# Hiding Data in DNA

Boris Shimanovsky[1], Jessica Feng[2], and Miodrag Potkonjak[2]

[1] XAP Corporation
[2] Dept. Computer Science, Univ. of California, Los Angeles

**Abstract.** Just like disk or RAM, DNA and RNA can store vast amounts of information, and just like data stored in digital media, DNA data can be easily copied or tampered with. However, unlike in the digital realm, there are no techniques for watermarking, annotating, or encrypting information in DNA and RNA. The ability to catalogue genes, place checksums, watermark, and otherwise protect intellectual property in such a medium is of profound importance. This paper proposes the original idea of hiding data in DNA and RNA. Moreover, it defines two new and original techniques for hiding the data. The first is a simple technique that hides data in non-coding DNA such as non-transcribed and non-translated regions as well as non-genetic DNA such as DNA computing solutions. The second technique can be used to place data in active coding segments without changing the resulting amino acid sequence. Combining codon redundancy with arithmetic encoding and public key cryptography, this robust technique can be used simultaneously for encryption and authentication. Protecting genetic discoveries, gene therapy drugs, and a great deal of other intellectual property in medicine, genetics, molecular biology, and even DNA computing, could be made possible by the techniques presented in this paper.

## 1 Introduction

Learning new methods for storing data is of fundamental importance to us. Science constantly seeks novel ways of retaining and laying claim to information. And what information storage medium could be of more profound importance than the one that details the construction of life as we know it? DNA and RNA, over millions of years, have demonstrated their effectiveness as a coding medium for the instruction set that governs and propagates living things. Of late, their medicinal usefulness and propensity for solving complex, highly parallel computational problems have also been demonstrated. The ability to hide, watermark, and annotate information within this medium is clearly meaningful.

Molecular biology and genetics are heavily researched fields with a great deal of intellectual property and colossal amounts of money to protect. Academic and commercial institutions may feel more comfortable conducting years of costly research for isolating and recombining a gene responsible for some vital enzyme, if

F.A.P. Petitcolas (Ed.): IH 2002, LNCS 2578, pp. 373-386, 2003.

there was some way to embed a watermark identifying their contribution and claim to the work. DNA sequences, themselves, cannot be patented, and rightly so. However, the nature of DNA is such that, once the real work of isolating and identifying a useful sequence has been done, copying is trivial.

Moreover, DNA has demonstrated that it has significant computational power [1], being able to perform trillions of parallel operations, using little energy, space, and money. Still in developmental infancy, DNA computing is already getting a lot of attention, not only due to the novelty of the whole thing, but also because theoretically, it can do things in hours or days that today's digital computers cannot do in a lifetime. There is already a lot of attention paid to watermarking digital circuitry [2, 3] and solutions made by digital computers. It seems useful to consider the same kinds of things with regard to DNA.

DNA is a coding medium. Just like a disk or RAM, DNA strands contain information that can be interpreted and copied. However, rather than a binary representation of zeroes and ones, which are assigned for human comprehension, DNA contains sequences of 4 nucleic acids—adenine, thymine, guanine, and cytosine. These nucleotides could be used to encode binary information. Continuing the analogy, hiding a secret message in a binary sequence can be accomplished either by adding the message and increasing the overall size of the sequence, or by altering some portion intelligently such that the data is not functionally or perceptibly altered. To embed such a message, one would not arbitrarily add or intersperse information. It requires a complete understanding of the original message and the machinery that processes it. Similarly, one would not blindly change a sequence of nucleotides simply to facilitate a hidden message.

The goal of this paper is to propose the theoretical notion of hiding information in DNA and RNA, and offer some general schemes for embedding data.

## 2 Background – Molecular Biology

To understand the possible avenues for hiding information, it is necessary to establish the biological background for the material. The amount of knowledge in DNA processing and genetics is intractable, even in broad swipes, however, since this paper is written with computer scientists and engineers in mind, a general discussion of the central dogma of molecular biology, that is, how a DNA sequence eventually leads to a protein, is warranted. To biologists, the material will seem trite; they are advised to skip ahead.

### 2.1 Basics

DNA molecules are arranged as two oppositely oriented strands with sugar-phosphate backbones joined together in an alpha-helical structure by hydrogen bonding between the complementary nitrogenous bases. The bases, adenine(A), thymine(T), guanine(G), and cytosine(C) represent the "genetic code". A bonds with T and G bonds with C. This pairing is, among other things, useful for the error detection/correction machinery in the cell. Through a long and complex process, these

bases are “read” and eventually translated into chains of amino acids, which form a protein. Proteins are responsible for nearly everything that goes on in our cells, from providing structure, to helping digest nutrients, to catalyzing a reaction in a pathway leading to brown hair pigmentation. The arrangement of the amino acids dictates the structure and function of the protein. The DNA sequence determines the arrangement of amino acids. Through complex interactions, DNA determines the “what”, “when”, and “how much” of everything that our cellular machinery produces. The two main steps in protein genesis are transcription and translation.

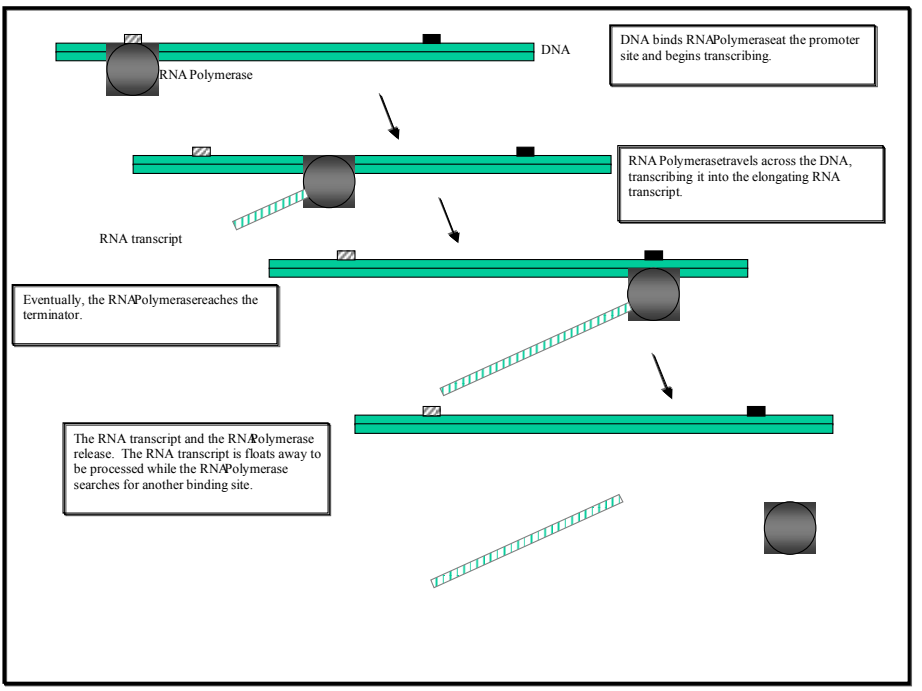



**Fig. 1.** Transcription

## 2.2 Transcription

Transcription is the process in which an intermediary copy of the instructions contained in DNA is created. This intermediary copy, RNA, is single stranded, and contains the nucleotide, uracil, where thymine would appear in DNA. The RNA is what is actually interpreted by the cellular machinery. In this manner, the DNA is retained safely in the cell’s nucleus, while RNA copies, or transcripts, can be created in abundance and translated into proteins.

Transcription begins with the binding of an enzyme called RNA polymerase to the DNA. The polymerase binds to a thermodynamically favorable region of the DNA called a promoter. This promoter acts as the start signal for transcription. Its effectiveness at attracting RNA polymerase is one of the methods used to control how much of a protein is produced. Many deficiency diseases and cancers are linked to

problems with promoter sites, so clearly, data hiding schemes must be sensitive to promoters, which can lie over a thousand bases upstream of the gene. Once bound, RNA polymerase facilitates the binding of bases complementary to the ones on the DNA template it is transcribing. As it elongates, the RNA strand peels away from the DNA. The RNA polymerase eventually reaches a region called a terminator, which causes it to release.

In eukaryotic cells (cells that have a nucleus—except for some simple bacteria, cells are generally eukaryotic), the RNA transcript undergoes a processing step in which INTervening sequences (introns) are cut out and discarded, leaving the EXpressed sequences (exons). These intervening sequences have structural functions and, thus, are not innocuous, however they do not code for protein [4].

Once processed, the RNA transcript is referred to as messenger RNA (mRNA). mRNA leaves the nucleus and binds a structure called a ribosome. The ribosome facilitates the translation of the mRNA sequence into protein.

### 2.3 Translation

mRNA binds two ribosomal subunits, forming a complex. The ribosomes step linearly along the mRNA strand, "reading" it. There are 20 distinct amino acids that can be chained together during protein synthesis. On the mRNA, a grouping of three nucleotides, called a codon, indicates which amino acid will be attached next. The codon binds a group of three nucleotides on a tRNA molecule called an anticodon. There are about 40 distinct tRNA molecules. Each has a binding site for one of the amino acids. In this way, tRNA acts as the medium for translating from nucleic acid code to protein.

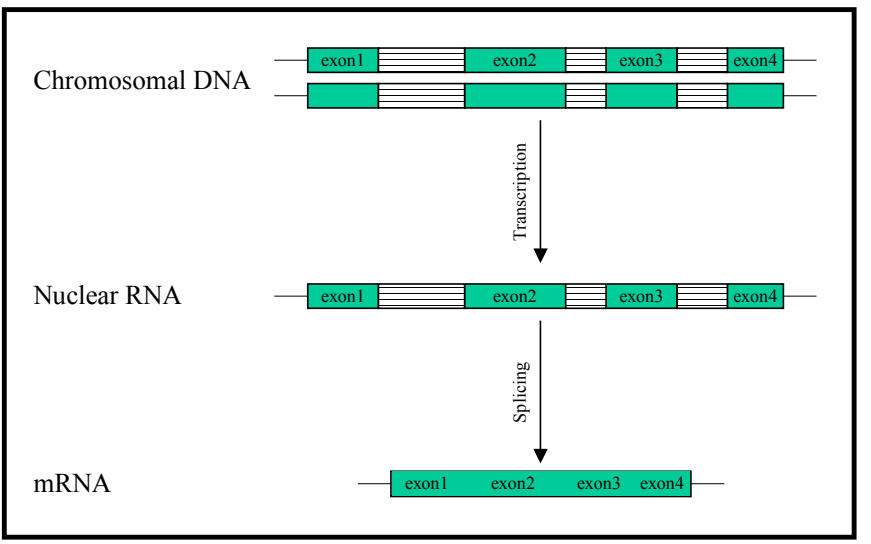



**Fig. 2.** Splicing

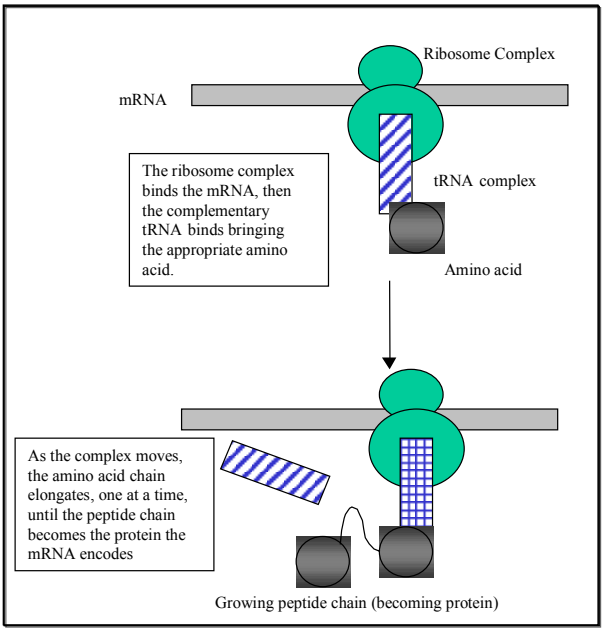



**Fig. 3.** Translation

The appropriate tRNA, bearing the appropriate amino acid, binds to the codon on the mRNA, extending the amino acid chain. Translation is completed when the ribosome encounters a STOP codon and the protein is released [5, 6].

### 2.4 Gene Therapy and Retroviruses

Gene therapy is a field that can benefit greatly from a data hiding scheme, and a bit of background in retroviral gene therapy is required to understand the examples. One approach to gene therapy is to use a retroviral vector to deliver a functional gene to cells with a faulty one. The retrovirus is an extremely simple RNA virus that has the ability to reverse the transcription process and get its RNA converted into DNA, which can then insert itself in the host DNA [7, 8].

The process involves the removal of the virulent components from the retrovirus genome and leaving in the components that are necessary for infiltrating the target cells and delivering the genetic payload. The gene sequence that needs to be delivered is added to the retroviral RNA, completing the "drug". The virus is unleashed upon the target cells, its RNA is released and reverse-transcribed, and the newly formed DNA, which would have ordinarily coded for the production of the virus, encodes the correct sequence for a defective gene.

This results in the ability for a cell to produce a protein that it lacks, and the deficiency of which led to the expression of a particular congenital disease.

## 3 Intellectual Property Protection and Data Hiding

As with digital information, DNA-encoded information can be copied such that the possibility of theft of intellectual property is high. Thus, it makes sense to observe the basic requirements and principles of digital data hiding,

The fundamental requirements when hiding data in digital information are [9]:

- reasonable ease of placement and detection by the legitimate party
- sufficient difficulty in detection and erasure for attackers
- credibility in case of a dispute
- robustness against filtering, compression, or truncation
- reasonable overhead
- no significant change in meaning or function of the original data
- the data being hidden should have error correction/redundancy

Unlike, audio or video, which can be altered in such a way that it is qualitatively noticeable when the signal has been too degraded, DNA is hard to filter for the purposes of defeating a watermark. It is important to bear in mind that DNA, especially if it codes for some biological product is not altered easily without a good understanding of the sequence, at which point, an attacker has been forced to do substantial original work. As such, the criteria of robustness against filtering, compression, or truncation, is not applicable. A DNA and RNA data hiding technique should meet the other basic requirements.

## 4 Hiding Data

To hide data, we need one of three things: the ability to insert a sequence containing the data, to alter an existing innocuous sequence, or to find redundancy in an existing sequence and leveraging it to hide data.

Again, the fundamental principles of digital data hiding techniques are extremely useful in the conception and evaluation of techniques for hiding data in DNA. In fact, encryption and compression techniques that are already proven and widely accepted for digital data can and should be used when storing data in DNA. For one thing, they're proven and widely accepted, which makes for fewer things that have to be proven credible in a dispute. Second, the work has already been done which makes the process simpler. Recall that these considerations are among the key criteria in creating data hiding schemes.

Another key consideration is the purpose of the data hiding scheme. If the goal is to annotate and help catalog sequences, then encryption and resistance to attack is of minimal concern, but ease of placement and retrieval are of primary importance. On the other hand, if the data is being hidden specifically for the purpose of protecting intellectual property, then easy placement and extraction steps are of less importance than a robust and well-hidden encrypted message. The two techniques presented in this paper will be evaluated based on their capacity for these distinct purposes.

Before proceeding, it is also important to clarify some nomenclature. Cellular DNA, gene therapy sequences, or other DNA that ultimately ends up in a living organism, will be called "live DNA". DNA sequences that are not intended to

undergo processing by cellular machinery, but are just chemical messages such as DNA computing solutions, will be called "chemical DNA". This is important since live DNA processing is extremely complex with elaborate structure and subtle interactions that must be fully understood before any modifications are made. For the purposes of this paper, unless stated otherwise, DNA should be thought of as a theoretical construct.

### 4.1 Non-coding DNA, Non-transcribed DNA, or Non-translated RNA

Adding or hiding data in chemical DNA sequences does not require us to be very clever. For most tasks, a flat encoding of 2 bits/nucleotide, assigned in alphabetical order would be a sufficient starting point.

A=00
C=01
G=10
T=11

With this basic foundation, we can add binary segments to DNA that could be used to store interspersed hidden data, annotate existing DNA sequences, watermark a DNA computing solution, and so on. Depending on the task, different operations could be performed on the binary sequence while the DNA encoding scheme remains constant. For the purposes of data storage or transport, a compression step on the binary data would be useful. For a watermark, an encryption step would be appropriate. For annotation, plain text or simple codes could be used. After each important sequence, perhaps it would be useful to add a checksum to verify that the DNA strand hasn't degraded or been altered. In all cases, the application would drive the operations performed on the hidden data and in all cases, existing well-known binary techniques could be used for this purpose. The motivation for this is simple. The fewer complex and unprecedented steps are performed on data, the easier it is to process, embed, extract, and explain in court.

Extending this notion to live DNA requires great care. Unlike, chemical DNA where changes can be made wherever necessary to hide data, with live DNA, besides the obvious dangers of active genetic segments, there are other complications. Although, only a small percentage of DNA codes directly for genes, in addition to genes, there are regulatory and structural regions. Altering or adding sequences that seem innocuous may have profound effects when processed by cellular machinery. Nevertheless, with enough study, this will be feasible. Mutations in live DNA happen with some frequency, but most of the time they happen to non-coding, non-regulatory regions, and are not expressed. So, clearly, it must be possible to make changes without repercussions.

In live DNA, there are two possible areas for data placement: Non-transcribed DNA and non-translated RNA regions. Non-transcribed regions are all DNA that is not transcribed to RNA. Most of the live DNA is not transcribed, and represents a vast space to hide data. After DNA is transcribed, in eukaryotic cells, certain portions of the RNA transcript are edited out before translation. Moreover, not all of the mRNA is translated into amino acid sequences. All transcript RNA that isn't translated into protein makes up the non-translated regions.

For chemical DNA, the level of robustness that this flat encoding scheme offers depends entirely on what data is hidden. If this method is used to watermark a DNA

computing solution for something like a traveling salesman problem, the watermark could be placed by choosing an almost-optimal path instead of an optimal one, hiding the watermark inside the sequence of destinations. In such a case, the watermark is in the answer to a problem. The ability to remove it would mean knowing an answer that is at least as good. Such a watermark derives its robustness from the difficulty of solving the problem. Likewise, this scheme would fare well for annotations or cataloging since the data only needs to be resistant to degradation or truncation. In this case, redundancy and error correcting encoding would be used, but efforts to prevent attack are unwarranted.

Conversely, using a technique like this for live DNA watermarking, where the goal is to protect intellectual property, is not the strongest solution. This is particularly the case for non-transcribed DNA segments. Active genetic regions have characteristic sequences surrounding them, acting as a demarcation for an attacker. This makes the process of isolating the important sections from a small DNA segment relatively simple. With this information, an attacker would be able to isolate the key elements of the gene, and begin to search for the watermark, or even easier, they could simply cut the active segment out and splice it into a neutral sequence that otherwise matches the original. At this point, the watermark that may have surrounded the active segments is effectively removed. A watermark in non-translated, but transcribed, sections is somewhat stronger since it would lie between the start and end markers for the active portion of the DNA segment. However, a watermark in these regions can also be removed with a bit of backtracking. Since the final mRNA product has neither the non-transcribed nor the non-translated regions, it could be used to exactly identify the portions of DNA that are part of the gene. Removing or altering the rest of the DNA segment can eventually be accomplished. This method does offer substantial delay, but it can be defeated with some work.

### 4.2 Codon Redundancy

An ideal watermark stays with the data it is protecting without altering it functionally. In audio and video, one would prefer to place the watermark in the active signal rather than in header or trailer segments. This improves robustness since headers and trailers can be easily chopped out. The same is true for DNA. The strongest watermarking scheme for live DNA is one where the watermark is embedded in the actual sequence coding for the gene. However, how can one change DNA that is part of an active genetic sequence and still make the protein product come out the same? The answer is redundancy.

mRNA codons are 3 bases long. There are 4 possible bases, U, C, A, and G. That means that there are $4^3$ or 64 unique codes that can be generated. However, there are only 20 amino acids that are encoded. Therefore, there is redundancy in the mRNA to amino acid mapping. This redundancy can be leveraged to embed additional information into the sequence without altering its function or length. Any time we are offered a choice, the choice that is made can be used to convey information.

Observing the chart, it is clear that a flat binary encoding scheme will not be optimal, since there can be sets of 1,2,3,4 or 6 codons representing the same amino acid. An encoding scheme that does not waste space for each of these possible set sizes is required to store a more optimal amount of data. One such scheme is arithmetic encoding, since it works with an arbitrary amount of subdivisions.

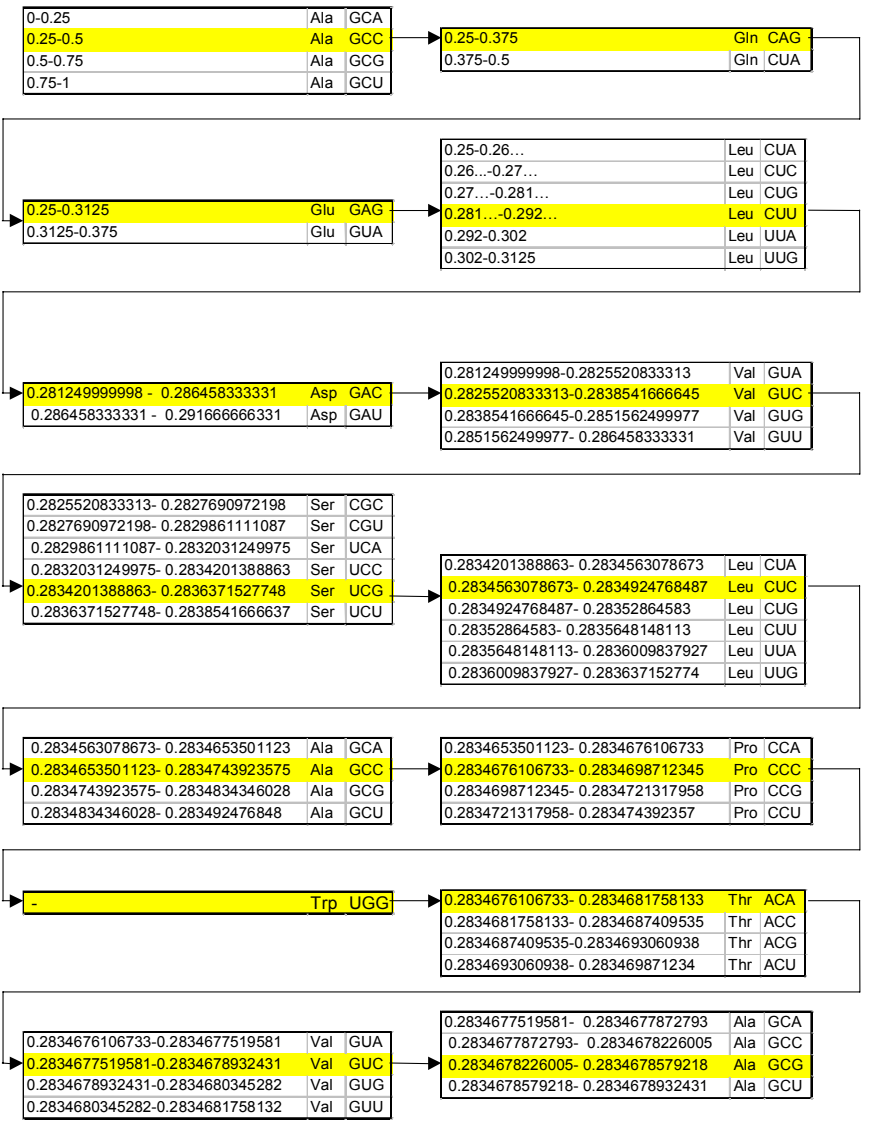
Arithmetic encoding would involve taking a binary sequence containing the encrypted message and converting it to a decimal number between 0 and 1 [10]. Then one would take the amino acid sequence that would be generated by the original mRNA, and generate a list of all possible codons that lead to the same sequence, listing the codons in alphabetical order. The number can then be generated by continually subdividing by the number of codon combinations for each amino acid in the sequence. Putting the codons in alphabetical order serves to standardize the process. This way, there is no ambiguity to the scheme, thus strengthening the credibility of the watermark.

To demonstrate how this works, let's take a sample message and embed it into a short sequence. The first step is choosing the binary sequence we wish to embed. For the purposes of the example, we choose the following binary sequence: "01001000100100010101100100110100110111 01". Now that we have the sequence we wish to store, we take the given mRNA sequence (chosen arbitrarily for the example) "AUG GCG CAG GUA UUA GAC GUA CGC CUA GCU CCA UGG ACC GUU GCC GGU CUG CCC GGG AUA CCU GAU" and translate it into the amino acid sequence it represents. Note that for this example we start in reading frame. "AUG" is the start codon so we do not count it. The amino acid sequence "Ala Gln Glu Leu Asp Val Ser Leu Ala Pro Trp Thr Val Ala Gly Leu Pro Gly Val Ser Ile Pro Asp" is generated from the translation. Our watermarked mRNA sequence must also translate exactly into this sequence. Now, we take advantage of the redundancy. There are a number of ways to generate that same amino acid sequence. We will use arithmetic encoding to choose the path we take. Converting the above binary sequence into a decimal number between one and zero yields ".283467841536".

**Table 1.** Codon to amino acid mapping

| | U | C | A | G | |
|---|---|---|---|---|---|
| U | UUU→Phe | UCU→Ser | UAU→Tyr | UGU→Cys | U |
| | UUC→Phe | UCC→Ser | UAC→Tyr | UGC→Cys | C |
| | UUA→Leu | UCA→Ser | UUA→STOP | UGA→STOP | A |
| | UUG→Leu | UCG→Ser | UAG→STOP | UGG→Trp | G |
| C | CUU→Leu | CCU→Pro | CAU→His | CGU→Arg | U |
| | CUC→Leu | CCC→Pro | CAC→His | CGC→Arg | C |
| | CUA→Leu | CCA→Pro | CUA→Gln | CGA→Arg | A |
| | CUG→Leu | CCG→Pro | CAG→Gln | CGG→Arg | G |
| A | AUU→Ile | ACU→Thr | AAU→Asn | CGU→Ser | U |
| | AUC→Ile | ACC→Thr | AAC→Asn | CGC→Ser | C |
| | AUA→Ile | ACA→Thr | AUA→Lys | CGA→Arg | A |
| | AUG→Met, Start | ACG→Thr | AAG→Lys | CGG→Arg | G |
| G | GUU→Val | GCU→Ala | GAU→Asp | GGU→Gly | U |
| | GUC→Val | GCC→Ala | GAC→Asp | GGC→Gly | C |
| | GUA→Val | GCA→Ala | GUA→Glu | GGA→Gly | A |
| | GUG→Val | GCG→Ala | GAG→Glu | GGG→Gly | G |

We use this as the target number in the repeated subdivision steps of arithmetic encoding, as shown in the figure below.



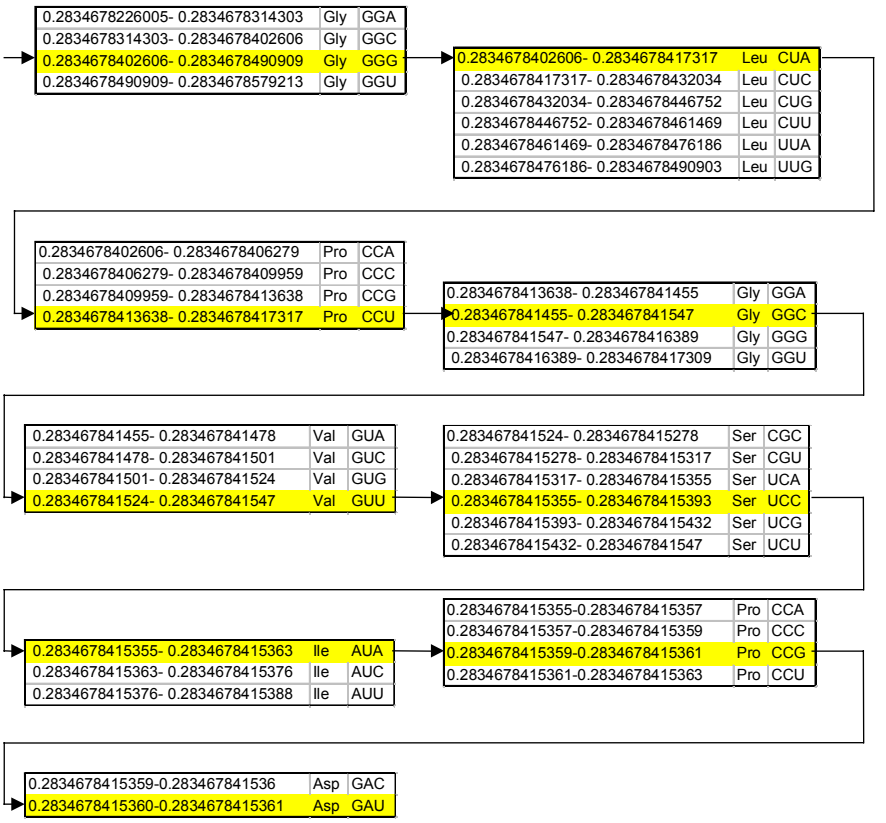



**Fig. 4.** Arithmetic coding example

This process has generated the new mRNA sequence. Following the highlighted path, we get "AUG GCC CAG GAG CUU GAC GUC UCG CUC GCC CCC UGG ACA GUC GCG GGG CUA CCU GGC GUU UCC AUA CCG GAU", which codes for the same amino acid sequence as the original mRNA, and now, also contains the intended watermark. The ability to embed 40 bits into a sequence this short should make it clear that there is adequate space to encrypt and embed a strong and repetitive watermark into live sequences, which consist of many thousands of base pairs.

The example given is purely theoretical. In live DNA, there are various species-specific restrictions to which codons can be used. As a result, the chosen scheme must degrade optimally as well, always working with the full potential of the redundancy available. This arithmetic encoding scheme degrades nicely, continuing to embed data without waste. As such, even if the amount of available redundancy is significantly reduced by real-life constraints, the assumption is that some will remain, and a long sequence can still retain a significant message.

Strengthening this technique to make it a good watermarking solution requires some additional work. Revisiting the notion of using the power and credibility of

existing digital techniques gives us a good starting point, especially since we are encrypting a binary sequence anyway. Also, we need to overcome the fact that an attacker can apply the same process to our watermarked sequence and replace the watermark with their own.

First, we try to generate a highly credible, difficult to detect type of binary sequence. Plain text can be discovered very quickly since the algorithm for placing it there is publicly known. A good solution for dealing with this is using private/public key cryptography since it is well known and can act simultaneously as an authentication scheme and an encryption scheme [11]. To generate the watermark, first the placing party would encrypt with their private key. This ensures that they were the only party that could have placed it there. Then they encrypt again, with their public key. This ensures that they are the only party that can read it. Now the message appears random to anyone that doesn't know their private key, and can be proven with high probability that they placed it there, since the encryption required a combination of information that only they could know and information that is known publicly to identify them.

Now the trick is to prevent someone else from doing the same thing. If we take as given that only the placing party knows the original sequence prior to the watermark, it could be used to settle a dispute. In the example above we arranged the nucleotide subdivisions alphabetically and started subdividing from the top. Instead, we could start with the codon that is in the original sequence and subdivide from there, wrapping around as necessary. For example, instead of starting from the first codon chosen alphabetically as presented earlier (left side), we can start numbering from the original codon before the watermark (right side). In the original sequence, GCG was the first codon, so that is where the zero point is. When we reach the bottom, we wrap around to GCA and GCC.

This would result in a completely different sequence, but the information conveyed would be the same. This step keeps the process unambiguous, and at the same time ensures that the original mRNA sequence is required to place and decode the watermark. Therefore, unlike the first example, an attacker would not be able to simply repeat the watermarking process.

Using this technique in a retrovirus vector, means that the message will transfer through the reverse transcription pathway and be embedded in the target cells' DNA in complementary form.

In the other direction, with some effort, one can generate an appropriate DNA sequence that, when transcribed, would create the mRNA strand with the embedded message.

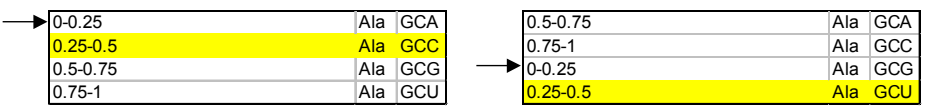



**Fig. 5.** Arithmetic coding, more robust variation

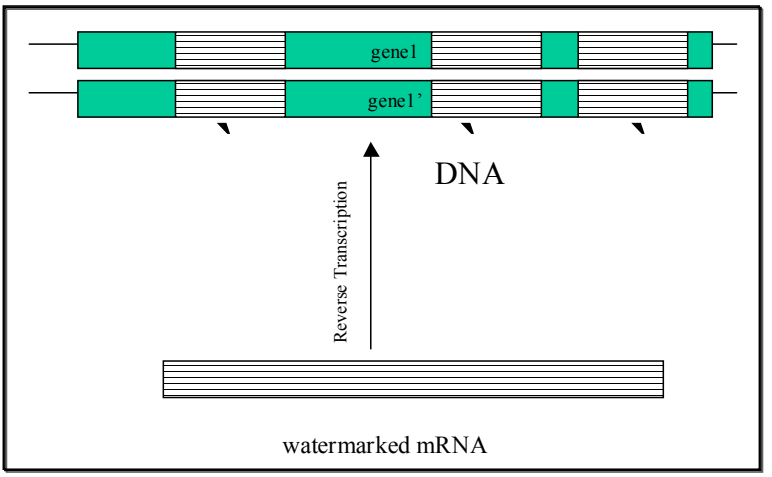



**Fig. 6.** Reverse transcription of watermark

## 5 Conclusion

DNA as a storage medium is extremely effective. It is compact, biodegradable, and consumes very little energy. Today it is used to propagate species, encode protein synthesis, and solve complex computational problems. Who knows what it will do in the future? Recognizing this, techniques for hiding data to catalog, annotate, watermark, and/or encrypt information in this medium can have tremendous purpose.

This paper proposes the original idea of hiding data in DNA and RNA. Moreover, it defines two new and original techniques for hiding the data, along with an evaluation and analysis of the utility of each for the different functions of hidden data. The first technique hides data in non-coding DNA such as non-transcribed and non-translated regions as well as non-genetic DNA such as DNA computing solutions. The second, in theory, can actually be used to embed information directly into active genetic segments. In addition to a straightforward set of steps for embedding the data, this paper also addresses the codon redundancy technique's susceptibility to attack and offers several additional steps to strengthen a watermark. The practical usefulness of such a technique can be enormous.

The techniques presented in this paper can be applied to protecting intellectual property in the realms of gene therapy, transgenic crops, tissue cloning, and DNA computing. Data can also be hidden to verify the integrity of DNA sequences, annotate sequences for easier cataloging and study, or simply to convey data inconspicuously in biological carriers.

## References

1. Adelman L., “Molecular computation of solutions to combinatorial problems”, Science 266, Nov. 11, 1994
2. J. Lach, W. Mangione-Smith, and M. Potkonjak Fingerprinting Digital Circuits on Programmable Hardware. Information Hiding Workshop, Portland, Oregon, 1998
3. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I.L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe Watermarking Techniques for Intellectual Property Protection. DAC-98 35th ACM/IEEE DAC Design Automation Conference, pp. 776-781, San Francisco, CA, June 1998
4. Campbell, N.A., Biology pp 322-335, Benjamin/Cummings Publishing, 1990
5. Felsenfeld, G. “DNA”, Scientific American, Oct 1985
6. Darnell, J., “RNA”, Scientific American, Oct 1985
7. Campbell, N.A., Biology pp 360, Benjamin/Cummings Publishing, 1990
8. Gallo, R. C., “The First Human Retrovirus”, Scientific American, December 1986
9. Bender, W., Gruhl, D., Morimoto, N., Lu, A., “Techniques for Hiding Data”, IBM Systems Journal, Vol. 35, NOS 3 & 4, 1996
10. Howard P., Vitter J., “Arithmetic Coding for Data Compression” " Proceedings of the IEEE, 82(6), June 1994, 857-865
11. Diffie, W. and Hellman, M. E. “Privacy and Authentication: An Introduction to Cryptography”, Proceedings of the IEEE, Vol. 67, No. 3, March 1979