CONTENT OF INTERNATIONAL PUBLICATIONS:

Content of International Publications:

The content of publications in International Journals and Conferences is given as follows. The content and format is unaltered and is presented here as available in a particular Journal or Conference Proceedings.

IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 1 (Mar. - Apr. 2013), PP 19-24
www.iosrjournals.org

Signal Processing Approach for Recognizing Identical Reads From DNA Sequencing of Bacillus Strains

Mamta C. Padole*[1], B. S. Parekh[2], D.P. Patel[3]
[1]*Department of Computer Science and Engineering, The Maharaja Sayajirao University of Baroda, India.*
[2]*Department of Computer Science and Engineering, The Maharaja Sayajirao University of Baroda, India.*
[3]*Department of Applied Mathematics, The Maharaja Sayajirao University of Baroda, India.*

Abstract *: DNA sequencing generates a large number of reads of lengths varying from 100bp to 1000bp, when sequenced using different methods of sequencing. These reads are further assembled to form contigs which are useful in annotation. The library generation using different amplification technique is involved in DNA sequencing process, which generates several identical reads, which are redundant, resulting in degraded quality of sequencing, besides also causing longer time for assembly. Existing computationally complex algorithms use string processing. The paper discusses the signal processing approach with application of Wavelet Transforms, designed to find exact and near exact identical reads. The string processing approach for pattern matching in search of similar patterns is computationally very expensive because the order of complexity of String comparisions is exponential in nature. Whereas Wavelet Transforms translates the sequence in co-efficients which are half of the length of the original sequence. On applying Wavelet Transforms repeatedly on the sequence, the sequence get transformed to half the length of the sequence used for transformations. Thus the order of complexity reduces to O(log n), which is much efficient compared to string processing.*

Keywords – *Haar wavelets, identical reads, pattern recognition, signal processing, wavelets,*

I. Introduction

DNA sequencing is the method of identifying the arrangement and order of nucleotides in a DNA sequence. The conventional widely used method of sequencing, the Sanger sequencing, implemented chain termination with di-deoxynucleotides [1], but has limitations in terms of throughput and cost of large genome sequencing[2]. The other methods are sequencing-by-hybridization (SBH), nanopore-sequencing and sequencing-by-synthesis [3]. "Sequencing-by-synthesis" involves taking a single strand of the DNA to be sequenced and then synthesizing its complementary strand enzymatically.

The process of DNA sequencing requires the library generation as one of the steps, which enable amplification of DNA [2] sequences which are available for sequencing. This process of amplification has a possibility of biased amplification of a DNA template causing large number of reads of same segment of DNA generated multiple times and thus causing large number of identical reads.

It is suggested that special attention should be paid to potential biases [4] introduced by these identical reads, especially in the cases of analyzing quantification and transcriptome profiling sequence data.

In this paper, we present an *a priori method*, for recognizing identical reads, which does not require any mapping reference for recognizing identical read, nor does it need to compare any string pattern as an input parameter for comparison [4] neither does it use clustering on basis of seeds [6]. The paper discusses the use of a heuristic approach of signal processing as a recognition criterion, for detecting identical reads from DNA sequencing reads, including exact and near exact identical reads. This paper emphasizes on the use of efficient Wavelet Transforms particularly the *Haar Wavelets* for identifying these identical reads. The time complexity of Wavelet transforms is *O(log n)*, n being the length of the transformed sequence.

II. Methods

The suggested algorithm for recognizing identical reads from the set of *DNA sequenced* reads is applied as in Fig. – 1 and the explanation following thereafter.

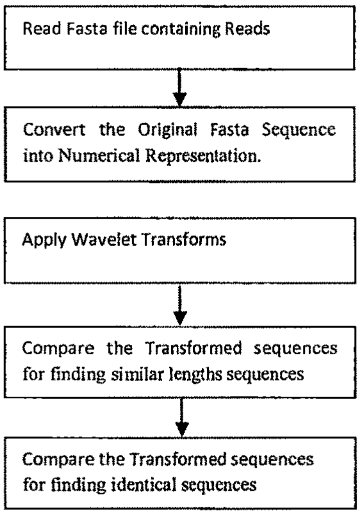

Read Fasta file containing Reads
Convert the Original Fasta Sequence into Numerical Representation.
Apply Wavelet Transforms
Compare the Transformed sequences for finding similar lengths sequences
Compare the Transformed sequences for finding identical sequences

Fig. 1. Steps to Identify the Identical Reads

Read the fasta file which contains the sequence $S_i = \{s_1,s_2,...s_n\}$ where $s_i \in \sum = \{ A,C,G,T \}$, i = 1, n and n is the length of S_i.

Convert the nucleotide sequence S_i into its numerical representation X_i. The single indicator sequence using Electron-ion interaction pseudo potentials - EIIP property of nucleotides, is used for numerical representation. EIIP values for A= 0.1260, C = 0.1340, G = 0.0806, T = 0.1335 [25]. The use of EIIP values for single indicator sequence representation reduces the computational overhead by 75% compared to the conventional four-base binary sequence representation of nucleotide sequence [25]. Only numerical representations can be applied for Wavelet transformations.

The next step is to perform multi-level Wavelet transforms on the numerical representation of the sequences. We performed four-level Haar Wavelet transforms on the sequences. The Haar Wavelet Transform applied up to fourth level, reduces the length of the original sequence to one-eighth. This reduced length transformed sequences can be efficiently used for comparison.

Compare the length of transformed sequences, to check whether the sequences are comparable.

If the lengths of the transformed sequences are same, then the element by element equality of the two transformed sequences for finding the identical reads is performed.

Thus, data-reduction without loss of information using Wavelet transform is applied to recognize identical reads. If a single element of a four-level Haar Wavelet Transform is found to be equal, it means, eight nucleotide bases in a given read are found to be similar. Thus it is much efficient to perform a single comparision on signal processed data, instead of eight comparisions while implementing string processing. Since, the computational complexity of Haar Wavelet is *O(log n)*, it is much faster than any other string processing based methods of finding identical reads. Haar transforms are also memory efficient, as computations are performed in place.

2.1 Wavelet Transforms

A wavelet transform is a transformation of a signal or data into time-scale domain on a basis of wavelet functions [7][8]. The wavelet transform representations enable exploring the hidden information about the signal. Two co-efficient vectors [9] are generated, the approximate and the detail co-efficient vectors, after Wavelet transform is performed of the original signal.

When a signal x is passed through low pass filters (scaling functions) and high pass filters (wavelet functions) simultaneously, it is defined as performing the discrete wavelet transform (DWT) which along with down-sampling, generates co-efficients with half the length of the original input to each filter.

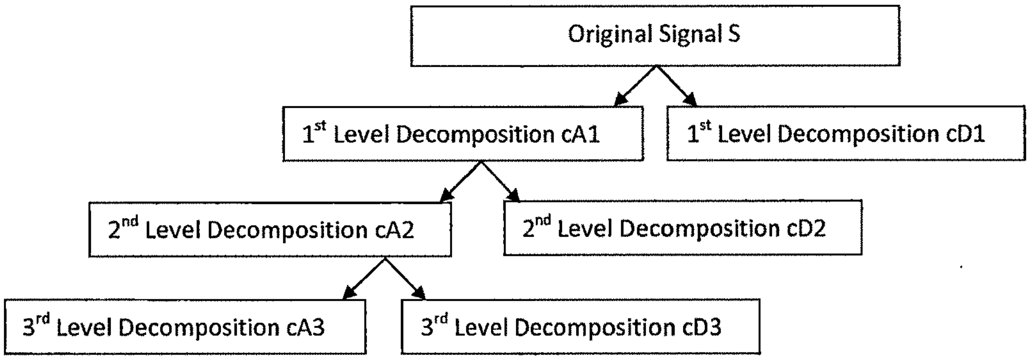

Original Signal S
1st Level Decomposition cA1
1st Level Decomposition cD1
2nd Level Decomposition cA2
2nd Level Decomposition cD2
3rd Level Decomposition cA3
3rd Level Decomposition cD3

Fig. 2. The Decomposition Phase in Discrete Wavelet Transforms. After each level of transform and down-sampling, half the length of co-efficient are generated at each pass.

Wavelet Transform W_T can also be represented as in (1),

$$W_T = X.W, \quad where\ W = [\varphi(x); \psi(x)] \qquad (1)$$

As in (Equation 1.),

$\varphi(x)$ is called scaling function to find the approximate co-efficients and

$\psi(x)$ is called wavelet function to find the detail co-efficient

Wavelet decomposition can be applied to Haar wavelets are related to a mathematical operation called the Haar transform. All the wavelet transforms refer to Haar Wavelets as its prototype. [15]. any sequential data, including strings, where, in case of strings, the position of a character in string represents the time series data. Wavelets are tools used to study regularity and to conduct local studies [27]. The zero moments [24] of the function are related to the regularity of scaling function & wavelets [14].

2.1.1. Haar Wavelets

Haar wavelets are conceptually simple, fast and memory efficient [17], [18], can be computed without a temporary buffer, are exactly reversible, can be perfectly reconstructed and are defined to be orthonormal [16].

Applications of Haar wavelets are dimensionality reduction [11], approximate querying of database [12], image processing [10], selectivity estimation tasks, digital network synthesis [19], binary logic design [20] [21] [22].

The Haar transform decomposes a discrete signal x into two sub-signals of half its length. One sub-signal is a running average or trend (C_a) as in Table-1; the other sub-signal is a running difference or fluctuation (C_d) as in Table-1. [23].

1.1.2. Computation of Haar Wavelet Transform of Time Series Data

Consider a one-dimensional data vector X containing the N = 8 data values X = [8,8,0,8,12,20,16,16]

Table 1. Representation Of Computations Of Haar Wavelet Transform

Transformation Level or Decomposition Level (n)	**Resolution or Granularity (Order k)**	**Length of signal (L)**	**Averages / Approximate Co-efficients (Ca)** **Ca = (xi + xi+1)/2**	**Differences / Detail Co-efficients (Cd)** **Cd = (xi - xi+1)/2**
Original signal	3	8	[8,8,0,8,12,20,16,16]	-
1	2	4	[8,4,16,16]	[0, -4, -4, 0]
2	1	2	[6, 16]	[2,0]
3	0	1	[11]	[-5]

Haar wavelet transform, are computed by iteratively performing pair-wise averaging and differencing [13].

The values are first averaged together pair-wise to get a new "lower-resolution" representation of the data with the following average values [8, 4, 16, 16]. To restore the original values of the data array, additional detail coefficients must be stored to capture the information lost due to this averaging. In Haar wavelets, these detail coefficients are simply the differences of the second value and the first value, of the pair from the computed pair-wise average, divided by 2, i.e., [8-8, 4-8, 16–20,16–16] = [0, –4, –4, 0].

There is no information loss in this process; it is simple to reconstruct the eight values of the original data array from the lower-resolution array containing the four averages and the four detail coefficients.

Recursively applying the above pair-wise averaging and differencing process on the lower-resolution array containing the averages, gives the full transform, which can be explained as follows :

The Haar wavelet transform WT of the original signal X is the single coefficient representing the overall average of the data values followed by the detail coefficients [Table 1.] in the order of increasing resolution, i.e., WT = [11, -5, 2, 0, 0, -4, -4, 0]
Each entry is called a wavelet coefficient.

III. Results

The algorithm defined is tested on the Short Reads Archive (SRA) data. The SRA data downloaded from NCBI site ftp://ftp.ncbi.nlm.nih.gov/sra/ containing short reads. The sequenced reads are for various strains of Bacillus. Table 2. and Table 3. show the results of Identical Reads recognized using Wavelet Transforms. The tables represent the output in terms of reads as well as nucleotide base pairs.

Table 2. Result Showing Number and Percentage of Identical Reads Recognized Using the Wavelet Transforms based Algorithm from various Strains of Bacillus

SRA Accession No.	**Total No. of Reads**	**Total No. of Copies of Identical Reads**	**Total Percentage of Identical Reads (%)**	**Total No. Unique Reads amongst Identical Reads**	**Total No. of Redundant Reads**	**Total Percentage of Redundant Reads %**
a	**b**	**c**	**d**	**e**	**(c-e)**	**((c-e) * 100) / b**
SRR149222	2182	249	11.4115	95	154	7.0577
SRR065619	3404	410	12.0447	156	254	7.462
SRR153778	3670	417	11.3624	158	259	7.0572
SRR393844	10932	681	6.2294	254	427	3.906
SRR052290	74076	12634	17.055	4291	8343	11.263

Table 3. Time Taken to find the Identical reads Using Wavelet Transforms based algorithm

SRA Accession No.	**Total No. of Reads**	**Total No. of Copies of Identical Reads**	**Time Taken for Recognizing Identical Reads**
a	**b**	**c**	
SRR149222	2182	249	9.4601 secs.
SRR065619	3404	410	15.3080 secs.
SRR153778	3670	417	16.6342 secs.
SRR393844	10932	681	82.5258 secs
SRR393839	12563	58	98.9385 secs.
SRR052290	74076	12634	1933.6 secs.

Table 4. The subset of records of the Result generated by Matlab program using Wavelet Transforms Algorithm, which represents the Read Nos. of all Identical Reads found in the Bacillus with SRA Reference Id. SRR149222

Read No.	1st Identical Read No.	2nd Identical Read No.	3rd Identical Read No.	4th Identical Read No.	5th Identical Read No.	6th Identical Read No.	7th Identical Read No.	8th Identical Read No.	Total No. of Identical Reads
300	543	867	909	1327					5
313	514	538	624	1192					5
317	359								2
325	494								2
327	479	931	1258	1607					5

328	433	628	748	832	974	1173	1312	1378	9
329	434	629	749	833	1174	1313			7
330	933	1063							3
347	616								2
351	353								2
356	772	1866							3

Table 5. Count of Redundant Reads and the No. of Redundant Base Pairs, for the subset of records of Reads found in SRR149222

1st Read No. whose Identical Reads are found	Sequence Length in Base Pairs	Total No. of Copies of Identical Reads	Total No. of Redundant Copies of Reads, after preserving 1 copy of Identical Reads	Total No. of Redundant Base Pairs, after preserving 1 copy of sequence of Identical Reads
(a)	(b)	(c)	(d) = (c) - 1	(e) = (d) * (b)
300	236	5	4	944
313	259	5	4	1036
317	218	2	1	218
325	191	2	1	191
327	417	5	4	1668
328	138	9	8	1104
329	54	7	6	324
330	227	3	2	454
347	144	2	1	144
351	117	2	1	117
356	77	3	2	154

From the **TABLE 4 & 5**, it is observed that there are several copies of identical reads found from DNA sequenced data. If the entire result is stored, without verification, than lot of redundant data may be preserved unnecessarily, occupying lot of disk space, at the same time causing increased processing time during annotation due to irrelevant data.

From DNA sequenced data of Bacillus with SRR149222 reference id, The total number of redundant reads are 154 and total number of redundant bases are 27536 resulting in wastage of storage space up to 7.0577% in terms of reads [**TABLE** 2] and 3.8738% in terms of bases.

Also, it is interesting to know that the SRA sequence with SRA Reference Id. SRR393844 contained the Read No. 62 with length 6, whose total number of identical copies were 52 copies [Fig. 3.].

62	64	207	209	821	847	896	1003	1071	1387

1434	1497	1832	2036	2104	2572	2579	2774	2924	3005

3120	3266	3487	3546	3877	4214	4378	5695	5899	5925

6016	6116	6765	7098	7274	7438	7471	7723	7751	7906

8207	8527	8559	8577	9136	9304	10013	10171	10282	10414	10542	10905

Fig. 3. List of Read Numbers of Identical Reads (Starting Read Number is 62)

So, this category of reads with irrelevant length and large number of copies can cause increased processing time during further analysis of these reads. The Wavelet Transform based algorithm defined in this paper also helps in removing this type of identical and insignificant reads from the generated sequencing output.

IV. Discussion

Thus, Signal Processing Approach can be used to compare the two reads generated from DNA sequencing, for verifying their resemblance. Using Wavelet Transforms we can reduce the data for comparision to one-eighth size of the original sequence. This data reduction using Wavelet Transforms optimizes the computational complexity to logarithmic order and hence provides improved algorithm for recognizing identical reads amongst the DNA sequenced data. Once the similar sequences are identified, it is not necessary to store the entire sequence, instead can store only the references to the strings for further annotation. This also optimizes the space requirement for storage of reads in the database. Also Wavelet Transforms are performed in place and hence memory requirement is reduced. Thus the proposed algorithm optimizes both space and time complexity involved in recognizing identical reads from DNA sequenced data.

Further, if it is possible to apply distributed computing on this algorithm, improvement in processing time is possible, particularly when data is very large.

V. Conclusion

The results reflect that the Wavelet Transforms can be applied to identify Duplicate Reads from DNA sequencing reads. It is also helpful in improving the efficiency in applying search for identical reads and is memory efficient.

References

[1] Sanger, F.*et al.* DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci.* 1977, 74: 5463–5467.

[2] Ronaghi, M. Pyrosequencing Sheds Light on DNA Sequencing *Genome Res.* 2001, 11: 3-11

[3] Metzker, M. L. *et al.* Emerging Technologies in DNA Sequencing. *Genome Res.* 2005, 15: 1767-1776

[4] Dong, H. *et al.* Artificial duplicate reads in sequencing data of 454 Genome Sequencer FLX System *Acta Biochim Biophys Sin* 2011, 43: Issue6: 496–500

[6] Gomez-Alvarez V. *et al.* Systematic artifacts in metagenomes from complex microbial communities. *ISME J*, 2009, 3 : 1314–1317

[7] Meher,J. K. et al. Wavelet Based Lossless DNA Sequence Compression for Faster Detection of Eukaryotic Protein Coding Regions , I.J. Image, Graphics and Signal Processing 2012, 7, 47-53

[8] Daubechies, I. Ten Lectures On Wavelets, 1992

[9] Aggarwal, C. C. On the Use of Wavelet Decomposition for String Classification, *Springer - Data Mining and Knowledge Discovery*, 2005, 10, 117–139

[10] Castleman, K.R. Digital Image Processing, (Englewood Cliffs: Prentice-Hall, 1996)

[11] Keogh E. *et al.* Dimensionality reduction for fast similarity search in large time series databases, *J Knowledge Information Systems*, 2001, 3:263–286

[12] Garofalakis, M. Discrete Wavelet Transform and Wavelet Synopses, Springer Science+Business Media, LLC *Encyclopedia of Database Systems*, 2009, 10.1007/978-0-387-39940-9_539

[13] Sacharidis, D. Constructing Optimal Wavelet Synopses , *Proceedings of the 2006 International Conference on Current Trends in Database Technology* EBDT, 2006, 10.1007/11896548_10 Pg 97-104

[14] Burrus, C.S. *et al.* Introduction to Wavelets and Wavelet Transforms (*Prentice Hall*, 1998)

[15] Mohamed M. I. *et al.* Comparison between Haar and Daubechies Wavelet Transformtions on FPGA Technology, *Proceedings of World Academy Of Science, Engineering And Technology*, 2007, Volume 20 ISSN 1307-6884

[16] Moharir, P.S. Pattern recognition transforms, New York: (Wiley 1992)

[17] Radomir S. *et al.* The Haar wavelet transform: its status and achievements, Elsevier Science Ltd. *Computers and Electrical Engineering*, 2003 29 25–44

[18] Ruiz, G. *et al.* Switch-level fault detection and diagnosis environment for MOS digital circuits using spectral techniques, *IEE Proc Part E*, 1992, 139(4):293–307

[19] Hurst, S.L. The Haar transform in digital network synthesis, *Int Symp Multiple-valued Logic*, Proc, 1981, 11th. p. 10–8.

[20] Falkowski, B.J. Mutual relations between arithmetic and Harr functions, *Proceedings of IEEE Int Symp Circ Syst*, ISCAS, 1998, vol. V. p. 138–41.

[21] Falkowski, B.J. *et al.* Efficient algorithm for forward and inverse transformations between Haar spectrum and binary decision diagrams, Int Phoenix Conf Comput Commun,, 1994, p. 497–503.

Recognizing Artificial Duplicate Reads in 454 Pyrosequencing Using Wavelet Transforms

Mamta C. Padole
Department of Computer Science and Engineering,
The Maharaja Sayajirao University of Baroda, India
Email: mpadole29@rediffmail.com

ABSTRACT

Pyrosequencing technique generates a large number of short reads of length 300bp to 500bp. These reads are further assembled to form contigs and scaffolds which are useful in annotation. The pyrosequencing technique, during library generation, through emulsion PCR, generates several artificial duplicate reads, which are redundant. These redundant reads result in degraded quality of sequencing, besides also causing longer time for assembly. Existing computationally complex algorithms use string processing to identify these duplicates. The paper discusses the de novo algorithm using signal processing approach, applying Wavelet Transforms, to identify exact and near exact artificial duplicate reads from Pyrosequenced data.

Keywords- Pattern Recognition, Signal Processing, Wavelet Transforms, Pyrosequencing, Artificial Duplicate Reads

1. INTRODUCTION

DNA sequencing is the method of identifying the arrangement and order of nucleotides in a DNA sequence. The widely used method of sequencing,

Pyrosequencing [1], based on sequencing-by-synthesis, involves the mechanism of oil emulsion polymerase chain reaction (PCR) for DNA amplification [2]. A unique DNA template would only generate a unique sequence read after being amplified and sequenced on GS FLX. However, biased amplification of DNA templates might occur in the process of emulsion PCR, which results in production of artificial duplicate reads. Under the condition that each DNA template is unique to another, 3.49% to 18.14% of total reads in GS FLX-sequencing data were found to be artificial duplicate reads [3]. These duplicate reads may lead to misunderstanding of sequencing data and potential biases they introduce to the data [3]. It is suggested that special attention should be paid to potential biases introduced by artificial duplicate reads, to avoid misinterpretation of abundance of data, especially in the cases of analyzing quantification, transcriptome profiling sequence data and in metagenomics study [4].

In this paper, we present an ab initio method, for recognizing artificial duplicate reads, which does not require any mapping reference or string pattern for comparison [3] nor seed for clustering [5]. The paper discusses the heuristic approach of signal processing as a recognition criterion, for detecting artificial duplicates from raw 454 sequencing reads, including exact duplicates and near identical duplicates. This paper emphasizes on the use of efficient Wavelet Transforms particularly the Haar Wavelets for identifying the artificial duplicate reads. The time complexity of Wavelet transforms is O (log n), n being the length of the transformed sequence. Also, Haar Wavelet transforms are memory efficient, as computations are performed in place [7] [8]. Thus, Wavelet Transforms prove to be computationally efficient in both time and space. Thus, use of Wavelet Transforms for recognizing artificial duplicate reads would prove to be more efficient than other string processing approach.

2. METHOD

The suggested algorithm for identifying the artificial duplicate reads from the set of *Pyrosequenced* reads, Figure 1.and the explanation following thereafter.

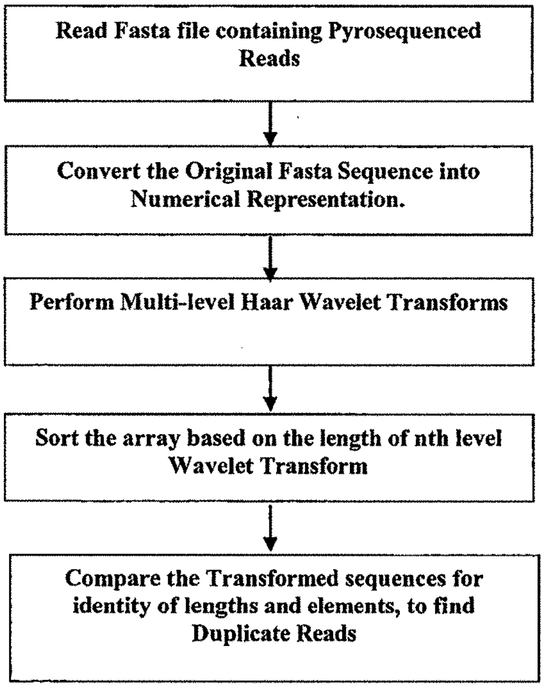

Read Fasta file containing Pyrosequenced Reads
Convert the Original Fasta Sequence into Numerical Representation.
Perform Multi-level Haar Wavelet Transforms
Sort the array based on the length of nth level Wavelet Transform
Compare the Transformed sequences for identity of lengths and elements, to find Duplicate Reads

Figure1. Steps to Identify the Duplicate Reads

Algorithm:

1. Read Fasta File containing Pyrosequenced Reads.
2. Convert each Fasta sequence to Signal i.e. Numerical Representation, using EIIP property of nucleotides.
3. Perform four levels Haar Wavelet Transform on Numerical Sequence.
4. Perform Steps 2 to 3 for all sequences in the Fasta File.
5. Sort the array of detailed co-efficient of fourth level Wavelet Transforms, based on the signal length of detailed co-efficient. Check for the similarity of lengths of the transformed sequences.
6. If the lengths are equal, perform element by element comparision of detailed coefficient of transformed sequences, pair wise, to identify duplicate reads.
7. Perform steps 6 to 7 for all pairs of similar length transformed sequences to identify all the sets of artificial duplicate reads.

Here, the sequences in Fasta file needs to be read, transform it using Haar Wavelets. Perform comparision of length of each transformed sequence, and if the lengths are same, then, compare detailed coefficients of the 4th level decomposition of the remaining sequences. If both are same, put a comparision mark. Compare remaining pairs of sequences, for duplicates.

2.1 Description of an algorithm

Read the Fasta file which contains sequences S.
$S = \{S_1, S_2, ... S_{k\}},$ where k = number of sequences in the Fasta file and
$S_i = \{s_1, s_2, ... s_n\}$, where $s_j \in \sum = \{A, C, G, T\}$, $j \in [1, n]$, n is the length of sequence S_i.

Convert the nucleotide sequence S_i into its numerical representation X_i. This numerical representation is now a Signal for each Read, which is used for Wavelet Transforms.

$X_i=\{x_1, x_2, ... x_n\}$, where $x_j \in \{0.1260, 0.1340, 0.0806, 0.1335\}$
The single indicator sequence using Electron-ion interaction pseudo-potential (EIIP) property of nucleotide [6] is used for numerical representation i.e. generating a Read Signal.

EIIP value for nucleotides is A = 0.1260, C = 0.1340, G = 0.0806, T = 0.1335 [6].

The use of EIIP values for single indicator sequence representation reduces the computational overhead by 75% compared to the conventional four-base binary sequence representation of nucleotide sequence [6]. Only numerical representations can be applied for Wavelet transformations.

The next step is to perform multi-level Wavelet transforms on the numerical representation of the sequences. We performed four-level Haar Wavelet transforms on the sequences. The Haar Wavelet Transform applied up to fourth level, reduces the length of the original sequence to one-eighth. This reduced length transformed sequences can be efficiently used for comparison Sort the array of detailed co-efficient of fourth level Wavelet Transforms, based on the signal length of detailed co-efficient. Sorting would help in comparing only a very small subset of array, instead of comparing the entire array, for similarity.

Compare the length of transformed sequences, to check whether the sequences are comparable.

If the lengths of the transformed sequences are same, then the element by element equality check, of the two transformed sequences for recognizing 100% identities of the duplicate reads is performed.

If the all the detailed coefficients of the fourth level decomposition of two sequences are same, then it is concluded that the two original sequences are also same. In comparing transformed sequences, one needs to perform only one-eighth number of comparisions, which can be performed in much lesser time than comparing the two original sequences. Also, while transforming, there is no data loss, hence the two transformed sequences are comparable.

Thus, data-reduction without loss of information using Wavelet transform is applied to recognize artificial duplicate reads. Since, the computational complexity of Haar Wavelet is *O (log n)*; it is much faster than any other string processing based methods of finding identical reads. Haar Wavelet transforms are also memory efficient, as computations are performed in place [7] [8].

3. WAVELET TRANSFORMS

A wavelet transform is a linear transformation of a signal or data into coefficients on a basis of wavelet functions [9]. The wavelet transformations represent the data in time-scale domain [10], from where hidden information can be explored. Two coefficient vectors [11] get generated, the approximate and the detail coefficient vectors, after Wavelet transform is performed of the original signal. The wavelet technique is applicable to any multi-dimensional or time-series technique. An important property of the wavelet technique is that it creates a hierarchical decomposition of the data which can capture trends at varying levels of granularity [11]. The performance efficiency of Wavelet Transforms, is logarithmic, in time and space.

When a signal x is passed through low pass filters (scaling functions) and high pass filters (wavelet functions) simultaneously, it is defined as performing the discrete wavelet transform (DWT).

The number of coefficients generated will be half the length of the original input to each filter, when a signal is passed simultaneously through low pass and high pass filters and on performing subsequent down-sampling

(Figure 2).

Wavelet Transform ω_T can also be represented as in (1),

$$\omega T = X.W, \textit{where } W = [\varphi(x); \psi(x)] \qquad (1)$$

Wavelets *function f (t)* consists of basic functions $\varphi(x)$ & $\psi(x)$ as in (2)

$$f(t) = \varphi(x).\psi(x) \qquad (2)$$

Φ (x) is called scaling function to find the approximate coefficients and

Ψ (x) is called wavelet function to find the detail coefficients

Wavelet analysis or decomposition can be applied to any sequential data, including strings, where, in case of strings, the position of a character in string represents the time series data. Wavelets are tools used to study regularity and to conduct local studies [12]. The zero moments [13] of the function are related to the regularity of scaling function & wavelets [14].

3. PERFORMANCE EVALUATION & DISCUSSION

The algorithm proposed is tested on the Short Reads Archive (SRA) data. The SRA data is downloaded from NCBI site ftp://ftp.ncbi.nlm.nih.gov/sra/. The SRA data used for experiment contain short reads obtained from Roche's 454 sequencer. The SRA data was either generated from GS20 or GS FLX Standard or GS FLX Titanium sequencer.

The SRA data primarily include metagenomics data. Identification of duplicate reads from metagenomics data, is more relevant, to avoid any type of bias in analysis, when used in further analysis.

The experiments did not apply any Filter to the SRA data. No filtering on the basis of read length or any other criteria was applied on the reads. The results generated are from the complete set of data, without any type of read removals.

The experiments were executed on a regular laptop with Intel Core2 Duo CPU, T6500 @ 2.10GHz 2.10 GHz processor and 32-bit Windows 7 Ultimate Operating System with 4.00GB RAM using MATLAB R2009a version 7.0.8

The results of identifying duplicate reads for SRA data have been displayed taking several aspects into consideration.

The Table 1 and Table 2 display the results of artificial duplicate reads with 100% identities recognized using Wavelet Transforms. The tables represent the output in terms of reads as well as nucleotide base pairs.

The Total number of reads identified in a given file with various SRA Accession numbers, (see column (b) of Table 1) whereas the total number of duplicate reads (see column (c) of Table 1). The Table 1 also identifies the redundant reads which should be removed before any further analysis of these reads is done. The redundant reads are identified by preserving one copy of read from all sets of duplicates, so as to avail proper mapping with the reference and avoiding any loss of data. The percentage of duplicate reads (see column (d), Table 1) from the experimental data ranges from 2% to 22%. The percentage of duplicates found for a given genome does not reflect any direct co-relation with the size of the genome. The percentage of redundant reads is also substantial in some genomes. It is appropriate to remove these substantial percentage of redundant reads, which otherwise, would create bias in analysis, and take more processing time during annotation. Thus, identification of duplicate reads is an essential pre-processing task in annotation. The entire set of information (see Table 1), is generated from sequence data after applying Wavelet Transforms. The study was conducted on a reduced data acquired after transforming the original nucleotide sequences, using multi level Haar Wavelet Transforms. The results generated using this algorithm does not apply any kind of string comparision through use of suffix arrays [16] nor does it use FM-Index [Ferraginna 2004] for indexing the strings and also no succinct data structures like Wavelet Trees [Oguzhan]. The algorithm uses a compressed form of the original nucleotide sequence, by making use of Haar Wavelets. The sequence is transformed by passing it through Low-Pass and High Pass Filters and further doing down-sampling. Thus, the algorithm makes use of signal processing for identifying duplicate reads from reduced data.

The result of experiments conducted using Wavelet Transforms for data reduction (see Table 1) is comparable to the result found using cdhit-454 algorithm used on datasets with SRA reference no.SRA001669, SRA001663, SRA000674 (here SRA000675), SRA000905 (here SRA000906) as in [4]. Actually results generated using Wavelet Transforms based algorithm seems to be more stringent. Higher percentage of duplicate reads obtained through proposed algorithm compared to cdhit-454 algorithm, reflects it.

The duplicate reads in metagenomics range from 2% to 22% (see Table 1). This result is also comparable to the results generated by *Niu B. et al., 2010*[4]

Table 2 represents the total number of redundant bases and percentage of redundancy. The Total No. of Redundant Bases (see column (e) of Table 2). The table does not display the Length of each Read, which is used in calculating column (f).

Table 3 represents additional information to Table 2. It represents the redundancy in terms of number of bytes, which is essential from storage aspect. Here the redundant bytes are calculated considering that each base requires 1 byte of storage space.

Table1.Result Representing Number and Percentage of Artificial Duplicate Reads with 100% identities, Recognized Using the Wavelet Transforms based Algorithm

SRA Accession No.	Total No. of Reads	Total No. of Copies of Duplicate Reads	Total Percent-age of Duplicate Reads	Total No. Unique Reads within Duplicate Reads	Total No. of Redundant Reads	Total Percentage of Redundant Reads
a	b	c	d	e	(c-e)	((c-e) * 100) / b
SRR000907	5133	124	02.4157	61	63	01.2274
SRR001669	41649	3362	08.072	1486	1876	04.504
SRR001670	55292	12431	22.482	4831	7600	13.746
SRR000675	142545	5210	03.655	2588	2622	01.8394
SRR077225	25698	127	00.494	63	64	00.249
SRR000906	200557	12127	06.0467	5739	6388	03.1851
SRR001663	369811	64164	17.3505	25434	38730	10.4729
SRR065619	3404	410	12.045	156	254	07.462
SRR052290	74076	12634	17.055	4291	8343	11.263

Table 2 - Representing details of Redundant Reads (after retaining one copy each) in terms of Number and Percentage of Base Pairs.

SRA Accession No.	Total No. of Reads	Total No. of Bases	Total No. of Redundant Reads	Total No. Redundant Bases	Total Percentage of Redundant Bases
a	b	c	d	e	(e*100) / d
SRR000907	5133	583662	63	6889	1.1803
SRR001669	41649	4422480	1876	194779	4.4043%
SRR001670	55292	5856963	7600	789129	13.47335%
SRR000675	142545	30932262	2622	423206	1.3682
SRR077225	25698	13166319	64	26939	0.2046%
SRR000906	200557	22735927	6388	723900	3.1839
SRR001663	369811	39152427	38730	4187877	10.6963
SRR065619	3404	1081048	254	37937	3.5093%
SRR052290	74076	22356978	8343	1211079	5.4170

Table 3.Representing byte conservation, using proposed algorithm

SRA Accession No.	Total No. of Reads	Total No. of Duplicate Reads	Total No. Unique Reads within Duplicate Reads	Total No. of Redundant Reads	Total No. Redundant Bases	Total Bytes which can be conserved by removing Duplicate Reads
A	b	c	D	e=c-d	f= (e) * Length of each Redundant Read	g
SRR000907	5133	124	61	63	6889	6889
SRR001669	41649	3362	1486	1876	194779	194779
SRR001670	55292	12431	4831	7600	789129	789129
SRR000675	142545	5210	2588	2622	423206	423206
SRR077225	25698	127	63	64	26939	26939
SRR000906	200557	12127	5739	6388	723900	723900
SRR001663	369811	64164	25434	38730	4187877	4187877
SRR065619	3404	410	156	254	37937	37937
SRR052290	74076	12634	4291	8343	1211079	1211079

Table 4.Representing the SRA Read No., with highest number of duplicate reads in a given SRA Read, along with Length and Maximum number of copies (SRA No. of read as specified in column (d))

SRA Accession No.	**Total No. of Reads**	**Total No. of Duplicate Reads**	**SRA Read No. of Read with Max No. of Copies (Refers to the 1st occurrence of the duplicate read)**	**Length of a Read with Max No. of Copies (bp)**	**Max No. of Copies of a given Read (units)**
a	**b**	**c**	**d**	**e**	**f**
SRR000907	5133	124	SRR000907.865.2	106	3
SRR001669	41649	3362	SRR001669.29.2	98	18
SRR001670	55292	12431	SRR001670.7715.2	103	19
SRR000675	142545	5210	SRR000675.3409.2	143	5
SRR077225	25698	127	SRR077225.1404.3	436	3
SRR000906	200557	12127	SRR000906.3468.2	95	7
SRR001663	369811	64164	SRR001663.17896.2	124	48
SRR065619	3404	410	SRR065619.64.3	8	18
SRR052290	74076	12634	SRR052290.3.3	5	394

Table 5.Representing Computational Time for Identifying Artificial Duplicate Reads with 100% identities

SRA Accession No.	**Total No. of Reads**	**Total No. of Duplicate Reads**	**Computational Time (in secs)**
a	**B**	**c**	**d**
SRR000907	5133	124	34.6411
SRR001669	41649	3362	1440.2
SRR001670	55292	12431	2900
SRR000675	142545	5210	7815.9
SRR077225	25698	127	453.8735
SRR000906	200557	12127	32447
SRR001663	369811	64164	112910
SRR065619	3404	410	15.0854
SRR052290	74076	12634	1933.6

Table5. Represents the time of computation involving all phases of computation (see Figure 1), that includes both, the phase of Wavelet Transforms of the sequences as well as identifying the duplicate reads (third and fourth phase of algorithm, see Figure 1) by performing naïve comparisions of Wavelet transformed sequences. The actual time involved in performing Wavelet Transforms is much less (see Table 6). If observed minutely, it can be concluded that the Wavelet Transforms takes almost same or sometimes more time, if the number of total reads is approximately 5000. But, if the number of reads is large i.e. in few hundred thousand then, time involved in

Wavelet Transformation is comparatively very less. This observation is an important note because most of biological data, particularly eukaryotes, is few billions large. (The discussion is only an observation. Statistical Information for larger data is not shown here). Algorithms with usage of suffix arrays [15]; [16], entropy-compressed indexes [16], FM-index [17] or even wavelet trees [18] take very large time for just creating the compressed data structure, besides computation time for finding the duplicates, which probably is much less in case of the proposed algorithm of using Wavelet Transforms (see Table 6).

Table 6. Result displaying Computational Time for Identifying Artificial Duplicate Reads with 100% identities Recognized Using the Wavelet Transforms based Algorithm used for Data Reduction, prior to comparing the sequences for finding exact or near exact match.

SRA Accession No.	**Total No. of Reads**	**Total No. of Duplicate Reads**	**Time Taken To Perform Four Level Wavelet Transforms (in secs)**	**Time Taken To Recognize Duplicate Reads (in secs)**	**Total Computational Time (in secs) inclusive of generating other statistical information**
a	**b**	**c**	**D**	**e**	**f**
SRR000907	5133	124	17.5457	17.0194	34.6411
SRR001669	41649	3362	181.9955	1257.5	1440.2
SRR001670	55292	12431	263.3025	2899.5	3163.4769
SRR000675	142545	5210	1016.9	6797	7815.9
SRR077225	25698	127	111.2499	300.9164	453.8735
SRR000906	200557	12127	1681.6	30761	32447
SRR001663	369811	64164	4679.3	108220	112910
SRR065619	3404	410	11.956	3.8123	15.9854
SRR052290	74076	12634	412.2294	1517.8	1933.6

The computational time was calculated based on experiment conducted on regular laptop with Intel Core2 Duo CPU, T6500 @ 2.10 GHz x 2.10 GHz processor and 32-bit Windows 7 Ultimate Operating System with 4.00GB RAM using MATLAB R2009a version 7.0.8.

Thus, the given algorithm can be smoothly executed on a simple desktop machine too. The computational time represented in Table 5, can be optimized to a great extent, if special purpose computers are used instead of shared resources.

4. ENHANCEMENTS

The computational time involved in recognizing artificial duplicate reads after applying data reduction using Wavelet Transforms, can further be reduced, if the better and more optimal search algorithm for comparing two wavelet transformed sequences, can be applied. Further, if the concept of scoring matrix is applied, for each compared co-efficient, we can use the Wavelet Transformed signal, for sequence alignment also. The time involved in comparision can also be achieved if the algorithm can be distributed across several processors through distributed computing.

5. RESULT

The application of Wavelet Transform reduces the data for comparision to *one-eighth* of the original data size and hence reduces the number of comparision to be performed for searching for duplicates. The algorithm optimizes time and memory requirement and hence can be executed on an ordinary personal computer or laptop also.

The algorithm is tested on metagenomics data and results are concluded in the paper.

6. CONCLUSION

Wavelet Transforms can be applied to recognize 100% identities of duplicate reads. The order of complexity is *O(log n),* which is much less to linear or exponential complexity when using other string comparision based algorithms. Hence, wavelet transform based algorithm can be applied to optimize time and memory requirement.

ACKNOWLEDGEMENTS

The author would like to thank Prof. B.S. Parekh, Department of Computer Science and Engineering, The M.S. University of Baroda, Dr. D.P. Patel, Department of Applied Mathematics, The M.S. University of Baroda and Dr.C.G.Joshi and Dr. P.G. Koringa, Ome Research Centre, Anand Agriculture University, Anand, Gujarat, India, for providing the resources and guidance during the study.

REFERENCES

[1] Ronaghi Mostafa 2001 Pyrosequencing Sheds Light on DNA Sequencing Genome Res. 11 (2001) 3-11

[2] Huse, S.M. et al. 2007 Accuracy and quality of massively parallel DNA pyrosequencing, Genome Biology, 8 (2007) R143

[3] Dong, H. et al. 2011 Artificial duplicate reads in sequencing data of 454 Genome Sequencer FLX System Acta Biochim Biophys Sin 43 (2011) Issue6: 496–500

[4] Niu, B. et al., 2010 RAesretaricfhi caritiaclel and natural duplicates in pyrosequencing reads of metagenomic data, BMC Bioinformatics, 11 (2010), 187

[5] Gomez-Alvarez V, Teal T K, Schmidt T M, 2009 Systematic artifacts in metagenomes from complex microbial communities. ISME J, 3 (2009) 1314–1317

[6] Nair A S, Sreenathan S P, 2006 A Coding Measure Scheme Employing Electron-Ion Interaction Pseudo potential (EIIP), Bioinformation, Vol. 1, No. 6, (2006), pp. 197-202.

[7] Radomir S. et al., 2003 The Haar wavelet transform: its status and achievements, Elsevier Science Ltd. Computers and Electrical Engineering, 29 (2003) 25–44

[8] Ruiz, G, Michell JA, Buron A, 1992 Switch-level fault detection and diagnosis environment for MOS digital circuits using spectral techniques, IEEE Proc Part E, 139(4) (1992) 293–307

[9] Meher, J. K. et al. 2012 Wavelet Based Lossless DNA Sequence Compression for Faster Detection of Eukaryotic Protein Coding Regions , I.J. Image, Graphics and Signal Processing 7 (2012 , 47-53

[10] Daubechies, I. 1992 Ten Lectures on Wavelets

[11] Aggarwal, C. C. 2005 on the Use of Wavelet Decomposition for String Classification, Springer - Data Mining and Knowledge Discovery, 10 (2005) 117–139

[12] MATLAB manual: http://www.mathworks.in/products/wavelet

[13] Zhang, B. 2006 Fast Poisson Noise Removal by Biorthogonal Haar Domain Hypothesis Testing Elsevier

[14] Burrus, C.S. et al. 1998 Introduction to Wavelets and Wavelet Transforms, Prentice Hall

[15] S. Kurtz et al.1999 REputer : fast computation of Maximal Repeats in Genome sequences, BioInformatics Applications Note, Vo. 15. No 5, (1999) Pg 426-427

[16] Grossi, R. et al., 2005Compressed suffix arrays and suffix trees with applications to text indexing and string matching. SIAM Journal on Computing, 35(32) (2005) 378–407, 2005.

[17] Ferragina, P. et al., 2004 An alphabet friendly FM-index. In International Symposium on String Processing and Information Retrieval (SPIRE), (2004) pages 150–160

[18] O˘guzhan M, Vitter J S, Xu B, Efficient Maximal Repeat Finding Using the Burrows-Wheeler Transform and Wavelet Tree, IEEE/ACM Transactions On Computational Biology And Bioinformatics.

Distributed Computing for Structured Storage, Retrieval and Processing of DNA Sequencing Data

Mamta C. Padole
Department of Computer Science and Engineering
The Maharaja Sayajirao University of Baroda
Baroda, India
Email:mpadole29@rediffmail.com

ABSTRACT

The technologically break-through in DNA sequencing process, and the current trend of DNA re-sequencing, both, generate high-throughput data, also referred to as Big Data. This Big Data demands increased computation power and storage space. Hence, the big data needs to be stored, retrieved and processed efficiently. The paper discusses the use of Distributed Application developed using Client-Server model and M-V-C Architecture for implementing distributed computing for efficient and secured storage, optimized and customized retrieval of Big Data, generated through DNA sequencing. The paper proposes the use of web application for data retrieval and storage, which provides total transparency of where and how the execution of various tasks is happening. The data and the business logic are independent components located on separate computers and are loosely coupled. Data dictionary designed, for storing the DNA sequencing reads generated using 454 Roche's GSFLX system and the results of analysis based on this DNA sequenced reads like contigs, repeats, coding regions, genes etc, is also discussed in brief.

Keywords –Distributed Application, Web Application, Client Server Architecture, M-V-C Architecture, Big data

1. INTRODUCTION

The ultra-high throughput data generated at an accelerated rate through next generation DNA sequencing [1] techniques need secured storage, speedy retrieval, quick and efficient processing, to gain genetic knowledge at appropriate and requisite time. Cost effectiveness and technical viability of high-performance, high-throughput computer in an organization, where both the data storage and processing is happening on a single machine, are often the matters of concern. To such queries, distributed computing is the best alternative. Hence, a distributed web applications are developed, where the data storage and business logic required for querying the stored data, are, located on different machines. Primary purpose of improving efficiency of storage, processing and analysis of DNA sequences is, to get timely genetic information of an individual or organism, because, their major application is for identification and treatment of diseases like cancer and HIV, drug designing, controlling spread of epidemics by identifying micro-organisms involved in disease causing and prohibiting their growth, enhancing crop breeds, to meet the drought situation, improving animal breeds that provide food, nourishment, medicines or income; for phylogeny identification and in forensics.

Globally accessible, centralized repository of NCBI (National Centre for Biotechnology Information) [2] is available for storage (albeit, physically they are scattered like Swiss-Prot or DDBJ. Centralized storage is only a logical view that presents transparency to user) and processing of genomic data. But, at times, it is not possible to upload and disclose the confidential data of some genomes into the centralized repository of NCBI, particularly, when the project is sponsored and controlled under some specified guidelines and contract. Although, one can upload data and keep it private in NCBI, but that is permissible, only till the manuscript describing the data gets published. If the sequencing data is uploaded at NCBI, eventually, it becomes an open data in public domain.

To avoid this disclosing, but at the same time use all the web based facilities for storage, retrieval and customized algorithms for processing of DNA sequencing data within an organization, a distributed application has been developed, which is discussed in this paper.

Moreover, in a spatially located organization or research organization where the data generation experiments like DNA sequencing are conducted at scattered locations or laboratories with the need to exchange data between various locations, it is advisable to maintain the central repository (at least, the common interface, which provides the single point of access to distributed data). This central repository maximizes of utilization of DNA data, at the same time relieves the biological researcher from obtaining the knowledge and responsibility of storage or security of data. The backup and recovery activity, botheration of security or availability of data, then becomes the responsibility of database administrator. Also, the biological researcher need not get involved in knowledge of computer science, in particular, database administration or network administration, and hence can concentrate in his actual research domain, which usually would be biology related.

The paper discusses development of an application for DNA sequencing data, based on distributed computing concepts, which initiates right, from uploading of the raw data to database and files servers, through web, retrieving that data when essential and query processing.

The web application provides data abstraction for the data stored in remote databases. Transparency of data location and processing is maintained. Open standards for loose coupling between components like business logic and data, facilitates data exchange, irrespective of data formats. Heterogeneous resources like file systems, database management systems and network protocols are also an important characteristic of the web application. The thin-client concept has been another approach to web application development. Pull technology is the core facet, but for intimation purpose, push technology is also implemented.

The entire approach of distribution in the application and various issues tackled for different purposes has been discussed in this paper.

2. DISTRIBUTED COMPUTING AND ISSUES

Distributed computing deals with all aspects of information access and computing across multiple processing elements. These processing elements may be connected by any form of communication network either local or wide-area, in coverage [3]. Distributed Computing is defined as computing or performing the processing using spatially distributed systems. Thus, Distributed Computing makes use of Distributed system. Distributed system is "A collection of independent computers that appear to users of the system as a single computer" [4]. Thus, distributed computing is any computing that involves multiple computers remote from each other, where each have a role in a computation problem or information processing. Distributed processing also includes the operations that occur on a database application which typically distributes front-end presentation tasks to client computers and allows a back-end database server to manage shared access to a database. Consequently, a database application processing system that is more commonly referred to as a client-server database application system is also a distributed application [5].

A. The features to be dealt with in distributed computing include: [6]
   - Transparency of distribution from an end user
   - Loose-coupling of systems
   - Heterogeneous computing
   - Scalability i.e. increasing the computational capacity
   - Fault-tolerance or handling of failure
   - Concurrency control

B. The motivation for implementing distributed computing takes into account:
   - Possibility or need for inherent distributed computations, when designing applications
   - Requirement of resource sharing
   - Security of data
   - Accessibility to remote data or remote resources
   - Availability of resources
   - Need of scalability
   - Purpose to improve performance to cost ratio

C. To implement the distributed system various models of distribution and architecture available are:
   - Client Server model
   - Processor Pool model
   - N-tier architecture
   - M-V-C architecture

D. The issues that need to be tackled during development and implementation of distributed system are :
   - Transparency to the user about resource location and processing environment
   - Resource availability on user requests
   - Appropriate controls for accessibility of resources
   - Heterogeneity of processing environments in terms of hardware architectures, operating systems, database servers, file server etc.
   - Scalability of data and resources
   - Coupling of components between clients and business logic
   - Granularity of data and code hiding
   - Protocols to be used in terms of proprietary, open or standard protocols
   - Data and information storage in terms of flat, structured or semi-structured form.
   - Data formats which are customized to applications
   - Development methodologies in terms of sequential, structured, object oriented, component based or service oriented development
   - Technologies and standards for Middleware to be used like Remote Procedure Calls (RPC), Java RMI, COM/DCOM, MPI interface, .Net Remoting etc.
   - Communication and scheduling platforms available like Matlab, Globus, Hadoop, Condor etc.

3. METHODOLOGY AND IMPLEMENTATION

Distributed Application which has been designed and developed comprises of two parts. The first portion consists of a web application for storage and retrieval of DNA sequencing data. The web application uses the Client-Server model as shown in Fig. 1, with M-V-C architecture as a design pattern [7]. The application can be used to upload the FASTA files containing DNA sequencing data to the Application Server. The privileged user can then upload the data file to Database Server, which provides the abstraction of central repository of the data with secured storage.

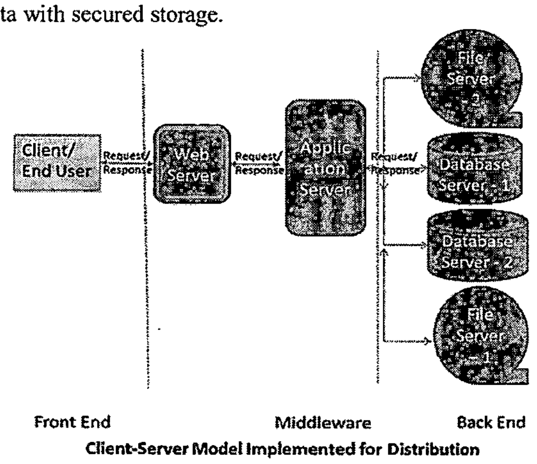


Fig. 1. Client Server Model Implemented for Distributed Application

In implementation as shown in Fig. 2., there are multiple database servers which store the data in duplicate, which guarantees data availability. The centralized File-Server is also configured as a part of the distributed system, that allows storage of original files as generated by DNA sequencing equipments. The application allows retrieving the data back from the File Server in unstructured form. The database server provides facility to store the data in the structured form with relational integrity. The relational integrity enables the user to acquire the information in terms of joins and sub-queries, thus, fetching more than just a raw data as was available in a FASTA file. The execution of SQL queries is done for a request received across the Web. The POJO's are written to read the raw FASTA files using org.biojavax [8] and java.util.regex API's, to convert the data, so that they can be appropriately stored as tuples in database. The POJO's are also developed using Java RMI, to transfer the data to remote File Server and back, when needed. The end-client can request to download the data stored in the database in form of web-page or can request to download the FASTA file from file server. The data can be requested using various criteria like a particular species, or sequencing run, or reads of a specific length. The table storing the reads generated from DNA sequencing is stored organism-wise and for each organism or species, many sequencing runs may be executed. The reads table maintains records along with this species and run information. The database has also been designed to store analysis data generated after analyzing these raw DNA reads. The tables for storing contigs, repeats, coding regions and genes are created. Data can be stored in these tables and retrieved across the web. Table 1, 2 and 3, displays the data dictionary for storing this data. (This is just the very minimal set of data dictionary used in storing various DNA sequencing and analysis data. The complete list is not shown due to lack of space.) Multithreading [9] is used for handling multiple requests from various clients.

Table 1. Table structure for storing the species details, for the species or organisms, whose DNA sequencing is performed

Column Name	Type	Constraints	Description
Species_id	Integer(10)	Primary key	Id of a species
Species_name	Varchar(100)	Not Null	Name of species

Table 2. Partial snapshot of the table structure for storing the details of the runs or DNA sequencing experiments that are performed for the species

Column Name	Type	Constraints	Description
Species_id	Integer(10)	Foreign key	Id of a species
Run_id	Integer(2)		Id of run (Number of runs executed for the given species)
Dt_of_run	Date	Not Null	The Date on which run was executed

Table 3.Table structure for storing the details of the reads generated from DNA sequencing of the given species and a given run.

Column Name	Type	Constraints	Description
Id	Integer(10)	Auto generated	Id of the read
Species_id	Integer(10)	Foreign Key	Id of a species
Run_id	Integer(2)	Foreign Key	Id of Number of runs
Read_id	Integer(10)		Id of a each read
Read_name	Varchar(100)	Not Null	Id generated by the DNA sequencer for each read
Read_rank	Integer(10)		Rank of each read
Read_x	Decimal(10,3)	Not Null	X attribute of each read
Read_y	Decimal(10,3)	Not Null	Y attribute of each read
Read_length	Integer(6)	Not Null	Length of each read
Read_sequence	Clob	Not Null	Sequence of each read
File_name	Varchar(100)		Name of the file

The features of Distributed Application are:

- Web-based application – The software for Distributed application can be accessed using any Web-browser. The application has been tested using Internet Explorer and Mozilla Firefox
- Open Standard – XML open standard is used for parameterization and flexibility of data exchange between loosely coupled components, separation of business-logic layer from data-layer [10].
- M-V-C Architecture – The software has been developed using Model-View-Controller design pattern

- Distributed approach - Application has been developed and tested such that all the components i.e. Web-Server, Database-Server, File-Server and client are placed on different computers of a network.
- Transparency - The end-user of the system i.e. the client, submits the request to the Web-server through web-browser. All the back-end distributed usage and execution is unknown to the client. Hence, transparency feature of distributed computing is implemented.

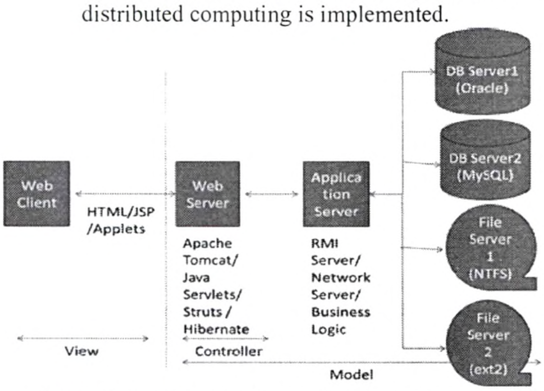


Fig. 2. Technologies used to implement the Distributed Application

- The distribution of component has loose-coupling of systems, heterogeneous computing capability and fault-tolerance or handling of failure.
- Thin client application – The application developed does not require any programs/software to be installed on the client machine, for executing the application. Only a request through web-browser is submitted using a URL of service running on the web server.
- Multiple Client Accessibility – Many clients can access the Distributed Application concurrently. This facility is provided by implementing the application in multithreaded Web-Container environment. The requesting client may be on same LAN or can access the system through Internet. Multithreading also facilitates inter-process communication, synchronization and mutual-exclusion
- Database Management System Independent – The data storage can be done on any Database. It has been tested to work on Oracle 10g and MySQL 5.5 Database Server.
- Availability of data is assured through data duplication at multiple database servers
- Operating System Independent – Can be executed on any Operating System. It has been tested on Windows and Linux operating systems.
- File System Independent - The File-Server has been developed for centralized storage of DNA sequencing data files. The File-Server can be implemented on FAT 32, NTFS and ext3 file system.
- Location Independent – The application being Web-based is accessible from any location on the local area network or internet from wherever the URL is accessible with appropriate security privileges provided.
- Open Source – All the tools & technology used in developing the Distributed Application like Java Development Kit, Apache Tomcat Web Server, MySQL Database Server, Struts Web Framework, Hibernate ORM tool, Linux OS are Open Source under General Public License. (Although testing is done on proprietary or licensed software also)
- The Distributed application uses Java POJO's for business logic.
- Apache-Tomcat Web Server [11] for managing web application and services.
- Java Servlets on the Server side as a controller, JSP as a view.
- Struts Web Framework [12] is used to implement the M-V-C architecture.
- Hibernate is used for Object-Relational Mapping [13].
- Spiral Model was adopted as a system development life cycle model, which is an iterative model.

The facilities provided by the Distributed Application for BioInformatics data are:

- Uploading of Data Files which are in FASTA format as shown in Fig. 3

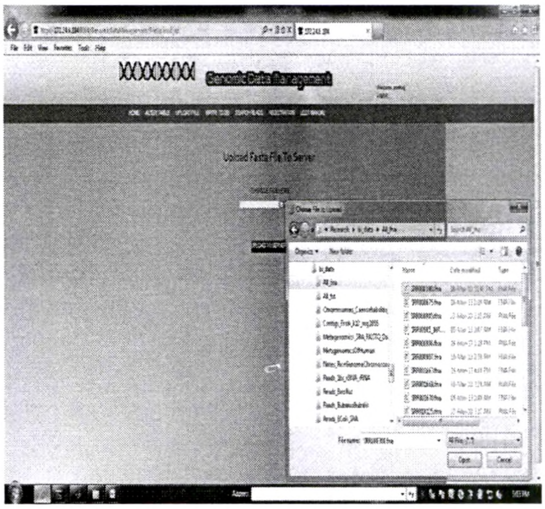


Fig. 3. User Interface with the Distributed Web Application, to Upload the FASTA file containing DNA sequencing data.

- Storage of Data Files on central Database Server
- Storage of Data Files on central File Server
- Duplication of data on multiple File Servers and Database servers simultaneously for data duplication and availability

- Search of DNA data on various criteria
- Possibility of firing customized queries or native SQL
- Download the search result in Web-Page or in FASTA file as shown in Fig. 4
- Download the original data file, if required for further processing
- Store DNA analysis data in structured form into database server like contigs, repeats, coding regions, genes etc.
- Data sharing within the organization
- Web accessibility
- User Registration for open accessibility
- User-wise accessibility for security/privacy of data
- The data stored in the file-server can later on be read by Matlab software [14] and used for further analysis, like finding the duplicate reads using signal processing approach [15]. This is the later part of the distributed application, which uses processor-pool model for processing the data.

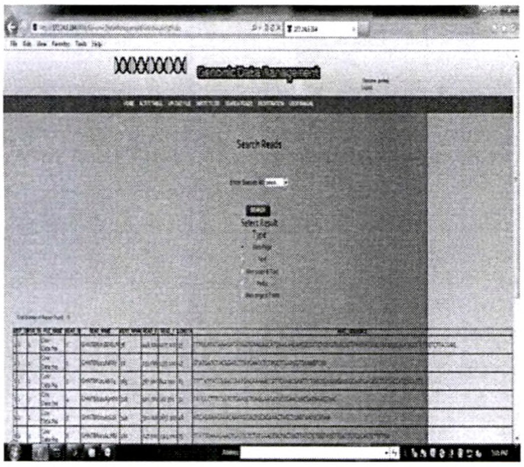


Fig. 4. Web Page displaying the result of the query fired on the database residing on the remote machine.

The implementation issues tackled during the development of the Distributed Application are:

- Making use of Open Standards, so that appropriate collaboration can take place between the distributed components.
- Tackling with transparency, heterogeneity and failure handling, through appropriate message passing and exception handling concept implementation.
- Identifying and Designing the components of a Distributed Application, so that they could be appropriately placed/ distributed at different locations/ computers on a network.
- Developing all the collaborating interfaces, so that each component can communicate with each other as required.
- Installing and Configuring the Web-Server, the Database Server, the File Server for various facilities on separate machines.
- Providing appropriate privileges across all the components and computers to make the application working
- Dealing with Firewalls implemented on different computers of the network
- Writing POJO's using regular expressions, to read and extract the information from the data files. This was important because data files are unique in format for the biological data. These programs were essential for converting the unstructured data into objects, which, could then be stored in a structured format in the database using Object-Relational Mapping.

4. ENHANCEMENTS

The extension to this work is the use of data files stored on File Servers, to do processing using distributed processor pool model. The distributed processing is applied using MATLAB Distributed Computing Toolbox and Parallel Computing Toolbox. The application used is to recognize duplicate reads from DNA sequencing data, using Wavelet Transforms. The work is in progress, and the comparative results are yet to be concluded. Hence, the second part of this Distributed Application will be communicated in separate paper.

5. CONCLUSION

The use of web-based application using client-server model and M-V-c architecture has proved to be an efficient way of handling the large data as that of DNA sequencing data. A single high-throughput system, required the use of proprietary software for each task. Also, upgrading the system was cost incurring. The use of distributed concept has been beneficial from scalability aspect as well as utilized the processing power of several personal computers. Of, course, from security aspect, the database server that was created was a dedicated server.

REFERENCES

[1] Ronaghi Mostafa 2001 Pyrosequencing Sheds Light on DNA Sequencing Genome Res. 11 (2001) 3-11

[2] http://ncbi.nlm.nih.gov/

[3] A.D. Kshemkalyani et. al., Distributed Computing : Principles, Algorithms and Systems, UK, CUP, 2008.

[4] Andrew S. Tanenbaum et. al., Distributed Systems: Principles and Paradigms, NJ, Prentice-Hall, 2003.

[5] http://www.oracle.com

[6] G. Coulouris, J. Dollimore, T. Kindberg, Distributed Systems – Concepts and Design, Pearson Education, Asia, 2001.

[7] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns:Elements of Reusable

Object-Oriented Software, Addison Wesley, USA, 1994

[8] http://www.biojava.org

[9] Guy Steele, Gilad Bracha, Bill Joy, James Gosling, The Java Language Specifications, SunMicrosystems.

[10] J. McGovern, S.Tyagi, M. E. Stevens, S.Mathew, Java Web Services Architecture, Morgan Kauffman Publishers, An Imprint of Elsevier, 2005.

[11] http://www.apache.org

[12] http://www.struts.apache.org

[13] http://www.hibernate.org

[14] www.mathworks.com/help/matlab/

[15] M. C. Padole, "Recognizing Artificial Duplicate Reads in 454 Pyrosequencing Using Wavelet Transforms" International Journal of Advanced Computing, ISSN:2051-0845, Vol.46, Issue.2, pp 1205-1211, Recent Science Publications, May 2013. 27702498.

Dimensionality Reduction of DNA Sequences through Wavelet Transforms

MAMTA C. PADOLE
Department of Computer Science and Engineering,
The Maharaja Sayajirao University of Baroda
Kalabhavan, Baroda,
Gujarat, India – 390 001
mpadole29@rediffmail.com

Abstract: - In recent years, development in molecular biology and sequencing technology has led to large scale sequencing of genomes, metagenomics and transcriptome. As a result sequencing data is growing exponentially. This mounting data needs secured storage and frequent analysis to resolve several mysteries of nature. Although, cost of storage capacities in computers is reducing and processing power enhancing with time, there still arises the need of storing *Big* data efficiently and applying various analysis techniques optimally. This paper discusses, the use of Haar Wavelet Transforms, *a signal processing approach,* for lossless data compaction for Genomic data. The use of Wavelet Transforms also guarantees perfect reconstruction. The advantage of Wavelet Transform is its computational complexity being *O(log n)* and is memory efficient as it can be computed in place, without a temporary buffer.

Key-Words: - *Wavelet Transforms, Signal Processing, Haar Wavelets, Dimensionality Reduction, Data Compaction*

1 Introduction

High throughput sequencing techniques has led to large scale sequencing of genomes, metagenomics and transcriptome. As a result sequencing data is growing exponentially. This mounting data causes lot of challenges to Computer Scientists to store this data efficiently and analyse it optimally.

Applying various techniques for storage or analysis of raw data in terms of DNA sequences or protein sequences, involves string processing or textual processing, sometimes along with different data structures for string processing. String processing algorithms are time consuming and usually involve linear or exponential computational complexity.

The existing compression tools are primarily based on the classic Lempel-Ziv algorithm [1], the bwt (Burrows-Wheeler transform) [2], Huffman Coding [3], FM-Index [4] and Wavelet Trees [5]

These algorithms or tools for DNA compression take statistical or dictionary based approaches. Statistical based algorithms need prior knowledge of the data in concern and hence are data dependent. “formatdb” [6] and GRS [7] uses Huffman coding, but deals efficiently only with minor symbols [8]. The Huffman code is an optimal prefix code in the case where exact symbols probabilities are known in advance and are integral powers of 1/2 [9][10]. The algorithms based on Arithmetic coders are slow in decompression process [8]. Arithmetic Coders and Huffman Coding use probability based compression which is difficult to predict, as DNA sequences comprises of nucleotide bases belonging to an alphabet size of 4, and hence each nucleotide base has almost similar probability [11]. The other algorithms applied in tools like GSR [12] or BioCompress [13] take into account characteristics of DNA like reverse complement or point mutation or characteristic structures of sequences like SNP’s or repeat regions [14]. But SNP map may not be available, SNP may not be the common bi-allele [15], Index of repetitiveness may vary in different genomes [16], or organisms have very little variation and hence may become difficult to compress. If, a *priori* knowledge of the characteristics of the given DNA sequence is not available, then statistical approach becomes irrelevant and hence the problem of data compression becomes difficult [1]. Reference sequence is used for data compression of DNA [17]. Another algorithm used in software tool ‘coil’ [18] uses Levenstein distances and encoding trees for compressing entire database of DNA data, but takes into account an initial DNA sequence to compare length-k substrings amongst all the other DNA sequences in database. This again is data dependent and will work efficiently only if all the sequences are nearly similar. For widely varied sequences in the database, the algorithm may prove inefficient [18] Techniques implemented on hardware based acceleration device, have also been used for data compression to accelerate DNA sequence alignment [19].

In this paper, we present an *ab initio* method of data transforms which can be applied for optimal analysis. This paper emphasizes on the use of Wavelet Transforms particularly the *Haar Wavelets* for data compaction of genomic data, principally the DNA sequences. After single level of Wavelet Transformation, there are always N elements, same as the original input sequence . But the fact is, that we do not need all N elements when we want to compare two sequences or need to perform sequence analysis. For local alignment, fine comparision is needed and this can be achieved through detailed co-efficient C_d. For global alignment of two DNA sequences, the coarse components i.e. approximate co-efficient C_a are enough. So, in all N/2 elements are sufficient for actual use. Thus, the paper proposes to reduce the number of elements used for further analysis, with an *apriori* approach. The content and positional information is preserved even after transform on which the analysis of data can be performed [20][21]. The time complexity of Wavelet transforms is *O(log n)*, n being the length of the input sequence. Also, use of Haar Wavelets is proved to be memory efficient.

2 WAVELET TRANSFORMS

A wavelet transform is a linear transformation of a signal or data into coefficients on a basis of wavelet functions [22]. The wavelet transformations represent the data in one domain into another, from where hidden information can be explored. Wavelet transform provides the time-scale information[23].

A signal is decomposed through Wavelet transforms into primarily two co-efficient vectors, which capture trends at different resolution levels [24]. These coefficient vectors of different resolution levels represent the characteristics of the data, at each different scale. Its performance efficiency is logarithmic, in time and space.

When a signal x is passed through low pass filters (scaling functions) and high pass filters (wavelet functions) simultaneously, it is defined as performing the discrete wavelet transform (DWT). The number of coefficients generated will be half the length of the original input to each filter, when a signal is passed simultaneously through low pass and high pass filters and on performing subsequent down-sampling. Therefore, the wavelet transform of a signal when passed through low pass filters and high pass filters generates the Approximate co-efficient C_a (1) and Detail co-efficient C_d (2), as represented in equations (1) and (2) [25].

$$Ca = ylow(\mathrm{n}) = \sum_{k=-\infty}^{\infty} \mathrm{x}[\mathrm{k}] \,.\mathrm{g}[2\mathrm{n}-\mathrm{k}] \quad (1)$$

$$Cd = yhigh(\mathrm{n}) = \sum_{k=-\infty}^{\infty} \mathrm{x}[\mathrm{k}] \,.\mathrm{h}[2\mathrm{n}-\mathrm{k}] \quad (2)$$

Here g is a scaling function and h is the wavelet function

The property of the wavelet transform is that the lower order coefficients of the decomposition represent the broad trends or coarse scales in the data, whereas the more localized trends or fine scales are captured by the higher order coefficients.

Filtering of the signal at each pass results in, a convolution of the signal x and the impulse response *g* of the filter. y(n) is the scalar product of functions x and g. Mathematically, this result can be represented as in (3):

$$\mathrm{y}(\mathrm{n}) = \sum_{k=-\infty}^{\infty} \mathrm{x}[\mathrm{k}] \,.\mathrm{g}[\mathrm{n}-\mathrm{k}] \quad (3)$$

Thus Wavelet Transform W_T can also be represented as in (4),

$$W_T = X.W, \quad where\ W = [\varphi(x); \psi(x)] \quad (4)$$

Wavelets *function f(t)* consists of scalar product of the basis functions $\varphi(x)$ and $\psi(x)$ as in (5)

$$f(t) = \varphi(x) \,.\psi(x) \quad (5)$$

$\varphi(x)$ is called scaling function to find the approximation and
$\psi(x)$ is called wavelet function that computes the details.

Wavelet decomposition can be applied to any sequential data, including strings, where the position of a character in string represents the time series data. Wavelets are tools used to study regularity and to conduct local studies [25]. The zero moments of the function are related to the regularity of scaling function and wavelets [26].

2.1 Haar Wavelets

The Haar transform is the simplest and oldest compact, dyadic, orthonormal wavelet transform [27]. The Haar function being an odd rectangular pulse pair, with compact support provides good possibility for local analysis of signal [28].

Haar wavelets are conceptually simple, fast and memory efficient [28], [29], can be computed in place, without a temporary buffer, are exactly reversible and can be perfectly reconstructed. The Haar transform decomposes a discrete signal x into two sub-signals of half its length. One sub-signal is a running average or trend (C_a) as in Table 1; the other sub-signal is a running difference or fluctuation (C_d).

2.1.1 Computation of Haar Wavelet Transform of Time Series Data

Consider a one-dimensional data vector X containing the N= 8 data values X = [18, 16, 6, 6, 12, 20, 4, 12}

Where, X is having number of elements n = 8, which is usually the power of 2. Haar Wavelet Transform of the signal X can be computed by iteratively performing pair-wise averaging and semi differencing [30], more precisely, convolving with the basis vector $< \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}} >$. As shown in Table 1, we can observe that, the Haar Transform W_T of the original signal X is given as:

$$W_T = [\ 47/\sqrt{2}, -1/\sqrt{2}, 11, 8, 1/\sqrt{2}, 0, -8/\sqrt{2}, -8/\sqrt{2}]$$

TABLE 1. Representation of Computations of Haar Wavelet Transform

Transformation Level or Decomposition Level (j)	Scale (Scale $a = 2^j$)	Resolution (Resolution is inverse of Scale *i.e.* 1/a)	Length of Signal/ (L)	Averages / Approximate Co-efficient (C_a)	Differences / Detail Co-efficient (C_d) C_d =
				$C_a = (X_i + X_{i+1}) / \sqrt{2}$	$(X_i - X_{i+1}) / \sqrt{2}$
(a)	(b)	(c)	(d)	(e)	(f)
0 (Original Signal)	2^0 = 1	$1/2^0 = 1$	8	[18, 16, 6, 6, 12, 20, 4, 12]	-
1	2^1 = 2	$1/2^1$ = 1/2	4	[34/√2, 12/√2, 32/√2, 16/√2	[1/√2, 0, -8 / √2, -8/√2]
2	2^2 = 4	$1/2^2$= 1/4	2	[23, 24]	[11, 8]
3	2^3 = 8	$1/2^3$ = 1/8	1	[47/√2]	[−1/√2]

The Haar Wavelet Transform consists of basis functions ϕ and ψ. The scaling function ϕ determines the scale or dilation/expansion and mother wavelet function ψ determines the translation or shift. ϕ and ψ for Haar Wavelet Transform can be written as,

$$\phi(t) = h(0)\sqrt{2}\phi(2t) + h(1)\sqrt{2}\phi(2t-1) \quad (6)$$

$$\psi(t) = g(0)\sqrt{2}\phi(2t) - g(1)\sqrt{2}\phi(2t-1) \quad (7)$$

Where,
h(0), h(1) and g(0), g(1) are known as the low-pass and high-pass filters respectively.
These filters are used to convolve with the scaling function and mother wavelet functions respectively to generate the wavelet transform of a function.
In Haar Wavelet Transform, the values of these low-pass and high–pass filters are expressed as given in (8) to (11)

$$h(0) = 1/\sqrt{2} \quad (8)$$

$$h(1) = 1/\sqrt{2} \quad (9)$$

$$g(0) = 1/\sqrt{2} \quad (10)$$

$$g(1) = -1/\sqrt{2} \quad (11)$$

Where,
h(0) and g(0) are analysis filters and h(1) and g(1) are synthesis filters.

In other words, these filter values can also be represented in terms of basis vectors as $< \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} >$ and $< \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} >$.

Since, the basis vectors used in Haar Discrete Wavelet Transform are the smallest possible basis vectors; it is not possible to do Haar Transform in one-pass. Thus, it becomes essential to recursively transform the input signal using these basis vectors [31]. This concept is clearly shown inTable 1.

The scaling function in Haar Wavelet Transform is defined as in (12)

$$\emptyset(x) = \begin{cases} 1, if\ x\ \in [0,1) \\ 0, if\ x\ \notin [0,1) \end{cases} \qquad (12)$$

The mother wavelet function or translation function in Haar Wavelet Transform is defined as in (13), [23]

$$\psi(x) = \begin{cases} 1, if\ x\ \in [0, 0.5) \\ -1, if\ x\ \in [0.5, 1) \end{cases} \qquad (13)$$

3 METHODOLOGY

Algorithm:

1. Read Fasta File containing Pyrosequenced Reads.
2. Assign binary values to nucleotide bases, thus converting nucleotide sequence into single indicator sequence.
3. Combine four binary values and put it as a single byte. Thus compressing the sequence one-fourth of the original sequence.
4. Perform four levels Haar Wavelet Transform on Numerical Sequence, which in turn reduces the sequence by 2^4.
5. Perform Steps 2 to 4 for all sequences in the Fasta File.
6. If at each level of transform, repeating sequences are removed, it would give further reduction of data.

The suggested algorithm for creating a compressed form of a given sequence is applied as follows:

Read the Fasta file which contains the sequence $S_i = \{s_1, s_2, \dots s_n\}$ where s_i □ $\sum = \{A,C,G,T\}$, $i = 1$, n and n is the length of S_i.

Convert the nucleotide sequence S_i into its numerical representation X_i. The single indicator sequence using binary representation of {A, C, G, T} as {00, 01, 10, 11} is used. This binary representation of the nucleotide bases is than combined into a group of four and each of these four binary representations of the four nucleotide bases is than stored in a single byte. Thus, only one byte is used to represent four consecutive nucleotide bases. The use of binary representation in a single byte reduces the computational overhead to 25% compared, to the other representation of nucleotide sequences like dipole moments [22] or EIIP values [32]. Only numerical representations can be applied for wavelet transformations.

Thus, if a given nucleotide sequence is:

S_i = GTGCAGGATGCTGCAA $i = [1, n]$, n being length of nucleotide sequence

Then, its numerical representation i.e. a digital signal can be given as:

$X_j = 185, 40, 231, 144$, $j = [1, m]$, m being length of numerical representation of binary values, as a single byte for four nucleotides. $\therefore m = 1 \dots n/4$

Perform first-level decomposition of this signal, using Haar Wavelet Transform. This would reduce the signal into one-half of the original numerical representation of the nucleotide sequences. Thus 50% compaction is achieved by single level wavelet transform.

Further decomposition of a signal, to 2nd level, 3rd level and 4th level would further reduce the size of the signal by two.

Thus, applying repeated wavelet transforms reduces the signal size and hence requires, less storage space compared to storing the original nucleotide sequence. This reduced size would also reduce the time involved in time.

Also, Searching on the transformed signal is equivalent to searching on a reduced data set of the original sequence and hence is more efficient. If the correct mapping/transformation is applied, the Parseval Theorem implies that frequency distance FD is less than or equal to edit distance ED (Distance preserving transformation) [33].

With every transform, we are discarding half the values as per Nyquist's rule [22] and hence optimizing the search, with time complexity of *O(log n)*.

The Haar wavelet transforms Ca i.e. running average represents the trend; the other sub-signal, Cd a running difference, represents fluctuations [34]. Hence, Haar wavelet, can be used to transform the signals for data analysis too, besides applying it for compaction.

Thus, use of this algorithm does compression at two levels. Initially converting four nucleotide bases into a single byte through binary representation compresses the data to one-fourth. Further, applying wavelet transforms four times reduces the data to $1/2^4$ times i.e. one-sixteenth of the numerical representation of the original nucleotide sequence. Thus, the proposed algorithm applies compression twice i.e.

The size of the compressed signal C_k becomes,

$C_k = 1/2^2 + 1/2^4 = 1/2^6 = 1/64.$

Thus, $C_k = S_i \cdot 1/64$.

So, 64 times compaction of data can be achieved using the proposed algorithm.
Storing this compact data would result in storage efficiency as well, will be much optimal for any type of analysis.
Overall, the complexity of the proposed algorithm applied for compression would be linear.

Complexity of proposed algorithm for compression = Complexity of converting nucleotide sequence to binary representation + Complexity of Wavelet transform

$= O(n) + O(\log n)$
$= O(n).$

4 RESULT AND DISCUSSION

Example-1: Example of Synthetic Nucleotide sequence, before and after applying the proposed algorithm. The sequence can be perfectly reconstructed as demonstrated in Fig. 1.

If, the Original Read (Total 121 characters) S_i is:

TTTGGGGCGTGCAGGATGCTGCAAAACGTT
ATTTCGGCAAGAATGCGAGCCAACTGGATG
CCAGTGAAGGTGCTATTTAGGGATGTTACGT
AACCCGACGTATTATAATACCGGCCGTGAT

Numerical Representation through Binary Representation of {A, C, G,T} as {00, 01, 10, 11} respectively and storing four consecutive characters as binary form in a single byte. The following are the decimal values of the converted binary form: (Total 30 elements i.e. 4 times compression) i.e. X_j:

254, 169, 185, 40, 231, 144, 6, 243, 246, 144, 131, 152, 148, 30, 142, 82, 224, 174, 115, 242, 163, 188, 108, 21, 134, 207, 48, 197, 165, 184

After applying Fourth Level Wavelet Transform, Data appears (Total 2 elements i.e. 2^4 times thus $30/16 \sim 30/15 = 2$ elements) as C_k:

49.2500000000001, -12.2500000000001

The graphical representation of the numerical form of original sequence, 4-levels of decompositions and the reconstructed sequence, is given in Fig. 1.
The X axis represents the Time or Space Localization and the Y axis represents the Scale or Frequency Localization.
Fig. 1 represents the output of the proposed algorithm for a sample sequence as explained in Example 1. The four level decompositions clearly display reducing length trend, each time being the half of the previous signal, using multi-level Haar Wavelet Transform. The graph labeled "reconstructed seqn" clearly represents that the signal can be perfectly reconstructed, as it is exactly same as the graph labeled as "original seqn" using multi-level Haar Wavelet Transform. The graph labeled "reconstructed seqn" clearly represents that the signal can be perfectly reconstructed, as it is exactly same as the graph labeled as "original seqn"

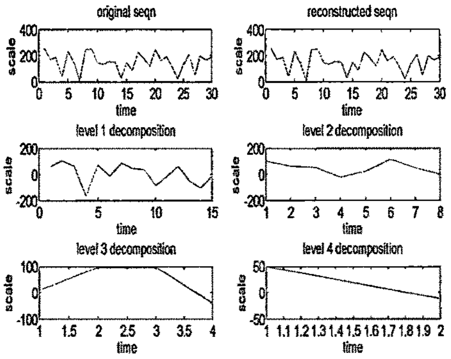


Fig. 1: Visual representation of four level decompositions displaying reducing length trend, of a sequence as explained in Example 1,

Example-2: SRA Read of Metagenomics sequence having Accession Number >gnl|SRA|SRR000675.1.2 EXHS9OF01EH7NX.2 is used to apply the proposed algorithm. (Refer Fig. 2)

Example-3: Contig 46 of E. Coli K-12 strain having Accession Number gi|305690406|gb|AEFE01000046.1 is used to apply the proposed algorithm.

Any type of Nucleotide sequence may be used for compression through application of the proposed algorithm. The sequence can be perfectly reconstructed, and can be compressed to almost 64 times, as demonstrated in Fig. 2.

Table 2, further supports the compression efficiency of the proposed algorithm because the cases considered for compression, show nearly 64 times of compression ratio which is worth for a big data of genomics.

The computational time was calculated for an experiment conducted on regular laptop with Intel Core2 Duo CPU, T6500 @ 2.10 GHz x 2.10 GHz processor and 32-bit Windows 7 Ultimate Operating System with 4.00GB RAM using MATLAB R2009a version 7.0.8.

The computing environment did not consider any special arrangement; instead the experiments were performed, while all routine services including Oracle Database service and Apache Tomcat Application Server was running on an average personal machine.

This type of environment used for computing, simply proves that special purpose computer or clustered computing environment is not essential for executing this algorithm. It can be smoothly executed on a simple desktop machine too. The computational time can be optimized to a great extent, if special purpose computers are used instead of shared resources.

Thus, the proposed algorithm can be applied on any form of genomic data.

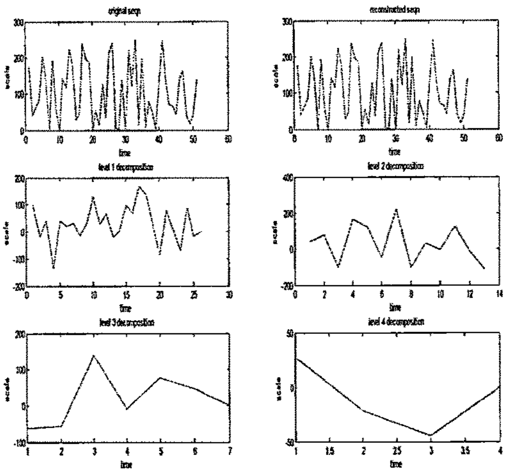


Fig. 2: Visual representation of compressed form metagenomics sequence having Accession Number >gnl|SRA|SRR000675.1.2 EXHS9OF01EH7NX.2, using proposed algorithm applying multi-level Haar Wavelet Transform.

TABLE 2. Represents the Compression Ratio of Original Sequence and Compressed Sequence after binary representation and after four levels of Wavelet Transforms, as proposed in the algorithm

Accession Number	Description of Sequence	Original Nucleotide Sequence Size (In base pairs)	Size in Decimal values of Binary Form of Nucleotide (In Number of elements)	Size 4th level Wavelet Transformed Sequence (In Number of elements)	Compression Ratio
(a)	(b)	(c) = S_i	(d) = X_j	(e) = C_k	(f) = (c)/(e)
-	Sample Read	121	30	2	60.5
gnl|SRA|SRR000675.1.2 EXHS9OF01EH7NX.2	Metagenomic Read	208	51	4	52

gi|305690406|gb|AEFE01000046.1|	Contig 46 of E.Coli K12 strain	14516	3540	222	65.387
gi|224514930|ref|NT_007741.14	Chromosome 7genomic Contig of Homo Sapiens	4758040	1189507	74345	63.999
gi|392976480|ref|NC_003281.9|	Chromosome 3 of Caenorhabditis Elegans	13980611	3445924	215371	64.914
gi|194719403|ref|NC_007327.3|NC_007327	Chromosome 26 of Bos Taurus	51750744	12937686	808606	63.999

5. CONCLUSION

The research concludes that discrete wavelet transform, particularly Haar wavelets, can be used to compress the genomic data. The proposed algorithm using signal processing, compresses the DNA sequences in terms of the number of elements, preserving the details of the original signal. The information of original DNA sequence is preserved in a signal compressed after wavelet transform, be it content of the sequence or the positional information. Thus, using transformed data in place of original sequence for data analysis would also be possible, because there is no loss of data due to transformation. The wavelet transforms have a time complexity of *O(log n)*. Hence using Haar wavelet transforms is much efficient than other classic string or regex based algorithms, with exponential time complexity. In all, the proposed algorithm using Haar Wavelet Transforms, is an efficient way to compress nucleotide sequences, in linear time, without requirement of complex data structures and can be used for data analysis.

The enhancement to this work is, of applying wavelet transforms on large sequences using distributed tool box of MATLAB. The research also aims to apply this transformed data using the Wavelet Transforms, for different types of sequence analysis.

REFERENCES

[1] Ziv J, Lempel A. A universal algorithm for sequential data compression, IEEE Transaction of Information Theory 1977; 23(3):337-343.

[2] Burrows M, Wheeler D: A block sorting lossless data compression algorithm. Tech. Rep. 124, Digital Equipment Corporation 1994.

[3] Huffman D.A. A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the I.R.E., September 1952, pp 1098–1102. Huffman's original article.

[4] Paolo Ferragina, Giovanni Manzini, Veli M¨akinen, and Gonzalo Navarro, "An Alphabet-Friendly FM-index?" In International Symposium on String Processing and Information Retrieval (SPIRE), (2004) pages 150–160

[5] O˘guzhan M, Vitter J S, Xu B, Efficient Maximal Repeat Finding Using the Burrows-Wheeler Transform and Wavelet Tree, IEEE/ACM Transactions On Computational Biology And Bioinformatics.

[6] http://www.ncbi.nlm.nih.gov

[7] Congmao Wangl and Dabing Zhang, A novel compression tool for efficient storage of genome resequencing data

[8] Hisahiko Satol Takashi Yoshioka, Akihiko Konagaya3 Tetsuro Toyoda,DNA Data Compression in the Post Genome Era, Genome Informatics 12: 512–514 (2001)

[9] Ana Balevic, Lars Rockstroh, Marek Wroblewski, and Sven Simon, Using Arithmetic Coding for Reduction of Resulting Simulation Data Size on Massively Parallel GPGPUs

[10] Sayood, K., Lossless Compression Handbook. Academic Press (2003)

[11] Toshiko Matsumoto Kunihiko Sadakane, Hiroshi Imai, Biological Sequence Compression Algorithms, Genome Informatics 11: 43–52 (2000) 43

[12] Chen, X., Kwong, S., and Li, M., A compression algorithm for DNA sequences and its applications in genome comparison, *Genome Informatics* , 10:52–61, 1999.

[13] Grumbach, S. and Tahi, F., A new challenge for compression algorithms: genetic sequences, *Information Processing & Management*, 30:875–886, 1994.

[14] Kenny Daily, Paul Rigor, Scott Christley, Xiaohui Xie, Pierre Baldi, Data structures and compression algorithms for high-throughput

sequencing technologies, BMC Bioinformatics 2010, 11:514

[15] Samantha Woodward, A Critical Analysis of DNA Data Compression Methods BIOC 218, Winter 2011-2012

[16] Bernhard Haubold andThomas Wiehe, How repetitive are genomes?, BMC Bioinformatics 2006, 7:541 doi : 10.1186/1471-2105-7-541

[17] Markus Hsi-Yang Fritz, Rasko Leinonen, Guy Cochrane, and Ewan Birney, Efficient storage of high throughput DNA sequencing data using reference-based compression, Genome Research, 21:734–740_2011 by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/11;

[18] W Timothy J White* and Michael D Hendy,Compressing DNA sequence databases with coil, *BMC Bioinformatics* 2008, 9:242 doi:10.1186/1471-2105-9-242

[19] Al Junid, S. A. M. Tahir, N.M. Haron, M.A. Abd Majid, Z., Idros, M.F. Osman, F.N., Development and Implementation of Novel Data Compression Technique for Accelerate DNA Sequence Alignment Based on Smith–Waterman Algorithm, IJSSST, Vol. 11, No. 3 ISSN: 34 1473-804x online, 1473-8031

[20] Mamta C. Padole, B. S. Parekh, D. P. Patel, Signal Processing Approach for Recognizing Identical Reads From DNA Sequencing of Bacillus Strains, IOSR Journal of Computer Engineering, Mar-Apr 2013, pp 19-24.

[21] Mamta C. Padole, Recognizing Short Tandem Repeat Regions in Genomic Sequences Using Wavelet Transforms, unpublished

[22] J. K. Meher, M. R. Panigrahi, G. N. Dash, P. K. Meher, "Wavelet Based Lossless DNA Sequence Compression for Faster Detection of Eukaryotic Protein Coding Regions ," *I.J. Image, Graphics and Signal Processing 2012, 7, 47-53*

[23] I. Daubechies, "Ten Lectures On Wavelets", 1992

[24] Charu C. Aggarwal, "On the Use of Wavelet Decomposition for String Classification," *Springer - Data Mining and Knowledge Discovery*, 10, 117–139, 2005

[25] MATLAB manual: http://www.mathworks.in/products/wavelet

[26] C.S.Burrus, R.A. Gopinath, Haitau Guo, "Introduction to Wavelets and Wavelet Transforms", Prentice Hall, 1998

[27] Moharir PS, "Pattern recognition transforms,"New York: Wiley, 1992

[28] Radomir S. Stankovic, Bogdan J. Falkowski, "The Haar wavelet transform: its status and achievements,"Elsevier Science Ltd. Computers and Electrical Engineering, 2003 29 (2003) 25–44

[29] Ruiz G, Michell JA, Buron A, "Switch-level fault detection and diagnosis environment for MOS digital circuits using spectral techniques," IEE Proc Part E, 1992; 139(4):293–307

[30] Dimitris Sacharidis, "Constructing Optimal Wavelet Synopses", Proceedings of the 2006 International Conference on Current Trends in Database Technology EBDT '06, Pg 97-104, 2006 10.1007/11896548_10

[31] Musawir Ali, An Introduction to Wavelets and the Haar Transform, http://www.cs.ucf.edu/~mali/haar/

[32] A. S. Nair and S. P. Sreenathan, "A Coding Measure Scheme Employing Electron-Ion Interaction Pseudopotential (EIIP)," Bioinformation, Vol. 1, No. 6, 2006, pp. 197-202.

[33] Aghili, Alireza A, Agrawal, D El Abbadi, A, "Sequence Similarity Search Using Discrete Fourier and Wavelet Transformation Techniques," Postprints, UC Santa Barbara, 2005 http://www.escholarship.org/uc/item/9w7094b9

[34] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets,"Comm. Pure Appl. Math, Vol 41, 1988, pp. 906-966.

"PyroSequencing"
Sequencing Technique in Roche's GSFLX System

Mamta Padole,
Department of Computer Science,
The M.S. University of Baroda.
mamta.padole@gmail.com

Abstract:

The breakthrough research in genomics, particularly plant genomics & metagenomics and in health sciences related to, causes of genetic diseases & infectious diseases is possible due to the current methods of DNA sequencing. 454 sequencing technology can produce biological sequence data on a scale that exceeds traditional Sanger sequencing by orders of magnitude. The paper discusses "Pyrosequencing", a DNA sequencing method, used in 454 Life Sciences GSFLX Titanium System, the format of the data generated by the system and the possible scopes of research in allied subjects.

Introduction:

In 1986, around the time the Human Genome Project was initiated, the cost of sequencing was around $10 per base. By 2001, the cost had fallen to about 10 to 20 cents per nucleotide [1]. Nowadays, Sanger sequencing can be approached at a cost of around 0.5 cents per nucleotide (that's a 2000-fold drop) but a recent technology breakthrough, pyrosequencing, is likely to drop the costs even further, while simultaneously increasing the throughput by an order of magnitude or more.

Pyrosequencing [2-5], first described in the literature by Hyman (1988) is, unlike Sanger technique, a non-fluorescence technique. Pyrosequencing also termed as a SNA method (Single Nucleotide Addition method) is a real-time, sequencing by synthesis method based on the detection of released pyrophosphate during DNA synthesis. Pyrosequencing is the high throughput sequencing method, generating upto 10 megabases/hour. On the other hand, the sequenced fragments have reduced lengths compared to Sanger ones, being approximately 100 bases.

The recent dramatic increase in sequencing throughput together with the reduction of costs in turn demands for increased computation power, as well as increased storage space. Also, it is to be considered that most bioinformatics tasks such as genome assembly, inversion distance computation, genome rearrangement analysis and molecular dynamics have got a quadratic or higher complexity. This poses the challenge to allied areas of research like Computer Science, Mathematics, and Statistics etc. To reduce this complexity, the new techniques are designed for optimal storage, analysis and interpretation of the raw data generated from the newly available pyrosequencing method of DNA sequencing. Before going into the details of various analytical techniques applied on the generated data, it would be preferable to discuss the pyrosequencing technique itself and the format of the data generated, which can then be further appropriately stored, analyzed and interpreted for the use in genomics research.

Pyrosequencing:

Principle:

Pyrosequencing technology is sequencing by synthesis, a simple to use technique for accurate and quantitative analysis of DNA sequences.

Technique:

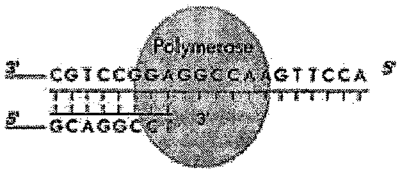
Pyrosequencing unarguably the most successful non-Sanger method, is a DNA sequencing technique that is based on the detection of released pyrophosphate (PPi) during DNA synthesis. In a cascade of enzymatic reactions, visible light is generated that is proportional to the number of incorporated nucleotides. The cascade starts with a nucleic acid polymerization reaction in which inorganic PPi is released as a result of nucleotide incorporation by polymerase. The released PPi is subsequently converted to ATP by ATP sulfurylase, which provides the energy to luciferase to oxidize luciferin and generate light. Because the added nucleotide is known, the sequence of the template can be determined.

The release of inorganic pyrophosphate is measured, which is proportionally converted into visible light by a series of enzymatic reactions (Ronaghi et al. 1996, 1998). Unlike other sequencing approaches that use 3_-modified dNTPs to terminate DNA synthesis, the pyrosequencing assay manipulates DNA polymerase by single addition of dNTPs (hence called SNA method), in limiting amounts. Upon addition of the complementary dNTP, DNA polymerase extends the primer and pauses when it encounters a noncomplementary base. DNA synthesis is reinitiated following the addition of the next complementary dNTP in the dispensing cycle. The light generated by the enzymatic cascade is recorded as a series of peaks called a pyrogram, which corresponds to the order of complementary dNTPs incorporated and reveals the underlying DNA sequence. Applications for pyrosequencing have been reviewed by Ronaghi (2001) and Langaee and Ronaghi (2005).

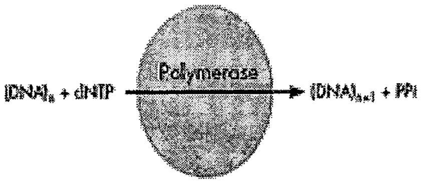
The following are the stepwise implementation of the technique to determine the DNA sequence.

Step 1

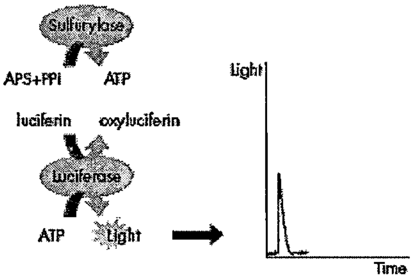
A sequencing primer is hybridized to a single-stranded PCR amplicon that serves as a template, and incubated with the enzymes, DNA polymerase, ATP sulfurylase, luciferase, and apyrase as well as the substrates, adenosine 5' phosphosulfate (APS), and luciferin.



Step 2
The first deoxribonucleotide triphosphate (dNTP) is added to the reaction. DNA polymerase catalyzes the incorporation of the deoxyribo-nucleotide triphosphate into the DNA strand, if it is complementary to the base in the template strand. Each incorporation event is accompanied by release of pyrophosphate (PPi) in a quantity equimolar to the amount of incorporated nucleotide.

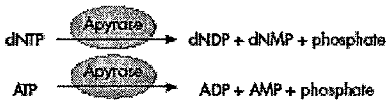


Step 3
ATP sulfurylase converts PPi to ATP in the presence of adenosine 5' phosphosulfate (APS). This ATP drives the luciferase-mediated conversion of luciferin to oxyluciferin that generates visible light in amounts that are proportional to the amount of ATP. The light produced in the luciferase-catalyzed reaction is detected by a charge coupled device (CCD) chip and seen as a peak in the raw data output (Pyrogram). The height of each peak (light signal) is proportional to the number of nucleotides incorporated.

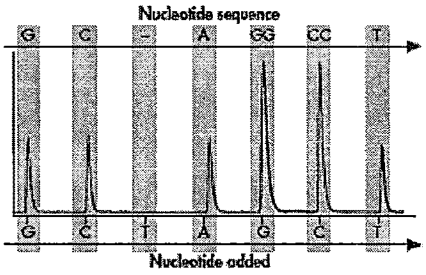


Nucleotide incorporation generates light seen as a peak in the Pyrogram trace

Step 4
Apyrase, a nucleotide-degrading enzyme, continuously degrades unincorporated nucleotides and ATP. When degradation is complete, another nucleotide is added.

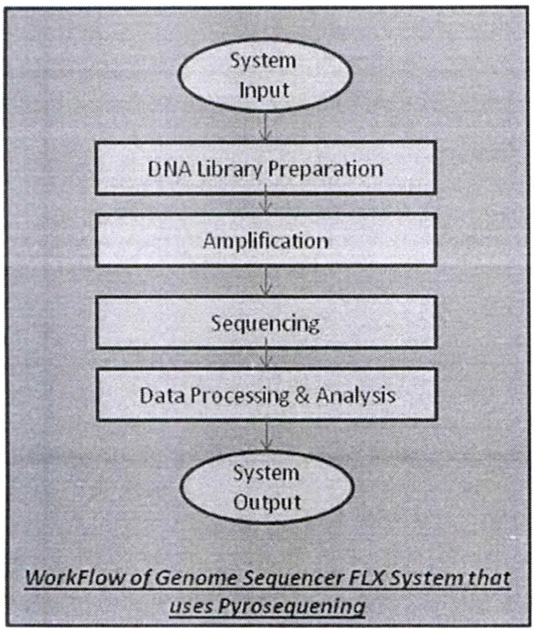


Step 5
Addition of dNTPs is performed sequentially. It should be noted that deoxyadenosine alfa-thio triphosphate (dATP·S) is used as a substitute for the natural deoxyadenosine triphosphate (dATP) since it is efficiently used by the DNA polymerase, but not recognized by the luciferase. As the process continues, the complementary DNA strand is built up and the nucleotide sequence is determined from the signal peaks in the Pyrogram trace.



The above 5 steps involves only the first 3 stages of the Workflow of 454 GSFLX system.

Application of Pyrosequencing in 454 Life Science's GSFLX System



WorkFlow of Genome Sequencer FLX System that uses Pyrosequening

The 454 Corporation has recently introduced a whole genome sequencing strategy by integrating pyrosequencing with their PicoTiterPlate (PTP) platform, which has been shown to amplify and image approximately 300,000 PCR templates captured on Sepharose beads (Leamon et al. 2003). The PTP is manufactured by anisotropic etching of a fiber optic faceplate with a well diameter of approximately 40 µm. The 454 group has developed a solution-based emulsion strategy to create microreactors for clonal amplification of single DNA molecules and attachment to these beads. One advantage of the clonal amplification strategy is that it addresses the dependence issue of dispensing order for sequencing of heterozygous bases. Following an enrichment step, DNA positive beads are loaded into individual PTP wells, which contain additional beads coupled with the necessary enzymes to perform the pyrosequencing chemistry (Margulies et al. 2005).

The output of a sequencing Run consists of raw or processed DNA sequencing data which is processed by the GS Run Processor application which carries its work in two steps: image processing and signal processing. [8]

The image processing step extracts the raw signals for each nucleotide flow, in each active well of the PicoTiterPlate device, from the raw images captured by the Genome Sequencer FLX Instrument during the sequencing run. The signal processing step corrects these flow signals for optical and chemical artifacts, and produces base calls.

The output of the GS Run Processor is packaged into a set of Composite Wells Format files (.cwf) and Standard Flowgram Format (.sff) files. The GS Reporter application is evoked by default to extract information from the CWF and SFF files output by the GS Run Processor, and generate a set of metrics and FASTA files that can be used to assess the quality of a Run.

The output of a sequencing Run consists of raw or processed DNA sequencing data (up to basecalls) that can be further analyzed using the post-Run data analysis software of the Genome Sequencer System (GS *De Novo* Assembler, GS Reference Mapper, or GS Amplicon Variant Analyzer), or other third party software.

The assembly of non-Sanger sequencing data will represent new challenges because the input read will differ in length, quantity, and quality. The complexity of the genome under analysis may also prove more difficult for assemblies compared with Sanger data, even when the offset is higher coverage of shorter reads. Compared with Sanger data, the read-length is inversely proportional to the number of contigs in the assembly (i.e., longer reads gave fewer contigs) Chaisson et al. (2004). Increasing genome complexity, on the other hand, directly increases the number of contigs.

Observed errors for real sequence data will undoubtedly decrease assembly performance for short reads. Thus, the success of the non-Sanger strategies for whole-genome sequencing applications will be highly dependent on the degree of its complexity.

Strengths of Pyrosequencing:

- It presents an analyzed mutation in the context of the neighbouring genetic sequence – we call this "Built-in Quality Control". It is a foolproof means of guaranteeing that the assay worked correctly.
- It delivers the "gold standard" of genetic analysis: the sequence itself. Other methods only provide a "Yes/No" signal. Unlike a fluorescent signal, sequence information is intelligible; therefore it is easy to communicate these results in the literature, and easy to transfer meaningful data between research labs.
- With Pyrosequencing technology, there is great flexibility in primer placement. Therefore it is easy to design a Pyrosequencing assay to analyze virtually any genetic marker. So far, Pyrosequencing technology has an unbeaten track record in analyzing any SNP.
- Pyrosequencing assays are mutation-tolerant. Unlike hybridization-based assays, Pyrosequencing analysis generates a correct sequence regardless of the appearance of a new, unexpected mutation. This is very important to microbiological applications: hybridization-based assays can give false negatives in the presence of a new mutation.
- With Pyrosequencing analysis, you not only obtain sequence information, but the data is also fully quantitative, ideal for measuring the relative amounts of alleles. This property allows the quantification of DNA methylation, heterozygosity, ploidy levels, multi-copy

genes, pooled DNA samples, hematopoeitic chimerism, and mixed genotypes in heterogeneous samples (e.g. tumor and normal cells).

- Pyrosequencing technology easily addresses the many and varied applications that are typical for clinical research labs.

Drawbacks of Pyrosequencing:

Although elegant in design, the pyrosequencing approach has several limitations.

- Sequence reads are typically fewer than 100 bases in length, which has application in sequence tag identification such as serial analysis of gene expression (SAGE) (Velculescu et al. 1995), mini-sequencing for known SNPs, and mapping related genomes to a reference sequence, but limited application for whole-genome sequencing.
- Homopolymer repeats greater than five nucleotides cannot be quantitatively measured. This is attributed to incomplete extension by DNA polymerase, which results from limiting the dNTP concentration to minimize nucleotide disincorporation effects.
- The dispensing order of dNTPs determines the pyrogram profile, which must be carefully designed to avoid asynchronistic extensions of heterozygous sequences.

The SFF format:

The output generated by 454's GSFLX system sequencing process, uses Standard Flowgram Format (SFF).
SFF was designed by 454 Life Sciences, Whitehead Institute for Biomedical Research and Sanger Institute.

The definition of a Standard Flowgram Format (SFF), similar to the SCF format, to hold the "trace" data for 454 reads

The proposed SFF file format [6] is a container file for storing one or many 454 reads. 454 reads differ from standard sequencing reads in that the 454 data does not provide individual base measurements from which basecalls can be derived. Instead, it provides measurements that estimate the length of the next homopolymer stretch in the sequence (i.e., in "AAATGG", "AAA" is a 3-mer stretch of A's, "T" is a 1-mer stretch of T's and "GG" is a 2-mer stretch of G's). A basecalled sequence is then derived by converting each estimate into a homopolymer stretch of that length and concatenating the homopolymers.

The file format consists of three sections, a common header section occurring once in the file, then for each read stored in the file, a read header section and a read data section. The data in each section consists of a combination of numeric and character data.

Following is the sample of the contents of SFF file [7]

```
Common Header:

Magic Number:  0x2E736666
 Version:      0001
  Index Offset:  96099976
  Index Length:  1158685
  # of Reads:    57902
  Header Length: 440
  Key Length:    4
  # of Flows:    400

  Flowgram Code: 1

  Flow Chars:
TACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGT
ACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTA
CGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTAC
GTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACG
TACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGT
ACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTA
CGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTACGTAC
G

  Key Sequence:  TCAG



>FIQU8OX05GCVRO

  Run Prefix:  R_2008_10_15_16_11_02_
  Region #:    5
  XY Location:  2489_3906
  Run Name:      R_2008_10_15_16_11_02_FLX04070166_adminrig_1548jinnescurtisstanford
 Analysis Name:
/data/2008_10_15/R_2008_10_15_16_11_02_FLX04070166_adminrig_1548jinnescurtisstanfor
d/D_2008_10_15_15_12_26_FLX04070166_1548jinnescurtisstanford_FullAnalysis
  Full Path:
/data/2008_10_15/R_2008_10_15_16_11_02_FLX04070166_adminrig_1548jinnescurtisstanfor
d/D_2008_10_15_15_12_26_FLX04070166_1548jinnescurtisstanford_FullAnalysis

  Read Header Len:  32
  Name Length:      14
  # of Bases:       104
  Clip Qual Left:   5
Clip Qual Right:  85
Clip Adap Left:   0
Clip Adap Right:  0


Flowgram:    1.06   0.08   1.04   0.08   0.05   0.94   0.10   2.01   0.10   0.07   0.96
      0.09   1.04   1.96   1.07   0.10   1.01   0.13   0.08   1.01   1.06   1.83   2.89
      0.18   0.96   0.13   0.99   0.11   1.94   0.12   0.13   1.92   0.21   0.07   0.94
      0.17   0.03   0.97   2.76   0.15   0.05   1.02   1.14   0.10   0.98   2.54   1.13
      0.96   0.15   0.21   1.90   0.16   0.07   1.78   0.22   0.07   0.93   0.22   0.97
      0.08   2.02   0.15   0.19   1.02   0.19   0.09   1.02   0.17   0.99   0.09   0.18
      1.84   0.16   0.91   0.10   1.10   1.00   0.20   0.09   1.11   3.01   1.07   1.98
      0.14   0.22   1.09   0.17   1.99   0.15   0.20   0.92   0.17   0.07   1.01   2.96
      0.15   0.07   1.06   0.20   1.00   0.10   0.12   1.00   0.15   0.08   1.90   0.19
      0.10   0.99   0.18   0.09   0.99   1.08   0.15   0.07   1.06   0.14   1.84   0.13
      0.11   0.95   1.05   0.13   1.04   1.10   0.18   0.94   0.14   0.10   0.97   1.08
      0.12   1.08   0.18   0.08   1.00   0.13   0.98   0.15   0.87   0.13   0.19   1.01
      3.06   0.17   0.11   1.04   0.09   1.03   0.10   0.11   2.02   0.16   0.11   1.04
      0.04   0.09   1.87   0.13   2.09   0.13   0.10   0.97   0.17   0.08   0.08   0.04
      0.12   0.05   0.08   0.07   0.08   0.05   0.07   0.06   0.07   0.03   0.05   0.04
      0.09   0.04   0.07   0.04   0.07   0.06   0.03   0.06   0.06   0.06   0.06   0.07
      0.09   0.04   0.05   0.08   0.05   0.04   0.09   0.06   0.03   0.02   0.08   0.04
      0.06   0.05   0.08   0.03   0.08   0.05   0.05   0.05   0.10   0.05   0.05   0.07
      0.06   0.04   0.06   0.05   0.03   0.04   0.05   0.06   0.04   0.04   0.07   0.04
      0.04   0.05   0.05   0.04   0.07   0.06   0.05   0.03   0.08   0.05   0.06   0.04
      0.06   0.05   0.04   0.04   0.04   0.05   0.06   0.04   0.05   0.04   0.05   0.05
      0.06   0.05   0.06   0.04   0.06   0.07   0.06   0.05   0.05   0.05   0.06   0.06
      0.04   0.05   0.06   0.03   0.06   0.04   0.06   0.05   0.03   0.06   0.06   0.05
      0.06   0.04   0.03   0.06   0.06   0.06   0.03   0.04   0.05   0.05   0.07   0.04
      0.05   0.06   0.07   0.07   0.05   0.07   0.06   0.05   0.06   0.05   0.07   0.06
      0.05   0.06   0.07   0.05   0.06   0.04   0.06   0.05   0.05   0.06   0.04   0.06
      0.04   0.03   0.06   0.05   0.05   0.04   0.05   0.05   0.04   0.04   0.05   0.06
      0.06   0.04   0.04   0.05   0.06   0.04   0.04   0.04   0.05   0.05   0.04   0.05
      0.05   0.03   0.06   0.06   0.06   0.04   0.07   0.05   0.05   0.04   0.06   0.06
      0.05   0.05   0.07   0.04   0.06   0.06   0.06   0.04   0.06   0.03   0.06   0.04
      0.06   0.04   0.09   0.05   0.05   0.05   0.07   0.06   0.05   0.05   0.06   0.05
      0.05   0.05   0.04   0.04   0.06   0.05   0.05   0.05   0.05   0.04   0.05   0.05
      0.06   0.04   0.05   0.05   0.05   0.05   0.05   0.04   0.06   0.04   0.05   0.05
      0.04   0.05   0.05   0.05   0.04

Flow Indexes: 1      3      6      8      8      11     13     14     14     15     17
      20     21     22     22     23     23     23     25     27     29     29     32
      32     35     38     39     39     39     42     43     45     46     46     46
      47     48     51     51     54     54     57     59     61     61     64     67
      69     72     72     74     76     77     80     81     81     81     82     83
```

```
        83    86    88    88    91    94    95    95    95    98    100   103
        106   106   109   112   113   116   118   118   121   122   124   125
        127   130   131   133   136   138   140   143   144   144   144   147
        149   152   152   155   158   158   160   160   163

Bases:
        tcagGCTAACTGTAACCCTCTTGGCACCCACTAAACGCCAATCTTGCTGGAGTG
TTTACCAGGCACCCAGCAATGTGAATAGTCActgagcgggctggcaaggc

Quality Scores:         37    37    37    37    37    37    37    37    37    37
        37    37    37    40    40    40    40    37    37    37    37    37
        39    39    39    39    24    24    24    37    34    28    24    24
        24    28    34    39    39    39    39    39    39    39    39    39
        39    39    39    40    40    37    37    37    37    37    37    37
        37    37    37    37    37    37    37    37    37    37    37    37
        37    37    37    37    37    37    37    37    37    37    37    37
        37    37    37    37    37    37    37    37    37    37    37    37
        37    37    37    37    37    37    37    37    37    37
```

Conclusion :

Looking to the details of pyrosequencing and the format of the SFF file the output generated by pyrosequencing in 454 GSFLX system, we can conclude that there is a lot of scope of research for the allied subjects. In Computer science, the output of the pyrosequencing data needs to be stored & retrieved efficiently, giving rise to research in area of Data Warehousing & Mining. The signal processing has good amount of contribution in understanding and interpreting the data as well as reducing the incorporated homopolymeric errors. Mathematics needs to be used for doing sequence alignment, through various techniques like distance editing, fuzzy logic, decision trees & graph theory. The output needs to be efficiently compared using statistical methods. Thus Pyrosequencing opens the large vista for research areas in allied fields.

References :

1. National Science Foundation. Cost of genomic sequencing. http://www.nsf.gov/news/speeches/colwell/rc03_dallas/sld016.htm
2. Ronaghi M, Uhlen M, Nyrén P. A sequencing method based on real-time pyrophosphate. *Science.* 1998;281:363–365. doi: 10.1126/science.281.5375.363. [PubMed]
3. Ronaghi M. Pyrosequencing sheds light on DNA sequencing. *Genome Res.* 2001;11:3–11. doi: 10.1101/gr.11.1.3. [PubMed]
4. 454 Life Sciences. GS 20. http://www.454.com/
5. Biotage. Principle of Pyrosequencing. http://www.biotagebio.com/
6. SFF Format. http://www.ncbi.nlm.nih.gov/
7. Quince et al. "Noise and the Accurate Determination of Microbial Diversity from 454 Pyrosequencing Data".
8. Roche's GSFLX System Manuals

Distributed Computing for Structured Storage, Retrieval and Processing of DNA Sequencing Data

Mamta C. Padole
Department of Computer Science and Engineering
The Maharaja Sayajirao University of Baroda
Baroda, India
Email:mpadole29@rediffmail.com

ABSTRACT

The technologically break-through in DNA sequencing process, and the current trend of DNA re-sequencing, both, generate high-throughput data, also referred to as Big Data. This Big Data demands increased computation power and storage space. Hence, the big data needs to be stored, retrieved and processed efficiently. The paper discusses the use of Distributed Application developed using Client-Server model and M-V-C Architecture for implementing distributed computing for efficient and secured storage, optimized and customized retrieval of Big Data, generated through DNA sequencing. The paper proposes the use of web application for data retrieval and storage, which provides total transparency of where and how the execution of various tasks is happening. The data and the business logic are independent components located on separate computers and are loosely coupled. Data dictionary designed, for storing the DNA sequencing reads generated using 454 Roche's GSFLX system and the results of analysis based on this DNA sequenced reads like contigs, repeats, coding regions, genes etc, is also discussed in brief.

Keywords –Distributed Application, Web Application, Client Server Architecture, M-V-C Architecture, Big data

1. INTRODUCTION

The ultra-high throughput data generated at an accelerated rate through next generation DNA sequencing [1] techniques need secured storage, speedy retrieval, quick and efficient processing, to gain genetic knowledge at appropriate and requisite time. Cost effectiveness and technical viability of high-performance, high-throughput computer in an organization, where both the data storage and processing is happening on a single machine, are often the matters of concern. To such queries, distributed computing is the best alternative. Hence, a distributed web applications are developed, where the data storage and business logic required for querying the stored data, are, located on different machines. Primary purpose of improving efficiency of storage, processing and analysis of DNA sequences is, to get timely genetic information of an individual or organism, because, their major application is for identification and treatment of diseases like cancer and HIV, drug designing, controlling spread of epidemics by identifying micro-organisms involved in disease causing and prohibiting their growth, enhancing crop breeds, to meet the drought situation, improving animal breeds that provide food, nourishment, medicines or income; for phylogeny identification and in forensics.

Globally accessible, centralized repository of NCBI (National Centre for Biotechnology Information) [2] is available for storage (albeit, physically they are scattered like Swiss-Prot or DDBJ. Centralized storage is only a logical view that presents transparency to user) and processing of genomic data. But, at times, it is not possible to upload and disclose the confidential data of some genomes into the centralized repository of NCBI, particularly, when the project is sponsored and controlled under some specified guidelines and contract. Although, one can upload data and keep it private in NCBI, but that is permissible, only till the manuscript describing the data gets published. If the sequencing data is uploaded at NCBI, eventually, it becomes an open data in public domain.

To avoid this disclosing, but at the same time use all the web based facilities for storage, retrieval and customized algorithms for processing of DNA sequencing data within an organization, a distributed application has been developed, which is discussed in this paper.

Moreover, in a spatially located organization or research organization where the data generation experiments like DNA sequencing are conducted at scattered locations or laboratories with the need to exchange data between various locations, it is advisable to maintain the central repository (at least, the common interface, which provides the single point of access to distributed data). This central repository maximizes of utilization of DNA data, at the same time relieves the biological researcher from obtaining the knowledge and responsibility of storage or security of data. The backup and recovery activity, botheration of security or availability of data, then becomes the responsibility of database administrator. Also, the biological researcher need not get involved in knowledge of computer science, in particular, database administration or network administration, and hence can concentrate in his actual research domain, which usually would be biology related.

The paper discusses development of an application for DNA sequencing data, based on distributed computing concepts, which initiates right, from uploading of the raw