Linear error correcting codes and their application in DNA analysis

S.Y. Korabelshchikova[1], B.F. Melnikov[2], S.V. Pivneva[3], L.V. Zyablitseva[1]

[1]Northern (Arctic) Federal University named after M. V. Lomonosov, Severnaya Dvina Emb., 17, Arkhangelsk, Russia, 163002

[2]Russian State Social University, Wilhelm Pieck str., 4, Moscow, Russia, 129226

[3]Togliatti State University, Belorusskaya str., 14, Togliatti, Russia, 445020

Abstract. The application of various areas of the theory of coding in the problems of analysis of the genome includes different areas. For instance, the method for detecting linear block codes, which takes into account possible insertions and divisions in DNA sequences. For a lot of such problems, we shall consider the number of codes of the required dimension. One of the most important such tasks is counting their number. We study this area it in this paper. We solve the problem of defining the number of cyclic codes over finite field with two arbitrary fixed parameters. These parameters are n, i.e., the length of the code, and k, i.e., the number of information symbols. The algorithm to solve this problem in general is described in this paper. Also the results at some fixed values are given. We have counted some constants connected with coding theory, which can be used in some problems of genome analysis.

Keywords: genome analysis, error correcting codes, algorithm, counting, C++ program.

1. Introduction and motivation

The application of various areas of the theory of coding in the problems of analysis of the genome began a long time ago; we are unlikely to be able to indicate the very first of the publications on this topic. In 2007, an attempt was made to somehow generalize the material accumulated to this moment, see [1]. According to the author, one of the first models for the application of coding theory in the analysis of the genome was proposed by [2]; let us also mention [3].

Some authors, being motivated by the excessive structure of the genetic code, the presence of large evolutionistic conserved non-coding regions among species and also the presence of special sequences in coding regions, attempt to apply coding theory models to understand the structure of DNA and the operation of various genetic processes. For example, the paper [4] has developed the first effective method of scanning the DNA sequence to determine whether any of the structures of the linear block code is present in the genome. And we are considering linear codes in this article.

A few years later, Rosen developed a method for detecting linear block codes, which takes into account possible insertions and divisions in DNA sequences [5, 6]; this method is connected with metrics similar to the Levenshtein distance (or metric) [7]. However, none of the listed works is capable of supporting the existence of such simple codes for correcting errors in DNA. So, according to the authors of this article, no one has yet proven the hypothesis of the existence of embedded error correcting codes in DNA. Despite this, this hypothesis is supported by a variety of biological observations: for example, it is well-known, that the size of the human genome is

much larger than the size required to indicate each characteristic of any given person: [8] and many other.

With the topic discussed in this article is also related to the MacDonaill's interpretation of the parity check code of the nucleotide composition [9]. Also worth mentioning are possible algorithms for highlighting white noise (AWGN, so called "additive white Gaussian noise") from the genome [1]; however, for these works we also cannot yet consider, that they have already received encouraging results.

Also in our previous works related to algorithms of DNA analysis, one can see the connection with the problems of coding theory considered in this paper. For example, in one practical implementation of the Panin's metric [10, 11], when heuristic calculation of the pseudo-distance between two given genomes, the exact value of the shift of both pointers to the positions of the genomes under consideration is made on the basis of calculations with skipping k positions from n considered ones; this leads to the use of cyclic codes. We propose to give a more detailed description of this version of Panin's metric in one of our next publications.

Thus, the authors believe that communication work with genome analysis and coding theory needs to continue. For mentioned problems, we shall consider the number of codes of the required dimension.

Thus, in the opinion of the authors of this work, for the analysis of genomes, problems arise related to linear error-correcting codes. Apparently, one of the most important such tasks is counting their number. We will study this problem it in this paper.

2. Preliminaries

Firstly, let us consider briefly some algebraic terms necessary for this article. Let us assume that symbols, of which the initial and the code messages consist, are elements of some finite field F_q, where q is the potency of the field. The structure of finite fields and their contemporary application have been described in [12, 13] and other. Let us also assume that the noise involves replacing some symbols with different elements of the field F_q, which is equivalent to adding the error vector $e = (e_1, e_2, \ldots, e_n)$ (i.e., AWGN mentioned above) to the channel code vector $x = (x_1, x_2, \ldots, x_n)$.

Linear codes are used to correct this kind of errors. The initial message is divided into k-symbol long blocks and each block $(a_1, a_2, \ldots, a_k)$ is replaced with a corresponding code vector $(x_1, x_2, \ldots, x_n)$ of length $n > k$, where k symbols are informative, i.e., $x_(i_1) = a_1$, $x_(i_2) = a_2$, $\ldots$, $x_(i_k) = a_k$ and extra $n - k$ symbols are for verification and they are defined based on the following set of equations:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = 0, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = 0, \\ \cdots\cdots\cdots\cdots\cdots \\ a_{(n-k)1}x_1 + a_{(n-k)2}x_2 + \cdots + a_{(n-k)n}x_n = 0. \end{cases} \tag{1}$$

The matrix $H = (a_{ij})$ of coefficients for the set (1) is called a check-matrix of the linear (n, k)-code and it uniquely defines the set of code vectors, which is the right null-space of the matrix H. A code is called systematic if the initial k symbols are informative, i.e., $x_1 = a_1$, $x_2 = a_2$, $\ldots$, $x_k = a_k$, and the final $n - k$ symbols are the check ones. A check-matrix of a systematic code can be reduced to the form $H = (A|E_{n-k})$, where E_{n-k} is an identity $n - k$ matrix, by elementary manipulations with rows.

Thus, a linear (n, k)-code C over the field F_q is defined by the $(n - k) \times n$ check-matrix $H = (a_{ij})$ of the rank $n - k$. The code C consists of all the n-dimensional vectors $x = (x_1, x_2, \ldots, x_n)$ that satisfy the set of homogeneous linear equations (1).

The Hamming distance between the n-digit long vectors is the number of components, at which these words are different. Minimal pairwise distance of different code vectors is called the minimal distance of a code and is denoted by $d_{\min}(C)$ or just d. The error-correcting capability of a code depends on this number: a code with minimal distance d can recognize $d-1$ and correct $((d-1))/2$ errors. (Here, the square brackets [] mean the integral part of the quotient.)

The columns of the check-matrix H fulfill the following condition: the minimal distance of a code equals d if and only if there exist d linearly dependent columns in the matrix H and any $d-1$ columns are linearly independent. For the code to correct at least one error, the minimal distance must meet the requirement $d \geq 3$. This means that any 2 columns of the check-matrix must be linearly independent. If the binary codes are considered, the linear independence of two columns means that these columns are different and non-vacuous. Rows of the matrix H must also be different and non-vacuous. Moreover, they must be linearly independent as the rank of matrix H equals $n-k$.

To summarize the given above, the binary $(n-k) \times n$ matrix H is a check-matrix for some binary error-correcting code if and only if its rows are linearly independent and its columns are pairwise different and non-vacuous. Let us count such matrices.

3. An algorithm for counting the number of check-matrices

Let us denote for convenience $n-k=m$. Further, there is the given algorithm for finding the quantity of binary $m \times n$ matrices, all the columns of which are non-vacuous and different and rows of which form a linearly independent system, consequently, the rows are also non-vacuous and different.

The algorithm for counting check-matrices consists of the following steps.

(i) Turning on the counter of the above mentioned matrices: $S := 0$.

(ii) We consider all the sequences of m different numbers from 1 to 2^n-1 (i.e., the numbers in the sequence are placed in ascending order). This can be done, for instance in the following way: a subset of the set can be described by specifying if an element of the initial set belongs to the considered subset. To do this thing, each element of the set is assigned to 0 or to 1. Thus, we generate all the (2^n-1)-digit long binary sequences (they are matched with all the subsets of the set $\{1, 2, \ldots, 2^n-1\}$). From all the binary sequences, we chose the ones containing m unities. Looking through the sequence from the left to the right, we associate each unity with number of the position it holds in the sequence.

(iii) Let $P = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ be the above mentioned sequence. For each number of the sequence, let us find a binary n-digit long representation: these ones would be m different rows of a matrix.

Let us find out if the rows of the obtained matrix are linearly independent. To do this thing, we reduce the matrix to the echelon form. If there appears a vacuous row, proceed to the next sequence of m numbers and do step 3 with it. Otherwise continue to step (iv).

(iv) Assuming, that

- binary representation of the number α_1 is $(a_{11}, a_{12}, \ldots, a_{1n})$;
- binary representation of the number α_2 is $(a_{21}, a_{22}, \ldots, a_{2n})$;
- .....;
- and binary representation of the number α_m is $(a_{m1}, a_{m2}, \ldots, a_{mn})$,

let us find the numbers β_1, β_2, ..., β_n, matched with the columns of the matrix, relying on the binary representations of these numbers, where

- $(a_{11}, a_{21}, \ldots, a_{m1})$ is the binary representation of the number β_1;
- $(a_{12}, a_{22}, \ldots, a_{m2})$ is the binary representation of the number β_2;
- .....;

- $(a_{1n}, a_{2n}, \ldots, a_{mn})$ is the binary representation of the number β_n.

If the following conditions hold:

- all the numbers β_1, β_2, …, β_n are different and nonzero;
- the matrix associated with the sequence $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ is a binary $m \times n$ matrix;
- all the columns of which are non-vacuous and different;
- and the rows of which form a linearly independent system,

then we set $S := S + 1$.

If all the sequences have been considered, continue to step (v). Otherwise continue to the to the next m-digit long sequence and do step (iii) with it.

(v) For each sequence $(\alpha_1, \alpha_2, \ldots, \alpha_m)$, the $m!$ different matrices can be generated changing the position of the rows. As this takes place, the columns remain different and non-vacuous. That is why the quantity of all the matrices that meet the condition equals $S - m!$. □

The output of the program based on this algorithm is given in Table 1; i.e., we write there the quantity of binary $m \times n$ matrices, all the columns of which are non-vacuous and different and rows of which form a linearly independent system.

Table 1. The quantity of counted matrices

m/n	1	2	3	4	5	6
1	1	0	0	0	0	0
2	0	6	6	0	0	0
3	0	0	168	840	2520	5040
4	0	0	0	20160	322560	3528000

4. Estimating the quantity of considered codes

We can see that the number of matrices found in Section 3 rises rapidly as the dimension slightly increases. And among the above mentioned matrices, there are some ones that define the same linear code, as elementary manipulations with the set of linear equations lead to its transition to a solution of equivalent set. Rank of the considered matrices equals to $m = n - k$. That is why in such matrices, there must be m linearly independent columns. By elementary manipulations with the rows of the matrix, these m columns can be replaced with columns that form an identity m matrix. Two linear (n, k)-codes are called equivalent if their check-matrices can be derived one from another by elementary manipulations with the rows or by repositioning the columns. Repositioning the columns is tantamount to corresponding rearrangement of the components in all the code vectors.

In the sense described above, every linear (n, k)-code is equivalent to some systematic linear code with a check-matrix of the form $H = (A|E_m)$, where A is a $m \times k$ matrix and E_m is an identity m matrix. Now, let us estimate the quantity of different matrices H of such form that define different systematic error-correcting codes.

Whatever be the matrix A, the rows of the matrix $(A|E_m)$ are different and linearly independent. Consequently, for the code to correct errors it is sufficient that the rest of the columns be non-vacuous, different from the columns of the identity matrix and pairwise different. Let us subtract the quantity of columns that have been used and the vacuous column, $(n - k) + 1 = m + 1$, from the number 2^m, the total quantity of binary m-digit long columns. I.e., we can use $2^m - (m+1)$ columns. The quantity of various positions of $2^m - (m+1)$ columns in place of k columns of the matrix A equals to the number of arrangements of $2^m - (m + 1)$ digits in k without repetition. Hence, the following theorem has been proved.

Theorem 1 *The quantity of different check-matrices of the form $H = (A|E_m)$ defining different systematic binary linear error-correcting (n, k)-codes equals to*

$$P^k_{2^m-m-1} = \frac{(2^m - m - 1)!}{(2^m - m - k - 1)!},$$

where $m = n - k$ is the number of check digits of a linear code.

Among the codes counted according to the theorem 1, there are equivalent ones, and we found by repositioning the columns of the matrix A. If the order of the columns in the matrix A is not important, the number of possible ways to arrange $2^m - (m + 1)$ in place of k columns of the matrix equals the number of combinations from $2^m - (m + 1)$ digits in k. But there may be matrices defining equivalent codes (in the sense mentioned above) among the rest of the matrices. That is why the following theorem is true.

Theorem 2 *The quantity of binary check-matrices of the form $H = (A|E_m)$ defining inequivalent systematic linear error-correcting (n, k)-codes is no more than*

$$C^k_{2^m-m-1} = \frac{(2^m - m - 1)!}{k! \cdot (2^m - m - k - 1)!},$$

where $m = n - k$ is the number of check digits of a linear code.

Thus, since the dimension of the check-matrices for linear $(6, 2)$-codes is 4×6, the quantity of inequivalent systematic binary linear error-correcting $(6, 2)$-codes is less than or equal to $C^2_{11} = 55$.

5. Cyclic codes

Cyclic codes are a special case of linear codes. They are successfully implied in so called "antinoise coding" because of easy hardware representation based on a shift register with regenerative connection. Such codes are resistant to cyclic shift in addition to the structure of linear space.

Let the following parameters be known:

- n, the code length;
- q, the potency of the finite field, over which the code is built;
- and k, the quantity of informative symbols.

It is reasonable that the question of quantity of different cyclic codes with values n, q and k fixed arises. Educational materials provide a range of special problems to illustrate this topic [12]. The formula for the quantity of irreducible monic polynomials of a given degree over a finite field is also known, but it does not solve our problem as a generator polynomial of a cyclic code must fulfill the additional requirements listed below and may or may not be irreducible. Coding principles and contemporary application of binary cyclic codes are described in [14], the same for non-binary ones is given in [12] and some others.

Let F_q be a finite field of q elements. It is known that q is a power of a prime number. A generating polynomial $g(x)$ of a cyclic (n, k)-code over a finite field F_q meets the following requirements: $g(x)$ is a monic polynomial; a degree of $g(x)$ equals to $n-k$; the polynomial x^n-1 is divisible by $g(x)$ in the polynomial ring $F_q[x]$.

The converse statement is also true [12]: any nontrivial monic factor $g(x)$ of the polynomial $x^n - 1$ in the polynomial ring $F_q[x]$ generates some cyclic code with the quantity of check-digits that equals a power of $g(x)$.

Let us assume that there is polynomial factorization of $x^n - 1$ into irreducible monic polynomials over the field F_q

$$x^n - 1 = f_1(x)f_2(x)\dots f_s(x).$$

Then the quantity of different n-digit long cyclic codes over the field F_q equals the number of different nontrivial monic factors of the polynomial $x^n - 1$, i.e., equals to $2^(s) - 2$ (from the total number of factors, 2 trivial ones were subtracted). We have also counted all the possible codes with arbitrary quantity of informative symbols. We would like to note that there may be inequivalent codes among these.

6. Conclusions

Thus, we have found a general solution using computing machinery. We have used methods of dynamic programming as well as reducing the given problem to the knapsack one in practice. We also have counted some constants connected with coding theory, which can be used in some problems of genome analysis.

As we said before, the authors believe that communication work with genome analysis and coding theory needs to continue.

7. Acknowledgements

The authors of the article express their gratitude to Ksenia Krasheninnikova (St. Petersburg State University, Russia) and Vasily Dolgov (Togliatti State University, Russia) for their detailed review of the books and articles cited in this paper.

The reported study was partially supported by RFBR according to the research project No. 16-47-630829.

8. References

[1] Akay, M. *Genomics and proteomics engineering in medicine and biology* / M. Akay (Ed.) — N.Y.: John Wiley & Sons, 2007. — 450 p.

[2] Yockey, H. *Information theory and molecular biology* / H. Yockey — Cambridge: Cambridge University Press, 1992. — 273 p.

[3] Yockey, H. *Information theory, evolution, and the origin of life* / H. Yockey — Cambridge: Cambridge University Press, 2005. — 259 p.

[4] Liebovitch, L. S. *Is there an error correcting code in DNA?* / L. S. Liebovitch, Y. Tao, A. Todorov, L. Levine // Biophys. J. — 1996. — Vol. 71. — P. 1539–1544.

[5] Moore, J. *Investigation of coding structure in DNA* / J. Moore, G. Rosen // In: ICASSP-2003, IEEE International Conference on Acoustics, Speech, and Signal Processing. Hong Kong — 2003.

[6] Šponer, J. *Computational studies of RNA and DNA* / J. Šponer, F. Lankaš (Ed.) — Berlin: Springer, 2006. — 638 p.

[7] Levenshtein, V. *Binary codes capable of correcting deletions, insertions, and reversals* / V. Levenshtein // Soviet Physics Doklady. — 1966. — Vol. 10 (8). — P. 707–710.

[8] Hayes, B. *The Invention of the Genetic Code* / B. Hayes // American Scientist. — 1998. — Vol. 86. — P. 8–14.

[9] MacDonaill, D. *A parity code interpretation of nucleotide alphabet composition* / D. MacDonaill // Chemistry Communication. — 2002. — Vol. 18. — P. 2062–2063.

[10] Melnikov, B. *On a parallel implementation of the multi-heuristic approach in the problem of comparison of genetic sequences* / B. Melnikov, A. Panin // Vektor Nauki of Togliatti State University. — 2012. — Vol. 4 (22). — P. 83–86. — (in Russian). `https://elibrary.ru/item.asp?id=18755384`

[11] Melnikov, B. *Various algorithms, calculating distances of DNA sequences, and some computational recommendations for use such algorithms* / B. Melnikov, S. Pivneva, M. Trifonov // CEUR Workshop Proceedings. — 2017. — Vol. 1902. — P. 43–50.

[12] Lidl, R. *Applied Abstract Algebra* / R. Lidl, G. Pilz — Berlin: Springer, 1984. — 744 p.

[13] Ignatieva, I. *Approximation of a semigroup of characters with homomorphisms into a multiplicative semigroup of a finite field* / I. Ignatieva, S. Korabelschikova // Arctic Environmental Research. Series: Science. — 2011. — Vol. 1. — P. 107–110. – (in Russian). `https://elibrary.ru/item.asp?id=16389001`

[14] Berlekamp, E. R. *Algebraic Coding Theory* / E. R. Berlekamp — N.Y.: McGraw-Hill Book Company, 1968. — 474 p.