iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** V **Month of publication:** May 2020

DOI: http://doi.org/10.22214/ijraset.2020.5181

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

DNA Fragment Assembly using Deep Recurrent Neural Network

Sk. Syeed. Ahmad[1], P. V. R. Anantha. Krishna[2], SK. Johnvali[3], V. V. N. Sandeep[4]
[1, 2, 3, 4]*B.Tech IVth Year, Department of CSE, KHIT, Guntur, AP, India.*

Abstract: ***The technique of DNA fragment assembly tries to reconstruct the original DNA sequence from a large number of DNA fragments where each fragment consists of several hundred base-pairs long. DNA fragment assembly technique is much needed because the current technology like gel electrophoresis, cannot produce direct and accurate sequence of DNA molecules longer than 1000 bases.. However, most genomes are much longer. Originally, the short fragments of DNA molecule was assembled by hand, which results in error-prone as well as inefficient. The shotgun sequence technique first cuts the original DNA molecule into several pieces.. But the problem with shot gun sequence technique is that the division of DNA molecule into pieces is achieved by cutting the molecule at random places. These short pieces are then processed with gel electrophoresis technique to find the DNA sequence of that piece. Once all the pieces are identified with the sequence, they need to be reassembled to get the original DNA sequence of the molecule. This problem is called DNA fragment assembly problem. Most of the assembly algorithms tries to examine all the pairs thereby creating a set of candidate overlaps followed by forming an approximate layout of DNA fragments, and finally creating a consensus sequence. All the existing methods of DNA fragment assembly rely on heuristics because the fragment assembly problem is NP-hard. The algorithms developed to solve this problem till now are based on genetic algorithms, greedy algorithms and nature inspired algorithms such as ant colony optimization, artificial bee optimization technique, cuckoo search algorithm.***
The proposed project aims at developing a deep recurrent neural network which takes the DNA fragments as input and produces the DNA Sequence.
Keywords: fragment, sequence, assembly, neural network

I. INTRODUCTION

A. Structure of DNA

DNA (Deoxy Ribo Nucleic Acid) is a polymer. The structure of a polymer molecule consists of repeated subunits.Those repeated subunits in DNA are called Nucleotides Each nucleotide is composed of 3 parts: a 5-carbon sugar, a phosphate group and a nitrogen base. Nucleotides join together to form two long chains, one on each side of the molecule, with the phosphate group and sugar molecules alternating to form the sides or backbone of the DNA molecule. The sugar molecule of each nucleotide bonds with the nitrogen base of the same nucleotide. And, the nitrogen base of one nucleotide bonds with a nitrogen base of another nucleotide on the opposite side of the molecule, which forms the rungs of the ladder. The result is a structure that looks like a twisted ladder as shown in Figure 2.

Nucleotide – each nucleotide has three parts(see Fig 1.): a 5-carbon sugar, a phosphate group and a nitrogen base. The structure of DNA molecule consists of four possible nucleotides .There are:

1) Four different nitrogen bases.
2) 5-carbon sugar (Pentose) – Deoxyribose (sugar's name)
3) Phosphate group – is composed of one atom of phosphorous which is surrounded by the four oxygen atoms.
4) Nitrogen base –

a) Adenine(A)
b) Thymine (T)
c) Cytosine (C)
d) Guanine (G)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

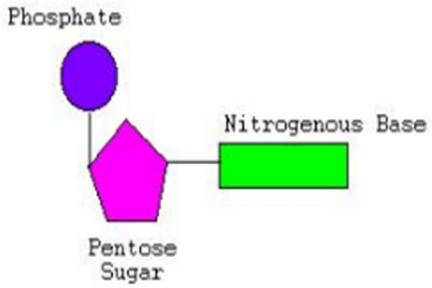


Fig1. A Nucleotide

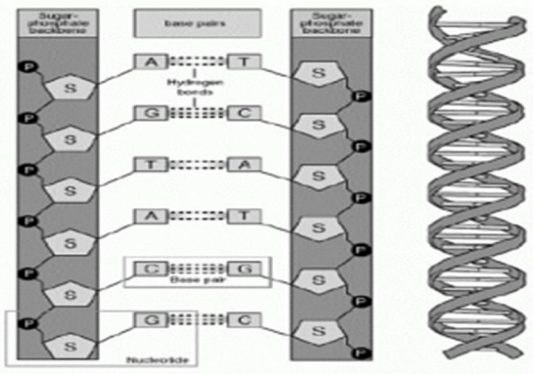


Fig2. A DNA Molecule

The structure of DNA molecule is encoded with the hereditary information in the chemical language. All cells in the structure of a DNA molecule stores the genetic information in the base sequence of DNA . The genotype which is determined by the sequence of bases of the DNA .Generally,the DNA molecule is the genetic material of living organisms and it is located within the nucleus of all cells, which are apart from the red blood cells.The DNA structutre consists of a long chemical sequence which holds the information needed by the living organism to develop, survive and to pass its genetic information on to the next generation. The DNA chemical sequence differs between individuals. The pattern of this sequence which is made up of A,T,C,G is called the genotype.

B. DNA Sequencing

1) The process of DNA sequencing determines the precise order of nucleotides within the DNA molecule. It invloves any method or kind of technology, helps in determining the order of the four bases like adenine, guanine, cytosine, and thymine in the strand of the DNA molecule. The rapid development of the DNA sequencing methods made a great impact leading to accelerate the biological and medical research and their discoveries.

2) The Knowledge of DNA sequences has become crucial for the basic level of biological researchs, and it is applied in various fields like diagnostic, biotechnology, forensic biology, and biological systematics. The rapid increasing speed of sequencing methods succeed with the latest technologies of DNA sequencing has been instrumental in the sequencing of the complete DNA sequences, or genomes of the numerous types and species of living organisms, including the human genome and other species like animal, plant, and microbial species of the complete DNA sequences.

C. Uses of DNA Sequencing

DNA sequencing technique may be used to determine the sequence of individual genes, larger genetic regions (i.e.group of gene), full chromosomes or entire genomes.. Sequencing provides the order of individual nucleotides in DNA or RNA (commonly represented as A, C, G, T, and U) isolated from cells of animals, plants, bacteria or virtually any other source of genetic information. This is useful for:

1) *Molecular Biology:* Deals with the study of the genome itself, how the proteins are made, what kind of proteins are made and to identify new genes which are associated with several diseases and phenotypes, to identify the potential drug targets

2) *Evolutionary Biology:* Deals with the study of how different organisms are related to eachother and how they are evolved

3) *Metagenomics:* Deals with study of identifying the species which are present in the body of water, sewage, dirt, debris filtred from the air, or swab samples of organisms. This is helpful in various fields like ecology, epidemiology, microbiome research, and other fields.

The non-sequencing techniques like DNA fingerprinting produces less-precise information which is eassy to obtain and it is useful for:

a) Detecting the presence of known genes for medical purposes (see genetic testing)

b) Forensic identification

c) Parental testing

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

D. DNA Fragment Assembly Problem

Biologists have developed very clever laboratory methods for the sequencing of DNA molecule.But, those methods only work on short DNA fragments i,e it fails for larger fragments, on the order of hundreds of units.Generally, a complete strand of DNA is made up of millions of bases.Biologists break the DNA strands into short fragments which is of the size that the sequencing machine can handle that short fragments.(While the Strands which are too short or too long are not sequenced at all.)

There are some major problems with this. First, when the scientist break or cut a strand of DNA then, they are not able to control it exactly, instead the DNA is chopped into pieces at random locations in the string. Then, the fragments are small enough which could be able to be read by the sequencing machine. The second difficulty is that there is no way to keep tracking of how the resulting fragments are ordered in the targeted strand. So, it will end up with the sequences of hundreds of thousands of DNA fragments, but no way to piece them together.

Therefore, algorithms are needed to solve this fragment assembly problem.

II. LITERATURE SURVEY

A. TIGR

The TIGR Assembler, developed at the Institute for Genomic Research, was used to assemble the 1.8 mb Haemophilus infuenzae genome (Sutton et al. 1995). In the TIGR assembler, first a pairwise comparison of all reads in a data set identifies potential overlaps between reads. The pairwise comparison for n reads is a function of n2. To speed up this step, rather than executing a full Smith-Waterman alignment (Smith & Waterman 1980) on each pair, an evaluation of the number of substrings common to both reads is performed. Only those pairs with a sufficient number of substrings in common are fully aligned and checked for similarity.

During the pairwise comparisons, putative repeated regions are identified in reads that have an overly abundant number of potential overlaps.

Assembly of these reads is deferred until last and is carried out with a higher match stringency. In addition, distance constraints are used to help properly place sequences that have been identified as potential repeats. The distance between some pairs of reads containing a repeat may be known. In that case, if a read containing a repeat already has its paired read in a contig, then only regions at the given distance from the paired read are considered for overlap with the repeat read.The consensus sequence is generated by examining the base calls in an aligned column.

A profile is produced that indicates the total number of reads in the column with calls of A, C, G, T, and gap. Allowable consensus calls include: upper and lower-case bases (A, C, G, T, a, c, g, and t), gap, two-base ambiguity codes (r, k, s, w, m, and y), and n. A small set of rules determines the consensus call based on the profile. If the largest component in the profile is greater than two-thirds the total, an upper-case base or gap is called. If two non-gap components are significant, a lower-case ambiguity is called. If the largest component is between one-half and two-thirds the total, a lower-case base or gap is called. In all remaining cases, a lower-case n is called. Lower case letters indicate when the confidence in the call is not high. In addition, if the confidence in a gap call is not high, the call following the gap is lower-case. Using this method, lower-case letters in the consensus sequence pinpoint calls that require examination by human editors.

B. GAP

The Genome Assembly Program (GAP) uses a greedy approach to fragment assembly (Bonfield, Smith, & Staden 1995). First a companion program, PREGAP assigns quality values to base calls and trims reads to remove poor quality data and vector. Then, one at a time, each read is added to a contig if it is sufficiently similar. First the read is compared against all other reads for matching subsequences. An alignment is then made between the read and each other read for which it has a match, and the quality of each alignment is noted. The read is overlapped with the read with which it has the highest-quality alignment (over some minimum threshold). If the read aligns sufficiently well with more than one read, then after it is overlapped with the best-aligning sequence, it is used to join the contigs of the two matching reads.

The GAP program can also use distance constraints. To use the constraints, the placement of a read may be restricted to an approximate given distance from an anchor read. If an above threshold alignment can be found for the read within the specified region, the read is added to the contig. The program also allows for a variety of tags to be used. One tag is used to label repeated regions in reads. A region that is tagged as having a repeat is not used in searching for subsequence matches but is aligned during assembly.

C. AP2

CAP2 is an improved version of the Contig Assembly Program (CAP) (Huang 1996). The assembly methods developed for CAP are at the core of the ABI Prism AutoAssembler. This program assembles reads in three phases: 1) overlap detection, 2) contig formation, and 3) consensus sequence determination. In the first phase, a filter identifies which pairs of reads are likely to overlap. Between all such pairs, the match similarity is computed using a variant of the Smith-Waterman alignment algorithm (Smith & Waterman 1980). Error are computed and used to evaluate the strength of overlaps and to identify chimeric reads. In the second phase, a preliminary assembly is formed by joining pairs of reads in decreasing order of pairwise similarity. Inconsistent overlaps in the preliminary assembly are used to identify and partition repeated reads. The partitions are used to establish the final assembly. In phase three, the contigs are fully aligned and the consensus computed.

D. Alewife

A method called Alewife that is reported to run in time less than n2 is described on the Whitehead Institute/MIT Genome Sequencing Project web site (MIT 1998). The basic idea of this assembler is similar to the idea I use in my layout algorithm. The Alewife method uses 25-mers as tags to identify overlaps in fragment reads. By using a hash table, Alewife is reported to assemble fragments at a rate that is proportional to n log n. It thus appears to be based on an algorithm capable of performing sequence assembly more quickly than programs using n2 algorithms. However, the linear algorithm I have developed, SLIC, performs sequence assembly proportional to n, and is theoretically faster than Alewife. Since the details of the MIT algorithm are not yet published, other possible differences between my layout algorithm and Alewife are not known.

E. Genetic Algorithms

A unique approach to sequence assembly is to use genetic algorithms (Parsons & Johnson 1995, Parsons 1993). In this work, the layout of a data set with n fragments is represented as a bit string with n * k bits (k bits per fragment) where n 2k. For each fragment, the integer value of its k bits identifies the position of the read in the overall layout. Added to the n * k bit string is an additional k bits used to identify the starting fragment in the layout. The total length of the bit string is then (n + 1) * k. Standard operators are used while the mapping from bit strings to layouts ensures that legal layouts are generated. The fitness function evaluates the strength of overlaps between adjacent reads in the layout. Two variants of the fitness function were developed, one is proportional to n and the other to n2. Depending on which function is chosen, the overall time to run the algorithm is then proportional to n2 or n3. In initial work, although the genetic algorithm approach produced good layouts for small data sets under 20 kb, the large search space often led to the failure to find good solutions for larger data sets. Later work produced acceptable results with a 35 kb data set, but this size is still far from the size of sequencing projects undertaken by large sequencing centers and is certainly far less than most whole genomes. Another drawback of the method is its failure to deal with highly conserved repeat regions. With the presence of repeats, the resulting consensus tends to be shorter than expected, indicating a compression of the repeated regions.

III. PROPOSED METHODOLOGY

In this paper we will consider fragments randomly obtained through the shotgun procedure.The procedure starts by obtaining a large number of duplicates of the target segment with cloning techniques.Then many fragments are randomly extracted from each copy as samples of the original string, expecting that on each position of it ten to hundred fragments are available for reconstructing its content, as sketched.

Fig: System Architecture

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

The target segment is duplicated in many copies; from the various copies random samples are created. The overlapping of the initial readable parts of the samples makes reconstruction possible (the black color refers to the first readable part of the sample).

Fragment production is indeed not a free lunch operation. Management of this delicate nucleotide assembl8y affects fragments (up to 5% of their com- ponentes') through three types of elementary corruptions: substitutions (=change of a nucleotide with another), insertions (of a nucleotide) and deletions (of a nucleotide).

Starting from the most scored pair, whatever the orientation of each fragment, an optimal alignment is obtained by minimizing their mutual distance. A fragment contained in another one disappears in it; it joins it instead in the case of a mere overlap on one of its ends, thus concurring to define a larger fragment named contig. If no overlap- ping exists between contigs, they remain constituting separate parts of the reconstructed mother sequence, which in this case has some gaps. Contig assembling runs in two phases.

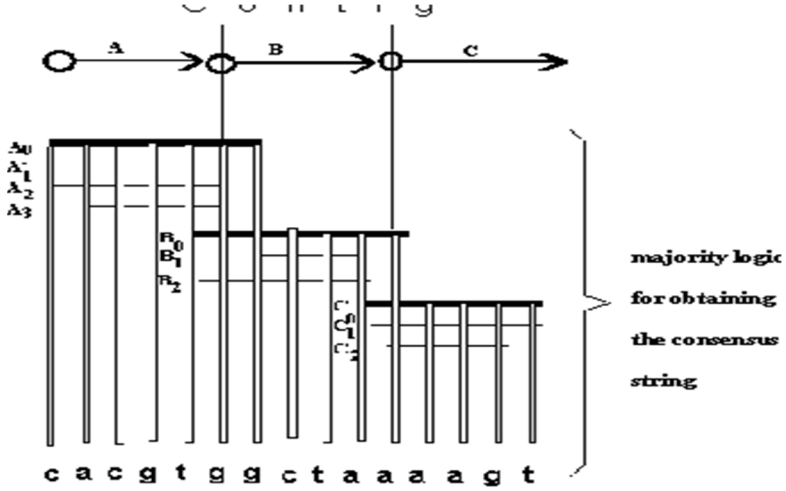
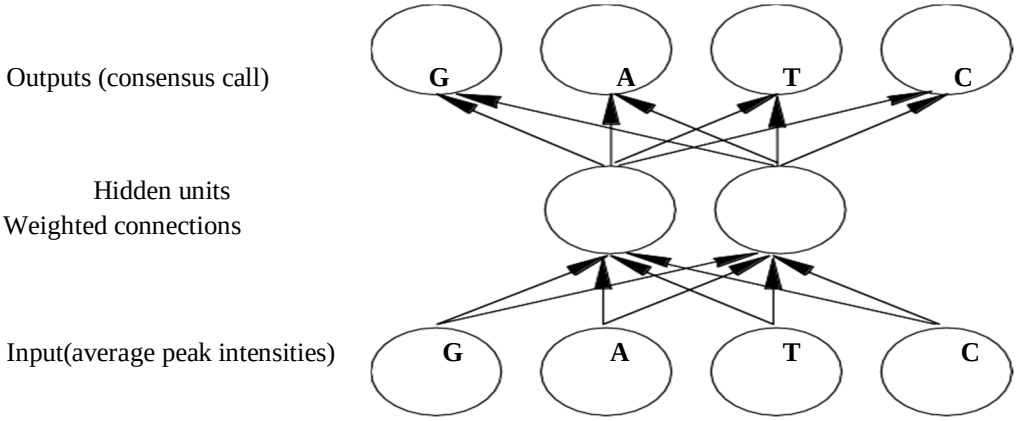


Fig: Contig

At the end of the assembly process, the consensus string at the bottom is obtained by the majority rule. Bold lines represent non contained fragments; thin line contained fragments. Capital letters denote fragment names, small letters base in the consensus string.

Neural networks often provide a good solution to biological problems such as these since the problems involve intricate interactions, and the strength of neural networks lies in their ability to learn to recognize complex patterns. Given their success in the computational research community, neural networks have the potential to be a powerful tool for data analysis in biological research labs. In spite of this, the usage of neural networks for the tasks in DNA sequencing have been scarcely explored.

Figure: contains a brief description of the operation of neural networks; details can be found in McClelland and Rumelhart (1986).



a) *Inputs:* Average relative G, A, T, and C trace peak intensities.
b) *Outputs:* A consensus call for the aligned column.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue V May 2020- Available at www.ijraset.com

A. Neural Networks

A feed-forward backpropagation neural network learns to categorize patterns of inputs. Here, Inputs are the features of a problem which are represented in numerics. Typically, there is one output for each category of the problem; the desired output is 1 for the correct category and is 0 otherwise. First the network is trained by processing a set of categorized examples (a training set). A categorized example is an instance of the problem that includes its inputs and desired outputs. During training, weighted connections in the network are adjusted so that the error in the actual output is reduced. When the difference between the desired and actual inputs is sufficiently low, training is halted and the network can be used to categorize previously unseen instances of the problem. Future accuracy of the trained network is estimated by measuring the trained network's performance which is on a disjoint set of testing examples. In the fig. consisting of an example of a neural network where the inputs are extracted using the function whose purpose is to call the consensus for a single aligned column of DNA bases from fluorescent traces. The network is given four inputs (the relative G, A, T, and C trace intensity averages), and outputs a consensus call (G, A, T, or C).

To determine the sequence of bases in a genome or large segment of DNA, researchers first produce and sequence small, overlapping fragments of the genome. The base-call sequences of the small fragments are commonly referred to as fragment reads, sequences, or simply reads. The overlapping regions of the fragment reads are aligned into one or more contigs (contiguous segments) and the resulting layout is used to determine the consensus sequence of the genome. The process of determining the layout of the reads is called fragment assembly or sequence assembly. Assembly is complicated by repeats, subsequences that occur more than once in a genome. When a read contains a repeat, its placement in the layout is often ambiguous.

Four major steps are required for this process. Firstly, the reads must be preprocessed in order to remove both the low-quality data and vector sequence. Second, the layout method is used which establishes the approximate offset for each read in a contig. Then the reads are gapped and aligned so as to produce an assembly. Finally, a consensus sequence for each contig is determined. The steps are summarized here. Preprocess to trim poor-quality ends and remove vector sequence.

1) Determine the layout of fragment reads into contigs using SLIC.
2) Align the layout of reads in each contig.
3) Compute the consensus sequence for each contig.

The layout algorithm usually relies on the subsequences of bases or the mers which occurs in the overlapping regions of fragment reads. The mers which are common to two or more fragment reads are aligned in order to determine the overall layout of reads.Here the premise is that the large DNA fragments may contain many mers that occurs only once, that can be used to tag the positions(relative) of the fragment reads**.** The idea of using mers to tag fragments and identify overlaps is illustrated.

B. Actual Sequence

CGAATGTCATATGGCAGTACACGGCGTACGTTAGGTTTCTGAGGGATTTTCGAG

Fragment Reads:

1. CGAATGTCATATGGCAGTA
2. TATGGCAGTACACGGCGTACGT
3. GGCGTACGTTAGGTTT
4. TTAGGTTTCTGAGGGATT
5. AGGTTTCTGAGGGATTTTCGAG

Fragment Read Layout:

1. CGAATGTCATATGGCAGTA
2. TATGGCAGTACACGGCGTACGT
3. GGCGTACGTTAGGTTT
4. TTAGGTTTCTGAGGGATT
5. AGGTTTCTGAGGGATTTTCGAG

Using Mer Tags to Identify Overlaps. The mer tags for each of the fragment read are underlined and align to the matching mer tags in order to determine the layout of the reads

IV. RESULT & ANALYSIS

```
In [1]: DNA_sequence_input= 'AAAATTGCGGCATGTGGGCTGACTCTGAAAGCGATGCTCACGAAAAGGGAACGCGGCGCCGTCGGGCGCCGCGCGCCGCTTAGGACTGCTGGCCTGCGGCCGGCGCC
        DNA_sequence = list(DNA_sequence_input)

In [2]: #Give fragment length
        split_input = 100

In [4]: composite_list = [DNA_sequence[x:x+split_input] for x in range(0, len(DNA_sequence),split_input)]
        # print (composite_list)

In [5]: fragments = pd.DataFrame(composite_list)

In [6]: fragments
```

Out[6]:

	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93	94	95	96	97	98	99
0	A	A	A	A	T	T	G	C	G	G	...	G	G	C	C	T	G	C	G	G	C
1	C	G	G	C	G	C	C	T	G	C	...	G	G	G	C	T	G	C	T	G	C
2	T	T	G	G	C	G	G	G	C	T	...	A	C	C	G	G	C	G	C	C	C
3	C	C	G	G	A	C	G	G	A	T	...	None	None	None	None	None	None	None	None	None	None

4 rows × 100 columns

```
In [7]: most_repeated_one = fragments.mode()

In [8]: #most_repeated_one

In [9]: final_DNA_sequence = most_repeated_one.loc[0,:]

In [10]: final_DNA_sequence = list(final_DNA_sequence)

In [11]: final_DNA_sequence = ''.join(final_DNA_sequence)
         final_DNA_sequence

Out[11]: 'CAGGACGGGTCTTGTGGGAGCCCTCGAAGACCACTACTGCTCAGCAGGCCACGAGGCGCCGCGCCCCCCCTCGCCCAGCTCAAGACTGGCGGCCTGCCGC'

In [12]: final_DNA_out = pd.DataFrame({'DNA_sequence':[final_DNA_sequence]})
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	DNA_sequence													
2	CAGGACGGGTCTTGTGGGAGCCCTCGAAGACCACTACTGCTCAGCAGGCCACGAGGCGCCGCGCCCCCCCTCGCCCAGCTCAAGACTGGCGGCCTGCCGC													
3														

A. *Advantages of Proposed System*

As the speed of producing sequencing data increases, computational methods for assembling large amounts of data in a time-efficient manner must be developed. Fragment assembly programs currently favored by large genome sequencing centers execute pairwise comparisons of fragment reads, resulting in assembly times that are proportional to n2 (where n is thenumber of reads). I have developed an algorithm for fragment layout into Contigs, that avoids explicit pairwise comparisons of all reads. By using a hash table to store and retrieve information on the subsequences that occur in reads, this algorithm is, in practice, a function of n. This represents a dramatic decrease in assembly time as the number of reads in an assembly increases.

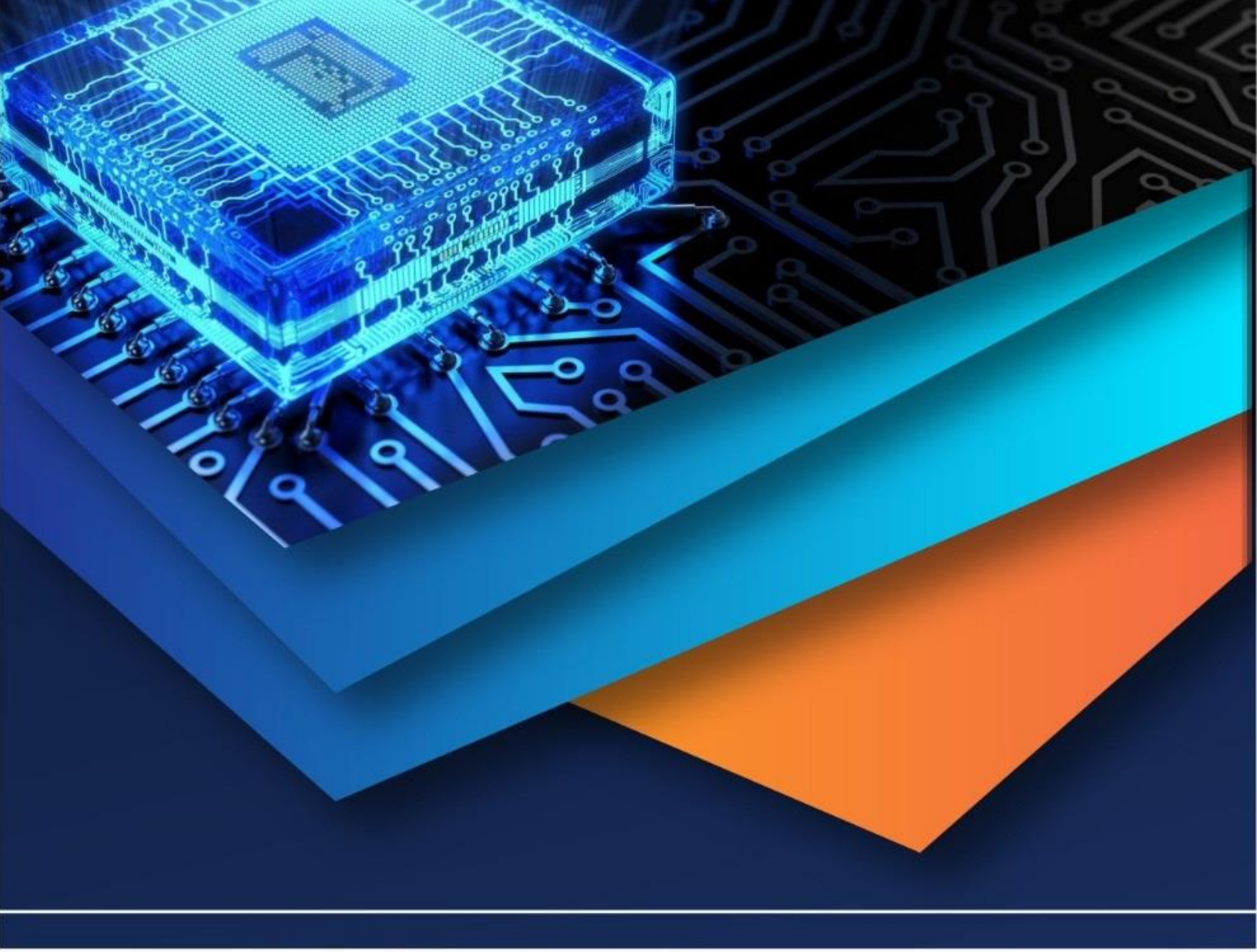
V. CONCLUSION

The goal of my work is to develop computational methods for increasing the speed and accuracy of DNA fragment assembly. As advances in technology result in the production of increasing amounts of sequencing data in decreasing amounts of time, it is imperative that computational methods are developed that allow data analysis to keep pace. In this dissertation, I presented a method that improve the speed and accuracy of fragment assembly and that lay a foundation for further research.

The work I have completed represents a real improvement in methods for DNA fragment assembly. Although the work has limitations, it still provides a solid ground for building more sophisticated solutions to unsolved problems.

REFERENCES

[1] J. Setubal and J. Meidanis 1977, Introduction to Computational Molecular Biology (PWS, Boston).
[2] A. M. Maxam and W. Gilbert 1977, "A new method for sequencing DNA," Proc. Natl. Acad. Sci. USA 74, 560–564.
[3] R. Staden 1981, "Automation of the computer handling of gel reading data produced by the shotgun method of DNA sequencing," Nucl. Acids Res. 10, 4731–4754. F. Sanger, A. R. Coulson, G. F. Hong, D. F. Hill and G. B. Peterson 1981, "Nucleotide sequence of bacteriophage l-DNA," J. Mol Biol. 162, 729–733.
[4] B. Lewin,"GENES IV," Oxford University (1990).
[5] T. F. Smith and M. S. Watermann 1981, "Identification of common molecular sequence," J. Mol Biol. 147, 195–197.
[6] O. Nerrand, P. Roussel-Ragot, D. Urbani, L. Personnaz and G. Dreyfus 1994, "Training recurrent neural networks: Why and how? An illustration in Dynamical Process Modelling," IEEE Trans. on Neural Networks 5(2), 178–184.
[7] B. Apolloni, D. Crivelli and M. Amato 1993, "Neural time warping," Proc. of Eurospeech 93, Berlin 1, 139–142.
[8] J. Galant, D. Maier and J. Storer 1980, "On finding minimal length superstrings," J. Comput. and Syst. Sci. 20, 50–58.
[9] J. D. Kececioglu, "Exact and approximation algorithms for DNA sequence reconstruction," Tech. Rep. 91–26, Dept. of Comp. Sci., Univ. of Arizona, Tucson (1989).
[10] K. Culik II and T. Harju 1989, "Dominoes and the regularity of DNA splicing languages," Lect. Notes on Comp. Sci. 372, 222–233.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY