Linear codes and some their applications

S Y Korabelshchikova¹, B F Melnikov², S V Pivneva² and L V Zyablitseva¹

¹Northern (Arctic) Federal University named after M. V. Lomonosov, Severnaya Dvina Emb. 17, Arkhangelsk, Russia, 163002
²Russian State Social University, Wilhelm Pieck str., 4, Moscow, Russia, 129226

e-mail: s.korabelsschikova@narfu.ru, bf-melnikov@yandex.ru, tlt-swetlana@yandex.ru

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.

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 evolutionist 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.
The MacDonaill’s interpretation of the parity check code of the nucleotide composition [9] is also related to the topic discussed in this paper. 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 in this paper.

The paper has a following structure.

In Section 2, we briefly consider some algebraic terms necessary for this paper. Let us assume that symbols, of which the initial and the code messages consist, are elements of some finite field \( \mathbb{F}_q \), where \( q \) is the potency of the field. The structure of finite fields and their contemporary application have been described in [14, 18] and other. Let us also assume that the noise involves replacing some symbols with different elements of the field \( \mathbb{F}_q \), 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:
The minimal distance of a code and is denoted by \( d \), where these words are different. Minimal pairwise distance of different code vectors is called the Hamming distance between the \( x \) 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
\[
H = (A \mid E_{n-k}),
\]
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 \( d_{\text{min}}(C) \) 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.\(^1\)

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. Below, we shall count such matrices.

3. Some problems in the data integrity control

The data integrity control, as well as the error correction, are important tasks at many levels of working with information. In order to detect and correct errors, noiseless block codes are used. The original message is divided into blocks of \( k \) symbols, and each block is replaced by the corresponding code vector of the length \( n > k \), where \( n - k \) positions are verified. One of the authors previously solved the problem of estimating the number of different linear and cyclic codes with given parameters in [12, 13].

Here, we shall not assume that the code has additional good properties, that is, the binary codes discussed in this article are not generally linear in the general case. For the fixed code of the length \( n \) and minimum distance \( d \), the problem arises of estimating the number \( A(n, d) \), i.e. the maximum possible number of vectors in the code. It is necessary to find the maximum subset of binary vectors of the length \( n \), such that the Hamming distance between any pair of

\(^1\) The square brackets \([\]\) mean the integral part of the quotient.
vectors in this set is not less than \( d \). The formulation of this problem and the results of its solution known at that time are given, see for example [14, p. 262].

It can be reduced to one of the following three problems:

(i) packing balls of radius \( d/2 \) into a Boolean cube;
(ii) search for the maximum independent subset in the graph, where:
- the vertices are all \( 2^n \) supposed variants of codes,
- the edges exist between those (and only those) vertices, which Hamming distance is less than \( d \);
(iii) the problem of finding the maximum clique; it is inverse to Problem 2.

These items are important because of the following fact: each of these (similar) formulations describes their own decision algorithms. The latter especially applies to heuristic algorithms that give a solution close to optimal.

For some values of parameters, the answer \( A(n, d) \) is a priori known, the values can be found in [15]. Since in the general case, the algorithms for solving these problems work exponentially, we need to search for more efficient algorithms.

For this thing, we propose a randomized algorithm that probabilistically tries to find a solution to Problem 2. The algorithm recursively considers all \( 2^n \) variants of binary vectors. In the event that the current binary vector does not conflict with the set of already selected vectors, it is added to the formed set.

This heuristic implementation gives a pseudo-optimal answer; however, as we said before, it is close to the optimal answer.\(^2\) To find a more effective solution, the following conditions should be introduced:

- the next bit in the recursive generation of a binary vector is selected randomly;
- if a binary vector is suitable, then we do not take it with a certain probability.

Thus, we have obtained an algorithm that quickly\(^3\) finds the answer for, e.g., \( n = 9 \) and \( d = 3 \).

The results of the program implemented on the basis of the described algorithm are given in Table 1 below. It is important to note that they coincide with the results from [14, p. 262], i.e. for small dimensions, optimal solutions are obtained.

| \( n \) | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|------|----|----|----|----|----|----|----|
| \( A(n, 3) \) | 2  | 4  | 8  | 16 | 20 | 40 |    |

Concluding this section, we note that a similar problem can be posed for non-binary codes, and studies in this direction can lead to new interesting results.

4. 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.

---

\(^2\) See [16, 17] etc. for the description of our approaches to the development of heuristic algorithms.

\(^3\) We mean a few minutes, on the “ordinary” modern computer technology: processor clock speed is of about 2 GHz.
Algorithm 1 (counting check-matrices).
The algorithm 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 the last 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. 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 $\alpha_1$ is $(a_{11}, a_{12}, \ldots, a_{1n})$;
- binary representation of the number $\alpha_2$ is $(a_{21}, a_{22}, \ldots, a_{2n})$;
- \ldots;
- and binary representation of the number $\alpha_m$ is $(a_{m1}, a_{m2}, \ldots, a_{mn})$,

let us find the numbers $\beta_1, \beta_2, \ldots, \beta_n$, 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 $\beta_1$;
- $(a_{12}, a_{22}, \ldots, a_{m2})$ is the binary representation of the number $\beta_2$;
- \ldots;
- $(a_{1n}, a_{2n}, \ldots, a_{mn})$ is the binary representation of the number $\beta_n$.

If the following conditions hold:

- all the numbers $\beta_1, \beta_2, \ldots, \beta_n$ 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 $S - m!$.

(End of the description of the algorithm)

The output of the program based on this algorithm is given in Table 2; 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 2. The number 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|

As a continuation of the present topic, we note the following possible direction. Obviously, the values in the table undergo the so-called “combinatorial explosion”, i.e., it is hardly possible to obtain a value for the algorithms described by us, for example, for the pair \( m = 8, n = 10 \). In this connection, the following problems arise:

1. A description of the most accurate upper and lower bounds for the number of counted matrices corresponding to large values \( m \) and \( n \);
2. A description of the heuristic algorithms that compute these values approximately – not necessarily randomized algorithms (as in the previous section), but also built on some other principles.

5. Estimating the number of linear error correcting codes

We can see that the number of matrices found in Section 4 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 $4 \times 6$, the quantity of inequivalent systematic binary linear error-correcting $(6,2)$-codes is less than or equal to $C^2_{11} = 55$.

6. 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 [14]. 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 [19], the same for non-binary ones is given in [14] 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 [14]: 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 $F_q$

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

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.
7. Conclusion
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, e.g., in the problems of genome analysis.
Let us repeat the short description of some problems for the following solution. The problem
similar to one considered in Section 3, can be posed for non-binary codes, and studies in this
direction can lead to new interesting results. And for Section 4, the following problems arise:
we should make a description of the most accurate upper and lower bounds for the number of
counted matrices corresponding to large values m and n. Also, we should make a description of
the heuristic algorithms that compute these values approximately.
As we said before, the authors believe that communication work with genome analysis and
coding theory needs to continue.
Here are the links to the recent works by authors related to the subject matter discussed in
this paper: [20, 21].

8. References
[1] Akay M (Ed.) 2007 Genomics and proteomics engineering in medicine and biology
(New York: John Wiley & Sons)
[2] Yockey H 1992 Information theory and molecular biology (Cambridge: Cambridge
University Press)
[3] Yockey H 2005 Information theory, evolution, and the origin of life (Cambridge: Cambridge
University Press)
[4] Liebovitch L S, Tao Y, Todorov A and Levine L 1996 Is there an error correcting code in DNA
Biophys. J. 71 1539-1544
[5] Rosen G and Moore J 2003 Investigation of coding structure in DNA ICASSP-2003, IEEE
International Conference on Acoustics, Speech, and Signal Processing Proceedings 2 361-364
[6] Sponer J and Lankas F (Eds.) 2006 Computational studies of RNA and DNA (Berlin: Springer)
[7] Levenshtein V 1966 Binary codes capable of correcting deletions, insertions, and
reversals Soviet Physics Doklady 10(8) 707-710.
[8] Hayes B 1998 The Invention of the Genetic Code American Scientist 86 8-14
[9] MacDonail D 2002 A parity code interpretation of nucleotide alphabet composition
Chemistry Communication 18 2062-2063
[10] Melnikov B and Panin A 2012 On a parallel implementation of the multi-heuristic
approach in the problem of comparison of genetic sequences Vektor Nauki of Togliatti
State University 4(22) 83-86 (in Russian)
[11] Melnikov B, Pivneva S and Trifonov M 2017 Various algorithms, calculating distances
of DNA sequences, and some computational recommendations for use such algorithms CEUR
Workshop Proceedings 1902 43-50
[12] Korabelshchikova S and Chesnokov A 2013 On the number of different cyclic codes of a
given length Vektor Nauki of Togliatti State University 4(26) 25-26 (in Russian)
[13] Zyublitsseva L, Korabelshchikova S and Chesnokov A 2014 On the number of different cyclic
codes of a given length Heuristic algorithms and distributed calculations 1(1) 47-59 (in Russian)
[14] Lidl R and Pilz G 1984 Applied Abstract Algebra (Berlin: Springer).
[15] Table of general binary codes (Access mode: https://www.win.tue.nl/ aeb/codes/
binary-1.html, free (title from the screen)
[16] Melnikov B 2006 Multiheuristic approach to discrete optimization problems Cybernetics and
Systems Analysis 42(3) 335-341
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 the research project of Russian State Social University. It was also partially supported by RFBR according to the research project No. 16-47-630829.

[17] Melnikov B and Tsyganov A 2012 The state minimization problem for nondeterministic finite automata: the parallel implementation of the truncated branch and bound method Proceedings. 5th International Symposium on Parallel Architectures, Algorithms and Programming (Taipei) pp 194-201
[18] Ignatieva I and Korabelshchikova S 2011 Approximation of a semigroup of characters with homomorphisms into a multiplicative semigroup of a finite field Arctic Environmental Research. Series: Science 1 107-110 (in Russian)
[19] Berlekamp E R 1968 Algebraic Coding Theory (New York: McGraw-Hill Book Company)
[20] Melnikov B Korabelshchikova S and Churikova N 2017 On verification algorithms for some binary relations on the general supermonoid of a free monoid Izvestiya of Higher Educational Institutions. Volga Region. Physics and Mathematical Sciences 3(43) pp 87-99 (in Russian)
[21] Melnikov B and Trenina M 2018 On a problem of the reconstruction of distance matrices between DNA sequences International Journal of Open Information Technologies 6 1-13 (in Russian)