Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 603–609
Progress in Natural Science
www.elsevier.com/locate/pnsc

Special report

DNA media storage

Christy M. Bogard, Eric C. Rouchka *, Benjamin Arazi

Computer Engineering and Computer Science Department, University of Louisville, Louisville, KY 40292, USA

Received 17 December 2007; received in revised form 21 December 2007; accepted 21 December 2007

Abstract

In 1994, University of Southern California computer scientist, Dr. Leonard Adleman solved the Hamiltonian path problem using DNA as a computational mechanism. He proved the principle that DNA computing could be used to solve computationally complex problems. Because of the limitations in discovery time, resource requirements, and sequence mismatches, DNA computing has not yet become a commonly accepted practice. However, advancements are continually being discovered that are evolving the field of DNA computing. Practical applications of DNA are not restricted to computation alone. This research presents a novel approach in which DNA could be used as a means of storing files. Through the use of multiple sequence alignment combined with intelligent heuristics, the most probabilistic file contents can be determined with minimal errors.

Keywords: DNA computing; Error reduction; Multiple sequence alignment

1. DNA representation of digital information

How one approaches a problem is often defined in how the problem is represented. Various representations lend themselves to a set of predefined actions that can easily shape one's perspective and approach in the quest for the solution. For example, when presented with the problem of determining the time at which a thrown ball is at a given height, represented by the following equation [1]

$$h = vt\sin(A) - (1/2)Gt^2 \quad (1)$$

where h is the height at time t, v is velocity, t the time in air, A the angle at which thrown, G the universal gravitational constant, it is possible to algebraically solve for the corresponding time values. However, it is easier to graph the associated equation of height as a function of time as shown in Fig. 1, referencing the graph for the corresponding solution. Additionally, from the graphical representation, it is apparent that there are two corresponding times at which the thrown ball is at a given height – one time in which the ball is traveling upward and an additional time in which the ball is traveling downward.

Computer scientists have long used the notion of a binary bit to represent a digital information, wherein 1 indicates that the given element is present and 0 indicates that the given element is not present [2]. Combining a series of binary bits enables more states to be represented for a given element; a two-bit binary sequence can represent four possible states – 00, 01, 10, 11 – where each element represents an associated state in the problem. In this same manner, geneticists represent the four possible DNA states with a quaternary alphabet, using the symbols A, C, G, and T to encode for each of the four states. Understanding the relationship among various representations, such as between the digital binary bit of computer scientist and the DNA quaternary character of the geneticists, enables one to easily translate between different representations to approach the same problem from a new perspective. For example, translating between the computer scientist's alphabet and

* Corresponding author. Tel.: +1 502 852 1695; fax: +1 502 852 4713.
E-mail addresses: Christy.Bogard@louisville.edu (C.M. Bogard), Eric.Rouchka@louisville.edu (E.C. Rouchka), Benjamin.Arazi@gmail.com (B. Arazi).

1002-0071/$ - see front matter © 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.
doi:10.1016/j.pnsc.2007.12.009

the geneticist's representation is easily accomplished through a direct substitution of two binary base pairs encoding for a single quaternary character, as shown in Fig. 2.

2. Adleman and the Hamiltonian path problem

A Hamiltonian path is defined as a route through an undirected graph which visits each vertex in the graph exactly once [3]. The Hamiltonian path problem (HPP) aims to find the lowest cost Hamiltonian path within the graph. One specific variant of the HPP is the Traveling Salesman Problem (TSP), where the vertices in the graph represent different cities, and the edges represent the cost to travel between a set of cities. For example, given the graph in Fig. 3 [4] where all edges are bidirectional and have an associated cost of one unit, a Hamiltonian path starting from city 0 would be $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ with a total cost of seven units.

In 1994, University of Southern California computer scientist, Dr. Leonard Adleman solved the Hamiltonian path problem using DNA as a computational mechanism [5,6]. Adleman began by using 20-mer oligonucleotide sequences to uniquely represent each city. Paths were represented using complementary 20-mer oligonucleotide sequences generated by combining the last 10 bases of the starting city with the first 10 bases of the ending city. When the oligonucleotide sequences were combined, DNA's desire to form a stable double helix structure enabled the paths to be constructed through the combination of the city

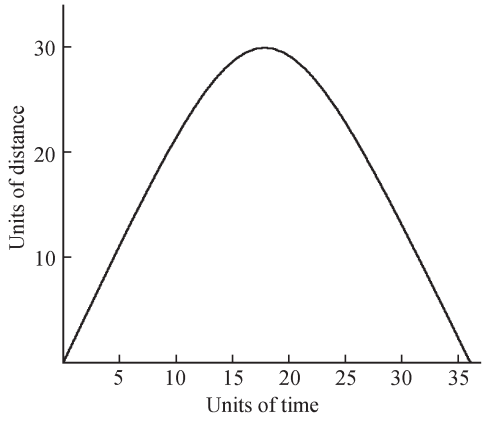

Fig. 1. Trajectory of a ball thrown as a function of time. Changing the representation of a problem changes the perspective in which it is approached. In solving for the time at which a thrown ball is at a given height, it is apparent from the graph that there will be two corresponding times, while this critical detail is hidden within the equation representation.

Digital → DNA			
00 → A	01 → C	10 → G	11 → T

Fig. 2. Conversion between digital bit-based and DNA-based alphabet. Knowing that DNA uses a base – four alphabets, it is possible to convert a two-bit digital sequence to the equivalent DNA base and vice versa.

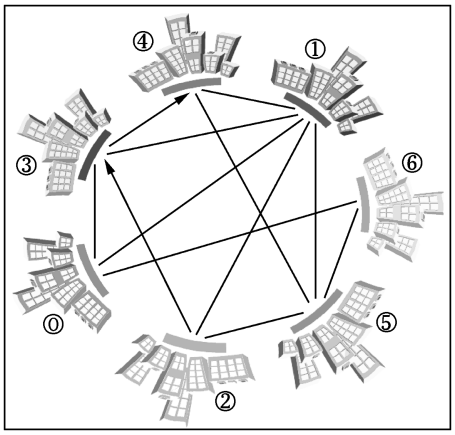

Fig. 3. Traveling Salesman Problem (TSP). TSP, a variant of the Hamiltonian path problem, aims to find the lowest cost Hamiltonian path within the graph, where the vertices in the graph represent different cities, and the edges represent the cost to travel between a set of cities. Image from Parker [4].

sequences with the complementary edge sequences. For example, as shown in Fig. 4, the first three sequences represent 20-mer oligonucleotide representations of three of the cities – cities 2, 3, and 4. Since a path exists from city 2 to city 3, the last 10 bases from city 2 are combined with the first 10 bases of the city and the complementary sequence of this new 20-mer sequence will enable the two cities to be combined. Since this is not a directed graph, meaning a path is bidirectional, it is also important to generate the converse path as well. In other words, the process must be repeated using the last 10 bases from city 3 with the first 10 bases of city 2, representing the directed path from city 3 to city 2.

Once all representations of the cities and corresponding paths were in place, a large number of copies were generated to produce all possible combinations of cities and edges, in effect generating all possible paths through the graph. Paths that did not meet all of the problem rules were systematically eliminated. A valid Hamiltonian path through the cities must have exactly seven vertices present; all generated paths that were not of this length, whether too short or too long, were eliminated. Since the path must visit each city exactly once, sequences with duplicated cities were also eliminated. Any remaining generated paths are valid Hamiltonian paths through the graph. If no generated paths remain, then the graph does not contain any Hamiltonian paths.

Adleman's solution to the Hamiltonian path problem proved that DNA could in principle be used to solve NP-complete problems. One of the primary benefits of DNA computing is its ability to make computations in parallel. This benefit comes at the cost of a lengthy discovery of the DNA solution. For Adleman's solution to the Hamiltonian path problem, all possible solutions were enumerated in only a few hours. However, it took approximately 7 days to eliminate all of the invalid paths. While Adleman's methodology was slow and inefficient when compared with

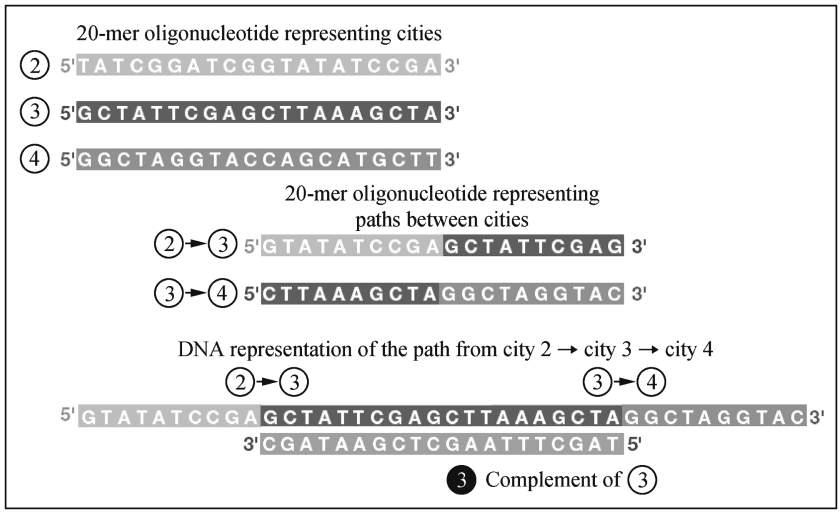

Fig. 4. DNA representation of the Traveling Salesman Problem. Strands of 20-mer oligonucleotide sequences are used to uniquely represent each of the seven cities in the graph. To represent that a path exists between two cities, the complementary 20-mer oligonucleotide sequence was generated. When strands were combined within a mixture, DNA's desire to form double helix structures enables the corresponding Hamiltonian Paths to be created. Image from Parker [4].

today's methodology, it is still a lengthy process to biologically find the DNA solutions among a given mixture.

DNA has the ability to store a vast amount of information. Current methods of data storage require approximately 10^{12} nm^3 of space to store a single bit, while DNA has the ability to store a single bit in only 1 nm^3 [4]. However, DNA representation of problems can be difficult. Adleman represented each city and edge with a 20-mer oligonucleotide sequence to ensure that there would be no errors in his calculations of the Hamiltonian paths. If one were to scale the Hamiltonian path problem from the original seven cities to 200 cities, the DNA required to represent all of the cities and the corresponding edges would be greater than the weight of earth.

Finally, since Adleman's experiment was limited to only seven cities, he could represent the cities with distinctly different sequences as to minimize the number of alignments that would result in solutions that do not exist. However, as the number of cities increase, it becomes more difficult to uniquely represent the cities in such a manner as to avoid mismatched alignments. Therefore, additional error-checking will be required to ensure accurate solutions.

3. Using multiple sequence alignment in error reduction

DNA allows for a drastic reduction in storage space per bit compared with traditional digital computing. As a result, redundant storage capabilities and parallel processing on the exact same data are feasible. However, if the storage or computation results in inconsistencies, determining which are correct and which are not is problematic. The bioinformatics technique of multiple sequence alignment yields insight into how the issue of data integrity can be solved.

3.1. Multiple sequence alignment

Multiple sequence alignment is the process of finding a representative, or consensus, model of the similarities between three or more sequences. Like pairwise sequence alignment, it finds an optimal solution for the model conditions placed upon it. If the conditions are changed, then the model may or may not hold. For a set of highly conserved sequences, the multiple sequence alignment is easily seen, even with the naked eye. As the sequences diverge, so does the complexity of finding the best alignment [7].

Multiple sequence alignment begins by finding the optimal pairwise sequence alignment between each pair of sequences. Once found, there are a number of approaches used to discover the underlying model. The top three approaches are progressive [7], iterative [8], and statistical or probabilistic modeling [9]. Progressive modeling begins with the alignment of the two most similar sequences and iteratively adds sequences to the alignment in descending order of their similarity. Iterative modeling aligns any pair of similar sequences or set of sequences, continually clustering until only one group remains.

Finally, statistical or probabilistic modeling selects the ordering of alignment based on a given statistically or probabilistic model believed to represent the given set of sequences. Once a multiple sequence alignment is in place, it can be described using a number of different approaches. The most useful of these represents the alignment as a statistical model, known as a profile Hidden Markov Model (HMM) [10]. HMMs have the power to represent the alignment through states for insertions, deletions, and matches/mismatches found within the alignment. For the match/mismatch and insertion states, an associated emission probability is given to the observed characters for a particular position.

3.2. Multiple sequence alignment for error reduction

Since multiple sequence alignment is sensitive to sequence similarities, it can be used to combine the multiple copies of the same file to find the most probabilistic contents. There are three scenarios that can be discovered: (1) areas completely conserved among all of the sequences, (2) areas highly conserved among the sequences, and (3) areas not conserved among the sequences. Each of these scenarios directly corresponds with the level of error within the region.

First, consider areas that are completely conserved among all of the sequences. In this case, no mutations have occurred in any of the file copies. Since the region is an exact clone of all other copies, there are no discrepancies introduced and as such, the region is completely 100% free of errors. For highly conserved areas, discrepancies indicate potential areas that have been introduced. Since a multitude of copies have been stored, then it is probable that the majority of sequences will be highly correlated. Thus, the emission properties of the associated Hidden Markov Model state will clearly indicate which one of the bases is most probable of being emitted as it will have a significantly higher emission over the remaining bases. It is important to note that pseudocounts should not be introduced within the Hidden Markov Model, as they will skew the emissions of the state.

Finally, consider areas that are not conserved among the sequences. It may not be possible to determine the most probabilistic emission because a significant number of discrepancies have been introduced into the region. Since there can be no determination as to what the sequence was originally, this region represents the system state of irrecoverable errors. In such circumstances, there are a number of external alternatives to be considered. An artificial intelligent agent could be introduced to make the final determination of the state. Conversely, all of the represented sequences could be presented to the end user to make the final determination as to what were the original contents of the file.

3.3. Improving the multiple sequence alignment

The genetic code allows for a three-base nucleotide sequence (codon) to encode for one of 20 amino acids within an organism. Since there are four possible bases (A, C, G, and T) for each of the three possible bases of the amino acid, there are a total of 64 possible combinations [7], meaning there are multiple codon representations that encode for a single amino acid. For example, from the translation table given in Fig. 5, Lysine (K) is encoded by AAA and AAG. Thus, a discrepancy between A and G in the third base would not have a difference in the resulting amino acid. Likewise, Threonine (T) is encoded by ACT, ACC, ACA, and ACG, meaning the third base is obsolete in the determination of the amino acid because all the four bases translate into Threonine. Consequently, alignment of the translated amino acid sequences has a greater probability of defining more highly conserved regions that may be indeterminate at a DNA sequence level. Alignment of regions of low conservation can potentially be improved by aligning the corresponding translated amino acid sequences.

While increased accuracy is possible, it comes at a cost of a dramatic increase in the computational time required to find the alignment. To translate a DNA sequence into

Amino Acid	Symbol	DNA codons					
Alanine	A	GCA	GCC	GCG	GCT		
Cystenine	C	TGC	TGT				
Aspartic Acid	D	GAC	GAT				
Glutamic Acid	E	GAA	GAG				
Phenylalanie	F	TTC	TTT				
Glycine	G	GGA	GGC	GGG	GGT		
Histidine	H	CAC	CAT				
Isoleucine	I	ATA	ATC	ATT			
Lysine	K	AAA	AAG				
Leucine	L	CTA	CTC	CTG	CTT	TTA	TTG
Methionine (START)	M	ATG					
Asparagine	N	AAC	AAT				
Proline	P	CCA	CCC	CCG	CCT		
Glutamine	Q	CAA	CAG				
Arginine	R	AGA	AGG	CGA	CGC	CGG	CGT
Serine	S	AGC	AGT	TCA	TCC	TCG	TCT
Threonine	T	ACA	ACC	ACG	ACT		
Valine	V	GTA	GTC	GTG	GTT		
Tryptophan	W	TGG					
Tyrosine	Y	TAC	TAT				
STOP	*	TAA	TAG	TGA			

Fig. 5. Translation table to convert from a three-base codon to an amino acid. A single amino acid can be encoded by multiple DNA codons. All 64 possible triple base codons are presented with their associated amino acids. Aligning the translated amino acid sequence has a greater probability of defining more highly conserved regions when compared to aligning the DNA sequence.

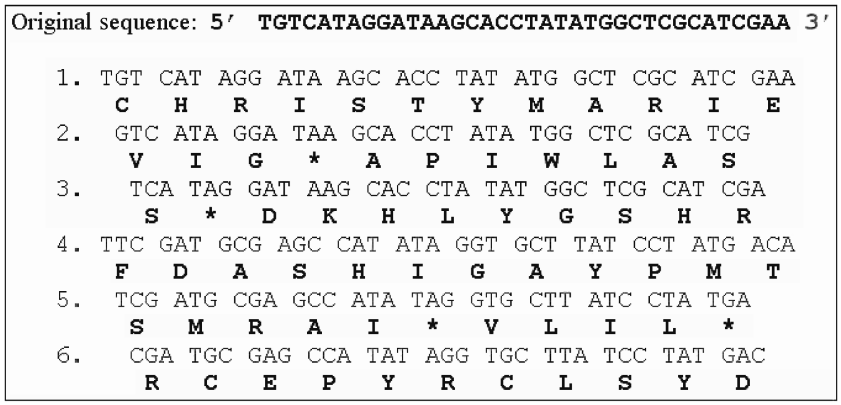

Fig. 6. Translation of a single DNA sequence. Translation results in six distinct amino acid sequences arising from each of the three reading frames of the original sequence in the 5′ direction and each of the three reading frames of the reverse complement.

its corresponding amino acid sequence results in six possible translations. This is the result of an unknown open-reading frame, or lack of knowledge as to which base is the correct starting location of the translation and not a carryover of the previous amino acid. As such, each of the first three bases of the DNA sequence must be considered as a possible starting codon location. Additionally, since the DNA forms a double helix, one must also consider the first three bases of the reverse complement sequence as possible codons since there is no decisive method of determining in which direction the sequence was originally read. An example translation from a single DNA sequence to its corresponding six distinct amino acid sequences is shown in Fig. 6.

Since the pairwise alignment between two nucleotide sequences is being sought, both sequences must be translated into their corresponding six amino acid sequences. All 36 combinations must be considered, aligning each of the six amino acid sequences from the first translated DNA sequence with each of the six amino acid sequences from the second translated DNA sequence. The pairwise alignment with the highest score is then deemed to be the best representation of the two sequences. The number of pairwise alignments between the sequences to be considered has increased the time and space complexity by a factor of 36.

3.4. Heuristic improvements of the algorithm

Knowing that the aligned sequences are very similar, if not identical, there are number of heuristics that can be applied to reduce the computational, storage, and time complexity required for the multiple sequence alignment. Continuing with the discussion of the storage of a file, it is reasonable to assume that the majority of sequences being aligned will be of the same length within a given threshold. Since a file will not produce or reduce the amount of information contained within it without some sort of external stimuli, one can quickly eliminate sequences which disproportionately longer or shorter than majority of sequences being aligned.

Additionally, since the sequences are highly similar, the alignment will probabilistically follow the diagonal of the dynamic programming alignment matrix [11,12]. Thus, performing a bounded alignment in which only cells within a given threshold above and below the diagonal of the dynamic programming alignment matrix are calculated will reduce the computational complexity and the storage complexity required for all of the pairwise sequence alignments performed. Determining the appropriate threshold is dependent upon the application; however, for any sequence set of substantial length, it is reasonable to assume that the threshold could be set between 5% and 10% and still produce highly accurate results.

To reduce these complexities even further, an intelligent agent could retain the probabilities of the identical alignments without requiring the actual storage of the alignments. Specifically, if two or more of the sequences are identical, it is inefficient to store the alignment, as the highest pairwise alignment is an exact copy of itself. However, the frequencies of the identical sequences must be retained in order for the Hidden Markov Model emissions to be representative of the aligned sequences. If these frequencies are not retained, then discrepancies in the alignment will be emphasized as the frequency of the dominate character is decreased.

4. Discussion

Duplicate copies of a file must be stored for accurate information retrieval. Fig. 7 shows eight generated strings to represent eight encoding sequences of a file. Changes in the sequences are introduced within the sequences to represent mutations that could occur within a biological environment.

Alignment of the nucleotide sequences in Fig. 8 reveals completely conserved, highly conserved, and indeterminate states within these eight sequences. Completely conserved states are indicated with bold, uppercase text; highly conserved and conserved states are indicated with lowercase text; indeterminate states are indicated with a solid circle. Using eight nucleotide sequences, a relatively small num-

```
TGC GCG CGT GAT ATT AAC TAG CTC TCT
TGC GCC AGG GAT ATT AAC TAG TTG TCA
TCT GCG CGG GAT ATT AAC TGA CTC TCT
TGC GCG CGG GAC ATT AAC TGA CTC TCG
TCG GCA CGT GAT ATT AAC TAG CTT TCT
TCG GCT CGG GAT ATT AAT TAA TTG TCT
TGC GCA AGG GAT ATT AAT TAG CTT TCT
TGT GCG AGA GAT ATT ACC TGA CTC TCA
```

Fig. 7. DNA sequences representing stored information. Generated strings are created to represent possible information stored in genetic sequences. Changes are introduced to mimic mutations occurring in a biological environment.

```
Alignment: Tgc GCg cGg GAt ATT Aac Ta● cTc TCt
```

Fig. 8. Alignment of the eight nucleotide sequences. Alignment of the eight nucleotide sequences reveals only fourteen of the twenty-seven bases are completely conserved, or just over half. Twelve bases are determined based on the highest emitted nucleotide. One base is indeterminate based on the emission probabilities.

```
1. TGC GCG CGT GAT ATT AAC TAG CTC TCT
    C   A   R   D   I   N   *   L   S
2.  GCG CGC GTG ATA TTA ACT AGC TCT
     A   R   V   I   L   T   S   S
3.   CGC GCG TGA TAT TAA CTA GCT CTC
      R   A   *   Y   *   L   A   L
4. AGA GAG CTA GTT AAT ATC ACG CGC GCA
    R   E   L   V   N   I   T   R   A
5.  GAG AGC TAG TTA ATA TCA CGC GCG
     E   S   *   L   I   S   R   A
6.   AGA GCT AGT TAA TAT CAC GCG CGC
      R   A   S   *   Y   H   A   R
```

Fig. 9. Translation of polynucleotide chain into corresponding amino acid chain. Translation results in six distinct amino acid sequences arising from each of the three reading frames in the 5′ direction and each of the three reading frames in the 3′ direction of the original sequence.

```
CAR DIN *LS
CAR DIN *LS
CAR DIN *LS
CAR DIN *LS
CPR DIN *LS
CAR DIN *LS
CAS DIN *LS
CAR DIN ALS
Alignment: Car DIN *LS
```

Fig. 10. Alignment of the converted amino acid sequences from Fig 7. Converting the eight nucleotide sequences to the corresponding amino acid sequences before alignment results in an increased confidence in the multiple sequence alignment.

ber, results in only 14 of the 27 bases being completely conserved, or 51.9%. While only one state is indeterminate, 12 states are determined based on the highest emission probability across the eight sequences, with the lowest confidence of 50%, the highest confidence of 87.5%, and an average confidence of 65.6%.

Using the amino acid translation table described above, the 27 base polynucleotide chains can be converted into the corresponding amino acid sequences, as shown in Fig. 9. Recognizing there are six reading frames per sequence means there are 36 comparisons for each pairwise alignment. This number goes exponentially when completing multiple sequence alignment; for eight sequences, 40,320 comparisons are required to complete the alignment.

Using the highest scores found in a progressive alignment methodology, the multiple sequence alignment results in significant reduction of discrepancies and eliminates the indeterminate state. The multiple sequence alignment of the amino acid chains, as shown in Fig. 10, results in six of the nine bases being completely conserved, or approximately 66.7%. Conserved regions, determined by the emission probabilities of the bases for the state, have a higher level of confidence; determination of the state has increased from a simple emission majority meeting a confidence of 50% to significant emission probability with a confidence of 87.5% in all three conserved regions.

5. Conclusions

This research presents a novel approach in which DNA could theoretically be used as a means of storing files. Through the use of multiple sequence alignment combined with intelligent heuristics, the most probabilistic file contents can be determined with the minimal errors. Completely conserved regions have no discrepancies and as such are 100% error-free. Highly conserved regions have minimal discrepancies, whose correct content can be determined based on the emission probabilities of the associated Hidden Markov Model. Finally, poorly conserved regions represent the most difficult areas because of the high discrepancies with low-emission probabilities. However, using the associated translated amino acid sequences, it is possible to improve the accuracy of the region's emission probabilities with multiple codons encoding a single amino acid.

Adleman hypothesized that "for the long-term, one can only speculate about the prospects for molecular computation" [5]. With each new theory introduced, we move closer to the practical applications afforded by DNA computing. It is unrealistic to predict DNA computing will form the sole basis of the next generation of technology; however, when combined with current technologies, could form a hybridization capable of achieving the fast computational benefits of DNA with the flexibility of current silicon.

Acknowledgements

This research is supported by the NIH-NCRR Grant P20RR16481 (Nigel G. F. Cooper, PI) and NIH–NIEHS Grant P30ES014443-01A1 (Kenneth S. Ramos, PI). Its contents are solely the responsibility of the authors and

do not represent the official views of NCRR, NIEHS, or NIH. The authors also acknowledge the support provided by Hank and Becky Conn.

References

[1] Halliday D, Resnick R, Walker J. Fundamentals of physics. 6th ed. New York: John Wiley and Sons; 2001.
[2] Brookshear JG. Computer science: an overview. 6th ed. Reading: Addison-Wesley; 2000.
[3] Baase S, van Gelder A. Computer algorithms: introduction to combinatorial problems. 3rd ed. Reading: Addison-Wesley; 2000.
[4] Parker J. Computing with DNA. Eur Mol Biol Org Rep 2003;4(1):7–10.
[5] Adleman LM. Molecular computation of solutions to combinatorial problems. Science 1994;266(5187):1021–4.
[6] Amos M. Theoretical and experimental DNA computation. Netherlands: Springer; 2005.
[7] Pevsner J. Bioinformatics and functional genomics. Hoboken: John Wiley and Sons; 2003.
[8] Mount DW. Bioinformatics: sequence and genome analysis. 2nd ed. Cold Spring Harbor: Cold Spring Harbor Laboratory Press; 2004.
[9] Durbin R. Biological sequence analysis: probabilistic models of proteins and nucleic acids. 11th ed. Cambridge: Cambridge University Press; 2006.
[10] Rabiner LR, Juang BH. An introduction to hidden Markov models. IEEE ASSP Mag 1986;3(2):4–16.
[11] Carillo H, Lipman D. The multiple sequence alignment problem in biology. Soc. Ind. Appl. Math. J Appl Math 1988;48(5):1073–82.
[12] Myers EW. An overview of sequence comparison algorithms in molecular biology. University of Arizona, Department of Computer Science, Technical Report TR 91-29; 1991.