, vol.13, no.4, pp.547–569, 1966.

[4] Chaitin, G.J., On the length of programs for computing finite binary sequences: statistical considerations, *Journal of the ACM*, vol.16, no.1, pp.145–159, 1969.

[5] Chen, X., Kwong, S., and M. Li, A compression algorithm for DNA sequences and its applications in genome comparison, *Proceedings of the 4th ACM Annual Conference on Computational Molecular Biology*, pp.107–117, 2000.

[6] Chuzhanova, N.A., Anassis, E.J., Ball, E., Krawczak, M., and D.N. Cooper, Meta-analysis of indels causing human genetic disease: mechanics of mutagenesis and the role of local DNA sequence complexity, *Human Mutation*, vol.21, pp.28–44, 2003.

[7] Curnow, R., and T. Kirkwood, Statistical analysis of deoxyribonucleic acid sequence data – a review, *Journal of Royal Statistical Society*, vol.152, pp.199–220, 1989.

[8] Grumbach, S., and F. Tahi, A new challenge for compression algorithms: genetic sequences, *Journal of Information Processing and Management*, vol.30, no.6, pp.875–886, 1994.

[9] Hancock, J.M., Genome size and the accumulation of simple sequence repeats: implications of new data from genome sequencing projects, *Genetica*, vol.115, pp.93–103, 2002.

[10] Harper, P.S., *Myotonic Dystrophy: The Facts*, Oxford University Press, Oxford, 2002.

[11] Harper, P.S., Engelen, B.V., Eymard, B., and D. Wilcox, *Myotonic Dystrophy: Present Management, Future Therapy*, Oxford University Press, Oxford, 2004.

[12] Ikeda, Y., Dalton, J.C., Day, J.W., and L.P.W. Ranum, Spinocerebellar ataxia type 8 (SCA 8), http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=gene&part=sca8.

[13] International Myotonic Dystrophy Organization, Information, http://www.myotonicdystrophy.com.

[14] Jacquemont, S., Hagerman, R., Hagerman, P., and M. Leehey, Fragile-X syndrome and fragile X-associated tremor/ataxia syndrome: two faces of FMR1, *Lancet Neurology*, vol.6, no.1, pp.45–55, 2007.

[15] Jasinska, A., Michlewski, G., de Mezer, M., Sobczak, K., Kozlowski, P., Napierala, M., and W.J. Krzyzosiak, Structures of trinucleotide repeats in human transcripts and their functional implications, *Nucleic Acids Research*, vol.31, no.19, pp.5463–5468, 2003.

[16] Keceioglu, J., Li, M., and J. Tromp, Inferring a DNA sequence from erroneous copies, *Theoretical Computer Science*, vol.185, pp.3–13, 1997.

[17] Kolmogorov, A.N., Three approaches to definition of information quantity, *Problems of Information Transfer*, no.1, pp.3–11, 1965.

[18] Kolmogorov, A.N., Logical basis for information theory and probability theory, *IEEE Transactions on Information Theory*, vol.14, no.5, pp.662–664, 1968.

[19] Koshelev, M., and L. Longprë, Algorithmic information theory may explain the pathogenic number of DNA repeats in Myotonic Dystrophy type 1 (and in similar diseases), *ACM SIGACT News*, vol.41, no.4, pp.61–64, 2010.

[20] Li, W., The measure of compositional heterogeneity in DNA sequences is related to measures of complexity, *Complexity*, vol.3, no.2, pp.33–37, 1997.

[21] Li, M., and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, 2008.

[22] Liu, J., and D. Li, Conditional LZ complexity of DNA sequence analysis and its application in phylogenetic tree reconstruction, *Proceedings of the IEEE International Conference on BioMedical Engineering and Informatics*, pp.111–116, 2008.

[23] Lodish, H., Berk, A., Kaiser, C.A., Krieger, M., Scott, M.P., Bretscher, A., Ploegh, H., and P. Matsudaira, *Molecular Cell Biology*, W. H. Freeman, 2007.

[24] Mielke, J., Refined bound on Kolmogorov complexity for ω-languages, *Electronic Notes in Theoretical Computer Science*, vol.221, pp.181–189, 2008.

[25] Mielke, J., and L. Staiger, On oscillation-free ε-random sequences II, *Proceedings of the Sixth International Conference on Computability and Complexity in Analysis*, pp.183–193, 2009.

[26] Mouse Genome Sequencing Consortium, Initial sequencing and comparative analysis of the human genome, *Nature*, vol.420, no.6915, pp.520–562, 2002.

[27] Myotonic Dystrophy Foundation, Disease information,
http://www.myotonic.com/go/mdf/disease-information/absout-the-disease.

[28] National Institutes of Health (NIH), *Myotonic Dystrophy: Overview of Condition*,
http://ghr.nlm.nih.gov/condition=myotonicdystrophy.

[29] National Institutes of Health (NIH), *Trinucleotide Repeat Expansion*,
http://www.nlm.nih.gov/cgi/mesh/2009/MB_cgi?mode=&term=Trinucleotide+Repeat+Expansion.

[30] Nobrega, M.A., Zhu, Y., Playzer-Frick, I., Afzal, V., and E.M. Rubin, Megabase deletions of gene deserts result in viable mice, *Nature*, vol.431, pp.988–993, 2004.

[31] Orlov, Y.L., and V.N. Potapov, Complexity: an internet resource for analysis of DNA sequence complexity, *Nucleic Acids Research*, vol.32, pp.W628–W633, 2004.

[32] Orr, H.T., and H.Y. Zoghbi, Trinucleotide repeat disorders, *Annual Review of Neuroscience*, vol.30, pp.575–621, 2007.

[33] Ranum, L.P.W., and T.A. Cooper, RNA-mediated neuromuscular disorders, *Annual Review of Neuroscience*, vol.29, pp.259–277, 2006.

[34] Rao, R.P.N., Yadav, N., Vahia, M.N., Joglekar, H., Adhikari, R., and I. Mahadevan, Entropic evidence for linguistic structure in the indus script, *Science*, vol.324, no.5931, p.1165, 2009.

[35] Richards, R.I., and G.R. Sutherland, Dynamic mutation: possible mechanisms and significance in human disease, *Trends in Biochemical Sciences*, vol.22, no.11, pp.432–436, 1997.

[36] Rozanska, M., Sobczak, K., Jasinska, A., Napierala, M., Kaczynska, D., Czerny, A., Koziel, M., Kozlowski, P., Olejniczak, M., and W.J. Krzyzosiak, CAG and CTG repeat polymorphism in exons of human genes shows distinct features at the expandable loci, *Human Mutation*, vol.28, no.5, pp.451–458, 2007.

[37] Ryabko, B.Y., Coding of combinatorial sources and Hausdorff dimension, *Soviet Mathematics Doklady*, vol.30, pp.219–222, 1984.

[38] Solomonoff, R., A preliminary report on a general theory of inductive inference, Report V-131, Zator Co., Cambridge, Ma. February 4, 1960.

[39] Solomonoff, R., A formal theory of inductive inference, *Information and Control*, Part I, vol.7, no.1, pp.1–22, 1964.

[40] Solomonoff, R., A formal theory of inductive inference, *Information and Control*, Part II, vol.7, no.2, pp.224–254, 1964.

[41] Staiger, L., On oscillation-free ε-random sequences, *Electronic Notes in Theoretical Computer Science*, vol.221, pp.287–297, 2008.

[42] Stern, L., Allison, L., Coppel, R.L., and T.I. Dix, Discovering patterns in plasmodium falciparum genomic DNA, *Molecular Biochemistry and Parasitology*, vol.118, pp.175–186, 2001.

[43] Troyanskaya, O.G., Arbell, O., Koren, Y., Landau, G.M., and A. Bolshoy, Sequence complexity profiles of prokaryotic genomic sequences: a fast algorithm for calculating linguistic complexity, *Bioinformatics*, vol.18, pp.679–688, 2002.

[44] Wan, H., Li, L., Federhen, S., and J. Wootton, Discovering simple regions in biological sequences associated with scoring schemes, *Journal of Computational Biology*, vol.10, pp.171–185, 2003.

[45] Wells, R.D., and T. Ashizawa, *Genetic Instabilities and Neurological Diseases*, Academic Press, Boston, Massachusetts, 2006.