Fuzzy Codebooks for DNA Word Design

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

Andrea Sgarro
Dept. of Mathematics and Informatics
University of Trieste
Trieste, Italia
CBM, Center for Biomolecular Medicine
Trieste, Italia
sgarro@units.it

Abstract

We develop a formal framework to deal with code constructions in a fuzzy setting. Strings are modeled as fuzzy sets and an adequate concept of distance is defined. Moreover, we study fuzzy codebooks from the point of view of their minimum distance. This fuzzy framework is then used to model the DNA word design problem, i.e. the construction of particular codes of DNA strings that are used in molecular computation.

Keywords: Fuzzy codebooks, fuzzy strings, DNA word design.

1 Introduction

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 Adlemann 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 hybridization 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 hybridize, i.e. they can attach one to the other, forming the famous double helix. Actually, hybridization 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 hybridizations occurring determine the output of the "algorithm" [8]. Therefore, one of the main concerns is to avoid that "spurious" hybridizations occur, leading straight to the so-called *DNA word design* problem.

DNA word design (cf. [7, 3]) consists of identifying sets of DNA strings of a given length, typically in a range from 10 to 20, called *DNA codes*, satisfying some constraints, usually related to distances between codewords, cf. next Section. In particular, the main concern of DNA word design is to identify maximal set of strings satisfying the above mentioned constraints.

All the current approaches to DNA word design deal with DNA strings as crisp objects. However, from the point of view of the applications, DNA computing above all, this assumption seems too strong. In fact, computation with DNA, from a coarse point of view, proceeds by creating many copies of the designed strands, putting them in a test tube, and letting them interact with each other (essentially, hybridize). Then the result is extracted by means of wet lab techniques. The process of creating many copies of a DNA string is not an error-free mechanism. Therefore, in the test tube, we do not have thousands copies of the same string, but actually a cloud of strands, (hopefully) close to the original one. As the error rate is not so high, we do not expect to find strings very different from the original one, and that is why this fuzzy property of DNA words is always forgot, as long as designed strings are far away from each other. In our approach, instead, we want to encapsulate this information in the code construction procedure, by relaxing the crisp requirement on strings, and modeling them as fuzzy sets.

The first steps in this direction are presented here, and consist in defining in a sound way the concept of distance between fuzzy strings, the concept of fuzzy codebook and its characterizing property, i.e. minimum distance. We stress that the theoretic framework below applies to any codebook and any string distance, but it is precisely a biological context which makes a "vague" (fuzzy) description of codewords especially appropriate.

The paper is organized as follows. In Section 2 classical DNA word design is presented in more detail. In Section 3 we introduce the desired notion of fuzzy distance between strings, while in Section 4 we comment on the concept of minimum distance. Finally, in Section 5 we go back to DNA word design, to see how it fits in our new framework.

2 Crisp DNA Word Design: a Reminder

DNA word design focuses on the construction of sets, or codes, of DNA strings satisfying certain constraints. A detailed description, with emphasis on the biological point of view, can be found in [3, 7].

We consider oriented strings built from DNA alphabet, $\Sigma = \{a, c, g, t\}$, of a fixed length n. Our task is to identify a code $\mathcal{C} \subseteq \Sigma^n$ such that all the strings of $\mathcal{C}$ are sufficiently "far away" from each other. Here been distant means that the reverse complement x^{RC} of one string $x \in \mathcal{C}$ hybridizes just with x and not with other strings of the code. In addition, we don't want that different strings of the code hybridize between themselves, and that a string self-hybridizes.

To formalize the above intuitions, first we define the reverse complement operation. Given a string $x = x_1 \dots x_n$, its reverse complement is $x^{RC} = x_n^C \dots x_1^C$, where $(\dot{})^C$ is the Watson Crick complement, defined as $a^C = t$, $c^C = g$, $g^C = c$ and $t^C = a$. Note that this function is an involution. The *Hamming distance* $d_H(x, y)$ between two strings x and y is defined in the usual way as the number of positions in which x and y differ, while the *reverse complement Hamming distance* between x and y is $d_H^{RC}(x, y) = d_H(x, y^{RC})$, i.e. the Hamming distance between x and the reverse complement of y.

On a codebook $\mathcal{C}$ we impose that two different strings must have Hamming distance and reverse complement Hamming distance greater than a certain threshold. The constraint on the Hamming distance guarantees that the reverse complement x^{RC} of the string x hybridized just with x (the reverse complement operation is an isometry in the space Σ^n, hence $d_H(x^{RC}, y^{RC}) = d_H(x, y)$). The other constraint, instead, accounts for the property that two strings of the code do not hybridize between themselves. Self-hybridization is avoided by putting $d_H^{RC}(x, x) \geq D$, for a threshold D, i.e. by restricting the choice of x from a set of strings sufficiently "non-palindromic".

The constraints introduced give a combinatorial formulation of the DNA code construction problem, which is very similar to the code construction of coding theory. In this light, there is some theoretical work [4] that gives upper and lower bounds to the dimensions of such codes. There are also some algorithms for constructing these codes, that are either based on stochastic local search [12], or on a branch and bound method [2].

Several other constraints for DNA word design can be found in literature, coming from thermodynamical and biological considerations, cf. [3]. However, without loss of generality, we deal here with a simplified version, using only the combinatorial constraints introduced above. Extensions to more general settings are straightforward.

Incidentally, we observe that the parallel between DNA code construction and coding theory gives rise to interesting questions related to information transmission and code corrections capabilities in the contest od DNA computing. Considerations in this sense can be found in [11].

3 Distances for Fuzzy Strings

Let a crisp distance $d(x,y) \in \{0,1,\ldots,n\}$ be given for n-length strings, e.g. quaternary strings as needed in DNA word design. By the *extension principle* [5], the fuzzy extension of $d(x,y)$ to fuzzy sequences X and Y (i.e. to fuzzy sets of crisp sequences) is the fuzzy distance $d(X,Y)$, which is a fuzzy quantity (a fuzzy integer). It is defined by the corresponding degrees of membership:

$$\begin{array}{l}\langle m \in d(X,Y)\rangle \;= \\ \max_{x,y:\, d(x,y)=m} \langle x,y \in X \times Y\rangle\end{array} \tag{1}$$

Here and below angular brackets denote degree of memberships, and so $\langle m \in d(X,Y)\rangle$ is the degree of membership of the integer m to the fuzzy quantity $d(X,Y)$ seen as a fuzzy set over $\{0,1,\ldots,n\}$; it answers the question: up to what degree the distance d is equal to m? Recall that, by *interactivity*, the degree of membership of a couple to the corresponding Cartesian set, i.e. $\langle x,y \in X \times Y\rangle$, is the minimum of the degrees of membership of the two coordinates, and so is equal to $\langle x \in X\rangle \wedge \langle y \in Y\rangle$ (the wedge stands for a minimum). A degree of membership is 0 if the corresponding maximization set is void.

After choosing a threshold ϵ, $0 \le \epsilon \le 1$, one may give the following "conservative" definition[1] so as to defuzzify the fuzzy quantity $d(X,Y)$:

$$\begin{array}{l}d^*_\epsilon(X,Y) \;=\; \min\,[d(X,Y)]_\epsilon \\ \;=\; \min\{m : \langle m \in d(X,Y)\rangle \ge \epsilon\}\end{array} \tag{2}$$

with $[d(X,Y)]_\epsilon$ equal to the ϵ-*cut* of $d(X,Y)$.

Example 1. *We take two binary triples X and Y, which are fuzzy, and take d equal to the usual Hamming distance. Choose e.g. $X = (000,1;111,1/2;001,1/3)$ and $Y = (001,1;101,1/2)$, with self-explaining notation (the triple 000 crisply belongs to X, the triple 111 belongs to X up to degree $1/2$, and so on). One has:*

$\langle 0 \in d(X,Y)\rangle = 1/3$, $\langle 1 \in d(X,Y)\rangle = 1$, $\langle 2 \in d(X,Y)\rangle = 1/2$, $\langle 3 \in d(X,Y)\rangle = 0$,

and so $d^*_{1/3}(X,Y) = 0$, $d^*_{1/2}(X,Y) = 1$, $d^*_1(X,Y) = 1$.

Choose Z quaternary, e.g. $Z = (atga, 1; atgt, 1/2)$, with reverse complement Hamming distance. One has:

$\langle 0,1 \in d(Z,Z)\rangle = 0$, $\langle 2,3 \in d(Z,Z)\rangle = 1/2$, $\langle 4 \in d(Z,Z)\rangle = 1$,

and so $d^*_{1/2}(Z,Z) = 2$, $d^*_1(Z,Z) = 4$.

Alternatively, to defuzzify $d(X,Y)$ we may give another equally conservative definition, recalling how a distance[2] between crisp sets is usually defined, when a distance between its elements is given. Now, ϵ-cuts $[X \times Y]_\epsilon$ *are* crisp sets of couples x,y:

$$\begin{array}{l}d_\epsilon(X,Y) \;=\; \min_{x,y\,\in\,[X\times Y]_\epsilon} d(x,y) \;= \\ \min_{\langle x\in X\rangle\wedge\langle y\in Y\rangle\ge\epsilon} d(x,y)\end{array} \tag{3}$$

For computations Definition (3) is more convenient than the starred Definition (2). As soon checked, in the example above one has $d_\epsilon(X,Y) = d^*_\epsilon(X,Y)$ and $d_\epsilon(Z,Z) = d^*_\epsilon(Z,Z)$: fortunately, this is no coincidence, as follows from the Proposition below, which by the way makes the starred notation useless:

Proposition 1.

$$d_\epsilon(X,Y) = d^*_\epsilon(X,Y)$$

Proof. Straightforward consequence of the lemma in the section below, with $d(X,Y)$ instead of $f(X)$, and choosing min as an operator. ■

Actually, given the generality of the lemma, one might have "aggregated" m-values by any operator, e.g. an arithmetic average, rather than the minimum. Our choice is just a first try, suggested by the biological applications we have in mind.

3.1 Side Note: An "Abstract" Lemma

However obvious, the lemma below is quite convenient in many situations. In it we consider only

[1] If the minimization set is void, which never happens if the two fuzzy strings involved are constrained to be *normal* [5], the minimum is set equal to the largest distance, which in our case will be n.

[2] Our "distances" are not necessarily *metric* distances; e.g. in the case of sets, as well known, the triangle inequality fails to hold.

functions f between *finite* sets; however, generalizations would be straightforward. Let

$$\langle y \in f(X) \rangle \; = \; \max_{x:\, f(x)=y} \langle x \in X \rangle$$

define the fuzzy extension $Y = f(X)$ of the crisp function $y = f(x)$. Think of the ϵ-cut:

$$[f(X)]_\epsilon \; = \; \{ y : \, \langle y \in f(X) \rangle \; \geq \epsilon \}$$

which of course is a crisp set of y's. Since requiring $\max \langle x \in X \rangle \; \geq \; \epsilon$ is the same as requiring that an x exists with $\langle x \in X \rangle \geq \epsilon$, one has also:

$$[f(X)]_\epsilon = \{ y : \, \exists x \in X_\epsilon \; \text{ s.t. } \; f(x) = y \}$$

Let $\odot$ be any operator, i.e. any function $\odot$, whatever its range, whose domain is made up of sets of y's (or multisets, i.e. sets with repeated elements, or K-tuples). The two expressions for an ϵ-cut soon imply the equality below:

Lemma 1.

$$\bigodot \, [f(X)]_\epsilon \; = \; \bigodot_{x \in X_\epsilon} f(x)$$

4 Minimum Distance for Fuzzy Codebooks

Let a fuzzy codebook $\mathcal{C}$ be assigned through its K fuzzy codewords $C_1, \ldots, C_K$, in this order. In the following $\underline{c}$ will denote a generic K-tuple of (not necessarily distinct) crisp strings $c_1, \ldots, c_K$. The (crisp) minimum distance $\delta(\underline{c})$ for such a K-tuple $\delta(\underline{c})$ is then well defined as the minimum distance between codewords c_i with distinct indices:

$$\delta(\underline{c}) \; = \; \min\nolimits_{i \neq j} \, d(c_i, c_j) \qquad (4)$$

Of course, this quantity is zero whenever one has $c_i = c_j$ for $i \neq j$. Cf. e.g. [6] for basics on coding theory and on the coding-theoretic significance of the minimum distance.

Equally well defined is the degree of membership $\langle \underline{c} \in \mathcal{C} \rangle$ of the K-tuple $\underline{c}$ to the fuzzy codebook $\mathcal{C}$: actually, by just recalling the definition of a fuzzy Cartesian power, $\langle \underline{c} \in \mathcal{C} \rangle$ is equal to $\min_i \langle c_i \in \mathcal{C}_i \rangle$. By the extension principle the fuzzy minimum distance of the fuzzy codebook $\mathcal{C}$ is then defined by:

$$\langle m \in \delta(\mathcal{C}) \rangle \; = \; \max_{\underline{c}:\, \delta(\underline{c}) = m} \langle \underline{c} \in \mathcal{C} \rangle$$

To defuzzify this (rather inconvenient) expression, once more we choose a threshold ϵ and once more we aggregate crisp minimum distances in the ϵ-cut $\delta(\mathcal{C})_\epsilon$ by a minimum; by so doing we obtain the *minimal minimum distance* of the fuzzy codebook $\mathcal{C}$, $\delta(\mathcal{C})$. The choice of the minimum as an aggregator is the most conservative one: in practice, *whatever crisp codebook which is extracted from the fuzzy codebook $\mathcal{C}$ cut at level ϵ has a (crisp) minimum distance $\geq \delta_\epsilon(\mathcal{C})$.*

Using the abstract lemma and the definition of crisp minimum distance, and observing that

$$\mathcal{C}_\epsilon = [C_1]_\epsilon \times \ldots \times [C_K]_\epsilon$$

one has:

Proposition 2.

$$\delta_\epsilon(\mathcal{C}) \; =_{\text{def}} \; \min \, [\delta(\mathcal{C})]_\epsilon \; = \; \min_{\underline{c}:\, \underline{c} \in \mathcal{C}_\epsilon} \delta(\underline{c})$$

With our conservative choice, an alternative and meaningful expression is available for $\delta_\epsilon(\mathcal{C})$, which uses $d_\epsilon(C_i, C_j)$ as defined in Section 2. We recall that the *level* of a fuzzy set is the greatest degree of membership of its elements; if the codewords are all *normal*, i.e. if their level is one, the constraint in Proposition 3 is certainly met.

Proposition 3. *Assume that all the fuzzy codewords C_i have level $\geq \epsilon$. Then:*

$$\delta_\epsilon(\mathcal{C}) \; = \; \min_{i \neq j} \, d_\epsilon(C_i, C_j)$$

Proof. Recalling (3) and the definition of the crisp minimum distance (4), it will be enough to prove that:

$$\min_{i \neq j} \; \min_{c_i, c_j \in [C_i \times C_j]_\epsilon} \; d(c_i, c_j) = \min_{\underline{c} \in \mathcal{C}_\epsilon} \, \min_{i \neq j} \, d(c_i, c_j)$$

Just observe that, if the constraint is met (and only in this case), one can "prolong" any couple c_i, c_j in $[C_i \times C_j]_\epsilon$ to a whole K-tuple $\underline{c}$ such that its degree of membership to $\mathcal{C}$ is still $\geq \epsilon$. Consequently, the second of the four minima in the equality can be replaced by a minimum over $\underline{c} \in \mathcal{C}_\epsilon$; then just swap the first two minima. ■

Example 2. *Consider the following (rather simple) fuzzy codebook $\mathcal{C}$, made up of three fuzzy binary strings of length 4: $\mathcal{C} = \{C_1, C_2, C_3\}$, with $C_1 = \{0000, 1; 0001, 1/3; 0100, 1/2\}$, $C_2 = \{1111, 1; 1011, 1/2; 1001, 1/6\}$, $C_3 = \{0101, 1; 0111, 1/3; 0100, 1/3\}$. If we use the Hamming distance, then the fuzzy distances between these strings are: $d(C_1, C_2) = \{0, 0; 1, 1/6; 2, 1/3; 3, 1/2; 4, 1\}$, $d(C_1, C_3) = \{0, 1/3; 1, 1/2; 2, 1; 3, 1/3; 4, 0\}$ and $d(C_2, C_3) = \{0, 0; 1, 1/3; 2, 1; 3, 1/2; 4, 1/3\}$. Starting from the ε-cuts of these sets, using Proposition 3, we can easily compute $\delta(\mathcal{C})$: $\delta_{1/3}(\mathcal{C}) = 0$, $\delta_{1/2}(\mathcal{C}) = 1$ and $\delta_1(\mathcal{C}) = 2$.*

The constraint in the last proposition is unavoidable, else the first side can be strictly smaller than the second side (or it they can be both undefined, if one chooses to leave undefined minima over void sets). To see this, take e.g. three codewords of length 1, $X = (a, 1)$, $Y = (b, 1)$, $Z = (c, 1/2)$, and choose $\epsilon = 1$.

5 Fuzzy DNA Word Design: Perspectives

In the introduction we noted that in real applications of DNA word design all the strings of a DNA code are duplicated in several copies (on the order of billions), and put in a test tube to perform the intended operations. As the duplication process is error prone, we do not expect to have just exact copies of the designed strings, but rather a “cloud” of strings centered around the original one. Therefore, a more realistic model can be obtained by considering the DNA strings as fuzzy rather than crisp. This decision implies that DNA word design should be tackled by constructing a fuzzy codebook.

In the following, we provide the details of a first attempt in this direction, just after making explicit some working hypothesis. The process of DNA strand synthesis is usually described by means of the observed frequency of errors, interpreted as a probability. In particular, assumptions are made on the independence of the occurrence of a transcription error, hence we can assume that the degree of membership depends only on the distance from the central string. Formally, if $x \in \Sigma^n$ is a DNA string, then we denote by $F(x)$ the fuzzy string defined by $\langle x \in F(x)\rangle = 1 = \mu_0$ and $\langle y \in F(x)\rangle = \mu_k$, with $k = d_H(x, y)$. Moreover, the μ_i satisfy the relation $1 = \mu_0 \geq \mu_1 \geq \ldots \geq \mu_n \geq 0$. The concrete value of those μ_i may depend on the particular synthesis mechanism used, but we will comment more on this at the end of the section.

The simple form of the fuzzy strings under consideration allows us to give a close form for the fuzzy distance and for its aggregated ε-cuts.

Proposition 4. *Let $x, y \in \Sigma^n$, with $d_H(x, y) = k$. For each $i \in [0, n]$, let $k_i = \lceil \frac{|k-i|}{2} \rceil$. Then:*

1. *$\langle i \in d_H(F(x), F(y))\rangle = \mu_{k_i}$;*
2. *$(d_H)_{\mu_i}(F(x), F(y)) = \max\{k - 2i, 0\}$.*

Proof. Point *1.* follows easily from the triangular inequality of the metric d_H. Suppose $i < k$ (the case $i > k$ is specular), and let x_i, y_i be a pair of strings at distance i, then $k = d_H(x, y) \leq d_H(x, x_i) + d_H(x_i, y_i) + d_H(y_i, y)$ and so $d_H(x, x_i) + d_H(y_i, y) \geq k - i$. To maximize $\langle x_i \in F(x)\rangle \wedge \langle y_i \in F(y)\rangle$, we have to minimize the maximum of $d_H(x, x_i)$ and $d_H(y_i, y)$ (it follows from the monotonic property of μ_i), and this is done by splitting evenly the distance between two strings realizing $d_H(x, x_i) + d_H(y_i, y) = k - i$. The value obtained is k_i. Point *2.* is soon derived from *1.*, observing that the minimum index j, if any, with degree of membership at least μ_i is given by $\frac{k-j}{2} = i$. ∎

Now we have to introduce the reverse complement Hamming distance, and integrate it with the Hamming distance. The first point is easily tackled by extending the reverse complement operation to fuzzy sets of the form $F(x)$, using its isometric property with respect to the Hamming distance ($d_H(x^{RC}, y^{RC}) = d_H(x, y)$). Concretely, we set $[F(x)]^{RC} = F(x^{RC})$, so that $d_H^{RC}(F(x), F(y)) = d_H(F(x), [F(y)]^{RC}) = d_H(F(x), F(y^{RC}))$, reducing the computation of the fuzzy reverse complement Hamming distance to the computation of the Hamming one.

On the other hand, the simplest way to combine together those two metrics is by taking their minimum, i.e. by defining

$d(x,y) = d_H(x,y) \wedge d_H^{RC}(x,y)$, and then by computing the minimum distance of the codebook with respect to this new distance d. A simple manipulation of minima shows that $d_\varepsilon(F(x),F(y)) = (d_H)_\varepsilon(F(x),F(y)) \wedge (d_H)_\varepsilon(F(x),F(y^{RC}))$, which for $\varepsilon = \mu_i$ becomes $d_{\mu_i}(F(x),F(y)) = \max\{d(x,y)-2i,0\}$. Now we are ready to state the following lemma, giving a simple expression for the minimum distance of a fuzzy codebook $\mathcal{C}$.

Lemma 2. *Let $\mathcal{C} = \{F(c_1),\dots F(c_m)\}$ be a fuzzy codebook w.r.t. distance d. Then:*

$$\delta_{\mu_k}(\mathcal{C}) = \min_{i\neq j}\max\{d(c_i,c_j)-2k,0\}.$$

Consider the case where we fix a threshold, and build a code $\mathcal{C}$ such that the distance between crisp strings is above a threshold D, i.e. $d(c_i,c_j) \geq D$ for all $c_i \neq c_j$ in $\mathcal{C}$. It is not restrictive to ask that such threshold is reached by some pairs of strings, and under this hypothesis, we have that $\delta_{\mu_k}(\mathcal{C}) = \max\{D-2k,0\}$. This means that, if we fix as reliability threshold any number greater than μ_1, than the fuzzy construction coincides with the crisp one, but this is no more true whenever the reliability threshold is less or equal to μ_1.

A little care must be taken in this approach, because we are not taking into account the self-distance constraint. The distance d, in fact, is such that $d(x,x) = 0$. However, self-distance is in a way "crisp": that's why the corresponding constraint can be dealt considering as an input space only sequences which are sufficiently "non palindromic", exactly as happens in the crisp case.

Let's try now to give a numerical expression for the μ_i that can be significative from a biological point of view. In general, error frequencies in the creation of DNA strands varies from 10^5 to 10^8, depending on the particular molecular machinery used to do the job. If we assume that the error level is 10^5, than we have a probability of $1/10^5$ of committing a transcription error, and the probability of having committed exactly one error while coping a string of length n can be approximated by $n/10^5$. Hence, we can model the μ parameters by setting $\mu_i = n/10^{5i}$. For a string of length 20, this means that $\mu_1 = 0.0005$, hence the reliability threshold above which classical and fuzzy code construction coincide is very small. This fact, by the way, confirms theoretically that the simplifications induced by the crisp constructions are safe enough.

Finally, we note that implicitly, by talking only about minimum distances, we have covered just reliability and *not* optimal transmission speed, i.e. optimal *code-rate*. Clearly, in absence of further constraints, optimal code constructions lead necessarily to *crisp* codebooks; not so, however, if one has to require that codewords should have a certain "degree of fuzziness" (cf. also [9]), which might make sense from a biological point of view. However, the first step in this direction is reasoning about the meaning of transmission and error correction, in the biological context, and a soft approach seems more adequate (cf. [11, 10]).

Acknowledgements

INDAM-GNCS and FIRB Bioinformatics have supported this research. We heartfully thank Federica Agostini for helpful discussions about biological issues.

References

[1] L. Adlemann. Molecular computations of solutions of combinatorial problems. *Science*, 266:1021–1024, November 1994.

[2] L. Bortolussi and A. Sgarro. Constraint satisfaction problems on DNA strings. In *Proceedings of 1th Int. Workshop on Constraint-based Methods in Bioinformatics , WCB 2005*, 2005.

[3] A. Condon and A. Brenneman. Strand design for bio-molecular computation. *Theoretical Computer Science*, 287(1):39–58, 2002.

[4] A. Condon, R.M. Corn, and A. Marathe. On combinatorial dna word design. *Journal of Computational Biology*, 8(3):201–220, November 2001.

[5] D. Dubois and H. Prade, eds. *Fundamentals of Fuzzy Sets*. Kluwer, 2000.

[6] J. van Lint. *Introduction to Coding Theory*. Springer Verlag, Berlin, 1999.

[7] 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.

[8] N. Pisanti. A survey on dna computing. *EATCS Bulletin*, 64:188–216, 1998.

[9] A. Sgarro. Possibilistic time processes and soft decoding. In *Soft Methodologies and Random Information Sustems*, ed. by M. Lopez-Diaz et alt., 249–256, Springer Verlag, 2004.

[10] A. Sgarro and L. Bortolussi. Codeword distinguishability in minimum diversity decoding. *Submitted to Journal of Discrete Mathematical Sciences and Cryptography*, 2005.

[11] A. Sgarro and L. Bortolussi. A possibilistic framework for DNA word design. *Preliminary*, 2006.

[12] D. Tulpan, H. Hoos, and A. Condon. Stochastic local search algorithms for dna word design. In *Proceedings of 8th Int. Workshop on DNA Based Computers , DNA 2002*, 2002.