Possibilistic channels for DNA word design

Luca Bortolussi¹ and Andrea Sgarro²

¹ Dept. of Mathematics and Informatics, University of Udine, Udine, Italia
luca.bortolussi@dimi.uniud.it

² Dept. of Mathematics and Informatics, University of Trieste, and Consorzio Biomedicina Molecolare, Area Science Park, Trieste, Italia
sgarro@units.it

Summary. We deal with DNA combinatorial code constructions, as found in the literature, taking the point of view of possibilistic information theory and possibilistic coding theory. The possibilistic framework allows one to tackle an intriguing information-theoretic question: what is channel noise in molecular computation? We examine in detail two representative DNA string distances used for DNA code constructions and point out the merits of the first and the demerits of the second. The two string distances are based on the reverse Hamming distance as required to account for hybridisation of DNA strings.

Keywords: possibility theory, possibilistic information theory, channel coding, DNA word design, reverse Hamming distance.

1 Introduction

Assume that an input word is chosen out of a list, or *codebook*, and is sent through a *noisy communication channel*; a distorted version of the input word is received at the output end of the channel. Several rational behaviour can be envisaged, according to the context. The observer can decode to the input word whose *diversity* or *distortion* from the output is minimum; or: the observer of the output can *decode* to the input word whose *similarity* to the output is maximum; or: the observer can decode to the input word such that the *likelihood* of having observed the output is maximum; or: the observer can decode to the input word such that the *possibility* of having observed right that output is maximum. All these approaches, even if maximum likelihood is definitely less powerful, can be conveniently lodged under the shed of *possibilistic information theory* and *possibilistic coding theory*, as were put forward in [11], [12], [13], [14]. So far, this theoretical framework had been applied to devising error-correcting telephone keyboards and to the problem of ensuring safety in a telephone network with sensitive users [8], [2]; below, we show that it can consistently and conveniently lodge “unusual” forms of coding, as is *DNA word design* (for which cf. e.g. [4], [9], [10]), and it allows one to answer an intriguing information-theoretic question of a subtle theoretical nature: what is *noise* in molecular computation, or more specifically in DNA word design?

Below, Sections 2 and 3 deal with possibilistic coding up to the formulation of our main problem: what is channel noise in DNA word design? In Section 4 we deal with two representative cases, i.e. DNA code constructions based on checking the reverse Hamming distance between codewords, and those based on checking *both* Hamming distances, direct and reverse. Only this “compound” DNA string distance passes the “possibilistic control” (theorem 1), and allows a satisfactory definition of what channel noise may be in biological computation, while pure reverse Hamming distance fails (theorem 2). The last section contains a short biological reminder on DNA word design and a few concluding remarks. In this paper we consider only “crisp” codes, even if from the “soft” point of view of possibility theory; fuzzy DNA codebooks and their biological adequacy have been discussed in [3].

2 Channel models

An input word-set $\mathcal{X}$ and an output word-set $\mathcal{Z}$ are given. A *codebook* $\mathcal{C}$, or simply a *code*, is a subset of the input set, whose elements are called *codewords*; $\mathcal{C} \subset \mathcal{X}$. The channel is specified through a *matrix* with rows headed to $\mathcal{X}$ and columns headed to $\mathcal{Z}$, and whose entries $\alpha(x, z)$ are non-negative real numbers. In this paper we shall cover explicitly two³ cases:

Possibilistic matrix (the absolute maximum in each row of the matrix is 1): its elements are *transition possibilities* from inputs to outputs.

Distortion matrix (the absolute minimum in each row of the matrix is 0): its elements specify the *distortion* between input and output.

The *decoding strategies* are: in the case of transition possibilities decode to the input codeword $\in \mathcal{C}$ which maximises the value in the matrix column headed to the output received; instead, one minimises the matrix value in the case of distortions. So, the first decoding strategy is a *maximising strategy*, while the second is a *minimising strategy*. The underlying implicit assumption is that the smaller the possibility (the higher the distortion, respectively), the less “likely” it is to occur during channel transmission, and this in a very uncommittal sense of the word “likely”, cf. [11], [13], [14]. In case of ties one declares a *detected error*; unfortunately, one can incur also into *undetected errors*, even if the possibilities of such events should be small for a good code.

Examples. In “standard” coding theory, one decodes by minimising Hamming distance, that is the number of positions in which the input sequence and the output sequence differ (both sequences are assumed to have the same length). The corresponding transition possibility is the frequency of positions where input and output

³ We apologise for this redundancy: we might have stucked to possibilities only, as in [11], but we are confident that this redundancy will make the paper easily readable a larger audience, inclusive of coding theorists and computational biologists.

sequences coincide; cf. below. In Shannon theory, instead, one minimises transition (conditional) probabilities, as arranged in a stochastic matrix. In [11], where the theoretical and practical bearing of the possibilistic approach to coding is discussed and vindicated, one mimicks the Shannon-theoretic approach, after replacing transition probabilities by transition possibilities (probabilistic channels and probabilistic noise by possibilistic channels and possibilistic noise). An even more general possibilistic framework would be a “bayesian-like” generalisation: assume one has a possibility vector on the input set $\mathcal{X}$, and a transition possibility matrix from $\mathcal{X}$ to $\mathcal{Z}$. Form a *joint* possibilistic matrix by taking minima between prior possibilities and the corresponding transition possibilities: then one decodes to codewords which must have both a large prior possibility and a large transition possibility; if the prior possibility is too low, a large transition possibility by itself will not always do. In this case the matrix would be normal, but not all of its rows (“normal” means that the maximum entry is 1, cf. e.g. [6]).

Actually, what matters in these matrices is not the actual numeric values, but rather their mutual order. More specifically, two possibilistic matrices (two distortion matrices, respectively) are *equivalent* if there is a strictly increasing one-to-one correspondence $\beta(x, z) = f(\alpha(x, z))$, between their entries $\alpha(x, z)$ and $\beta(x, z)$; a possibilistic matrix and a distortion matrix are equivalent when the one-to-one correspondence f is strictly *decreasing*.

Proposition. *Take a codebook $\mathcal{C}$. Decode according to two equivalent matrices. The first decoder makes a detected decoding error, or, respectively, an undetected decoding error, iff the second decoder makes an error, detected or undetected, respectively.*

(Proofs are omitted in this preliminary version, save for Theorem 2). A distortion matrix can be soon converted to an equivalent possibilistic matrix. In practice, we shall have to deal only with words which are strings of the same length n , with the distortion ranging from 0 to n in each row, and so we shall *always consider jointly* a distortion matrix $d(x, z)$ and the possibilistic matrix defined by the transition possibilities:

$$\pi(z|x) = 1 - \frac{d(x, z)}{n}$$

In the case of decoding by maximum possibility we are ready to incur only into decoding errors which correspond to negligible possibilities π ; if the decoding strategy is minimum distortion, we are ready to incur only into decoding errors which correspond to exorbitant distortions $d = n(1 - \pi)$.

3 Optimal code constructions for possibilistic channels

The *confusability* and the *distinguishability* between two input sequences x and y are defined, respectively, as:

$$\gamma(x, y) = \max_z \pi(z|x) \wedge \pi(z|y), \quad \delta(x, y) = \min_z d(x, z) \vee d(y, z)$$

The *maximum confusability* $\gamma_{\mathcal{C}}$ of the code $\mathcal{C}$ and its *minimum distinguishability* $\delta_{\mathcal{C}}$ are the the maximum confusability and the minimum distinguishability between any two distinct codewords, respectively. Of course:

$$\gamma(x, y) = 1 - n^{-1}\delta(x, y), \quad \gamma_{\mathcal{C}} = 1 - n^{-1}\delta_{\mathcal{C}}$$

The operational meanings of $\delta_{\mathcal{C}}$ and of $\gamma_{\mathcal{C}}$ are given by the following *reliability criterion*:

Reliability criterion ([13], [14]). *The maximum confusability $\gamma_{\mathcal{C}}$ is the highest possibility which is not always corrected when decoding by maximum possibility, while possibilities $> \gamma_{\mathcal{C}}$ are always corrected. Equivalently: the minimum distinguishability $\delta_{\mathcal{C}}$ is the lowest distortion which is not always corrected when decoding by minimum diversity, while distortions $< \delta_{\mathcal{C}}$ are always corrected.*

Optimal code constructions. The classical *optimisation problem* of channel coding is maximising its size subject to a specified reliability constraint, which in our case is that all “large” possibilities, or all “small” distortions, respectively, should be properly⁴ corrected.

Maximising the code size is the same as maximising its *transmission rate* $n^{-1} \log |\mathcal{C}|$. We stress that the constraints with respect to which one optimises should be expressed in terms of confusabilities, or distinguishabilities, and *not* in terms of transition possibilities or distortions, as the reliability criterion makes it clear.

In the case when the distortion is the *Hamming distance* $d_H(x, y)$, the distinguishability and the confusability are soon found to be:

$$\delta_H(x, y) = \left\lceil \frac{d_H(x, y)}{2} \right\rceil, \quad \gamma_H(x, y) = n^{-1} \left\lfloor n \frac{1 + \pi(y|x)}{2} \right\rfloor$$

and so they are (weakly) *increasing* functions of the distortion and of the transition possibility, respectively. Given this monotonic dependence, one can construct reliable codes as one does in algebraic coding, that is with respect to reliability constraints expressed in terms of the minimum Hamming distance between distinct codewords, rather than constraints on confusabilities or distinguishabilities, as one may and should do in full generality. In other words, the Hamming distance acts also as a very convenient “pseudo-distinguishability” between codewords⁵ and

⁴ Due to the maxitive nature of possibilities, this is the same as constraining *decoding error possibilities*, exactly as one checks decoding error probabilities in Shannon theory, cf. [11].

⁵ Since the monotonicity is only weak, by checking distances rather than distinguishabilities one ends up solving a more general combinatorial problem, and one found also “spurious” solutions. Such spurious solutions are made good use of for error detection rather than error correction, but we cannot deepen this point here; cf. [14].

not only as a distortion between input sequence and output sequence. In the literature of DNA word design [4], [5], [9], [10] one is interested in code constructions where one bounds suitable DNA string distances, out of which we shall consider two (representative) examples. Can one solve in a satisfactory way the *inverse problem* of constructing possibilistic channels (of describing suitable “channel noise”) which would “explain” those constructions, as one can do in the usual Hamming case of coding theory? Or: can these DNA distances be seen as suitable “pseudo-distinguishabilities” for a suitable noisy channel? As argued in next section, one of the two DNA distances will faithfully mirror the usual Hamming case, while the second will turn out to be totally unmanageable. It is no coincidence, we deem, that the second case is less justifiable also from a strictly biological point of view.

4 The inverse problem of channel noise in DNA word design

DNA word design is an “odd” form of coding used in molecular computation, where, based on biological facts (cf. section 5), one exhibits maximum-size code constructions relative to constraints of the form $\xi(x, y) \geq \lambda$ for a suitable DNA string distance ξ . An information-theoretic problem arises: what is the nature of the biological channel one is implicitly envisaging, or, equivalently: what sort of biological “noise” are we fighting against when we use these code constructions? Thinking of the above arguments, we can re-formulate the question as follows: can $\xi(x, y)$ be interpreted as a “pseudo-distinguishability”, i.e.: can one exhibit a transition possibility $\pi(z|x)$ between inputs and outputs such that the corresponding distinguishability function $\delta(x, y)$ is a non-trivial and non-decreasing function of $\xi(x, y)$? The possibilistic framework⁶ allows one to answer these question in a sensible way. We shall discuss two types of code constructions found in the literature: the answer will be positive in one case, which is better justified also from the biological point of view, and negative in the other.

We shall deal only with two DNA “distortions”, which however are very representative, the *reverse Hamming distance* and a variation thereof:

$$d_R(x, y) \quad \text{and} \quad d_{H \wedge R}(x, y) = d_H(x, y) \wedge d_R(x, y)$$

Here $d_H(x, y)$ is the usual Hamming distance, while the reverse Hamming distance is $d_R(x, y) = d_H(x, y^*)$, with y^* mirror image of y . In practice, in the case of d_R , codewords in a good code should have a large reverse Hamming distance, while they should have *both* a large Hamming distance *and* a large reverse Hamming distance⁷ in the case of $d_{H \wedge R}$. We recall that $d_{H \wedge R}(x, y)$ is a *pseudometric*; one has $d_{H \wedge R}(x, y) = 0$ when $x = y$ or when x and y are mirror images of each other. Nothing so tame happens in the case of d_R , which violates the triangle inequality.

⁶ Maximum likelihood is enough to account for “usual” code constructions, as those in [7], by referring to symmetric channels. The maximum likelihood approach, however, does not appear to be the right model for “odd” codes as the ones of DNA word design.

⁷ The reason why we do not discuss explicitly complementarity and self-hybridization are given in the last section.

Below we shall try to “explain” the corresponding DNA code constructions by exhibiting a suitable possibilistic noisy channel and a suitable noise-fighting decoder. To achieve this, let us begin by the “friendlier case”, and let us compute the confusability $\gamma_{H\wedge R}$ and the distinguishability $\delta_{H\wedge R}$ corresponding to the string distance $d_{H\wedge R}$ taken as the distortion between inputs and outputs. We decode the output z by minimum distortion, and so we are implicitly assuming that it is “unlikely” (i.e. possible only to a small degree) that z has both a large Hamming distance and a large reverse Hamming distance from the codeword c actually sent over the channel.

Theorem 1. . *Decode the output z by minimising $d_{H\wedge R}(c, z)$, $c \in \mathcal{C}$; the corresponding distinguishability and confusability functions are:*

$$\delta_{H\wedge R}(x, y) = \left\lceil \frac{d_{H\wedge R}(x, y)}{2} \right\rceil, \quad \gamma_{H\wedge R}(x, y) = n^{-1} \left\lfloor n \frac{1 + \pi(y|x)}{2} \right\rfloor$$

This is exactly the same situation as found with usual Hamming distances and the codes of algebraic coding. In practice, this means that a possibilistic channel based on the transition possibilities

$$\pi(z|x) = 1 - n^{-1} d_{H\wedge R}(x, z)$$

and the corresponding “noise” quite adequately “explain” the code constructions based on checking the pseudometric $d_{H\wedge R}$, as are those found in the literature.

Now, let us think of a DNA word design construction where one controls *only* the minimum reverse Hamming distance between codewords. The situation is less friendly, because if we decide to decode by minimum reverse Hamming distance, the corresponding distinguishability function turns out to be a non-decreasing function of the *usual* Hamming distance, and *not* of the reverse Hamming distance, as a simple computation shows. In other words, against this sort of noise one would need the *usual* codes of coding theory, and *not* the codes of DNA word design which we are trying to “explain”. So, the following problem is relevant:

Problem: Exhibit a non-trivial transition possibility $\pi(y|x) \geq 0$ with distinguishability function $\Xi(x, y)$ such as to be a non-decreasing function of the reverse Hamming distance.

Unfortunately one has the following negative result:

Theorem 2. *Let $\pi(y|x)$ be a transition possibility such that its distinguishability function $\Xi(x, y)$ is a non-decreasing function f of the reverse Hamming distance $d_R(x, y)$. Under such hypothesis, Ξ is trivial, in the sense that $\Xi \equiv 0$ when n is even, while, when n is odd, $\Xi(x, y) = 0$ whenever $d_R(x, y) \leq n - 1$. Moreover, if ξ is constrained to verify the triangle inequality, then $\Xi \equiv 0$ even for n odd.*

Proof. Formally, one has $\Xi(x, y) = f(d_R(x, y))$, with $f : \mathcal{N} \rightarrow \mathcal{N}$ such that $k \leq h \Leftrightarrow f(k) \leq f(h)$; we can also suppose without loss of generality that $f(0) = 0$. We know that $\Xi(x, y) = f(d_R(x, y))$. For all $x \in \mathcal{X}$, it holds that $d_R(x, x^*) = 0$, hence

$\Xi(x, x^*) = f(0) = 0$. But this means that $\exists \tilde{z} \in \mathcal{X}$ such that $\xi(x, \tilde{z}) = \xi(x^*, \tilde{z}) = 0$. Therefore $\Xi(x, x) = \min_{z \in \mathcal{X}} \xi(x, z) = 0$, and so each string has distinguishability zero from itself. Now, if x is a string such that $d_R(x, x) = m$, one has $f(m) = f(d_R(x, x)) = \Xi(x, x) = 0$; as $d_R(x, x)$ can assume all even values from 0 to n , one may take $m = n$ for n even and $m = n - 1$ for n odd. Hence the non-decreasing function f maps the integer interval $[0, n]$ to 0 for n even, and the integer interval $[0, n - 1]$ to 0 for n odd. If ξ is constrained to verify the triangle inequality, then $\xi(x, y) \leq 2\Xi(x, y)$ (cf. [14]), and so $\xi(x, y) = 0$ whenever $d_R(x, y) \leq n - 1$. Now, for $n = 3, 5, \dots$ one can always find a z at reverse Hamming distance $\leq n - 1$ from both x and y ; e.g. take z with the first digit equal to the last digit of x and with the last digit equal to the first digit of y ; this implies $\Xi \equiv 0$ even for n odd.

In practice, the theorem means that, at least within the possibilistic framework, ample as it may be, code constructions based on checking reverse Hamming distances have *no* counterparts in terms of noisy channels and channel decoders; *no* possibilistic matrix $\pi(z|x)$ exists which would adequately support those constructions.

5 A short reminder on DNA word design

In the last ten years, a new computational paradigm emerged from a very uncommon place, i.e. wet labs of biologists. The fact that DNA contains all the basic information necessary to build very complex living organisms convinced Adleman that it could also be used as a computational entity. In his milestone paper of 1994 [1], he proposed a computational model based on very simple manipulations of DNA that can be performed in a wet lab. This model is Turing-complete and bases its power on the massive parallelism achievable by using DNA. Moreover, one of the basic operations performed is the hybridisation of complementary DNA strings. Specifically, DNA strings are oriented strings over the alphabet $\Sigma = \{a, c, g, t\}$, where a - t and c - g are complementary letters. Two such strings are said to be complementary if they have the same length and if one can be generated by reversing the other and complementing each of its letters. Physically, complementary DNA strings can hybridise, i.e. they can attach one to the other, forming the famous double helix. Actually, hybridisation can occur also between strings that are not perfect complements, but close to it. In DNA computations, data is coded by short strings of DNA in such a way that hybridisations occurring determine the output of the “algorithm” [10]. Therefore, one of the main concerns is to avoid that “spurious” hybridisations occur, leading straight to the so-called *DNA word design* problem.

DNA word design (cf. [9, 4]) consists of identifying sets of DNA strings of a given length, called *DNA codes*, satisfying some constraints, usually related to distances between codewords. In particular, the main concern of DNA word design is to identify maximal set of strings satisfying the above mentioned constraints, cf. Section 3.

In the body of the paper complementarity has been forgotten out of simplicity, since it does not really change the problem, but makes notations and formulations

heavier; cf. also [5]. Seemingly, we have forgotten also about self-hybridisation, i.e. we have forgotten to check that the codewords should not “resemble” their own mirror images. Notice however that in Section 5 we have never used the assumptions that the input space is made of *all* the strings of length n : the constraint on the reverse Hamming distance between a codeword and itself has a nature of its own, and defines the input space for possible codewords, restricting it only to those strings whose self-distance is large enough with respect to a prescribed threshold. This approach is simpler, and in our context does not really make any special difference.

References

1. L. Adleman. Molecular computations of solutions of combinatorial problems. *Science*, 266:1021–1024, November 1994.
2. L. Bortolussi and A. Sgarro. Fuzzy integrals and error correction in a telephone networks with sensitive users. *Proceedings of IPMU-2004*, Perugia, Italy, 1503–1508, 2004.
3. L. Bortolussi and A. Sgarro. Fuzzy codebooks for DNA Word Design. To be presented at *IPMU-2006*, Paris, France, July 2006.
4. A. Condon and A. Brennen. Strand design for bio-molecular computation. *Theoretical Computer Science*, 287(1):39–58, 2002.
5. A. Condon, R.M. Corn, and A. Marathe. On combinatorial dna word design. *Journal of Computational Biology*, 8(3):201–220, November 2001.
6. D. Dubois and H. Prade, eds. *Fundamentals of Fuzzy Sets*. Kluwer, 2000.
7. J. van Lint. *Introduction to Coding Theory*. Springer Verlag, Berlin, 1999.
8. F.L. Luccio, A. Sgarro, Fuzzy graphs and error-proofs keyboards. *Proceedings of IPMU-2002*, Annecy, France, 1503–1508, 2002.
9. G. Mauri and C. Ferretti. Word design for molecular computing: A survey. In *Proceedings of 9th Int. Workshop on DNA Based Computers, DNA 2003*, pages 37–46, 2003.
10. N. Pisanti. A survey on dna computing. *EATCS Bulletin*, 64:188–216, 1998.
11. A. Sgarro. Possibilistic information theory: a coding-theoretic approach. *Fuzzy Sets and Systems*, 132-1, 11–32, 2002.
12. A. Sgarro. Possibilistic time processes and soft decoding. In *Soft Methodologies and Random Information Sustems*, ed. by M. Lopez-Diaz et al., 249–256, Springer Verlag, 2004.
13. A. Sgarro. Utilities and distortions: an objective approach to possibilistic coding. *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*, 13-2, 139–161, 2005.
14. A. Sgarro and L. Bortolussi. Codeword distinguishability in minimum diversity decoding. Submitted to *Journal of Discrete Mathematical Sciences and Cryptography*, 2005.