DNA Molecular Storage System

Citation for published version (APA):
Tavella, F., Giaretta, A., Dooley-Cullinane, T. M., Conti, M., Coffey, L., & Balasubramaniam, S. (2018). DNA Molecular Storage System: Transferring Digitally Encoded Information through Bacterial Nanonetworks. *ArXiv*.

Published in:
ArXiv

DNA Molecular Storage System: Transferring Digitally Encoded Information through Bacterial Nanonetworks

FEDERICO TAVELLA, University of Padua, Italy
ALBERTO GIARETTA, Örebro University, Sweden
TRIONA MARIE DOOLEY-CULLINANE, Waterford Institute, Ireland
MAURO CONTI, University of Padua, Italy
LEE COFFEY, Waterford Institute, Ireland
SASITHARAN BALASUBRAMANIAM, Tampere University of Technology, Finland and Waterford Institute of Technology, Ireland

Since the birth of computer and networks, fuelled by pervasive computing and ubiquitous connectivity, the amount of data stored and transmitted has exponentially grown through the years. Due to this demand, new solutions for storing data are needed, and one promising media is the DNA. This storage solution provides numerous advantages, which includes the ability to store dense information while achieving long-term stability. However, the question as how the data can be retrieved from a DNA-based archive, still remains.

In this paper, we aim to address this question by proposing a new storage solution that relies upon molecular communication, and in particular bacterial nanonetworks. Our solution allows digitally encoded information to be stored into non-motile bacteria, which compose an archival architecture of clusters, and to be later retrieved by engineered motile bacteria, whenever reading operations are needed.

We conducted extensive simulations, in order to determine the reliability of data retrieval from non-motile storage clusters, placed at different locations. Aiming to assess the feasibility of our solution, we have also conducted wet lab experiments that show how bacteria nanonetworks can effectively retrieve a simple message, such as "Hello World", by conjugation with non-motile bacteria, and finally mobilize towards a final point.

CCS Concepts: • **Computing methodologies → Modeling and simulation**; • **Computer systems organization → Other architectures**; • **Hardware → Communication hardware, interfaces and storage**; **Emerging technologies**;

Additional Key Words and Phrases: DNA Encoding, Data Storage, Bacterial Nanonetworks, Molecular Communications.

1 INTRODUCTION

Worldwide, the quantity of new data is rapidly increasing on a daily basis, due to the massive number of connected devices to the Internet. Indeed, many of the current software features that we are provided nowadays, such as navigation systems and social networks, heavily rely upon machine learning techniques which run over big data. These techniques create even more data, which leads to the increased demanding of storage that we experience since the last decade. To face this shortage, new infrastructures like data centres [15] and cloud computing infrastructures [4] are ceaselessly built; as an example, as part of *Internet.org* initiative, in 2013 Facebook built a new high capacity data center [12].

Besides the large-scale infrastructure, new solutions at the hardware level (HDDs, SSDs, servers and clusters) are also required. In recent years, besides silicon technology to store digital information, researchers investigated alternative storage media, and one example is the *Deoxyribonucleic Acid* (*DNA*). DNA, which is naturally found inside biological cells, encodes information that represents functionalities and characteristics of the cell itself: from a computer science perspective, where the cell represents the hardware, the DNA represents the software.

Since the '80s, techniques for encoding DNA has continually improved in terms of performance, overcoming even Moore's Law[6]. A good example of Digital DNA encoding was recently proposed by Goldman et al. [8]. However, a major question still remains as to how we can possibly automate the Reading process of this encoded information, especially if they are archived collectively. In this paper, we aim to address this Reading process through a bio-inspired communication mechanism, known as *molecular communications*.

In this paper, bacterial nanonetworks with engineered *E. Coli* pick up the plasmids with digitally encoded DNA information from an archive library, in order to deliver the information to a point where sequencing operations can run to decode it. Through simulations, we assess how much the movement randomness affects the engineered bacteria while mobilizing towards a destination point. Our aim is to determine the whole system reliability applied to tasks of information retrieval, by conjugation with clusters of motility-restricted bacteria.

The engineered motility is based on the positioning technique proposed by Moore et al. [13] [14], and we adopt a triangulation process that is inspired by the one commonly used in cellular mobile networks. The triangulation of mobile devices is obtained by measuring radio signal strength, while in this paper the trilateration is based on sensing chemical signals emitted by beacons. Our proposal relies on engineered bacteria ability to sense chemicals with different variations, as proposed in [14], in order to mobilize towards various spatial points of the archive. Each point reflects the location of a cluster of motility-restricted bacteria, where the desired information can be picked up by motile bacteria through conjugation.

1.1 Contribution

The contributions of this paper are manifold:

- **Molecular Positioning System (MPS)**: we propose a positioning system that enables the engineered bacteria to sense chemoemissions and, by programming different types of receptors, to mobilize towards a specific location described by a predefined concentration of chemicals;

- **Digitally enocded-DNA archive system**: we propose a way to enable bacteria to recover digital encoded DNA information from an archive system, and deliver such information to a target destination for later sequencing;
- **System validation**: we conduct a set of simulations to assess the precision of MPS, as well as the efficiency of DNA pick-up tasks, in order to evaluate the whole system performance;
- **Wet lab experiments**: finally, we also run a thorough set of wet lab experiments to demonstrate the feasibility of using bacteria to pick up information, digitally encoded into plasmids, from motile-restricted bacteria, and deliver this information to a destination point.

1.2 Organization

This paper is organized as follows: Section 2 briefly discusses the current state of art about DNA storage systems. Section 3 shows the architecture of our model and describes how the MPS is modelled and implemented, while Section 4 how data are encoded into DNA. Section 5 discusses our simulations and evaluates the related results. In Section 6 we describe how we conducted wet lab experiments to demonstrate the feasibility of our proposal. Finally, Section 7 presents our conclusions.

2 RELATED WORK

The field of molecular communications aims to develop artificial communication system from biological components that are found in nature [10] [1] [16]. Developing biological artificial communication systems at miniature scale can open numerous opportunities such as advanced healthcare solutions, as well as environmental monitoring and protection. In order to realize a fully functional network, a certain degree of control and engineering at nanoscale is required and this could be achieved through **Synthetic Biology**. The community has also proposed an extension of the Internet of Things that we know today as the **Internet of Bio-NanoThings (BNTs)** [2], where miniaturized engineered biological cells can transmit information to the cyberworld.

One particular form of molecular communication that has received considerable interest is known as *bacterial nanonetworks*, where engineered bacteria such as *E. Coli* uses its motility property as information carriers. Researchers proposed bacteria-based models for molecular communications due to bacteria inherent properties, such as the ability to communicate and signal between each other, as well as mobilizing within an aqueous environment. For example, Balasubramaniam et al. [3] proposed a multi-hop communication model, using bacteria as carriers for information encoded into plasmids to communicate between multiple nanomachines. Moore et al. proposed positioning systems for BNTs [13] [14], where the relative position could be inferred by concentration gradients. Other examples of proposed applications include engineering BNTs for cooperative target localization [18] and related techniques for countering security threats [7].

Given the popularity of digital encoding of DNA, a number of different works have been proposed both for simulations and wet lab experiments to demonstrate the feasibility of the idea. The University of Washington [6] developed an architecture for DNA-Encoding and Archiving system, which is structured as a key-value store that uses random access through the *Polymerase Chain Reaction* (*PCR*) technique. The storage system is composed of a DNA synthesizer responsible for encoding data, storage container divided in compartments that store DNA pools, and a DNA sequencer, which reads DNA sequences and converts them back to binary data. In order to implement key-based retrieving mechanism, researchers use selective DNA amplification with PCR. Taking advantage of DNA sequencing primers, they are able to amplify only strands corresponding to desired data, discarding the undesired parts. In a similar way, using primers enables this technique to give each strand a key before putting it into a pool. The encoding technique uses both a simple 2 bits-1 base match, which is possible from the four different nucleotides (A, C, G, T).

However, due to the large amount of errors that can result from the sequencing and synthesis, another proposed solution was the 3 based encoding using Huffman encoding combined with a rotational digit to base conversion [6]. Other researchers [21] focused on a DNA storage method that allows random-access to information encoding and rewriting. Each data block, of length 1000 *bpm*, contains an addressing block at the beginning and end, while the remaining 960 *bpm* is used for text encoding. The rewriting technique is based on replacing certain known blocks of information from the dictionary, and replacing with new words that are to be updated into the storage. In [9], the encoding work focuses on the longevity of the storage, in particular to handle the errors and this is achieved through physical storage that can provide maximum stability. The storage medium is based on silica, where predictions have been made that this could last for over 2 millions years.

Blawat et al. [5] developed a Forward Error Correction technique that could be used to counter various errors that occur from DNA encoding. These errors may range from the maximum run-length, which is the maximum length of identical nucleotides that is only limited to three same nucleotides, or errors that can emerge from deletion or insertion of nucleotides. Another research [20] presented an algorithm where information encoding and decoding of digital DNA can be performed into Nucleic Acid Memory (NAM). This process is achieved through protein translation that maps the three nucleotides in a row, also known as codon, to hexadecimal characters. The mapping process ensures that minimum errors result from the arrangement of the nucleotides, ensuring stability of the encoded information. A study [19] has demonstrated that storing DNA vectors into bacteria can ensure high reliability. Researchers showed a recording system that uses the nucleotide content, temporal ordering, in order to encode arbitrary information within the genomes of a population of cells, and has shown its reliability.

None of the related works examine in depth how encoded data can be transferred between different points and, at the same time, how to ensure reliability by storing such data in a biological cell. We try to address each of these issues through our proposal.

3 DNA ARCHIVAL - SYSTEM ARCHITECTURE

Apart from encoding data within DNA, the question as how to retrieve such information has to be addressed by any kind of proposed storage system. We envision a system where bacteria are the main actors, both for storage and reading purposes. Indeed, not only data is stored within motile-restricted bacteria, but also motile bacteria are responsible for moving towards pre-defined points, read the desired information through bacterial conjugation and, finally, deliver such information to an endpoint.

Figure 1 describes the overall system architecture. The digital information are first encoded into nucleotides, and for this we analyse two different encoding techniques, described in Section 4. The synthesized genes of the encoded nucleotides are then inserted into plasmids, and plasmids are taken up by bacteria through the process of transformation. To guarantee that such bacteria do not mobilize, we position them on solid agar (refer to Section 6 for further details about this process). We place these motility-restricted bacteria with at specific regions of the grid, as illustrated in Figure 1. To achieve our storage system, we position bacteria that contain the same chunk of information at the same spot in the grid. Therefore, each region of the grid identifies a cluster of bacteria that contain the same information, and every cluster stores different data from the others.

In the event that a read operation needs to be performed, motile bacteria are released from *A*, swim towards the compartment, and then conjugate with the motile-restricted bacteria to retrieve the plasmids with the encoded information. Once this is complete, bacteria swim towards position *C* to deliver the plasmids. Conjugation is a process where bacteria come together and form a physical connection that allows them to transfer plasmids between each other, and this process has

a probability associated with it. At position C, the plasmids are retrieved and sequenced to obtain the data and decoded back into digital format.
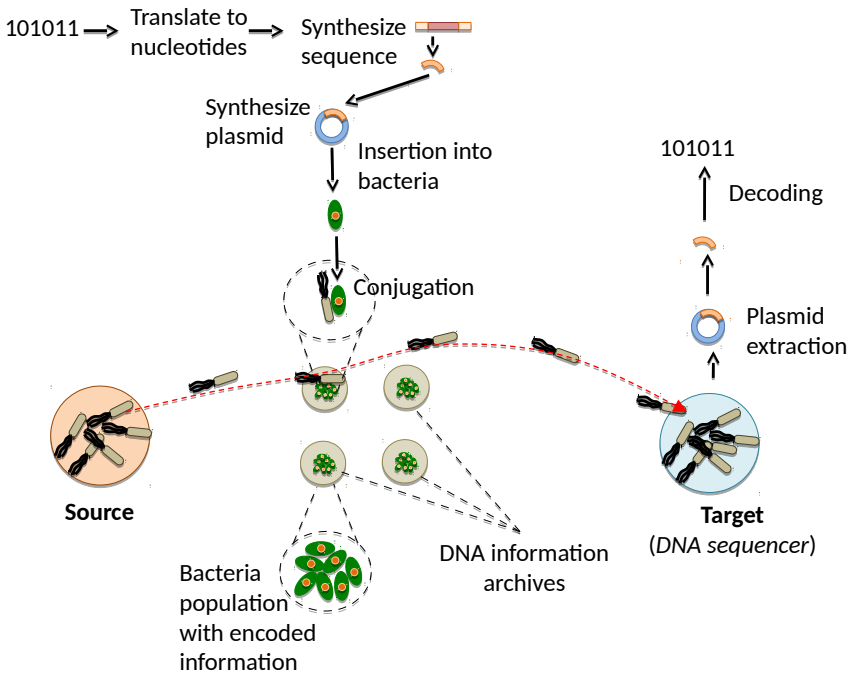

Fig. 1. Overall system architecture of the DNA archival system that enables reading using bacterial nanonetworks. Motile bacteria (in grey) swim from the Source towards each of the archive points, which contain different information. Reached the motile-restricted bacteria, conjugation starts to retrieve the encoded information contained into the plasmid. Last step, motile bacteria swim towards the Target, where the plasmid is retrieved through sequencing processes.

A key requirement for the motile bacteria is the capability to swim towards an accurate point to conjugate with the motile-restricted bacteria, in order to retrieve the right plasmid with the encoded information,. This is where our proposed Molecular Positioning System (MPS) plays a role. As suggested by Okaie et al. [18], it is possible to take advantage of engineered-bacterial chemotaxis in order for them to move towards a certain region, within the field of the chemoattractants, and this pairs well with our MPS proposal which is based on the receptor saturation addressing technique proposed by Moore and Nakano [13] [14].

While trilaterating an object is a difficult task, it is even more challenging when applied to chemical signalling environments. As shown in Figure 2, it is always possible to draw three circles (each one centred on a beacon) that intersect at a specific point inside the convex hull formed by the beacons (i.e., the B_1, B_2, B_3 triangle), without creating an overlapping area. On the contrary, it is not possible to intersect at a point outside the convex hull avoiding an overlapping area, and the farther the intersection point sits from the convex hull, the bigger the resulting overlap is.

This geometric fact is particularly important for our proposed approach, since it entails that the achievable precision is inversely proportional to the size of the overlapping area. Indeed, in our proposed solution the motile bacteria react chemotactically towards their destination, until the receptors for each of the three chemoemissions are saturated. Once that the saturation has been
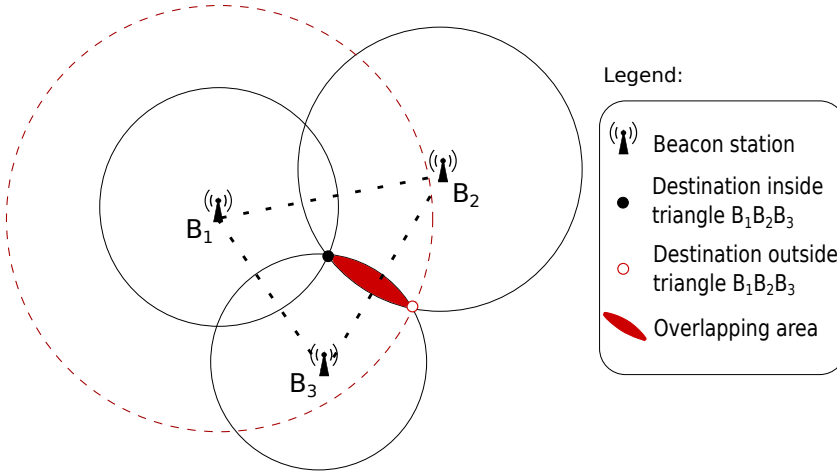
Fig. 2. An example of a general trilateration process. Drawing three circles that intersect at a point outside the $B_1 B_2 B_3$ convex hull, inevitably creates a red overlapping area, as shown here with the red circle centred in B_1 and the two black circles centred in B_2 and B_3.

achieved for all receptors, the motile bacteria perform random step-walks until they eventually drift too far away from the desired location.

3.1 Chemoreception and Mobility Model

In our work, we use the mobility model proposed for engineered bacteria by Okaie et al. [18]. At each time step, which occurs every Δt seconds, a bacterium senses the concentration of chemoattractants emitted from the beacons. We assume the concentration of chemoattractants is approximated as exponentially decreasing with respect to the square distance from the beacons.

The concentration of attractants sensed at current position (x, y), namely $C_A(x, y)$, is taken into account in the next time step only if it is lower than the value expected at the destination, namely $C_A(x_d, y_d)$. This criterion enables a bacterium to swim away from a beacon if it gets too close, with respect to its programmed destination, and this behaviour is represented as follows:

$$C_A(x, y) = \begin{cases} e^{\left(-d_B(x,y)^2\right)}, & if \ < C_A(x_d, y_d) \\ 0, & otherwise. \end{cases} \tag{1}$$

The engineered motile bacteria senses the chemoattractants concentrations within range $D_A = [-\psi_A, \psi_A]$ and select the angle Ψ_A which results from the highest summation of C_A:

$$\Psi_A = \max_{\psi \in D_A} \sum_{i=1}^{m} C_{A_i}(x_\psi, y_\psi), \tag{2}$$

where m is the number of *beacons*.

A random component Φ is necessary to emulate the typical run-and-tumble behaviour of the *E. Coli* bacteria, which swim in a straight line at speed v, and correct their direction at specific intervals. Therefore, the random addend is:

$$\Phi = \pm\sqrt{D2\Delta t}, \tag{3}$$

where Φ is randomly chosen and D is the rotational diffusion coefficient.

Finally, chemoattractants induced angle and random angle are summed up to compute a total drifting angle θ:

$$\theta_{t+\Delta t} = \theta_t + \Psi_A + \Phi. \tag{4}$$

Substituting Φ from Equation 3 we obtain:

$$\theta_{t+\Delta t} = \theta_t + \Psi_A \pm \sqrt{D2\Delta t}. \tag{5}$$

3.2 MPS - Molecular Positioning System

Our MPS aims to enable the engineered motile bacteria to approach (and remain within close vicinity of) a chosen position, leveraging the concentration of chemoattractants emitted by the beacons. We assume that the beacons are anchored at a known fixed position, and emit three discernible chemoattractants at a constant rate. The engineered motile bacteria are then programmed to move, targeting a specific molecular concentration for each of the three beacons, which means that matching the three different concentrations results in reaching the specific location desired. When a motile bacterium is released into the environment, the chemoattractant concentrations emitted by the beacons are used in two different ways: (i) to derive its relative position, and (ii) as a tool to move towards its target destination. Therefore, if a motile bacterium drifts too far away from a beacon, it is able to vary its own chemotactic behaviour and correct its position by following the specific chemoattractants.

In our proposal, we assume that, (i) the engineered motile bacteria know *a priori* the three beacons locations, (ii) each beacon emits a unique chemoattractant, to allow the motile bacteria to infer how far each beacon is, and (iii) the beacons do not move around the environment.

4 DNA ENCODING

In this section, we describe how digital data is encoded into DNA, as well as the process to package the data and the addressing. There are two basic operations to store and retrieve the encoded information, which could be a file or portion of a file:

- `Store (namespace(B1,B2,B3), filename`$_i$`)`: Stores the file into the location that is represented with concentration of (B1,B2,B3);
- `Retrieve (namespace(B1,B2,B3), filename`$_i$`)`: This operation requires that bacteria are encoded with certain concentration of receptors on their surface, namely (C1,C2,C3). Engineered bacteria tend to drift towards the chemoemissions, until the receptors are saturated for each of the chemoemitter, therefore by this feature we can mobilize them towards specific points.

The entire process of the retrieval is illustrated in Figure 3, where the algorithm is presented in Algorithm 1. The end process of the encoding results in the payload DNA that contains the information, and this is encapsulated through a virtual addressing. The virtual addressing represents the location of the cluster of non-motile bacteria, which contain the information stored through the plasmid. The virtual addressing allows the bacteria to accurately swim directly to the cluster to conjugate with the non-motile bacteria to retrieve the plasmid with the encoded DNA.

We analysed two different techniques to convert binary data into nucleotides of the DNA. As described in [6], there are plenty of encoding techniques used to transform digital data into strings of nucleotides. First, we describe the most simple conversion (2 bits-1 nucleotide match), that we call here *basic encoding*, and the approach proposed by Goldman et al. [8], which improves over the basic encoding. Second, we describe how our DNA archive system can be utilized for content managing purposes, where clusters of motility-restricted bacteria are arranged based on the information priority.
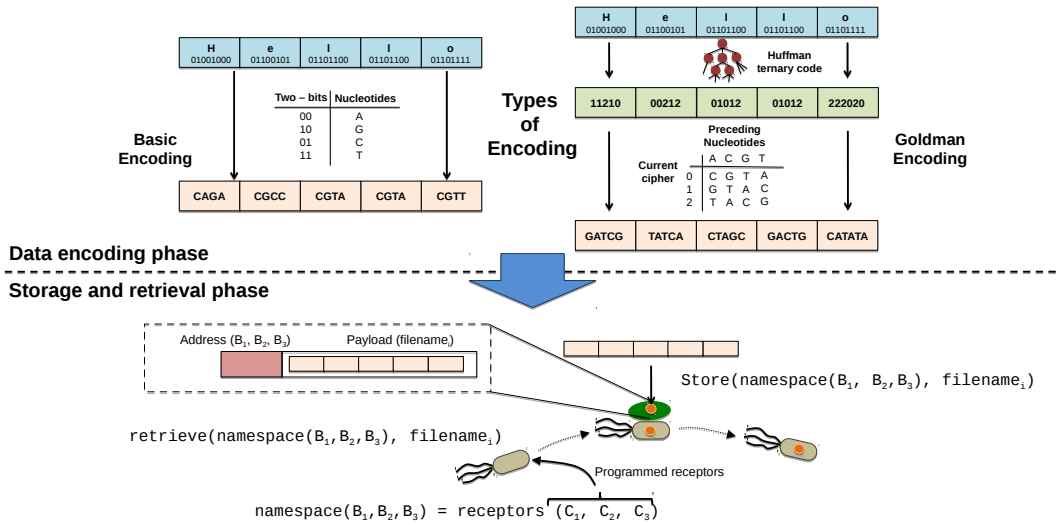
Fig. 3. The process demonstrating the types of encoding (Basic and Goldman encoding), and the production of digitally encoded DNA (payload), which is encapsulated through a virtual addressing that represents the location of the cluster. The virtual addressing encapsulation represents the concentration of receptors (C1, C2, C3, which corresponds to the beacon chemoattractants B1, B2, and B3) on the surface of the motile bacteria, which allows the bacteria to mobilize towards the cluster.

ALGORITHM 1: Retrieving plasmids from the clusters

Input: Positions of the clusters.
Output: Plasmids contained in the clusters.
$t \leftarrow 0, \Delta t \leftarrow 50$;
$conjugationTime \leftarrow 1500, timeLimit \leftarrow 7200$;
while $t < timeLimit$ **or** $data\ missing$ **do**
    forall $receivers$ **do**
        $Move\ by\ one\ step$;
        if $distance(retriever, cluster) < threshold$ **and** $shouldConjugate$ **then**
            $success \leftarrow tryConjugation()$;
            if $success = True$ **then**
                $Do\ not\ move\ for\ conjugationTime\ seconds$;
                $shouldConjugate \leftarrow False$;
            end
        end
    end
    $t \leftarrow t + \Delta t$;
end

4.1 Basic Encoding

As already mentioned, the Basic Encoding is based on a simple mapping. Since DNA is composed of 4 nucleotides (**A**denine, **C**ytosine, **G**uanine, **T**hymine; usually referred using the first letter). Using this technique we can encode $log_2(4) = log_2(2^2) = 2$ bits using a single nucleotide. In this way, we

are able to use the 4 bases that compose the DNA strand to encode each byte of data. An example of the whole process is illustrated in Figure 3.

Even though this is a simple technique, it is not efficient and it is weak against DNA replication errors and mutations. As described in the next section, other proposed techniques [8] [6] proved to be more efficient.

4.2 Goldman encoding

When we choose an algorithm to encode binary data into a sequence of nucleotides, we must keep in mind certain biological properties. For example, processes of DNA synthesis and sequencing are subject to a variety of errors, which include:

- **Insertion:** addition of one - or more - nucleotide to a DNA sequence;
- **Deletion:** removal of at least one nucleotide from a DNA sequence;
- **Substitutions:** mutation of one - or more - nucleotide. We can have two different types of mutation: *transition*, which are interchanges of two-ring purines (i.e., adenine and guanine) or of one-ring pyrimidines (i.e., cytosine and thymine), or *transversion*, interchanges of purine for pyrimidine bases which therefore involve exchange of one-ring and two-ring structures. These mutations can be differentiated in *synonymous mutations*, which is when DNA mutation does not lead to an amino acid change, and *non-synonymous mutations*, which can result in an amino acid change. This last kind of error can be split in two further categories:
  - **Missense mutation:** one amino acid replaced by another amino acid;
  - **Nonsense mutation:** amino acids replaced by a stop codon.

This nonsense mutation is the most problematic during translation (i.e., conversion of *mRNA* into a protein) because it leads to truncated amino acid sequences, which in turn results in truncated proteins. The probability of some forms of errors can be mitigated by encoding digital data into base 3, instead of using the technique illustrated in Section 4.1. A base 3 encoding is proposed by Goldman [8], which uses a rotational code to avoid *homopolymers*, which are continuous repetitions of the same DNA base that can increase the likelihood of sequencing errors[17]. The whole process is illustrated in Figure 3.

First, binary data is encoded in base 3 using a Huffman code [11], which compresses data based on the frequencies of the characters. Since each character is composed of 8 bits, we need at least $2^8 = 256$ base 2 representations. The closer we can get using base 3, results in $3^6 = 729$ different ternary strings (with $3^5 = 243$ representation we would not be able to convert all the characters). Thus, we are wasting $726 - 256 = 473$ states that are never used because for sequences of length five, we result in 13 characters missing. However, using the Huffman code, we can use five digits to represent the most common characters and six for the least common ones. In this way, we can maintain a reasonable overhead (i.e., one more digit) over a whole binary file. Finally, the ternary data representation in converted into a nucleotide sequence using a rotational code to avoid continuous sequences of the same DNA base.

One may notice that, in the end, this encoding is less efficient than the basic encoding approach. In fact, to encode one byte the most obvious encoding requires 4 nucleotides, while the one proposed in [8] uses 5-6 nucleotides (depending on the frequency of the character). On the other hand, we already pointed out that this last method is less error prone, and thus more reliable. On the other hand, this technique increases significantly the whole strand length (i.e., the file size). A trade-off between the two alternatives might be the XOR-encoding proposed in [6], which has a reliability similar to the one described by Goldman, but is twice as dense.

4.3 Priority Content Management

Once the encoding process is achieved, the resulting data is placed into motile-restricted bacteria clusters, where each cluster represents different libraries of information. As previously stated, to achieve a reliable behaviour clusters should be placed within the convex hull defined by the three chemoemitters. Given that there is only a finite space available, this limits the number of clusters that can be placed within the convex hull.

In this section, we propose a content management system that takes advantage of this limitations and provides the opportunity to prioritize information. In particular, we propose a content system where higher priority information is stored into clusters placed inside the convex hull, whereas the lower priority information is stored outside the convex hull, in less reliable clusters. The corresponding algorithm is shown in Algorithm 2. As an example, which is illustrated in Figure 4, we concentrate on four clusters of bacteria. For this we distribute the four clusters in two macro areas: one inside the triangle and one outside of it. Consequently, we can distribute different kind of content (i.e., data with different access priority) into the different clusters. From the model proposed in Section 3.2, we expect to have a higher chance of retrieving encoded information from the cluster placed inside the convex hull (i.e., inside the triangle). Thus, data with lower access priority are inserted in the clusters placed outside the convex hull.
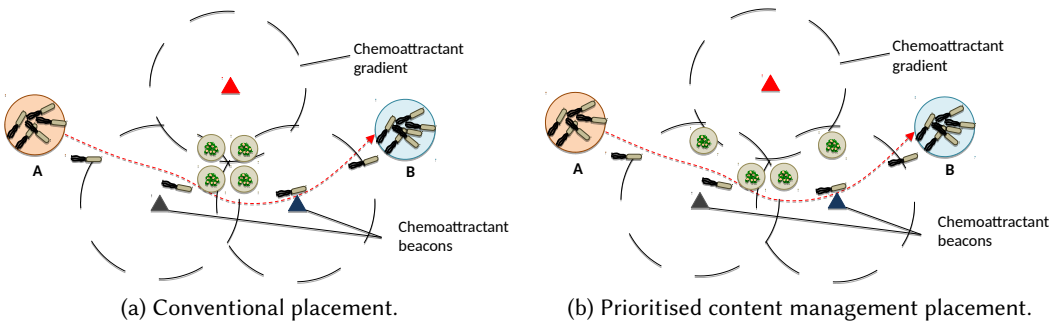

(a) Conventional placement.   (b) Prioritised content management placement.

Fig. 4. Comparision of the clusters positioning.

For example, we can imagine how a web browser can be stored using this mechanism. Firstly, we want to be sure that all the core functionalities (i.e., retrieving and presenting information from the Web) can be retrieved at each access to the storage system. Consequently, we place these features in the cluster inside the convex hull. Finally, we insert in the clusters outside the convex hull all the additions that are not fundamentals for the functioning of the browser (e.g., high-level CSS implementations, add-ons, plugins, etc.). This scenario is represented in Figure 4b.

5 SIMULATIONS

In this section, we discuss the variety of simulations that we run, aiming to establish the accuracy of our positioning system and the efficiency of the whole DNA archive system.

First, we conduct some analysis over the Molecular Positioning System (MPS) which is essential for our proposal, By running these simulations, we assess the ideal chemoemitters configuration, in order to achieve the most accurate positioning possible for the motile bacteria. Second, after learning that the best configuration is achievable by positioning all the clusters within a convex hull area, we assess the capability of retrieving a chunk of data from such a configuration, by using the aforementioned MPS. Last, we assess the feasibility of a content management system, by storing

---

ALGORITHM 2: Content distribution based on priority

Input: Number of clusters with low/high priority.
Output: Clusters containing the encoded data.
N + M = number of clusters;
Set priority of N clusters to low;
Set priority of M clusters to high;
while *inserted(data) < amount(data)* **do**
  stored ← False;
  forall *clusters* **do**
    if *notFull(cluster)* **and** *prority(data) = priority(cluster)* **and** *stored = False* **then**
      Store data into the cluster;
      stored ← True;
    end
  end
end

---

low-priority information into 2 out of 4 clusters, and positioning these clusters outside the convex hull.

5.1 Positioning System

We execute a number of simulations to asses the efficacy of MPS; every independent simulation uses 100 engineered motile bacteria, where there are no interaction signalling between them. Table 1 presents the default parameters used in our simulations. Figure 5 presents an example of a MPS positioning event. Initially, due to their random drifting angle, all the engineered motile bacteria take a different path as they mobilize towards the destination. However, as they approach the target, the paths tend to regroup and the engineered motile bacteria converge towards the target position.

Table 1. Default Parameters

Parameter	Default Value
t	$1000\ (s)$
Δt	$2 \cdot 10^{-2}\ (s)$
v	$5 \cdot 10^{-3}\ (cm/s)$
D	$5\ (rad^2/s)$
ψ_A	$3,49 \cdot 10^{-2}\ (rad)$

Our initial evaluation aims to determine the performance of the MPS based on variations of the target position. In particular, our goal is to compare the average positioning precision when the destination is placed inside and outside the convex hull. Furthermore, we want to verify if fluctuations appear in these two different cases.

First, we position three beacons such that their convex hull is an equilateral triangle and place seven concentric circles of different radii, as shown in Figure 6. For the four inner circles, we place the destination targets on the circle radii of 0.030 (cm), 0.058 (cm), 0.087 (cm) and 0.200 (cm), while the motile bacteria starting points are placed on the three outer circles, which radii are 0.300 (cm), 0.350 (cm) and 0.450 (cm). Each independent run consists of 100 engineered motile bacteria that share starting points and destination points, so that all the combinations of points are equally and separately evaluated. This results in a total of 576 independent simulations.
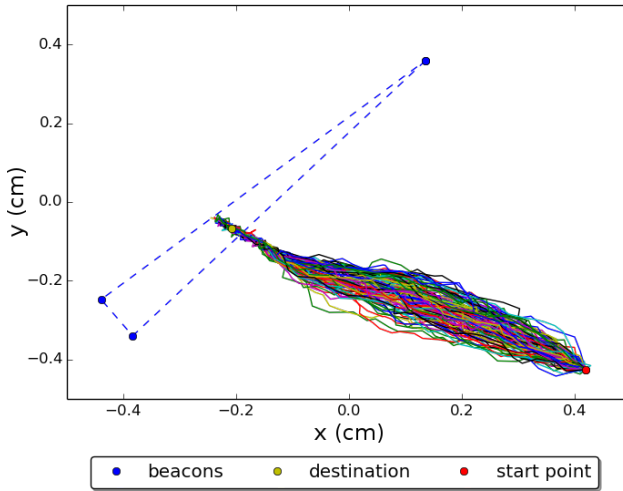
Fig. 5. Example of a simulation run. The engineered motile bacteria do not follow the same path, due to the random component in their movement algorithm [18].

Figure 7 shows that the choice of the destination has a high impact on the results. We can observe that if the target location is contained within the convex hull, the results are consistent throughout the simulations: *Circle A* and *Circle B* bars highlight that the average positioning error remain under $0,05$ (*cm*) with a very small standard deviation. On the other hand, *Circle C* and *Circle D* bars, which represent the destinations outside the convex hull, show that the positioning error grows considerably as the distance from the barycentre increases. As well as the average error, the standard deviation also increases and becomes more evident in *Circle D* bars. These last results are particularly relevant, since they confirm the geometric concept that we expressed in Section 3, where the farther the destination point is from the convex hull, the more overlapping area of the three chemoattractants, which entails a lower precision.

5.2 Retrieving Archived Information

After evaluating the feasibility of our MPS, we can use it to implement our Reading process from the DNA archive system, as described in Section 3. In order to prove the robustness of our model, we want to check if it is possible to retrieve the desired encoded file as we vary certain parameters in the simulator. Our simulations focus on three main points: the *number* of engineered motile bacteria necessary to retrieve the whole file in a limited time slot, *randomness* of engineered motile bacteria movement (related to factor D described in Section 3.1), and the *time* required to retrieve the whole file by varying the first two parameters.

In order to do so, we create a file of 18.4 (*KB*) and encode it into DNA, which is distributed in the bacteria placed into four different clusters (e.g., encoding of the file generates m plasmids, which are distributed in n clusters that consequently contains $\frac{m}{n}$ plasmids). Each bacteria has an average capacity of 100 plasmids (i.e., normal distribution with average of 100 and standard deviation of 10), where each of the plasmids is composed of 200 base pairs (bps). We impose a limit of 120 (min) to retrieve and deliver the whole file.. We also assume the probability associated to conjugation as a normal distribution $N(0, 1)$, and we use a threshold (0.5) to decide the occurrence of the event. In order to start conjugation, we suppose that two bacteria should have a distance less than 10 (*nm*). To improve the likelihood of retrieving the data, we vary the number of bacteria - both retrievers
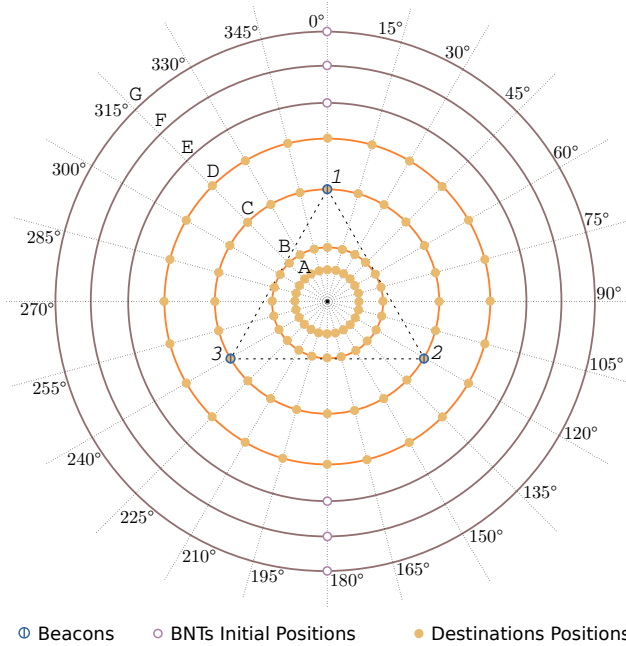
Fig. 6. Diagram that illustratess the performance inside and outside the beacons convex hull. For each initial bacterium position (spots on circles *E*, *F* and *G*) we investigated every final destination (spots on circles *A*, *B*, *C* and *D*), which totals up to 576 independent simulations. The circles are centred at the triangle barycentre and, in order to meet spacial requirements, the radii sketched in this figure do not represent their real ratio with respect to the whole picture.
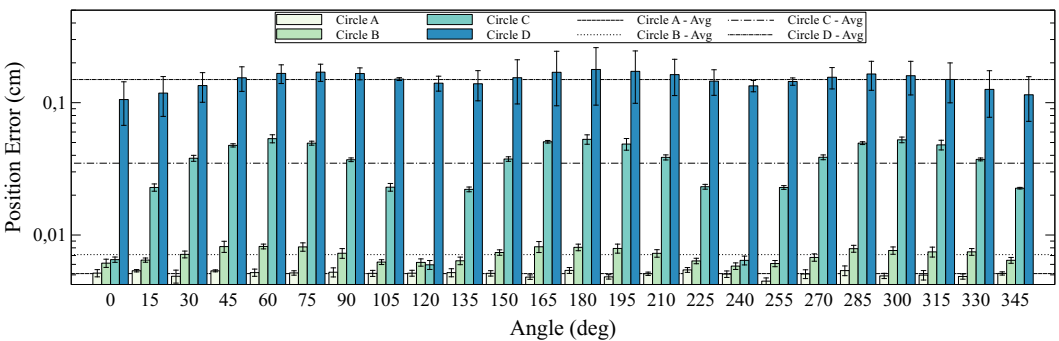

Fig. 7. Results are consistent and stable throughout all the simulations with destinations placed on *Circle A* and *Circle B*, whereas results clearly worsen when destinations are placed outside the convex hull, on *Circle C* and *Circle D*. We have chosen a logarithmic scale on Y axis to better depict such high variety of data. Each bar represents a set of 6 independent simulation, which totals up to 576 independent simulations.

for each cluster and bacteria used as storage inside the clusters - in the range [10, 150] with a step of 10 (e.g., 10, 20, 30). In this way, we keep a hypothetical ratio of 1 between the number of engineered bacteria used as retrievers and the number of the motility-restricted bacteria used as DNA storage. However, due to the random spatial distribution over the cluster area, we can not be certain that

all bacteria are in a position that can be reached by the retrievers. To emulate this uncertainty, we impose as 50 the maximum number of bacteria that can simultaneously conjugate inside each cluster.

The experiment is based on three main locations: the start point (A), the center of the triangle composed by the chemoemitters indicating the storage area (B) and the center of the triangle composed of chemoemitters that indicate the end point (C). These three points are vertically aligned; the distance between points A-B and B-C is 0.4 (cm). The clusters are placed at equal distance from point B (0.02 (cm) on both X and Y axis), composing a square. We place in point A the engineered bacteria used as retrievers, switching on their chemoattractants receptors for the chemoemitters around the storage area. In this way, bacteria are able to swim towards the clusters to conjugate with the motility-restricted bacteria inside. Once conjugation occurs, two bacteria take about 120 (min) to exchange genetic material. Once horizontal gene transfer has happened, we switch off the chemoreceptors for the storage area and we switch on the ones that enable the engineered bacteria to reach point C. Finally, when the bacteria reach their destination, they random-walk. In case that bacteria drift too much away from their destination, the receptors would be no more saturated and bacteria would try to approach the point again, as instructed by our MPS.

The evaluation is conducted by varying the following process and parameters of the simulator:

- *Engineered motile bacteria*: 10 to 150 with a step of 10, creating 15 different simulations;
- *Random motility factor*: from 5 to 32 with a step of 3, leading to $10 \cdot 15 = 150$ different runs;
- *Encoding algorithm*: Basic and Goldman encoding.

All these different simulations are executed 10 times. Consequently, the results discussed in this section are calculated at an average of 1500 different simulations for each encoding algorithm (3000 simulations in total).

As we expected, Figure 8a and Figure 8b shows that increasing the number of bacteria has a beneficial effect for the basic and Goldman encoding techniques. Indeed, the two figures show that increasing the throughput of our system, by increasing the number of engineered bacteria, leads to a lower time to deliver the whole file to the destination point. Unsurprisingly, both figures highlight the detrimental effect of increasing the bacteria randomness D, with respect to the time that it takes to retrieve the whole file at the destination. The same conclusions can be drawn by comparing Figure 9a and Figure 9b, where lower number of bacteria and a high randomness in their movement makes it harder to direct them to their destinations, therefore harder to retrieve the whole file.

Comparing Figure 8a with Figure 8b, and Figure 9a with Figure 9b, the basic encoding performs clearly better than the Goldman encoding. This result is easy to explain, in order to be more resilient to mutations, the Goldman encoding uses more nucleotides than the basic encoding, and this overhead increases the time needed to complete the conjugation process.

Considering that the default value for the random component D is 5 (rad^2/s), it is remarkable that with 14 rad^2/s our system is still able to retrieve the whole file within the set time threshold, if enough bacteria (i.e., more than 80) are used. As a matter of fact, Figure 8a and Figure 9a shows that the basic encoding is still functioning even with randomness of 17 (rad^2/s).

5.3 Content Management

In Section 4.3 we discussed about how different topologies of clusters can lead to a content management system for the DNA archive system. Based on the reliability of the MPS, this means placing the more frequently accessed data inside the convex hull and the less priority contents outside of the convex hull defined by the beacons. Here, we test the file retrieval ratio (in percentage)
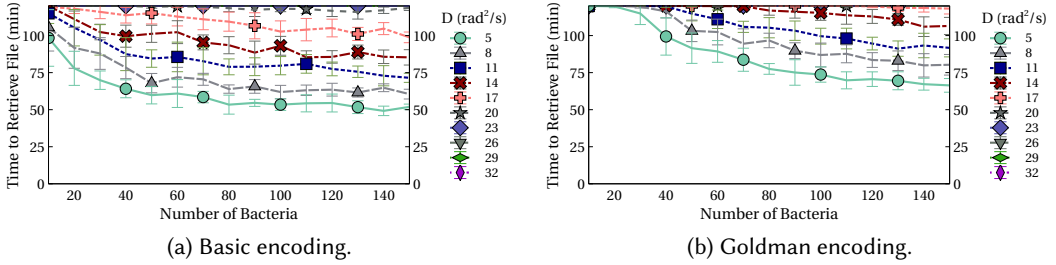
Fig. 8. Relationship between number of engineered motile bacteria, random factor D and time needed to retrieve the whole file, both with the Basic and the Goldman encoding. If the time is greater or equal to 120 minutes, then the engineered motile bacteria were not able to retrieve all the information.
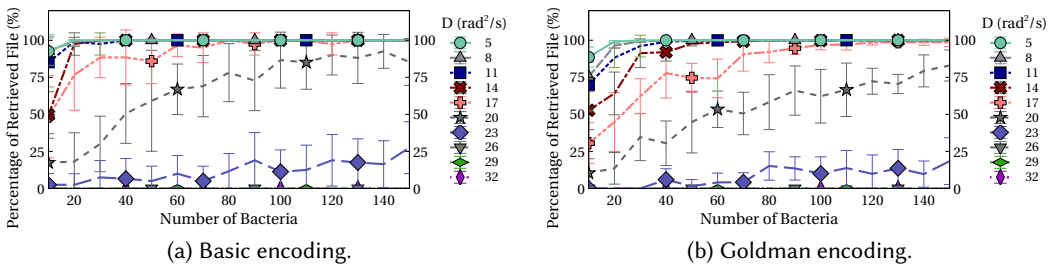

Fig. 9. Relationship between number of engineered motile bacteria, random factor D and percentage of file retrieved, both with the Basic and the Goldman encoding. If the percentage is lower than 100%, it means that the engineered motile bacteria were not able to retrieve the whole file in less than 120 minutes.

achieved by the bacteria and how the two different encoding techniques affect the whole system performance. We execute the same set of simulations described in Section 5.2, however, we place the two uppermost clusters outside the convex hull area and on the same abscissa of the previous simulations, but horizontally aligned to the two emitters composing the base of the triangular convex hull.

Analysing Figure 10a and Figure 10b, it is easy to verify again that the Basic encoding performs better than the Goldman encoding technique even for the content management system. This is based on the results that shows that with sufficient bacteria and a reasonably low D value, the content management system sometimes functions with the Basic encoding. However, the same does not hold for the Goldman encoding, which never enables us to retrieve the whole file within the 120 (min) threshold.

Looking at Figure 11a and Figure 11b, we want to remark that each graph line is the resulting average of 10 different simulations. Therefore, we can conclude that on average we cannot be sure that the whole file can be retrieved, but the standard deviation bars for the Basic encoding shows that with sufficient bacteria and a low randomness, the successful retrieval can be achieved.

On the contrary, with the Goldman encoding the whole file is never retrieved since not even a single standard deviation bar hits the 100% mark. However, with a content management system in mind, losing certain data is not inherently a critical issue, where this could be compensated by error detection and correction algorithms. These algorithms could be set at the destination point to assess the integrity of the file and, if possible, fix it. The other option is for the archive owner to
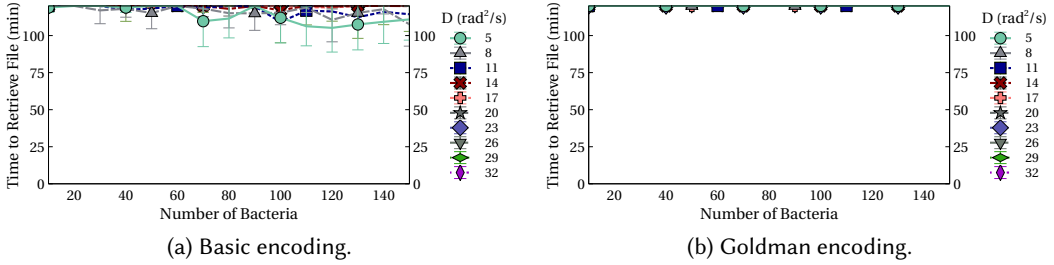
Fig. 10. Relationship between number of engineered motile bacteria, random factor D and the time required to retrieve the whole file, both with the Basic and the Goldman encoding. If the time is greater or equal to 120 minutes, then the engineered motile bacteria were not able to retrieve all the information.
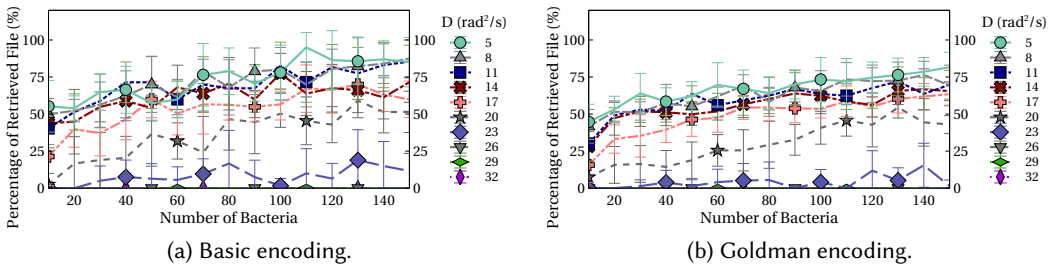

Fig. 11. Relationship between number of engineered motile bacteria, random factor D and percentage of file retrieved, both with the Basic and the Goldman encoding. If the percentage is lower than 100%, it means that the engineered motile bacteria were not able to retrieve the whole file in less than 120 minutes.

decide to place in the outermost clusters the lower priority data, which can be retrieved without any time constraints or hard deadlines.

6 WET LAB EXPERIMENTS

In this section we discuss wet lab experiments to demonstrate our concept of Reading from DNA message-encoded plasmids that are stored in motility restricted bacteria. Figure 12 illustrates our wet lab experimental setup that is based on engineering an agar plate to have a channel with motility agar. The surrounding portions of the agar plate, as well as the centre island (*B*), is hard agar that ensures no motility can occur. In this experiment, the motile bacteria are released from *A*, and swim towards *C*, and along the way conjugate with the motility-restricted bacteria in *B*. The conjugation process leads to the motile bacteria picking up the plasmid with the encoded information, which also contains the antibiotic resistance gene from the motility-restricted bacteria. The encoded information in the motility-restricted bacteria is **Hello World**. The successful pick of the plasmid allows the motile bacteria to survive the antibiotics in *C*, the destination of the message, as illustrated in Figure 13b.

The first step of the experimental work, after the *pWITGLO* plasmid (please see Appendix A for Material and Methods) was constructed, was to demonstrate the conjugation process, and this is illustrated in Figure 13 (a). Two *E.coli* K-12-derived commercially available recombinant cloning strains were used; *Novablue* and *HB101*. *Novablue* is Tetracycline resistant, *HB101* is *Streptomycin* resistant. *Oligonucleotide* primers were designed to amplify the full GFP gene from *pGlo* using
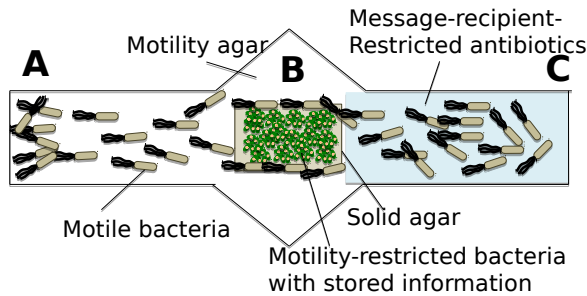
Fig. 12. Illustration of the wet lab experimental set up, where the motility-restricted bacteria with the encoded "Hello World" is placed at B. Non-motile bacteria swim from A towards C and pick up the message along the way.
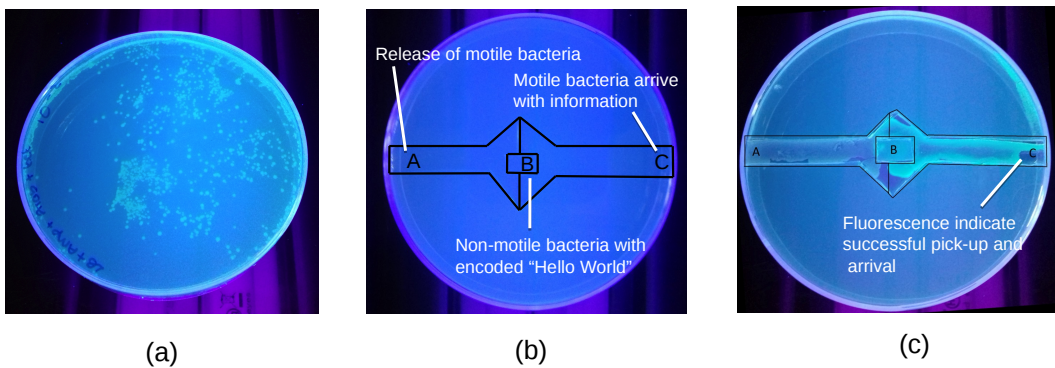

(a) (b) (c)

Fig. 13. Wet lab experimental result demonstrating the successful pick-up of the encoded "Hello World" message from the non-motile bacteria in B by the motile bacteria released at A. After picking up the message, the motile bacteria swims towards C to deliver the message. (a) demonstrates the successful conjugation process, (b) presents the engineered agar plate illustrating the placement of the non-motile bacteria with the encoded plasmid of "Hello World", and the channel for the motile bacteria to swim from A to C, (c) results of the experiment, where the fluorescence indicates the motile bacteria successfully conjugated with the non-motile bacteria to pick up the encoded plasmid with antibiotic resistance gene and surviving the antibiotics.

PCR, with the *"Hello World"* message contained in the reverse primer immediately after the stop codon, resulting in the message immediately downstream of the gene as a single stranded overhang. *Klenow* polymerase was then used to fill in this overhang, resulting in a complete double stranded PCR product; the GFP gene and the message. This "filled in" product was then prepared for ligation/insertion to a cloning vector(*TOPO pEXP5/CT*) before being transformed into *Novablue* cells. The resulting recombinant plasmid was named *pWITGLO*. For the conjugation demonstration experiment, fresh growing cells of each strain were mixed for 1 hour to encourage conjugation and then spread onto agar containing the correct inducer *IPTG* (induces GFP expression and fluorescence), *streptomycin* (*HB101* can tolerate this, *Novablue* cannot) and ampicillin (selects for the plasmid). All resulting colonies growing and fluorescing are *HB101* having taken up *pWITGLO* from *HB101*, now resistant to both antibiotics.

The next experiment is to demonstrate the DNA encoded information Reading process. LB agar plates containing *kanamycin*, so neither strain can grow, were used to flank motility channels.

These motility channels consisted of motility agar, which is softer, allowing motility and the spread of cells. Figure 13 (c) shows, after inoculation, the successful Reading process. Zone *A* contained *Streptomycin* so *HB101* (without *pWITGLO*) can only grow, and the plate was inoculated at *A* with *HB101*. This strain was seen to grow/move towards *B* throughout the motility agar. In the middle of the plate, *B*, was a square plug of motility agar containing *IPTG*, *tetracycline* and *ampicillin* inoculated with *Novablue* containing *pWITGLO*, which was fluorescent. This strain was motility-restricted to zone *B* as it is sensitive to the surrounding *Streptomycin*. *HB101* was unable to grow within zone *B* due to the presence of *Tetracycline*. Conjugation occurred at the interface of Zones *A* and *B*. In zone *C* was motility agar with *IPTG*, *ampicillin* and *streptomycin*. As illustrated in the picture, only *HB101* cells that have moved from *A* to *B*, picked up *pWITGLO* from *Novablue* have grown here, now resistant to both antibiotics and able to fluoresce. This occurred within 72 hours, at a distance of 10 cm from *A* to *C*. The fluorescence in *HB101* is higher than *Novablue*, with an increase of approximately 250 percent in fluorescence. *HB101* cells with *pWITGLO* were inoculated from *C* to fresh liquid broth and *pWITGLO* plasmid was purified from the resulting culture, with very high amounts of plasmid recovered; 380 *ng/ţl* concentration, 38 *mg* of total plasmid DNA was recovered from 3 *mls* of culture (10 *mg* of wet cells). DNA sequencing of the GFP gene and DNA message confirmed transfer of the message to point C. The performance results are presented in Table 2.

Table 2. Performance results

Factor	Results (unit)
Time taken for bacteria from *A* to reach *C*	72 (*hours*)
Relative fluorescence of *HB101* cells with *pWITGLO* versus *Novablue*	250 (*percent*)
Concentration of *pWITGLO* recovered from 10 *mg* of wet cells	380 (*ng/l*)

7 CONCLUSION

The continual growth of information created by the Internet has led to researchers searching new solutions for data storage. Digitially encoded DNA has recently been proposed as a promising solution for storing large quantity of data anticipated in the near future. In this paper, we combined the concept of digitally encoded DNA with molecular communications. The molecular communication system proposed is bacterial nanonetworks, where the bacteria mobilize and move towards spatial addresses based on engineered receptor saturation, in order to pick up the DNA encoded information from stationary non-motile bacteria. We term this directed movement as Molecular Positioning System, and is based on the concept of trilateration of a location placed inside and outside the convex hull of the chemoattractant field. Our positioning technique through simulations has shown that they can can consistently keep their positioning error under under $0,05$ (*cm*) if their destination is placed within the convex hull. Through our simulations, this has shown to very beneficial, where we compared the realiability of the bacteria to pick up information from two types of encoding techniques, which includes Basic and Goldman encoding. Our simulations has shown that the basic encoding basically achieves higher performance compared to the Goldman, given the lower overhead in the size of the DNA after encoding. We have also presented the concept of prioritizing data placement in the non-motile clusters at various locations depending on the overlapping chemoattractants. This allows the numerous data to be placed in different regions, in

order to cater for speed and reliability of retrieving the data in a timely manner. Lastly, the paper presented wet lab experimental work to demonstrate the validity of using non-motile bacteria to store a short message, where motile bacteria conjugate to pick up this message and deliver it to another location. Our work has demonstrated the feasibility of developing an archive system for digitally encoded DNA, which to date has remained as a big challenge for researchers.

REFERENCES

[1] I. F. Akyildiz, F. Brunetti, C. Blazquez, "Nanonetworks: A New Communication Paradigm," Computer Networks, June 2008.

[2] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, Y. Koucheryavy, "Internet of Bio-NanoThings," IEEE Communications Magazine, vol. 53, no. 3, pp. 32-40, March 2015.

[3] S. Balasubramaniam and P. Lio', "Multi-Hop Conjugation Based Bacteria Nanonetworks," in IEEE Transactions on NanoBioscience, vol. 12, no. 1, pp. 47-59, March 2013.

[4] A. Beloglazov, J. Abawajy, R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, Volume 28, Issue 5, 2012, Pages 755-768, ISSN 0167-739X, http://dx.doi.org/10.1016/j.future.2011.04.017.

[5] M. Blawat et al. "Forward error correction for DNA data storage," Procedia Computer Science, vol. 80, pp. 1011-1022, 2016.

[6] J. Bornholt, R. Lopez, and D. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-Based Archival Storage Sistem," at ASPLOS 2016 (International Conference on Architectural Support for Programming Languages and Operating Systems), April 2016

[7] A. Giaretta, S. Balasubramaniam and M. Conti, "Security Vulnerabilities and Countermeasures for Target Localization in Bio-NanoThings Communication Networks," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 4, pp. 665-676, April 2016.

[8] N. Goldman, P. Bertone, S. Chen, et al. "Toward practical high-capacity low-maintenance storage of digital information in synthesised DNA," Nature, vol. 494, no. 7435, pp. 77-80, 2013.

[9] R. N. Grass, "Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes," Angewandte Chemie International Edition, vol. 54, no. 8, pp. 2552-2555, 2015.

[10] A. Guney, B. Atakan and O. B. Akan, "Mobile Ad Hoc Nanonetworks with Collision-Based Molecular Communication," in IEEE Transactions on Mobile Computing, vol. 11, no. 3, pp. 353-366, March 2012.

[11] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," in Proceedings of the IRE, vol. 40, no. 9, pp. 1098-1101, Sept. 1952.

[12] R. Miller, "Facebook builds exabyte data centers for cold storage," http://www.datacenterknowledge.com/archives/2013/01/18/facebook-builds-new-data-centers-for-cold-storage/, 2013.

[13] M. Moore and T. Nakano, "Addressing by beacon distances using molecular communication," Nano Commun. Netw. (Elsevier) J., vol. 2, pp. 161-173, 2011.

[14] M. Moore and T. Nakano, "Addressing by concentrations of receptor saturation in bacterial communication," In Proceedings of the 8th International Conference on Body Area Networks (BodyNets '13), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 472-475, 2013.

[15] S. Murugesan, "Harnessing Green IT: Principles and Practices," in IT Professional, vol. 10, no. 1, pp. 24-33, Jan.-Feb. 2008.

[16] T. Nakano, M. Moore, F. Wei, A. T. Vasilakos, and J. W. Shuai, "Molecular communication and networking: Opportunities and challenges," IEEE Transactions on NanoBioscience, vol. 11, no. 2, pp.135-148, June 2012.

[17] T. P. Niedringhaus, D. Milanova, M. B. Kerby, M. P. Snyder, and A. E. Barron, "Landscape of next-generation sequencing technologies," Anal. Chem., 83:4327ÃćÂÂŞ4341, 2011.

[18] Y. Okaie et al., "Cooperative Target Tracking by a Mobile Bionanosensor Network," in IEEE Transactions on NanoBioscience, vol. 13, no. 3, pp. 267-277, Sept. 2014.

[19] S. L. Shipman, "Molecular recordings by directed CRISPR spacer acquisition", Science, vol. 353, no. 6298, 2016.

[20] K. Suyehira, "A Coding Scheme for Nucleic Acid Memory (NAM)," IEEE Workshop on Micro-electronics and Electron Devices (WMED), 2017.

[21] S.M. Yazdi, et al. "A rewritable, random-access DNA-based storage system." Scientic reports 5 (2015).

A MATERIALS AND METHODS

A.1 Construction of pWITGLO

Oligonucleotides were designed to amplify the complete GFP gene from *pGlo* plasmid (*BioRad*) via PCR, with the *Hello World* message contained in the reverse primer immediately after the stop codon in a reverse complement orientation (in bold) (GFP F: `atggctagcaaaggagaaga`, GFPM R: `tgcgtacgctagaacgagggttctaacgtacgtacgggcg`
`tctgttatttgtagagctcatcca`). *Oligonucleotides* were supplied by Eurofins, MWG operon, Germany. PCR conditions used to amplify the GFP gene and message were as follows; each 15 μL PCR reaction mixture contained 7.5 μL Q5Âő High-Fidelity DNA Polymerase Master Mix (NEB), 15 *pmol* of each primer and 15 *ng pGLO* plasmid DNA. PCR conditions: 1 cycle of 95 ℃ for 5 min, 30 cycles of 95 ℃ for 1 min, 56 ℃ for 1 min, 72 ℃ for 1 min, 1 final extension stage of 8 minutes. Negative controls contained water in place of DNA. PCR products were analysed by agarose gel electrophoresis.

PCR products were cleaned using the Clean & Concentrator$^{\text{TM}}$ 5 kit (Zymo Research) as per manufacturer's instructions. DNA polymerase I, large Klenow fragment, (Thermofisher Scientific) was used to fill in the DNA message overhang on cleaned PCR product as per manufacturer's instructions. This double-stranded GFP-message construct was then subjected to A-overhang addition; this μL reaction contained *Taq* DNA polymerase ((*Taq polymerase Amresco*)) and 10 *mM Datp* per reaction (*Amresco*), to ready the DNA for ligation to a cloning vector. The double-stranded A-overhang GFP-message construct was ligated to *pEXP5-CT/TOPO* vector (ThermoFisher Scientific) as per manufacturer's instructions. Plasmids were transformed into Novablue Gigasingles$^{\text{TM}}$ (*Novagen*) with positive clones identified by GFP-message PCR assay. Plasmid DNA (*pWITGLO*) was purified using the MonarchÂő Plasmid Miniprep Kit (New England Biolabs). Sequencing of plasmid insert was carried out in triplicate using the vector-specific universal primers *T7 promotor* and *T7 terminator*, with all sequencing carried out by GATC Biotech, Germany. Nucleotide sequences were analysed using Blastn software from the NCBI database.

A.2 Bacterial strains

E.coli strains used were *E.coli Novablue* (*Novagen*) (genotype (*endA1 hsdR17(rK12-mK12+) supE44 thi-1 recA1 gyrA96 relA1 lac F'[proA+B+ laclqZ'M15::Tn10] (TetR)*) and *E.coli HB101* (BioRad) (genotype (*F- Lambda- araC14 leuB6(Am) DE(gpt-proA)62 lacY1 glnX44(AS) galK2(Oc) recA13 rpsL20(strR) xylA5 mtl-1 thiE1 hsdS20(rB-, mB-)*).

A.3 Growth conditions

E.coli strains were grown on LB agar (Sigma Aldrich) with appropriate antibiotic; 12.5 μg/ml *Tetracycline* and *Streptomycin* 25 μg/ml (Sigma Aldrich) for *Novablue* and *HB101* respectively, at 37℃. *Novablue* cells transformed with *pWITGLO* were grown and maintained on LB media containing 12.5 μg/ml *Tetracycline*, 100 μg/ml *Ampicillin* (Sigma-Aldrich) and 0.5 *mM IPTG* (Sigma-Aldrich).

A.4 Motility and conjugation experiments

Motility experiments were carried out by stab-inoculating motility agar (Sigma Aldrich) and incubating at 37℃ for 3 days, observing motility. Conjugation experiments were carried out by harvesting Novablue containing *pWITGLO* and *HB101* cells from mid-log phase cultures (grown in LB broth and appropriate antibiotic), washing cells three times in 0.85 percent saline solution, resuspending both cell cultures together in LB broth and incubating with gentle agitation at 37 ℃ for one hour, before plating onto the surface on LB agar containing *Streptomycin* 25 μg/ml, 100 μg/ml Ampicillin and 0.5 *mM IPTG*.

A.5 DNA Message Transfer experiment

Message transfer agar channel plates were prepared as follows; channel zones *A*, *B* and *C* were removed from LB agar containing 30 μg/ul Kanamycin. Motility agar (Sigma-Aldrich) containing appropriate antibiotics were poured into channel zones A (25 μg/ml Streptomycin), B (100 μg/ml Ampicillin,12.5 μg/ml tetracycline, 0.5 *mM IPTG*) and *C* (*Streptomycin*, *Ampicillin* and 0.5 *mM IPTG*) in turn. Strain *HB101* was stab-inoculated to Zone *A* and *Novablue* containing *pWITGLO* was stab-inoculated to Zone *B* simultaneously. Plates were then incubated at 37℃ for 3 days.