



(51) International Patent Classification:

G11C 13/00 (2006.01) G06N 3/12 (2006.01)

(21) International Application Number:

PCT/EP2020/064648

(22) International Filing Date:

27 May 2020 (27.05.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

1907460.8 27 May 2019 (27.05.2019) GB

(71) Applicants: **VIB VZW** [BE/BE]; Rijvisschestraat 120, 9052 Gent (BE). **KATHOLIEKE UNIVERSITEIT LEUVEN, K.U.LEUVEN R&D** [BE/BE]; Waaistraat 6 - bus 5105, 3000 Leuven (BE).

(72) Inventors: **STIRPARO, Rocco**; Bondgenotenlaan 152 bus 01.01, 3000 Leuven (BE). **AMMIRATI, Antonio**; Martelarenlaan 11, box 401, 3010 Kessel-Lo (BE). **MOISSE, Matthieu**; Eugene Vanhoorenbekelaan 29, Bus 1, 3010 Kessel-Lo (BE).

(74) Common Representative: **VIB VZW**; Rijvisschestraat 120, 9052 Gent (BE).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

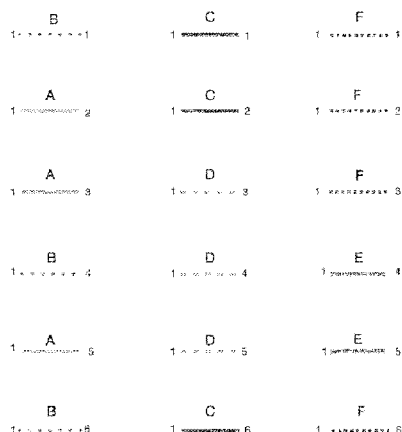
- with international search report (Art. 21(3))
- with sequence listing part of description (Rule 5.2(a))

(54) Title: A METHOD OF STORING DIGITAL INFORMATION IN POOLS OF NUCLEIC ACID MOLECULES

Figure 3

120 – Convert each binary element into combinations of data-oligonucleotides

130 – Add element IDs to each data-oligonucleotide for each combination



140 – Pool all combinations of nucleic acid molecules

(57) Abstract: The invention relates to data storage and retrieval. More particularly the invention provides means and method for storing files of digital of information in pools of individual nucleic acid molecules with a minimal DNA synthesis effort. The data-into-DNA storage system herein described is furthermore characterized by a high bit per nucleotide density and easy error detection and correction means.



A METHOD OF STORING DIGITAL INFORMATION IN POOLS OF NUCLEIC ACID MOLECULES**Field of the Invention**

The invention relates to data storage and retrieval. More particularly the invention provides means and method for storing files of digital of information in combinations of nucleic acid molecules with a minimal DNA synthesis effort. The data-into-DNA storage system herein described is furthermore characterized by a high bit per nucleotide density and easy error detection and correction means.

Background of the Invention

Deoxyribonucleic acid (DNA) molecules are a promising medium for storing digital data. Compared to the standard data storage systems, DNA as information carrier requires very low maintenance and limited physical space. Moreover, it is known that DNA molecules remain stable for hundreds of years. In the last years, there have been several studies and patent applications that have demonstrated that data storage is possible by using small DNA molecules (oligonucleotides with a length of less than 200 nucleotides) or larger DNA molecules (>200 nucleotides) (e.g. WO2013178801A2, WO2014014991A2, EP3173961A1, WO2015193140A1, US20160168579A1, WO2017011492A1, WO2017190297A1, WO2018005117A1, WO2018057526A2, WO2018094108A1, WO2018132457A1, WO2018156792A1, WO2019020059A1). However, in practice most approaches are not fully compatible with the technological problems which are inherently linked to current DNA synthesis and sequencing technology. For example, many data-into-DNA storage methods do not provide solutions to overcome the need to synthesize and/or sequence DNA molecules comprising homopolymers, repetitions or a misbalance of G/C content. Above all, none of the prior art methods provide a solution to make the data-into-DNA storage economically viable. Indeed, although the cost of DNA synthesis has dropped 1000-fold over the last 20 years, converting 1 gigabyte of data using the most advanced methods having the highest byte/nucleotide density would still cost around 20.000 US dollar. In order to make data-into-DNA storage economically appealing the DNA production cost should decrease at least 10.000 times. Additionally, the speed of DNA synthesis is a bottleneck as storage of 1 gigabyte would take several days of production.

WO2018081566 discloses a method wherein each single bit in a bit string is represented by the presence of a pre-synthesized short nucleic acid storage molecule and wherein said molecule also represents the position of that single bit within the bit stream. *De novo* DNA synthesis is thereby avoided and the presence of bits in a bit stream is translated to the presence of pre-synthesized DNA molecules in a pool of DNA molecules. WO2018081566 anticipates variations to said method wherein a nucleic acid storage molecule will represent a single byte within a byte stream. In that case, a higher number of starting DNA

molecules is required. To store a 50 byte digital fragment, 12800 different nucleic acid molecules would be needed. Besides the need of decreasing the number of these starting nucleic acid storage molecules, the WO2018081566 approach has additional limitations impairing the method to become economically appealing. A first limit is the density. In methods using *de novo* DNA synthesis, for example that of
5 Organick et al. (2018 Nat Biotech 36: 242-249 or US20180265921), each oligonucleotide can store about 15 bytes (or 120 bits) of information. According to WO2018081566 each oligo stores only 1 bit or alternatively 1 byte, hence a higher amount of resources is required for retrieving the digital information. A second limit is the limited error detection. The WO2018081566 method translates the presence of a binary digit into the presence of a specific oligonucleotide. As a result, "oligo-dropout" is not detectable
10 and thus cannot be corrected. Increasing the number of starting nucleic acid storage molecules and introducing redundancy might help to avoid this problem, but then the density is further decreased and the number of starting nucleic acid storage molecules that need to be synthesized is increased. It would thus be advantageous to develop means and methods of storing digital information into pre-synthesized and reusable short nucleic acid molecules, wherein the number of starting nucleic acid
15 molecules is low, the byte per nucleotide density high and wherein the method provides an easy and reliable error detection mechanism.

Summary of the Invention

Here, Applicants disclose a novel method to store digital information into a set of selected, diverse and
20 pre-synthesized nucleic acid molecules that are optimized for synthesis and sequencing purposes. Unique combinations of said nucleic acid molecules are translated into specific binary elements, which can comprise 2, 3, 4, 5, 6, 7, 8 or more bits.

Alternatively phrased, the application provides in a first aspect, a method of storing a file of digital
25 information using nucleic acid molecules, said file consisting of a plurality of binary elements, said method comprises:

- (a) converting each binary element from said file into a group of at least 2 individual nucleic acid molecules using a selected dictionary or conversion table, wherein said at least 2 individual nucleic acid molecules from one group differ from each other in nucleic acid sequence;
- 30 (b) providing to each of the at least 2 nucleic acid molecules of a group representing a binary element, the information of the position of said binary element in said file, wherein said information is represented by a nucleic acid sequence;
- (c) storing the file of digital information by pooling all the nucleic acid molecules representing the plurality of binary elements.

In one embodiment, said method further comprises a step wherein the information determining the selected dictionary from step (a) is converted to a nucleic acid molecule that is added to the pool of nucleic acid molecules of step (c).

In another embodiment, the positional information of step (b) is determined by an element ID added at the 5'-end and/or an element ID added at the 3'-end of said at least 2 nucleic acid molecules. More particularly, said element ID is between 15 and 25 nucleotides long. In other particular embodiments, the binary element consists of between 4 and 32 bits. Even more particularly, said binary element consists of 32 bits and said selected dictionary converts binary elements of 32 bits into groups of 8 individual nucleic acid molecules.

In another aspect, a device for storing digital information is provided comprising pools of nucleic acid molecules organized according to the above methods.

In another aspect, a method of retrieving a file of digital information from a pool of nucleic acid molecules is provided, said method comprises the following steps:

- (a) amplifying a plurality of nucleic acid molecules from said pool optionally comprising one or more nucleic acid molecules determining the dictionary needed to retrieve said file;
- (b) sequencing the amplified nucleic acid molecules;
- (c) grouping the sequenced nucleic acid molecules according to the positional information comprised in said nucleic acid molecules;
- (d) converting each group of nucleic acid molecules from step (c) to a binary element using the dictionary of step (a), wherein the positional information determines the position of said binary element in the file of digital information;
- (e) constructing the file of digital information by connecting said binary elements to each other in the correct order.

In one embodiment, said method of retrieving digital information further comprises a step of correcting errors.

In yet another aspect, a computer system for converting a file of digital information into nucleic acid molecules and/or for retrieving a file of digital information from a pool of nucleic acid molecules is provided, the computing system comprises one or more processors and is configured for performing one of the methods disclosed in current application for converting a digital file into nucleic acid molecules and/or one of the methods disclosed herein for retrieving said file of digital information.

Also a computer program for converting a file of digital information into nucleic acid molecules and/or for retrieving a file of digital information from a pool of nucleic acid molecules is provided, the computer program comprising instructions which, when the computer program product is executed by a computer, cause the computer to carry out one of the methods disclosed in current application for
5 converting a digital file into nucleic acid molecules and/or one of the methods disclosed herein for retrieving said file of digital information.

Description of the Drawings

Figure 1 shows a workflow to obtain unique combinations of data-oligonucleotides. In step 100 a
10 collection of data-oligonucleotides each with a different nucleic acid sequence is created. In this non-limiting example said collection consists of 6 data-oligonucleotides. In step 110 unique combinations of the data-oligonucleotides from step 100 are created.

Figure 2 shows a dictionary that can be used to translate combinations of data-oligonucleotides in binary
15 elements. Each binary element can also be written as a decimal number. In this non-limiting example binary elements of 4 bits and groups of 3 data-oligonucleotides are used. Hence, using a collection of 6 different data-oligonucleotides, 20 different combinations of 3 oligonucleotides can be created of which only 16 are needed to represent all possible binary elements of 4 bits.

Figure 3 shows a workflow of the encoding method. In step 120 the digital file is translated by converting
20 each binary element into a combination of data-oligonucleotides. In step 130 element IDs are added to the data-oligonucleotides determining the position of each binary element in the file of digital information. In step 140 all combinations of nucleic acid molecules, each nucleic acid molecule comprising a data-oligonucleotide and at least 1 element ID, representing all binary elements that make
25 up the digital file are pooled.

Figure 4 shows a workflow of the method of retrieving digital information for stored nucleic acid
molecules. In step 150 a pool of stored nucleic acid molecules is amplified and sequenced. In step 160 the sequenced nucleic acid molecules are grouped based on the element ID they comprise. In step 170
30 every group of data-oligonucleotides is converted into a binary element. In step 180 all binary elements are linked to each other based on their position within the digital file to construct said digital file.

Figure 5 shows how digital information can be stored in nucleic acid molecules without the need for adding element IDs. The initial number of nucleic acid molecules (here 384) is first divided in blocks (here

6 blocks of each 64 nucleic acid molecules or oligos). Each block of 64 oligos is used to compose a group of k oligos (here k = 8) that represents 1 binary element (here 4 bytes). The positional information of the binary elements is thus inherently enclosed in the sequence of the nucleic acid molecules.

- 5 Figure 6 illustrates how the use of 600 ID sequences (i.e. 6 different series (A1, A2, A3, B1, B2, B3) of 100 ID sequences each) can encode 100^6 different positions within a digital file.

Detailed Description of the invention

10 The present invention will be described with respect to particular embodiments and with reference to certain drawings. It will be understood that the embodiments and aspects of the invention described herein are only examples and do not limit the protective scope of the claims in any way. The invention is defined by the claims and their equivalents. It will be understood that features of one aspect or embodiment of the invention can be combined with a feature of a different aspect or aspects and/or embodiments of the invention. Any reference signs in the claims shall not be construed as limiting the scope. The drawings described are only schematic and are non-limiting. In the drawings, the size of some of the elements may be exaggerated and not drawn on scale for illustrative purposes.

Where the term "comprising" is used in the present description and claims, it does not exclude other elements or steps. Where an indefinite or definite article is used when referring to a singular noun e.g. "a" or "an", "the", this includes a plural of that noun unless something else is specifically stated. 15 Furthermore, the terms first, second, third and the like in the description and in the claims, are used for distinguishing between similar elements and not necessarily for describing a sequential or chronological order. It is to be understood that the terms so used are interchangeable under appropriate circumstances and that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein.

25 The terms or definitions used herein are provided solely to aid in the understanding of the invention. Unless specifically defined herein, all terms used herein have the same meaning as they would to one skilled in the art of the present invention. Practitioners are particularly directed to Sambrook et al. (2012 Molecular Cloning: A Laboratory Manual, 4th ed., Cold Spring Harbor Press, Plainsview, New York) and Ausubel et al. (2016 Current Protocols in Molecular Biology (Supplement 114), John Wiley & Sons, New York) for definitions and terms of the art. Unless defined otherwise, all technical and scientific terms 30 used herein have the same meaning as commonly understood by one of ordinary skill in the art (e.g. in molecular biology, biochemistry, structural biology, and/or computational biology).

Storage of digital information in DNA becomes more and more evident. However, the high cost and slow speed of DNA synthesis remain huge bottlenecks and data-into-DNA storage is nowadays limited to “boutique applications”. DNA sequencing on the other hand, needed to retrieve the stored data, is much cheaper, and prices dropped exponentially over the last years. Most sequencing methods rely on enzyme-mediated DNA amplification which is extremely cheap, even compared to silicon chip production. Indeed, polymerase chain reactions (PCRs) producing billions of copies of a certain nucleic acid template can be done for a cost of about 1\$ and in extremely small volumes as well.

Therefore, Applicants anticipate that the future of data storage in DNA will not lay in *de novo* synthesis of nucleotide strings which are linearly translated from bit strings, but instead in the use of pre-synthesized, reusable nucleic acid molecules that can be easily and cheaply amplified and that can be pooled in specific combinations. Bits or bytes making up the digital file will not be translated anymore in a nucleic acid sequence but instead said bits or bytes will be represented by specific combinations of pre-synthesized nucleic acid molecules that additionally comprise information of the position of said bits or bytes in the digital fragment. A digital file or sequence will therefore be represented by one or more pools of nucleic acid molecules.

The present application relates to such a method for storage of digital information in one or more pools of nucleic acid molecules, more particularly in pools of short nucleic acid molecules. Said nucleic acid molecules can have any length shorter than 200 nucleotides but based on the current state and costs of technology a length of between 45 and 65 nucleotides is envisaged, more particularly a length of about 60 nucleotides. Said nucleic acid molecules comprise a digital data part (from here on referred to as “data-oligo” or “data-oligonucleotide”) and a positional information part (from here on referred to as “element ID”) and/or an amplification part such as primer binding sites or adaptors. Said amplification part is located at the 5’ end and/or 3’ end of said nucleic acid molecule. Where a specific combination of data-oligos specifies which binary element is encoded by said combination, the element IDs define the position of said binary element in the digital data file. In some cases, it is not needed to add element IDs (see later).

The core of the invention is that a set of optimized data-oligos is generated, that a unique combination of at least 2 of said data-oligonucleotides is translated into a specific binary element consisting of 2, 3, 4, 5, 6, 7, 8 or more bits or consisting of 2, 3, 4 or more bytes, that each data-oligo from a specific combination is supplemented with an element ID defining the position of the binary element (for which the combination of data-oligos is encoding) in the digital data file and/or supplemented with an amplification part and that the information from said digital data file is converted into groups or combinations of nucleic acid molecules that are subsequently pooled together. Note that every binary

element is represented by a combination or group of nucleic acid molecules and not by one nucleic acid molecule alone. Hence, instead of converting binary data (comprising bits and/or bytes) to a nucleotide sequence, the bits and/or bytes that make up the binary data are represented by a number of individual nucleic acid molecules. Byte x at position y in the binary data is then encoded by the presence of at least 2 nucleic acid molecules jointly comprising a combination of data-oligos specific for byte x and optionally an element ID for position y. For example, a binary sequence 00001101011101000100101000101001 is not anymore translated to for example AATGCTGCAATTGCATAGCC but to a pool comprising 4 groups of at least 2 nucleic acid molecules. In a particular embodiment, the term “nucleic acid molecules” that is used throughout this document refers to DNA molecules.

The term “binary element” as used herein refers to a specific number of bits or bytes. In one embodiment, a “binary element” is a 2-bit, 3-bit, 4-bit, 5-bit, 6-bit, 7-bit or 8-bit fragment of a file of digital information. In another embodiment, a “binary element” is a 2-byte, 3-byte or 4-byte fragment of a file of digital information. The term “bit” as used herein refers to a binary digit, i.e. the smallest unit of binary data in a computing or data storage system. A sequence of bits is herein called a “bit string” or a “binary sequence”. A bit string can consist of 2, 3, 4, 5, 6, 7, 8 or more bits. A bit string consisting of 8 bits is called a “byte”.

The methods of current application which are described in more detail below have several advantages over the prior art. First, because a combination of at least 2 nucleic acid molecules is used to translate 1 binary element, a very low number of starting nucleic acid molecules is needed, which further reduces the data synthesis/storage cost. Second, the methods herein disclosed result in a data-into-DNA storage with high density. Third, the methods allow both error detection and error correction.

Current application is based on the use of pre-synthesized data-oligonucleotides. Therefore, a collection of these oligonucleotides should be available (Figure 1). Said data-oligonucleotides can have a variable length. In particular embodiments, the length of said data-oligonucleotides is between 6 and 30, between 7 and 25 or between 8 and 20 nucleotides. In more particular embodiments, the data-oligonucleotides have a length of 12, 13, 14, 15, 16, 17 or 18 nucleotides. In most particular embodiments all data-oligonucleotides have the same length.

Theoretically any oligonucleotide can be used, however the nucleotide sequence must be different enough to easily detect the different data-oligonucleotides. Preferably, at least 2 nucleotides are different between 2 different data-oligonucleotides. Also the collection of data-oligonucleotides can consists of a variable number of oligonucleotides depending on the storage capacity needs. In one embodiment, a collection of between 4 and 10, or between 8 and 25, or between 10 and 50, or between 12 and 80, or between 64 and 256, or between 128 and 384, or between 265 and 1024 data-

oligonucleotides is used. In the non-limiting example illustrated in Figure 1 (Step 100), a collection of 6 different data-oligonucleotides is designed (called A – B – C – D – E – F). In another non-limiting example, a collection of 64 different data-oligonucleotides is created or a multiple of it.

5 One of the crucial aspects of current application that makes it novel and inventive compared to the prior art is that every binary element (which can be anything between 2 bits and a plurality of bytes) is translated into a combination of different and at least 2 data-oligonucleotides. Therefore, from the above described collection of n data-oligonucleotides (wherein n is at least 3), unique combinations of oligonucleotides are made (Figure 1, step 110). These combinations or groups of data-oligonucleotides
 10 can comprise k different data-oligonucleotides, wherein $k < n$ and k is at least 2, at least 3, at least 4, at least 5, at least 6, at least 7, at least 8, at least 9 or at least 10. In each group, the order of the oligonucleotides is not relevant (e.g. "A - B - C" is the same as "C - B - A"), but repetitions are not allowed, and each data-oligonucleotide counts only once. Thus, considering that the pool contains n data-oligonucleotides that we want to combine in groups consisting of k data-oligonucleotides,
 15 mathematically, the number of such combinations is then $n!/k!(n-k)!$. In a non-limiting example illustrated in Figure 1 (step 110), 6 data-oligos are combined in groups of 3, therefore $n=6$ and $k=3$. In total, there are 20 possible different combinations (Figure 1, step 110). It is clear for the person skilled in the art that by changing the number of starting data-oligos in the collection (i.e. n) or the size of the groups (i.e. k or the number of data-oligonucleotides in the groups) a different amount of combinations
 20 can be generated. For example, a collection of 64 members that are combined in groups of 8 data-oligonucleotides can generate 4.426.165.368 possible combinations. Throughout the application "group" and "combination" will be used interchangeably.

To translate the obtained groups of data-oligonucleotides (see above) into binary elements, a plurality of dictionaries can be made. A dictionary is a type of hash table and defines which group of data-oligonucleotides is connected to which binary element, e.g. 4 bits, 1 byte, 4 bytes ... In the non-limiting
 25 example illustrated in Figure 2, a dictionary links 16 unique combinations (made of 3 data-oligonucleotides each constructed from a collection of 6) to the 16 possible binary elements consisting of 4 bits. The skilled person will appreciate that a plurality of dictionaries can be obtained. First, different ones of the dictionaries can be easily generated by changing the associations between the combinations
 30 of data-oligonucleotides and the binary elements in the dictionaries. Second, the number of bits in a binary element or the number of data-oligonucleotides per combination can be changed. It is important to note that although different dictionaries can be easily generated, a single dictionary or conversion table suffices. Yet, in a particular embodiment the term "selected one of a plurality of dictionaries" that is used throughout the application, refers to a selected dictionary or to a dictionary or conversion table.

In the non-limiting example illustrated in Figure 2, 16 combinations have been ordered and each group is associated with a different binary element, for example starting from 0000 to 1111. With such a selected one of a plurality of dictionaries, digital information can then be translated (binary element per binary element) in combinations of data-oligonucleotides. Logically, the information about the position of all binary elements within the binary data should be stored in the system, as well as information about which dictionary is used (see further).

In a first step of the storing method disclosed herein, the digital file (that has to be translated into nucleic acid molecules) is converted binary element after binary element (Figure 3, step 120). The length of the binary element depends on the length of binary elements used in the dictionaries as explained above. If a dictionary is constructed to translate combinations of data-oligonucleotides to a stretch of for example 4 bits, then the digital file needs to be read in fragments of 4 bits and converted 4 bits per 4 bits. Similarly, if the binary fragments in the dictionary to be used are bytes, the digital file needs to be converted byte per byte. As shown in the non-limiting example illustrated in Figure 2, with 16 different combinations of data-oligonucleotides, the binary element that can be converted into a combination of data-oligonucleotides is maximum 4 bits. The binary element 1100 can then for example be translated into the combination of data-oligonucleotides "B-C-F", while the binary element 0110 can be translated with the data-oligonucleotide combination "A-C-F". Which binary element will be translated into which combination of data-oligonucleotides entirely depends on the dictionary used.

In case more combinations are created, a different length of the binary element can be used. For example, each of the 4.426.165.368 combinations generated with $n=64$ and $k=8$ (see above) can translate strings of 32 bits of information into one combination of 8 oligonucleotides.

In one embodiment, to each data-oligonucleotide from a combination of data-oligonucleotides a "element ID" or a positional information part is added that specifies the position of the binary element (translated by said combination) within the digital file. Said element ID is a nucleic acid sequence as well.

In one embodiment, said element ID consists of 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 or more nucleotides. In particular embodiments, said element ID is a primer recognition site and can additionally be used to randomly accessing the digital information. As

2 primer recognition sites are needed to amplify a nucleic acid molecule with PCR, an embodiment is provided that 2 element IDs can be added to each data-oligonucleotide from each combination of data-oligonucleotides. In one embodiment, a first element ID is added at the start (or the 5'-end) of each data-oligonucleotide and a second element ID is added at the end (or the 3'-end) of said data-oligonucleotide. In most particular embodiments, said element ID has a length of between 15 and 30 nucleotides,

between 18 and 25 nucleotides or between 19 and 22 nucleotides. In a most particular embodiment, said element ID has a length of 20 or 21 nucleotides.

“Adding” as used herein refers to a merging, joining, uniting, aggregating, thus molecularly linking the element ID or IDs with the data-oligonucleotide in order to obtain one nucleic acid molecule. Said molecular linking can be done using several strategies, for example enzymatically or chemically. The most straightforward enzymatic method is the Polymerase Chain Reaction (PCR), which is well known by the skilled person. Briefly, PCR is an amplification method of a specific target DNA molecule. When PCR primers, i.e. small DNA molecules of about 20 nucleotides that anneal to the 5'- and 3'-end of the target DNA molecule, are added together with a DNA polymerase enzyme, said enzyme will extend the primer molecules using the DNA target as template following the 5'-to-3' direction and thus completely copy the DNA target sequence. Interestingly, the 5'-end of each primer can be different from the target DNA molecule, since there is no need of annealing in that extremity. For the purpose of current application, all element IDs are primer sequences that have different 5'-ends which define the element ID and a 3'-end that allows annealing to all the data-oligonucleotides from a collection.

Another enzymatic method is ligation. In this case, the target DNA (e.g. a data-oligonucleotide) is not amplified but the element ID is molecularly ligated to it by a ligase enzyme.

Besides above described linking methods, also chemical linking of data-oligonucleotides to one or more element IDs is envisaged, for example DNA crosslinking. It is clear that the method used to add element ID(s) to data-oligonucleotides is not limiting the invention.

In a non-limiting example as illustrated in Figure 2, the combination of data-oligonucleotides B, C and F represents the binary element 1100. If these data-oligonucleotides are supplemented with element ID “1” at the start (in the 5'-end of the oligo) and end (in the 3'-end of the oligo) (as in Figure 3 step 130), said group of nucleic acid molecules represents the binary element 1100 at position 1 of the digital file.

However and according to the non-limiting example in Figure 3 step 130, if oligonucleotides B, C and F are supplemented with element ID “1” at the start and element ID “6” at the end, they would jointly represent the binary element 1100 but at position 21 of the digital file or alternatively phrased 1100 would then be the 6th binary element in said file.

It should be clear to the skilled person that the longer the element ID is, the more different element IDs can be generated and/or the more correction possibilities will be present when an error in an element ID would occur. It is also clear that the amount of unique element IDs will determine how many binary elements (and thus the amount of digital information) can be stored in 1 pool of combinations of oligonucleotides.

In one aspect, current application thus provides a method of storing digital information using nucleic acid molecules, said digital information is encoded in a fragment consisting of a plurality of binary elements, said method comprises:

- (a) converting each binary element from said fragment into a group of at least 2 individual nucleic acid molecules (or data-oligonucleotides) using a selected one of a plurality of dictionaries, wherein said at least 2 individual nucleic acid molecules (or data oligonucleotides) from one group differ from each other in nucleic acid sequence;
- (b) adding or providing to each of the at least 2 nucleic acid molecules (or data oligonucleotides) of each group representing a binary element as obtained in step (a), the information of the position of said binary element in said fragment (or element ID), wherein said information (or element ID) is represented by a nucleic acid sequence;
- (c) storing the digital information encoded by said fragment by pooling all the nucleic acid molecules of step (b).

Only when the amount of data to be stored is too large to be stored in 1 group of nucleic acid molecules, positional information inherently present in the sequence of the nucleic acid molecules or in the form of additional nucleic acid molecules is needed and thus in the latter case each nucleic acid molecule comprises a data-oligonucleotide and at least one element ID.

The term “differ from each other in nucleic acid sequence” means that the nucleic acid sequences from said at least 2 individual nucleic acid molecules are not identical and have at least 1 nucleotide difference. In particular embodiment, said at least 1 nucleotide difference is at least 2, at least 3, at least 4, at least 5, at least 6, at least 7 or at least 8 nucleotide differences.

“Pooling” of nucleic acid molecules as used herein means collecting or assembling said nucleic acid molecules in the same physical location, for example in 1 recipient. The nucleic acid molecules are thus physically together but are present as individual molecules. They are not merged to one big molecule and can thus still be easily separated from each other or individually amplified.

In one embodiment, said method further comprises a step wherein the information determining the selected one of a plurality of dictionaries from step (a) is converted to a nucleic acid molecule that is added to the pool of nucleic acid molecules of step (c). Example 2 of current application illustrates a non-limiting example of how a selected one of a plurality of dictionaries can be encrypted in the pool of stored nucleic acid molecules in a cost-efficient way. In one embodiment, said selected one of a plurality of dictionaries is the dictionary described in the Example 2 or is the dictionary produced according the methods of the third aspect of current application (see further). Importantly, the invention of storing digital data in pools of nucleic acid molecules as herein disclosed is not limited to the selected one of a plurality of dictionaries.

In one embodiment, said nucleic acid sequence representing the positional information of a binary element is the “element ID” as described above and has a length of between 15 and 30 nucleotides, between 18 and 25 nucleotides or between 19 and 22 nucleotides. In particular embodiments, said positional information of a binary element is encoded by one or more additional nucleic acid sequences or element IDs that are added at the 5′-end and/or at the 3′-end of said at least 2 nucleic acid molecules of a group. In more particular embodiments, said positional information of a binary element is encoded by 2 nucleic acid sequences, one at the 5′-end and one at the 3′-end of each nucleic acid molecule of each group.

Alternatively phrased, the application provides a method of storing digital information using nucleic acid molecules, said digital information is encoded in a fragment consisting of a plurality of binary elements, said method comprises:

- (a) converting each binary element from said fragment into a group of at least 2 individual nucleic acid molecules, wherein every nucleic acid molecule from a group comprises an identical element ID but a different data-oligonucleotide, said different data-oligonucleotides from said group jointly represent a specific binary element determined by a selected one of a plurality of dictionaries, said element ID determines the position of said specific binary element in said fragment;
- (b) pooling all the groups of at least 2 individual nucleic acid molecules of step (a), thereby storing the digital information encoded by said fragment.

An extremely high amount of element IDs can be generated, for example 4^{20} if an element ID of 20 nucleotides is used. Moreover, if combinations of 2 elements IDs are used (for example to allow PCR amplification) the number increases to $4^{20} \times 4^{20}$. As the cost to synthesize all these primers would be too high, it is more cost efficient to limit the number of element IDs to for example 100 and combine several series of element IDs. Using 2 different series of 100 element IDs and adding 2 element IDs (1 of each series) to every data-oligonucleotide in every group of data-oligonucleotides, 10.000 (i.e. 100×100) different positions can be identified. If, in a non-limiting example, a collection would be used of 64 unique data-oligonucleotides and every 4 bytes of digital information is represented by a group of 8 nucleic acid molecules, 40 kilobyte (4 bytes per position $\times$ 10.000 positions) of digital information can be stored. According to the methods herein disclosed and for this particular non-limiting example, only 64 data-oligonucleotides (of for example 15 nucleotides each) and 200 element IDs (of for example 20 nucleotides each) are needed for almost 40.000 bytes.

Nevertheless, in some cases the digital information that needs to be stored will exceed the theoretically storable amount (e.g. 40 kilobyte) determined by the number of element IDs (e.g. 2×100), size of the binary element (e.g. 4 bytes) and the number of data-oligonucleotides per group (e.g. 8). In that case the digital information will be first fragmented in separate digital files.

5 Every digital file will then be translated and stored in a separate pool comprising groups of nucleic acid molecules according to the methods described above. Every said pool representing each a specific digital file or fragment can be stored on a different physical location, for example in different wells of a multi-well plate. Alternatively, all pools can be stored on the same physical location but then the nucleic acid molecules of a group should not only be supplemented with an element ID (representing information of
10 the position of a binary element in a fragment) but also with an additional fragment ID (representing information of the position of said fragment in the digital file). In this not limiting example, a second or even a third layer of ID sequences (e.g. consisting of 2×100 fragment IDs each) can be added to the data-oligo. Therefore, with only 600 ID sequences (element and fragment IDs) and 64 data-oligonucleotides, 4 terabytes of information can be stored ($100 \times 100 \times 100 \times 100 \times 100 \times 100 \times 4 \text{ bytes} = 4 \text{ terabytes}$) (Figure 6).

15 The invention as disclosed herein is compatible with all lengths of stored nucleic acid molecules. For illustrative and non-limiting purposes, the nucleic acid molecules to be stored comprise primer recognition sites at the end and at the start (e.g. element IDs) of 20 nucleotides each and a central part (i.e. data-oligonucleotide which allows the identification of a binary element) of between 8 and 20
20 nucleotides. In one embodiment of the application, the nucleic acid molecules from current application are between 48 and 65 nucleotides long. In particular embodiments, the data oligonucleotide is 18 nucleotides and the final length of the oligonucleotide is 58 nucleotides.

In a second aspect, the current application provides a method of storing digital information using nucleic
25 acid molecules, said method comprises:

- (a) fragmenting a file X of digital information into a plurality of N fragments, wherein each of said N fragments consists of or is converted into a plurality of binary elements;
- (b) performing for each of said N fragments, the following steps:
 - i. converting each binary element from one of said N fragments into a group of at least 2
30 individual nucleic acid molecules (or data-oligonucleotides) using a selected one of a plurality of dictionaries, wherein said at least 2 individual nucleic acid molecules (or data-oligonucleotides) from one group differ from each other in nucleic acid sequence;
 - ii. providing to each of the at least 2 nucleic acid molecules (or data-oligonucleotides) of each group representing a binary element, the information of the position of said binary element in

said one of said N fragments (or element ID), wherein said positional information (or element ID) is represented by a nucleic acid sequence;

iii. storing the digital information of said one of said N fragments by pooling all the nucleic acid molecules from all the groups representing all binary elements of which said fragment is composed and their position within said fragment into a single pool.

In one embodiment, said method further comprises a step wherein the information determining the selected one of a plurality of dictionaries from step (b) (i.) is converted to one or more nucleic acid molecules that are added to the pool of nucleic acid molecules of step (b) (iii.).

Alternatively phrased, the application provides a method of storing digital information using nucleic acid molecules, said method comprises:

(a) fragmenting a file X of digital information into a plurality of N fragments, wherein each of said N fragments consists of or is converted into a plurality of binary elements;

(b) performing for each of said N fragments, the following steps:

i. converting each binary element from one of said fragments into a group of at least 2 individual nucleic acid molecules, wherein every nucleic acid molecule from a group comprises an identical element ID but a different data-oligonucleotide, said different data-oligonucleotides from said group jointly represent a specific binary element determined by a selected one of a plurality of dictionaries, said element ID determines the position of said specific binary element in said one of said fragment;

ii. pooling all the groups of at least 2 individual nucleic acid molecules of step (b)(i), thereby storing the digital information encoded by said fragment.

In one embodiment, said method further comprises a step wherein the information determining the selected one of a plurality of dictionaries from step (b) (i.) is converted to one or more nucleic acid molecules that are added to the pool of nucleic acid molecules of step (b) (ii.).

In one embodiment, said method further comprises a final step in which all single pools of nucleic acid molecules obtained in a step (b) (iii) are stored at different positions in a recipient (e.g. multi-well plate) or on a surface, wherein the y^{th} position in the recipient or on the surface corresponds with the y^{th} fragment in said file X and wherein y is an integer between 1 and N.

Alternatively, the application provides a method of storing digital information using nucleic acid molecules, said method comprises:

(a) fragmenting a file X of digital information into a plurality of N fragments, wherein each of said N fragments consists of or is converted into a plurality of binary elements;

(b) performing for each of said N fragments, the following steps:

- i. converting each binary element from one of said N fragments into a group of at least 2 individual nucleic acid molecules (or data-oligonucleotides) using a selected one of a plurality of dictionaries, wherein said at least 2 individual nucleic acid molecules (or data-oligonucleotides) from one group differ from each other in nucleic acid sequence;
- ii. adding to each of the at least 2 nucleic acid molecules (or data-oligonucleotides) of each group representing a binary element, the information of the position of said binary element in said file X of digital information (or element ID), wherein said information (or element ID) is represented by a nucleic acid sequence;
- iii. storing the digital information of said one of said N fragments by pooling all the nucleic acid molecules from all the groups representing all binary elements of which said fragment is composed and their position within said file X of digital information into a single pool.

In one embodiment, said method further comprises a step wherein the information determining the selected one of a plurality of dictionaries from step (b) (i.) is converted to a nucleic acid molecule that is added to the pool of nucleic acid molecules of step (b) (iii.).

Alternatively phrased, the application provides a method of storing digital information using nucleic acid molecules, said method comprises:

(a) fragmenting a file X of digital information into a plurality of N fragments, wherein each of said N fragments consists of or is converted into a plurality of binary elements;

(b) performing for each of said N fragments, the following steps:

- i. converting each binary element from one of said fragments into a group of at least 2 individual nucleic acid molecules, wherein every nucleic acid molecule from a group comprises an identical element ID but a different data-oligonucleotide, said different data oligonucleotides from said group jointly represent a specific binary element determined by a selected one of a plurality of dictionaries, said element ID determines the position of said specific binary element in said file X of digital information;
- ii. pooling all the groups of at least 2 individual nucleic acid molecules of step (b)(i), thereby storing the digital information encoded by said fragment.

In one embodiment, said method further comprises a step wherein the information determining the selected one of a plurality of dictionaries from step (b) (i.) is converted to a nucleic acid molecule that is added to the pool of nucleic acid molecules of step (b) (ii.).

In one embodiment of the methods of the first and second aspect, said at least 2 individual nucleic acid molecules is at least 3, at least 4, at least 5, at least 6, at least 7, at least 8, at least 9 or at least 10 nucleic acid molecules and can be generally referred to as k . It goes without saying that k is smaller than n , the total number of data-oligonucleotides used in the methods of the application.

5 In other embodiments of the methods of the first and second aspect, the information of the position of the binary element in the file of digital information which is added in step (b) or in step (b) (ii.) is determined by a combination of an element ID added at the 5'-end and an element ID added at the 3'-end of said at least 2 nucleic acid molecules. In particular embodiments, said element IDs are between 15 and 25 nucleotides long.

10 In other embodiments of the methods of the first and second aspect, said binary element consists of between 4 and 32 bits. More particularly, said binary element consists of 32 bits and said selected one of a plurality of dictionaries converts binary elements of 32 bits into a combination of 8 individual nucleic acid molecules.

15 A storage-efficient dictionary

In the above, it is described that one of a plurality of dictionaries is needed. These dictionaries can be user-adapted. Logically, the information describing the used dictionary should also be provided along the digital information that is stored in nucleic acid molecules. In order to reduce this additional amount of information to be stored, a storage-efficient dictionary is disclosed herein. Said storage-efficient
20 dictionary can be one of the plurality of dictionaries described in the application.

To illustrate the method, let's take the non-limiting illustration of Figure 2. In Figure 2, three data-oligos from a collection of 6 data-oligos (A B C D E F) generate in total 20 combinations. The data-oligo are first numbered in a consecutive way: A is number 1, B number 2, etc. In a second step, combinations of said data-oligos are made and the combinations are ordered from low to high based on the number each
25 data-oligo was given in the previous step. Combination A B C thus comprises the lowest numbers, i.e. 1, 2 and 3. Combination A B D the one-but-lowest numbers, i.e. 1, 2 and 4. Next, based on said ordering, a decimal number starting from 0 is attributed to every combination of data-oligos. Combination A B C will thus be attributed decimal number 0, combination A B D decimal number 1, etc. (see Figure 2 for an illustration). According to general mathematical rules every sequence of bits (i.e. binary sequence) can
30 be converted to a ternary, quaternary, ... and also to a decimal number. With "decimal number equivalent to a specific binary element" is thus meant the decimal number to which every binary element can be converted to. As such every binary element can be converted in a combination of data-oligos or vice versa every sequenced combination of data-oligos can be converted into a binary sequence.

By making use of said dictionary a limited amount of information needs to be stored into nucleic acid molecules to retrieve the digital data (see later). Indeed, by providing the number of nucleic acid molecules in the collection used (i.e. n), the numerical order of said n nucleic acid molecules, the number of nucleic acid molecules in every combination (i.e. k), and the number of bits from every binary element (i.e. m), all combinations of nucleic acid molecules can be converted to binary information without the need of a stored dictionary.

In a third aspect, a method to generate or produce a dictionary is provided, said dictionary being used to convert a file of digital information comprising a plurality of binary elements each consisting of m bits into groups of k nucleic acid molecules (or vice versa), said method comprises the following steps:

- (a) provide a collection of n nucleic acid molecules and number every nucleic acid molecule from 1 to n (in a consecutive way);
- (b) make groups of k nucleic acid molecules wherein $2 \leq k < n$ and order said groups numerically according to the nucleic acid molecules they comprise and based on the numbers that were given to every nucleic acid molecule in step (a);
- (c) number the ordered groups from step (b) from low to high using decimal numbers and starting with the decimal number 0, wherein every decimal number is equivalent to a specific binary element consisting of m bits.

In one embodiment, said method further comprises a step of converting a binary element consisting of m bits into one of the groups of step (b) based on the decimal number equivalent to said binary element.

In another embodiment, said method comprises a step (d) of converting groups of nucleic acid molecules into binary elements each consisting of m bits, said binary elements being equivalent to the decimal number provided in step (b) for every specific group of nucleic acid molecules.

Said ordering in step (b) is done according to general mathematical rules wherein a group of nucleic acid molecules with the lowest numbers is given the number 1 and wherein a group of nucleic acid molecules with the highest numbers is given the highest number.

In an embodiment of the methods of the first and second aspect, said selected one of a plurality of dictionaries to convert binary elements into groups of k nucleic acid molecules (wherein k is at least 2 and $< n$) is generated or produced according to the method of the third aspect or according to the method comprising the following steps:

- (a) provide a collection of n different nucleic acid molecules and number every nucleic acid molecule in a consecutive way;

- (b) make groups of k nucleic acid molecules from said collection, wherein k is at least 2 but $< n$ and order said groups numerically based on the numbers that were given in step (a) to the nucleic acid molecules from said combinations;
- (c) number the ordered groups from step (b) from low to high with a decimal number starting from 0, wherein every decimal number is then converted to its binary number equivalent.

Element IDs

As explained above, the methods herein disclosed can be used to store large amounts of digital information in a cost-efficient manner. However, in some cases there is a need for storing small amounts of digital information. Indeed, to keep track of products or to authenticate luxury products and thereby detect or prevent counterfeiting, it is advantageous to label or tag these products. One possibility is the use of nucleic acid molecules, particularly of DNA. As demonstrated in Example 4, the methods herein disclosed are perfectly suited to provide DNA tags that can be used in nucleic acid based authentication of products.

Therefore, in a fourth aspect, a method is provided of producing a nucleic acid label for authenticating or tracking a product, said method comprises the steps of:

- a) providing a binary sequence that is unique for said product;
- b) providing a collection of n pre-synthesized nucleic acid molecules;
- c) providing groups of k nucleic acid molecules, wherein $2 \leq k < n$;
- d) fragmenting the binary sequence in maximum $n!/k!(n-k)!$ binary elements
- e) translating each binary element into a group of k nucleic acid molecules using at least one dictionary.

In one embodiment, $n!/k!(n-k)!$ groups of k nucleic acid molecules are provided. In particular embodiments, said k is identical for every of the $n!/k!(n-k)!$ groups.

If the digital information to be stored is small there is no need of using element IDs. But also with a larger amount of digital information to be stored the need to add additional nucleic acid molecules as element IDs can be overcome. Figure 5 illustrates that. First, n different nucleic acid molecules are provided and divided in blocks. Second, to compose the group of k nucleic acid molecules representing the first binary element, the nucleic acid molecules from block 1 are used. To compose the group of k nucleic acid molecules representing the second binary element, the nucleic acid molecules from block 2 are used, etc. By doing that the positional information of the different binary elements is inherently enclosed in the sequence of the nucleic acid molecules. Hence, the sequence of the nucleic acid molecules as such defines the positional information of the binary element in the digital file, while the combination of the nucleic acid molecules in a group defines the binary sequence.

The application thus discloses a method of storing a file of digital information using nucleic acid molecules, said file consisting of a plurality of binary elements, said method comprises:

- (a) converting each binary element from said file into a group of at least 2 individual nucleic acid molecules using a selected one of a plurality of dictionaries, wherein said at least 2 individual nucleic acid molecules from one group differ from each other in nucleic acid sequence;
- (b) providing to each of the at least 2 nucleic acid molecules of a group representing a binary element, the information of the position of said binary element in said file, wherein said positional information can be retrieved from the sequence of the least 2 nucleic acid molecules of a group;
- (c) storing the file of digital information by pooling all the nucleic acid molecules of step (b).

Storage device

In a fifth aspect, the application also provides a device for storing digital information, said device comprises a storage system for storing nucleic acid molecules pooled together according to the methods of the first and/or second aspect of the invention. In more particular embodiments, said device comprises nucleic acid molecules encoding digital information. Hence, a device for storing digital information is provided comprising pools of nucleic acid molecules organized according to any one of the methods of the first and/or second aspect of current application. In a particular embodiment, said device is a multi-well plate (for example a 196, 384, 1536 or 6144 well plate) or a silicon well plate (for example a 10.000 silicon well plate).

Retrieving digital information

In a sixth aspect, the current application provides a method of retrieving a file of digital information from a pool or a plurality of pools of nucleic acid molecules, wherein said nucleic acid molecules can be grouped based on the element IDs they comprise or based on the positional information that can be retrieved from the sequence of the nucleic acid molecules.

The method comprises the amplification of a plurality of stored nucleic acid molecules and the sequencing of said amplified nucleic acid molecules (Figure 4 step 150). The skilled person in the art is aware of molecular techniques that can be used to amplify and sequence nucleic acid molecules as referred herein. Within every amplified nucleic acid molecule, the data-oligonucleotide storing digital information represented by binary elements and the element ID storing information on the position of said binary elements in the digital file are first identified and subsequently all nucleic acid molecules comprising the same element ID or the same positional information are grouped (Figure 4, step 160). For example, all the sequenced nucleic acid molecules having the ID "1" at their 5'-end and ID "1" at

their 3'-end jointly identify the binary element on position 1 in the digital file. All the sequenced nucleic acid molecules having the ID "1" at their 5'-end and ID "2" at their 3'-end jointly identify the binary element on position 2 in the digital file, etc.

Every unique group of nucleic acid molecules is then converted into a binary element of which the element ID determines the position of the binary element with the original digital file (Figure 4, step 170). Said conversion is done by making use of a dictionary that was originally used to convert digital information in groups of nucleic acid molecules and thus by reversing the encoding step illustrated in step 120.

Finally, the file of digital information is constructed (Figure 4, step 180). All the binary elements are linked together (in correct order) and the file can be retrieved back.

A crucial step in retrieving the information is having access to a "dictionary" (e.g. as illustrated in Figure 2) that connects combinations of data-oligonucleotides to binary elements. Said dictionary can be physically stored in nucleic acid molecules as well. Depending on the number of data-oligos in the collection (i.e. n) and the number of data-oligos in one combination (i.e. k), the dictionary can be small or extremely big. For example, for $n=64$ and $k=8$ a table of more than 4.4 billion combinations should be stored. As explained above and in Example 2, Applicants of current application propose a dictionary that can be stored in nucleic acid molecules in a DNA-synthesis efficient way (for more details see storage-efficient dictionary or the third aspect of the invention).

The application thus provides a method of retrieving a file of digital information from a pool of nucleic acid molecules, said method comprises the following steps:

- (a) amplifying (150) from the pool of nucleic acid molecules a plurality of nucleic acid molecules comprising one or more nucleic acid molecules determining a dictionary needed to retrieve said file;
- (b) sequencing (150) the amplified nucleic acid molecules;
- (c) ordering (160) the sequenced nucleic acid molecules comprising the same element ID or representing the same positional information of a binary element in groups;
- (d) converting (170) each group of nucleic acid molecules from step (c) to a binary element, wherein the element ID or the sequence of the nucleic acid molecules determines the position of said binary element in the file of digital information, using the dictionary of step (a);
- (e) constructing (180) the file of digital information by connecting said binary elements to each other in the correct or logic order, more particularly according to their position in the file, even more particularly in an increasing order of their numerical position.

In one embodiment, said binary element consists of 2, 3, 4, 5, 6, 7, 8 or more bits and is defined by the dictionary that was used to convert digital information into groups of nucleic acid molecules.

In one embodiment of the methods of the sixth aspect, said dictionary to convert combinations of nucleic acid molecules to digital data is generated or produced according to a method comprising the following

5 steps:

- (a) provide a collection of n different nucleic acid molecules and number every nucleic acid molecule in a consecutive way;
- (b) make combinations of k nucleic acid molecules from said collection wherein $2 \leq k < n$ and order said combinations numerically based on the numbers that were given in step (a) to the nucleic acid molecules from said combinations;
- (c) number the ordered combinations from step (b) from low to high with a decimal number starting from 0, wherein every decimal number is then converted to its binary number equivalent.

15 Besides the reduction in number of starting nucleic acid molecules, using combinations of nucleic acid molecules to represent one binary element in a digital file has an additional advantage. It allows an easy error detection. Indeed, if the information storing method herein described makes use for example of 3 nucleic acid molecules that jointly represent one binary element, then for every group defined by an element ID, all 3 different nucleic acid molecules should be retrieved. If this would not be the case due to for example oligo-dropout, this will be immediately detected. In approaches using a one binary element to one nucleic acid molecule translation, oligo dropout immediately leads to data loss without detecting it. Therefore, in another embodiment, said methods of retrieving a file of digital information herein described further comprises a step of correcting of errors. In a particular embodiment, error correction is performed by the Reed-Solomon method which is well-known to the person skilled in the art. Details can be found in Reed & Solomon (1960) "Polynomial Codes over Certain Finite Fields", Journal of the Society for Industrial and Applied Mathematics, 8 (2): 300-304, doi:10.1137/0108018.

Computer-implementation

Some of the methods steps disclosed above (i.e. methods for storing and retrieving digital information and their respective embodiments) may be computer-implemented. The step of converting the plurality of binary elements into a plurality of groups of nucleic acid molecules using selected ones of a plurality of dictionaries is preferably computer-implemented. The step of determining with which element ID the data oligonucleotides from a group should be complemented with is preferably computer-implemented.

The methods according to the first and second aspect may therefore be computer-implemented methods.

Some of the steps from the methods for retrieving digital information may be computer-implemented.

The step of identifying the element IDs within the sequenced nucleic acid molecules is preferably

5 computer-implemented. The step of grouping the nucleic acid molecules comprising the same element ID is preferably computer-implemented. Also the step of converting a unique group of nucleic acid molecules to a particular binary element and the step of converting the element ID into the position of said binary element in said digital file is preferably computer-implemented. The step of constructing the digital file from the obtained information from the latter step is preferably computer-implemented as
10 well. The methods according to the sixth aspect may therefore be computer-implemented methods.

In a seventh aspect, the present invention provides a computer system for converting digital information into nucleic acid molecules. The computer system comprises one or more processors. The computer system is configured for performing a method according the first and/or second aspect of the present
15 invention.

In an eight aspect, the present invention provides a computer program product for converting digital information into nucleic acid molecules or for converting a plurality of binary elements into a plurality of groups of nucleic acid molecules using selected ones of a plurality of dictionaries. The computer
20 program product comprises instructions which, when the computer program product is executed by a computer, such as a computer system according to the third aspect of the present invention, cause the computer to carry out a method according to the first and/or second aspect of the present invention.

In a ninth aspect, the present invention may furthermore provide a tangible non-transitory computer-
25 readable data carrier comprising the computer program product of the eight aspect.

It is to be understood that although particular embodiments, specific configurations as well as materials and/or molecules, have been discussed herein for means and methods according to the present invention, various changes or modifications in form and detail may be made without departing from the
30 scope and spirit of this invention. The following examples are provided to better illustrate particular embodiments, and they should not be considered limiting the application. The application is limited only by the claims.

EXAMPLES**Example 1: Storing data in DNA with minimal DNA synthesis effort**

One of the technical effects of the methods disclosed in this application is that digital data can be stored in nucleic acid molecules with a minimal DNA synthesis effort. This is achieved by using combinations of nucleic acid molecules.

A collection of 64 different oligonucleotides was generated, wherein every oligonucleotide consists of 18 nucleotides (Table 1). These oligonucleotides are referred to data-oligonucleotides in the detailed description of current application.

Next, said data-oligonucleotides were combined in groups of 8. By constructing these combinations or groups of 8 different data-oligonucleotides, it is possible to generate more than 4.4 billion different combinations. This huge number of combinations is sufficient to cover all possible combinations of 32 bits, as 32 bits can be written in 2^{32} ways (i.e. 4.394.967.296), from 00000000 00000000 00000000 00000000 to 11111111 11111111 11111111 11111111.

Table 1. Exemplary collection of 64 diverse data-oligonucleotides of each 18 nucleotides.

data oligo number	sequence	data oligo number	sequence
1	AGCGGATCACGTGCGGAC	33	AATGGCTCGCATGGTAGT
2	ATGATCTATTACGCATTC	34	ATAAGGTTCTAAGCCTGC
3	AATAGTTGCTGTGCCGGA	35	AACGCATTCGGTGTGCGG
4	AGCCTGTATGCCGGCGTG	36	ACTGGATAAGGCGCTCGC
5	ACCACTTGACCTGGTAAC	37	ATCGATTTGGCTGTGGAT
6	AAGCACTCGTCAGTGATG	38	ATCATGTAGTCCGTGAAC
7	ATGGCCTAGACGGGTCTT	39	ACAATATCTAATGCCATT
8	ACTGTCTGAGGTGAAGCT	40	ACTCGTTCGAAGGATGGT
9	ACTCCAGTTGTCGCCGAG	41	ACGGCGTGAGCAGTTAGG
10	ACGCTTTTAGTTGCTAGA	42	ATGCATTTGCTGGGTGGC
11	ATACAATGGCGGGAGCTA	43	ACCTGTTGGTCTGTCTAC
12	ATGGAATAATGAGGACAT	44	ATGGTTTCTTGGGGTTAG
13	AGCTTATGCTTGCCAGG	45	AGCGTCTAGGTGGGATGT
14	AGTTACTCTAGCGACTTC	46	AATCGGTCATTAGAACGT
15	ACCGAATGAAGCGTCGAA	47	ATGAAGTCAAGGGGATCA
16	AAGTGCTCCGGCGAACTG	48	ACCGCCTACTGCGCTGCC
17	AACGGTTCTGCAGCTTAC	49	ATCCTTTAGATAGCACGG
18	ATAACATAGCTCGTCACT	50	AGGTCCTTCAATGACGTA
19	AACCAGTGCACTGATGCA	51	AGACCATCTTAAGGACGC
20	AGTCATGTTCTGCTTCGC	52	ACGTTCTCAGCTGACATG
21	ATCGGCTAGCCTGTCGTT	53	ATTAGCTACAGGGATAAT
22	ACGTGATATTGTGAATCG	54	ATGTACTGACGACTTATC
23	ATACTGTGTCCAGAAGTC	55	AGCAACTACGTCGTCCTA

24	AGTAGATGCGAACTTCCG	56	ATCCGGTGTTGCGAGCAT
25	ATCTAGTACCACGTGGCG	57	AGATGATCACTGGGAGTT
26	AACCGATTAGAAGGATAC	58	ACATCCTGACAGGGCAGC
27	AGGATAGTTGCTGCTGTT	59	ATTCAGTACGAGGAATTA
28	ATCGTATGGACAGGCTCC	60	ATGCCGTAGTGGGTGACA
29	AGGTGGTGCATCGGCTAA	61	AAGCCATATGAAGCGACG
30	AGTACTTATCTAGATCAG	62	ACAGGTTCATGCGCAAGT
31	ACTAGGTTGTGCGTCAAG	63	AGACTTCCATGGGAATC
32	ACCGTTTACCGTGGCAAT	64	AATTCTTGAGTGGGTATG

Example 2: Storing the dictionary

In Example 1 it is taught how starting from 64 different data-oligonucleotides, 4.4 billion different combinations can be made of 8 data-oligonucleotides each. Said 4.4 billion combinations are subsequently linked to all different combinations of 32 bits using a dictionary. In order to retrieve digital data that has been stored in a pool of DNA according to Example 1, the information describing which data-oligonucleotide combination is linked to which 32 bit string should also be stored in said pool of DNA. However, storing a dictionary of 4.4 billion combinations would require too many resources.

Therefore, an algorithm was developed that is able to quickly identify the correct binary element of every oligonucleotide combination in a “virtual hash table”, without “physically” storing it on a computer/server or DNA molecule.

The writing and reading algorithm is based on the fact that all the combinations of oligonucleotides are ordered following specific rules making it possible to mathematically predict the binary element corresponding to any given combination of data-oligonucleotides, without the need of an hash table or dictionary.

In Figure 2, a possible dictionary is illustrated consisting of all the 20 unique combinations of 3 data-oligonucleotides which are generated from a collection of 6 data-oligonucleotides (called A, B, C, D, E and F) and the binary elements consisting of 4 bits to which said collections can be converted. Besides the 4 bits sequence, the combination of data-oligonucleotides is also linked to the decimal equivalent of the binary element. In the non-limiting example of Figure 2, combination A, B and F is linked to the binary element 0011 having a decimal equivalent of 3.

The number of all the possible combinations of oligonucleotides can be calculated, following the formula $n!/k!(n-k)!$ wherein n is the number of data oligonucleotides in the collection and k the number of different data-oligonucleotides within a group or combination. With $n=6$ and $k=3$ the results is 20. Importantly, in the combinations of data-oligonucleotides, one particular data-oligo cannot be present

more than once, while the order of the data-oligonucleotides is not important, e.g. combination E A C is the same as A C E.

By using the alphabetic example (i.e. A, B, C, D, E and F oligos), the combinations of 3 oligos are then ordered from the first (starting with A B C) to the last (D E F). As a result, the first 10 combinations all comprise the oligonucleotide called A. Therefore, the combinations of oligos which are used to translate the binary element with a decimal equivalent between 0 and 9 will comprise oligo A. Vice versa, if we want to retrieve digital data from a combination of nucleic acid molecules, the binary element can be 0000 (decimal equivalent 0), 0001 (decimal equivalent 1), 0010 (decimal equivalent 2), 0011 (decimal equivalent 3), 0100 (decimal equivalent 4), 0101 (decimal equivalent 5), 0110 (decimal equivalent 6), 0111 (decimal equivalent 7), 1000 (decimal equivalent 8) or 1001 (decimal equivalent 9) if the combination comprises oligonucleotide A. The calculation then continues for the other nucleic acid molecules from said combination. If said combination also comprises oligo B then the possibilities are reduced from 10 to 4 binary elements. The previous information together with the identity of the last oligo will then reveal the decimal equivalent of the binary element for which the combination of oligos is translating for.

The above example is for illustrative purposes only. Since data-oligonucleotides within a collection do not have a relative order in contrast to the generally accepted alphabetic order of A, B, C, ..., an order should first be given to the n data-oligonucleotides from the collection that will be used. This order (e.g. as illustrated in Table 1) comprising predefined numerical positions of only n oligonucleotides can be easily stored in DNA in contrast to a dictionary of $n!/k!(n-k)!$ members.

If a group of nucleic acid molecules would comprise the data-oligonucleotides AGCGGATCACGTGCGGAC, ATGATCTATTACGCATTC and AGCCTGTATGCCGGCGTG, it can be retrieved from Table 1 that the data-oligos respectively have position 1, 2 and 4. Using the same mathematical calculation as with the alphabetic oligos, we would know that the combination has a decimal number lower than 10 (because of the presence of the oligo with position 1 in Table 1), lower than 5 (because of the presence of data-oligo with position 1 and 2). The presence of the data-oligo with position 4 in Table 1 reveals together with the 2 other data-oligos a decimal number of 2 with is equivalent to binary number 10 or because binary elements of 32 bits are encoded to 00000000 00000000 00000000 00000010.

Vice versa, if the latter 32-bit string is to be encoded in combinations of 3 data-oligonucleotides, the decimal equivalent "2" requires the presence of an oligo with position 1 from Table 1, with position 2 from Table 1 and with position 4 from Table 1.

Since the above algorithm is based on relatively simple mathematical calculations, any standard/commercially available computer can translate a file of for example 1.025.120 bytes of

information, into data-oligonucleotides combinations in about 1 minute. The reading process works in the same way but in the opposite direction, so the timing is comparable.

Example 3: Storing a txt file in pools of nucleic acid molecules

- 5 To practically demonstrate the methods disclosed herein the inventors have translated the following quote of the famous scientist Galileo Galilei: *"In questions of science, the authority of a thousand is not worth the humble reasoning of a single individual."* Galileo Galilei, in a pool of nucleic acid molecules. The text consists of 128 bytes in total. Using groups of 8 data-oligonucleotides each, wherein every group represent 4 bytes of digital information, said 128 bytes can be translated in 32 of said groups. Using
- 10 Table 1, the text file can be represented by the 32 groups as listed in Table 2.

Table 2. An overview of 32 groups of 8 data-oligonucleotides jointly representing a text file of 128 bytes.

Element ID	Oligo 1	Oligo 2	Oligo 3	Oligo 4	Oligo 5	Oligo 6	Oligo 7	Oligo 8
1	1	28	33	37	43	44	47	53
2	5	7	19	35	38	51	55	60
3	4	26	28	33	40	47	61	62
4	4	12	15	20	21	27	48	61
5	4	15	16	19	35	42	46	57
6	4	11	16	36	38	42	58	60
7	4	14	15	19	39	45	54	57
8	5	8	20	28	31	57	59	62
9	5	7	11	18	26	40	44	55
10	1	28	42	44	48	49	56	63
11	4	8	21	44	48	59	60	63
12	5	6	7	12	38	44	48	54
13	4	25	38	40	41	42	47	53
14	5	7	20	21	38	40	50	59
15	5	8	11	29	33	37	39	42
16	5	7	11	18	27	30	41	64
17	5	8	10	12	27	36	45	61
18	4	13	23	30	31	52	53	57
19	4	19	25	26	41	42	53	54
20	4	11	17	24	34	52	57	62
21	4	26	28	33	42	43	48	59

Element ID	Oligo 1	Oligo 2	Oligo 3	Oligo 4	Oligo 5	Oligo 6	Oligo 7	Oligo 8
22	1	28	31	34	36	37	48	53
23	4	8	21	44	48	59	60	63
24	4	26	27	33	35	44	46	49
25	1	28	30	39	43	44	56	60
26	4	15	16	20	24	31	57	58
27	5	8	20	32	50	52	54	60
28	6	23	33	38	39	45	52	63
29	4	19	24	45	53	57	62	64
30	5	6	7	12	15	37	62	63
31	4	19	24	45	53	57	62	64
32	4	15	16	20	33	50	58	59

As can be seen in Table 2, for the 32 different groups, 32 different element IDs have been generated. Every data-oligo of the same group is supplemented with the element ID correlated to the group. As described in the detailed description of current application, the element ID consists of a nucleic acid sequence. The numerical positions of the data-oligos within one group can be determined by Table 1.

5 Said numerical positions define which decimal number is represented by a group of data-oligonucleotides. Since every 32-bit string has a decimal equivalent, the corresponding binary elements can be retrieved.

Example 4. Storing an image and authentication labels

10 Next, we decided to store an image of 1080 bytes and 30 different 16-bytes long sentences. The latter represent what we define as authentication labels or nucleic acid tags, i.e. a nucleic acid ID code that can be stored into or on products that need to be authenticated during or after a production process.

In this exercise a stock of 384 nucleic acid molecules was designed. These 384 oligos were ordered according to Figure 2 and the third aspect described in the detailed description, and divided in 6 different

15 blocks of 64 different oligos each. From the second block onwards, the first oligo combination was artificially given the decimal number 0. Groups of 8 oligos were assembled representing 4 bytes. To encode a 16 bytes long label we thus needed 4 groups of 8 oligos. Based on the binary sequence, the first 4 bytes are represented by 8 oligos from block 1, the second 4 bytes are represented by 8 oligos from block 2, etc. Because the oligos are ordered in blocks of 64 oligos, the binary information can be

20 retrieved according to the principle described in aspect eight of the application. Another 2 sets of 8 oligos was added for error detection and correction based on the well-established Reed-Solomon method. Hence, 24 bytes of digital information is represented by 48 oligos or 6 groups of 8 oligos.

The encoding of the 1080 bytes image was performed in an identical manner. We used in total 1584 bytes of data (i.e. 1080 bytes of the image + 504 bytes for Reed-Solomon error correction), which resembles 66 times 48 oligos.

To challenge the efficiency and cost effectiveness of our method, we choose to read the image file and the 30 tags together in one single run of an Illumina Mi-Seq sequencing machine. Therefore, we used a 96-well plate, with 66 wells (each comprising 6 groups of 8 oligo) used for the image file and 30 wells (each comprising 6 groups of 8 oligos) for each individual DNA-tag.

The 384 oligos were 23 nucleotides long and each oligo different in at least 10 nucleotides from the other ones. The oligos were extended by a Forward Illumina overhang adapter sequence (TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG) and a Reverse Illumina overhang adapter sequence (GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAG).

The Illumina overhang adapters were needed in order to easily ligate the Illumina index, that we have used in order to define the well where each oligo has been dispensed.

The oligos were ordered and dispensed in the 96 wells-plate using Echo 650 liquid handler. Then, the 96-well plate has been indexed using Nextera indexes.

All oligos of all wells were pooled and sequenced on an Illumina MiSeq instrument, using the Reagent Kit v2 Nano 50 cycles (7.5 pM + 19.95% PhiX v3), paired end (26-8-8-26). The run was successful and yielded 638k PF clusters, 97.90% bases with $\geq Q30$ for Read1 and 96.20% bases with $\geq Q30$ for Read2. We obtained more than 100k reads, however the pooling was not optimal due to two less performant Nextera indexes (S510=CGTCTAAT, S511=TCTCTCCG found in wells in column G and H).

As a result, in 8 out of the 66 wells used to store the image file, we experienced oligo dropout. Surprisingly, despite that, the file could be perfectly recovered, due to our error-correction power.

The 30 DNA-tags were read by PCR-based methodology rather than pooling together multiple ones in a sequencing run.

CLAIMS

1. A method of storing a file of digital information using nucleic acid molecules, said file consisting of a plurality of binary elements, said method comprises:
 - (a) converting each binary element from said file into a group of at least 2 individual nucleic acid molecules using a selected one of a plurality of dictionaries, wherein said at least 2 individual nucleic acid molecules from one group differ from each other in nucleic acid sequence;
 - (b) providing to each of the at least 2 nucleic acid molecules of a group representing a binary element, the information of the position of said binary element in said file;
 - (c) storing the file of digital information by pooling all the nucleic acid molecules of step (b).
2. The method of claim 1 further comprising a step wherein the information determining the selected one of a plurality of dictionaries from step (a) is converted to one or more nucleic acid molecules that are added to the pool of nucleic acid molecules of step (c).
3. The method of any of the above claims, wherein the positional information of step (b) is provided by adding one or more additional nucleic acid sequences to the at least 2 nucleic acid molecules of a group or wherein the positional information of step (b) can be retrieved from the sequence of the at least 2 nucleic acid molecules of a group.
4. The method of claim 3, wherein the one or more additional nucleic acid sequences are added at the 5'-end and/or at the 3'-end of said at least 2 nucleic acid molecules of a group.
5. The method of claim 4, wherein said one or more additional nucleic acid sequences are between 15 and 25 nucleotides long.
6. The method of any of the above claims, wherein the binary element consists of between 4 and 32 bits.
7. The method of any of the above claims, wherein said binary element consists of 32 bits and wherein said selected one of a plurality of dictionaries converts binary elements of 32 bits into groups of 8 individual nucleic acid molecules.
8. A computer system for converting a file of digital information into nucleic acid molecules, the computing system comprising one or more processors, the computing system configured for performing the method according to one of the preceding claims.

9. A computer program for converting a file of digital information into nucleic acid molecules, the computer program comprising instructions which, when the computer program product is executed by a computer, cause the computer to carry out the method according to any one of claims 1-7.

5 10. A device for storing digital information comprising pools of nucleic acid molecules organized according to any one of claims 1 to 7.

11. A method of retrieving a file of digital information from a pool of nucleic acid molecules, said method comprises the following steps:

- 10 (a) amplifying (150) a plurality of nucleic acid molecules, optionally comprising one or more nucleic acid molecules determining a dictionary needed to retrieve said file, from said pool;
- (b) sequencing (150) the amplified nucleic acid molecules;
- (c) ordering (160) the sequenced nucleic acid molecules comprising the same element ID or representing the same positional information based on the sequence of the nucleic acid
- 15 molecules in groups;
- (d) converting (170) each group of nucleic acid molecules from step (c) to a binary element, wherein the element ID or the sequence of the nucleic acid molecules determines the position of said binary element in the file of digital information, using the dictionary of step (a);
- 20 (e) constructing (180) the file of digital information by connecting said binary elements to each other according to their position in the file.

12. The method of claim 11, further comprising a step of correcting of errors.

25 13. A computer system for retrieving digital information from a pool of nucleic acid molecules, the computing system comprising one or more processors, the computing system configured for performing the method according to any of claims 11-12.

14. A computer program for retrieving digital information from a pool of nucleic acid molecules, the

30 computer program comprising instructions which, when the computer program product is executed by a computer, cause the computer to carry out the method according to any of claims 11-12.

Figure 1

100 – Create a collection of data-oligonucleotides

A  D 

B  E 

C  F 

110 – Create unique combinations of data-oligonucleotides

0) A  B  C 

10) B  C  D 

1) A  B  D 

11) B  C  E 

2) A  B  E 

12) B  C  F 

3) A  B  F 

13) B  D  E 

4) A  C  D 

14) B  D  F 

5) A  C  E 

15) B  E  F 

6) A  C  F 

16) C  D  E 

7) A  D  E 

17) C  D  F 

8) A  D  F 

18) C  E  F 

9) A  E  F 

19) D  E  F 

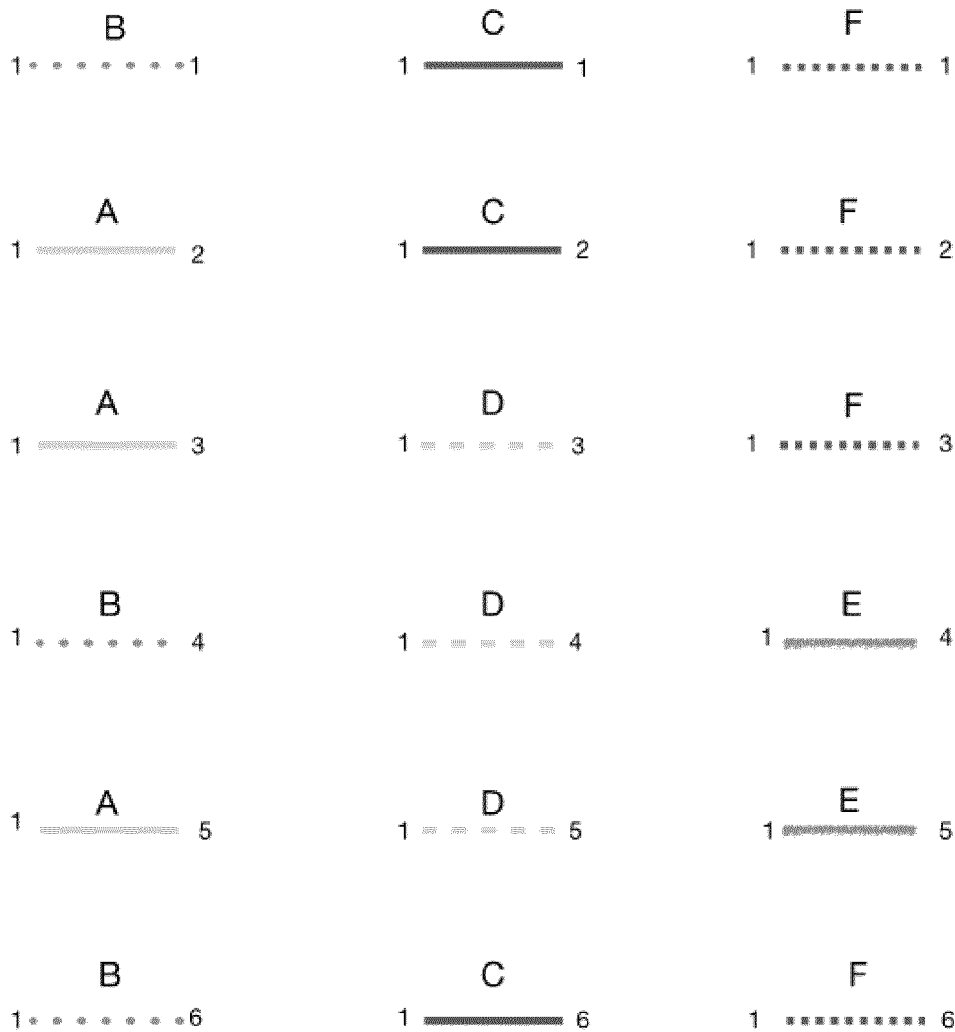
Figure 2

data-oligo combination	Decimal number	Binary element
A B C	0	0 0 0 0
A B D	1	0 0 0 1
A B E	2	0 0 1 0
A B F	3	0 0 1 1
A C D	4	0 1 0 0
A C E	5	0 1 0 1
A C F	6	0 1 1 0
A D E	7	0 1 1 1
A D F	8	1 0 0 0
A E F	9	1 0 0 1
B C D	10	1 0 1 0
B C E	11	1 0 1 1
B C F	12	1 1 0 0
B D E	13	1 1 0 1
B D F	14	1 1 1 0
B E F	15	1 1 1 1

Figure 3

120 – Convert each binary element into combinations of data-oligonucleotides

130 – Add element IDs to each data-oligonucleotide for each combination



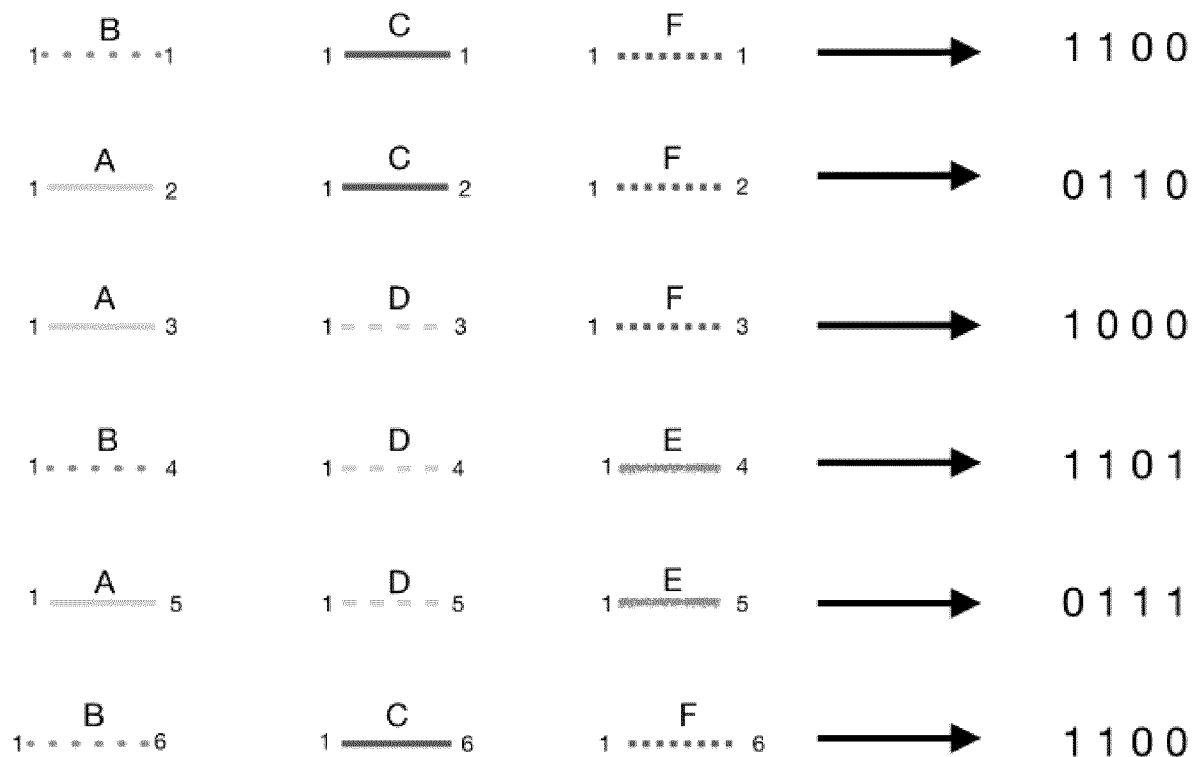
140 – Pool all combinations of nucleic acid molecules

Figure 4

150 – Amplify and sequence nucleic acid molecules

160 – Order the nucleic acid molecules in groups

170 – Convert groups of nucleic acid molecules to binary elements



180 – Link the binary elements and retrieve the data

Figure 5

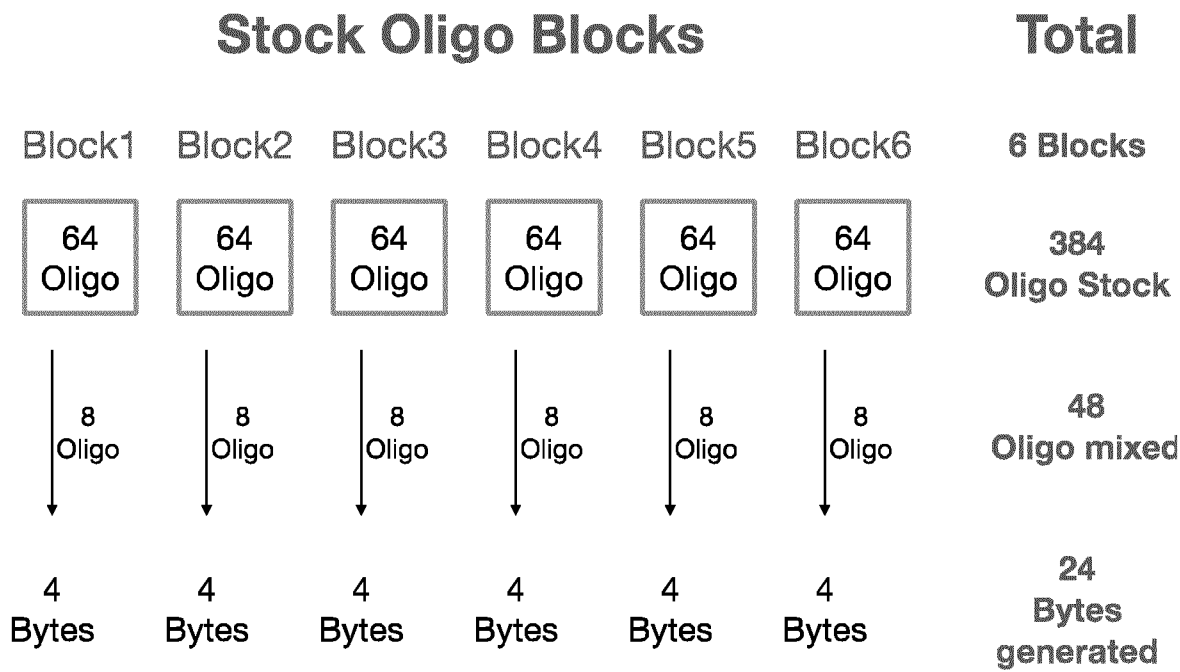


Figure 6



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/064648

A. CLASSIFICATION OF SUBJECT MATTER

INV. G11C13/00
ADD. G06N3/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G11C G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018/137418 A1 (ROQUET NATHANIEL [US] ET AL) 17 May 2018 (2018-05-17) paragraph [0137] - paragraph [0150]; figure 1	1-14
X	----- WO 2018/039938 A1 (UNIV TSINGHUA [CN]) 8 March 2018 (2018-03-08) paragraphs [0003] - [0004], [0175] - [0183]; figures 1-2	1-10
A	----- WO 2018/148260 A1 (THOMSON LICENSING [FR]) 16 August 2018 (2018-08-16) figure 15	1-14
A	----- US 2018/265921 A1 (CHEN YUAN-JYUE [US] ET AL) 20 September 2018 (2018-09-20) cited in the application figure 1	1-14
	----- -/-	



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

21 August 2020

Date of mailing of the international search report

01/09/2020

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Kontogiannis, K

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2020/064648

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2019/130280 A1 (ERDEN MEHMET FATIH [US]) 2 May 2019 (2019-05-02) figure 4 -----	1-14
A	EP 2 947 589 A1 (THOMSON LICENSING [FR]) 25 November 2015 (2015-11-25) figure 1 -----	1-14
A	CEZE LUIS ET AL: "Molecular digital data storage using DNA", NATURE REVIEWS GENETICS, NATURE PUBLISHING GROUP, GB, vol. 20, no. 8, 8 May 2019 (2019-05-08), pages 456-466, XP036837200, ISSN: 1471-0056, DOI: 10.1038/S41576-019-0125-3 [retrieved on 2019-05-08] page 458 - page 462 -----	1-14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2020/064648

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018137418 A1	17-05-2018	US 2018137418 A1	17-05-2018
		US 2020250546 A1	06-08-2020
WO 2018039938 A1	08-03-2018	EP 3509018 A1	10-07-2019
		US 2019311782 A1	10-10-2019
		WO 2018039938 A1	08-03-2018
WO 2018148260 A1	16-08-2018	WO 2018148257 A1	16-08-2018
		WO 2018148260 A1	16-08-2018
US 2018265921 A1	20-09-2018	NONE	
US 2019130280 A1	02-05-2019	NONE	
EP 2947589 A1	25-11-2015	EP 2947589 A1	25-11-2015
		EP 3146458 A1	29-03-2017
		WO 2015176987 A1	26-11-2015