This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

A MODEL for DISTRIBUTED PROCESSING and ANALYSES of NGS DATA under MAP-REDUCE PARADIGM

Sandip Samaddar, Rituparna Sinha, and Rajat K. De, *Senior Member, IEEE*

Abstract—Massively parallel sequencing technique, introduced by NGS technology, has resulted in an exponential growth of sequencing data, with greatly reduced cost and increased throughput. This huge explosion of data has introduced new challenges in regard to its storage, integration, processing and analyses. In this paper, we have proposed a novel distributed model under Map-Reduce paradigm to address the NGS big data problem. The architecture of the model involves Map-Reduce based modularized approach involving 3 different phases that support various analytical pipelines. The first phase will generate detailed base level information of various individual genomes, by granulating the alignment data. The other 2 phases independently process this base level information in parallel. One of these 2 phases will provide an integrated DNA profile of multiple individuals, whereas the other phase will generate contigs with similar features in an individual. Each of these 2 phases will generate a repository of genomic information that will facilitate other analytical pipelines. A simulated and real experimental prototypes has been provided as results to show the effectiveness of the model and its superiority over a few existing popular models and tools. A detailed description of the scope of applications of this model is also included in this article.

Index Terms—CNV, Hadoop, Personalised Medicine, Cancer, NGS, fault tolerant model, Bioinformatics analytical workflow.

1 INTRODUCTION

THE next generation sequencing (NGS) [1], [2] methodology has introduced the notion of short reads, massively parallel sequencing technique, which has revolutionized the sequencing capabilities with greatly reduced cost and increased throughput. The amount of output data generated by NGS technology has been increasing more than double in each year. In 2007, a single sequencing run could produce around one gigabase (Gb) of sequence data [3]. By 2011, it approached a terabase (Tb) of data produced in a single sequencing run [1]. The human genome is composed of approximately 3.2 billion base pairs [3]. At 30X coverage, the capacity to sequence these 3.2 billion bases of the human genome has increased exponentially [3]. Moreover, it is to be noted that the first human genome took 15 years to sequence, with cost nearly 3 billion dollars. HiSeqX Ten, released in 2014, is able to sequence over 45 human genomes in a single day for approximately US$1000 each [4]. Today's NGS machines can generate from a few hundred gigabytes (Gb) to a few terabytes (Tb) of read data per run [5]. Thus, the huge explosion of data is now introducing a new challenge in bioinformatics research.

Storage of this Big Data, its integration, processing and analyses have become more expensive than its generation. Thus, the issues are how to do efficient parallel accessing, and storing of NGS short read alignment information/files (Big Data). The second point is how to integrate the vast data present in different heterogeneous storage systems, and how to intelligently extract knowledge from the data tornado.

In order to overcome the bottleneck of performance in big data analysis, Google first proposed a data storage mechanism using big table in a special file system known as Google file System (GFS) [6]. This would act as a framework to process big data or distributable problems [7]. Later, Apache implemented this idea of GFS in developing an open source big data framework called Hadoop [8], which allowed parallel programming using Map-Reduce paradigm [9]. Hadoop uses a fault tolerant file system called HDFS (Hadoop Distributed File System) [1]. Hadoop has become very popular due to increasing attention of academia and industries [10] for its capability to implement fault tolerant system using HDFS [11] at low cost. Map-Reduce is a simple programming model that allows automatic parallelism and distribution of jobs ensuring high performance and fault tolerance over a cluster of commodity hardware. In this paradigm, a large distributed program is expressed as a sequence of parallel operations on datasets of $< key, value >$ pairs [12], where the data is kept in HDFS architecture as shown in Supplementary Figure S1. **An illustration of Map-Reduce programing is also depicted in Supplementary Figure S3.**

In recent years,Map-Reduce programming in Hadoop has been receiving a great deal of attention of both the research community and industry [7]. Development of NGS data analytical tools in Hadoop framework is also gaining immense popularity, because these tools can provide more accurate analysis of genomic alterations, like SNPs, CNVs and other structural variations, along with their corresponding role in various neurological

- *Rajat K. De is with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. E-mail: rajat@isical.ac.in*
- *Sandip Samaddar is with the Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata, India.*
- *Rituparna Sinha is with the Department of Information Technology, Heritage Institute of Technology, Kolkata, India.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

diseases and cancer [13]. In this regard, NGS tools like CloudBurst [14] has been implemented for mapping or aligning NGS short reads in parallel, using a seed-and-extend based alignment algorithm in Hadoop framework. However, it does not incorporate the quality information efficiently during the alignment process. SEAL [15] is another alignment effort in Map-Reduce framework, which incorporates Burrow Wheeler Alignment (BWA). **DistMap [16] has been designed in Hadoop framework to provide an integrated workflow for aligning NGS reads. Here, the user applications can exploit the flexibility of executing some or all modules of the alignment work-flow. Crossbow [17] implements a Map-Reduce based parallel version of an established alignment and SNP detection algorithms, called Bowtie [18] alignment and SOAP-SNP [19] techniques respectively. A Map-Reduce based read assembling framework, called Contrail [20], has been developed to provide supporting data structure and library to external applications.** SeqWare Query Engine [21] is a Cloud-database management system, using a NoSQL HBase backend and a Map-Reduce Hadoop infrastructure. **It provides a web-based front-end through which users can load several whole genome datasets, as well as query variants. Myrna [22] has provided a Map-Reduce based pipeline that analyses differential expression pattern of genes using RNA-Seq data. Genome Analysis Toolkit (GATK) [23] is a programming platform for NGS data analysis tools under Map-Reduce framework. This tool [23] can separate the complexity of accessing NGS data from the analytical logics. HADOOP-BAM [24] is another java based library tool, which acts as an integration layer between binary alignment (i.e., BAM) files [25] and analysis applications.**

However, all these efforts do not convincingly address the aspect of designing a generalized Map-Reduce based architecture that will allow applications with greater flexibility and control in execution. Besides, these models/ tools do not sufficiently address the evolution of the knowledge and their respective analytical processes with the acquirement of exponentially growing NGS data.

The model has been designed as a distributed analytical platform based on Map-Reduce framework for catering NGS big data issues. Three phases execute under batch processing mode, where the execution of Map-Reduce task of phase BaseProfiler will be followed by the execution of phase SeqClust and phase ContigClust independently. The latter two phases can execute in parallel to each other. Phase BaseProfiler processes NGS related alignment files , in the form of input splits, to generate information corresponding to each base position of the genome, which will be used by the other two phases. Phase SeqClust generates a multiple genome based reference sequence, whereas Phase ContigClust processes the output of the phase BaseProfiler to generate repository of contigs or segments having similar biological features or patterns.

The framework adopts a novel approach for processing $< key, value >$ pairs in both Map and Reduce tasks using a balanced search tree based index of intervals, where an interval corresponds to a genomic segment with start and end coordinates. Traditional Map-Reduce framework, like Hadoop, does not provide this freedom to use a customized tree based index structure or customized hashing for retrieving data in the form of $< key, value >$ pairs, as per the need of the application. The model thus provides a repository generated by analysing and summarizing the NGS short read data of human individuals, providing diverse range of knowledge base. The overall description of all the phases, along with input and output of each Map-Reduce procedures, are depicted in the Supplementary Table S1.

The implementation or prototype of the model facilitates generating several types of genomic information of a particular queried region in a summarized manner, which further enables diverse range of downstream genome analyses, including mappability measure or score, base coverage, and, GC Content data. These data, pertaining to a queried region, have important biological effects on each other. The prototype identifies such type of significant co-relations between several summarized information of a particular queried region, providing superior insight of the information base. In addition a comparative study has been done among the implemented prototype of the model, HADOOP-BAM and GATK tools, for evaluating the performance of the model. The superior performance of the model, in terms of speedup, has been presented in this regard.

2 METHOD

The proposed methodology primarily designs a scientific workflow model that provides analytical pipelines executable in parallel, for processing NGS based big data. The workflow mainly will provide an integrated DNA profile of similar/diverse human genomes which will get modified with increasing scalability. In addition, the model will generate a distributed repository of DNA segments based on similar features, like structural variations, GC content profiles and gene expression level, which can be utilized by other analytical tools. These tools could access the repository efficiently through an indexed based retrieval, unlike HADOOP-BAM and GATK. These pipelines would be deployed on a distributed framework based on Map-Reduce style, during processing of NGS alignment data of a population. The alignment data of each individual sample is divided into smaller data fragments (input splits), which are managed using a distributed framework on commodity hardwares (nodes). An input split is a collection of records where each record corresponds to a read associated with alignment information. Since the data fragments of each sample file are distributed independently across several computing nodes, the same set of analytical tools will process these data fragments simultaneously in parallel, establishing data parallelism. The analytical pipelines, illustrated in Fig. 1, allow each read of a sample to get processed independently through the pipelines, introducing task parallelism, which incorporates further parallelism in each computing node of the model. In addition, since each computing node handles input fragments originating from different samples, the same tool can simultaneously process these different input splits in parallel. The workflow model designed in this work will realize these pipelines through serial execution of three phases based on Map-Reduce paradigm. Phase BaseProfiler primarily focuses on generating a DNA profile of an individual human genome,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

and processes the NGS related alignment files, in the form of input splits, to generate an overall information corresponding to each base position of the genome. Phase SeqClust will generate integrated profiles based on the profiles of multiple individuals in the population, i.e., to generate a repository of genomic segments, associated with a cluster of samples sharing the segment, while phase BaseProfiler processes the output of the first Map-Reduce phase, to generate repository of contigs or segments having similar sequence, and biological features or patterns.

2.1 Phase BaseProfiler: Generating per base DNA profile of an individual in Map-Reduce paradigm

In this phase of the work-flow model, NGS based alignment files are processed in a Map-Reduce style to summarize information associated with each genomic base position of each individual in the population. This phase is primarily divided into map and reduce tasks. On each of the input splits, the same map task will be performed in parallel. The output of each map task will then be accumulated to reducer nodes, where the reduce task will be performed in parallel. The overall flow of execution steps of Map-Reduce phase BaseProfiler is shown in Fig. 2.

2.1.1 Map task

Map task is performed in parallel on each input split on each of the slave nodes. It introduces data parallelism across the slave nodes. This data parallelism may also be invoked in a single slave node processing several input splits. The Map Task is divided into 3 subtasks: Mapper, Combiner and Partitioner. Now, we describe 3 procedures corresponding to these 3 subtasks, which are executed on every split on each of the slave nodes.

Mapper procedure

Mapper procedure receives $< key, value >$ in the form of $< k_{1M}, v_{1M} >$ pairs as input, generated through preprocessing the alignment information files. For each read, the combination of reference-id, sample-id, start-coordinate and end-coordinate form an input key (k_{1M}), while value (v_{1M}) comprises sequence quality, cigar, read sequence and mapping quality. Since the values of these fields are differently represented in various file formats, a conversion of these values to a consistent form is necessary. The main purpose of mapper procedure 1 is to isolate base level information of all the reads and represent them in terms of key K_{1M} values. Each K_{1M} value comprises reference-id, sample-id, base-position, nucleotide, base quality and cigar. The alignment information of each base of a read, pertaining to a particular individual, is extracted from the alignment information of the read. Thus, a key K_{1M} in the output $< key, value >$ pair will represent the association of a nucleotide (i.e., A, T, C, G) with its aligned position at the reference genome, along with an integrated quality score and cigar of the nucleotide all of which are obtained by isolating the alignment information of the nucleotide from its parent read. The integrated quality score, formulated in this article, is defined based on two different types of quality information, i.e.,mapping quality and the sequencing quality. The corresponding value V_{1M} indicates the initial frequency or count of occurrence of this nucleotide at the base position. For each base of a record (read) in an input, the value V_{1M} is set to 1.

It is to be mentioned here that the integrated base quality in K_{1M} assumes any of four discrete values 0, 1, 2 and 3. This value is obtained by combining both mapping quality of a read and the sequence quality of a base in the read. Mapping quality provides the chance that the read is misaligned in the genome, while sequence quality provides base wise quality achieved during the chemical process. The quantities $10^{-seqQual}$ will be used as estimated probability of nucleotide being chemically sequenced incorrectly, where seqQual denotes phred score representing the sequencing quality of the base, as obtained from the alignment information of the read. On the other hand, $10^{-mapQual}$ of the whole read, containing the nucleotide base, will be used as an estimated probability of incorrect alignment of the read, where mapQual stands for the mapping quality of the read. Thus a smaller value for both the estimated probability will reflect a better sequencing and alignment of the read. As both the events of sequencing a read and aligning a read are independent of the other, thus the joint probability (JP) will be estimated as a product of these two estimated probabilities, using the following equation: $JP = 10^{-seqQual} \times 10^{-mapQual}$ This estimated joint probability (JP) of a base pair denotes the probability of the corresponding nucleotide being incorrectly sequenced and mapped. A transformation of this value JP into phred scale will be done, which will be referred to as base quality of the nucleotide in the Algorithm 1. This phred value will be discretised into four different categories, as to facilitate summarization in the reducer phase. Finally $< K_{1M}, V_{1M} >$ values constitute the output of a mapper procedure. It may be mentioned here that one may retain seqQual and mapQual values in K_{1M} or V_{1M}, if required. In that case, discretization needs to be done appropriately so that combiner procedure can be executed conveniently.

Combiner procedure

In combiner procedure, all the $< K_{1M}, V_{1M} >$ values emitted from the above mapper procedure will be combined on the basis of identical K_{1M} values. In this procedure let us represent these $< K_{1M}, V_{1M} >$ pairs as $< k_{1C}, v_{1C} >$, an input to combiner procedure. In other words, $< k_{1C}, v_{1C} >$ values act as input to the combiner procedure.

Now, a grouping of $< k_{1C}, v_{1C} >$ pairs with the same k_{1C} value will take place. A single $< key, value >$ pair will be generated against each group, in the form $< K_{1C}, V_{1C} >$, where the value of K_{1C} is the same as k_{1C} values of the group, and V_{1C} is the total number of such pairs, i.e., sum of corresponding v_{1C} values. In other words, V_{1C} represents the total number of reads that have got aligned to the same reference (i.e., k_{1C}.reference-id) at the position k_{1C}.base-position, with a particular nucleotide (i.e., k_{1C}.nucleotide) of similar integrated base-quality and cigar information. These $< K_{1C}, V_{1C} >$ values form the output of combiner procedure.

Partitioner procedure Let the entire range of genomic coordinates be partitioned into n non-overlapping partitions/intervals $[p, q]$, $1 \leq p, q \leq M$, $p < q$, where the positions of the nucleotides in the whole genome lies in $[1, M]$. For each partition, there will be a reducer node, as described in the next section, for performing reduce task. In partitioner procedure (2), let us represent each $< K_{1C}, V_{1C} >$ pair (obtained by combiner procedure), as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

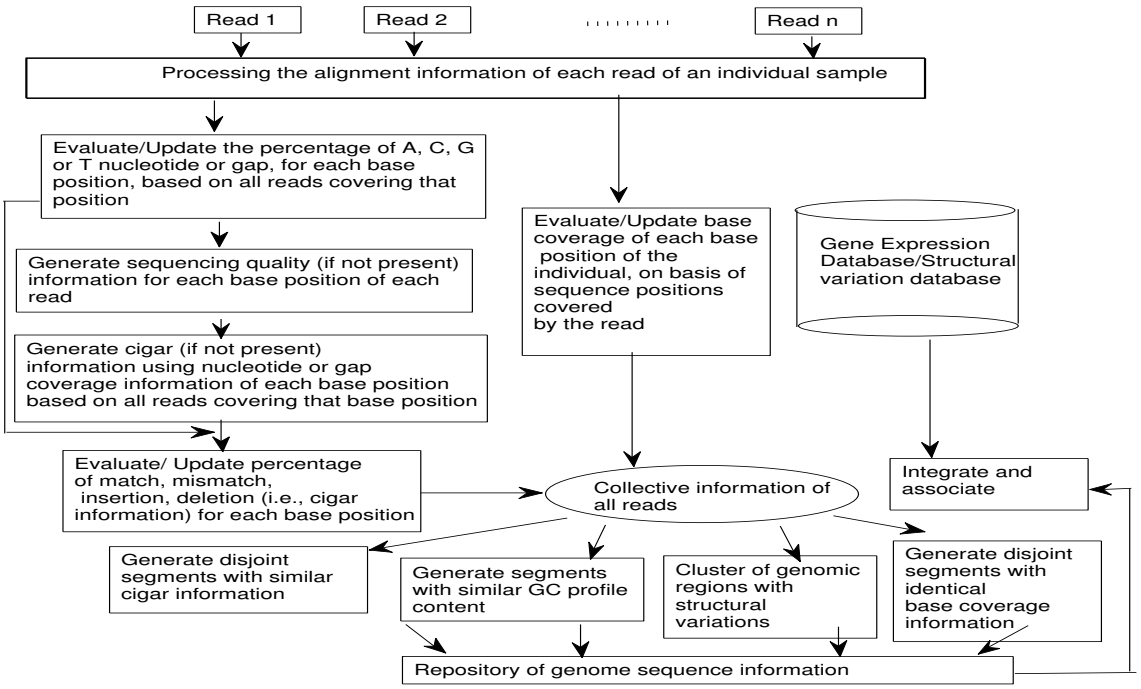

Fig. 1. The execution workflow of the analytical pipelines.

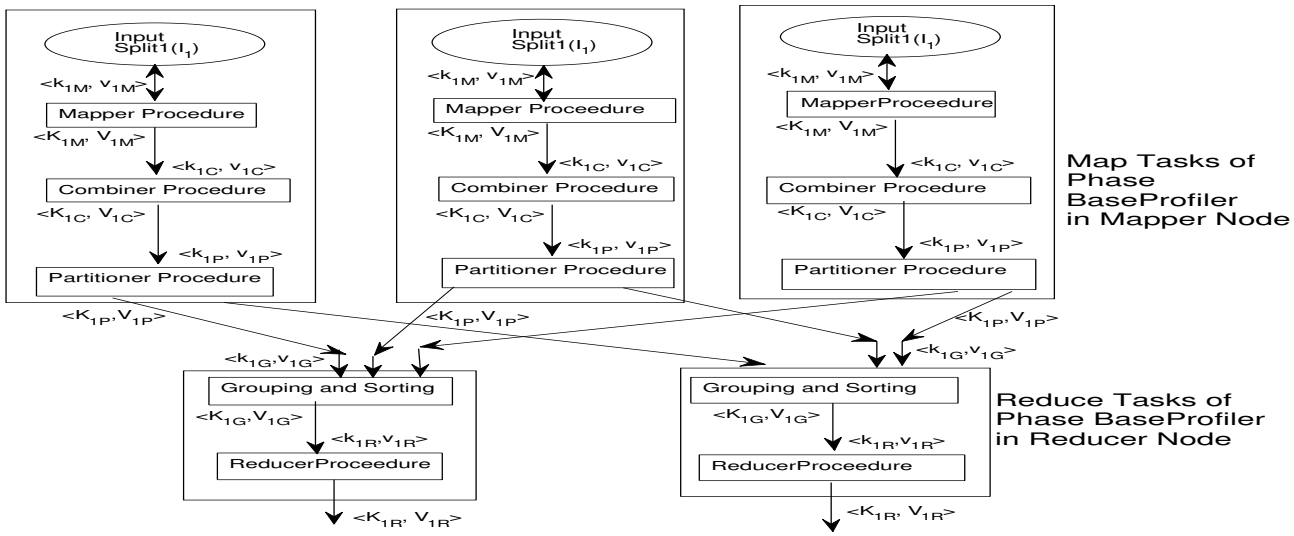

Fig. 2. The execution flow of Map-Reduce phase BaseProfiler. Each Map Task, i.e., the same mapper, combiner and partitioner procedure executes in parallel in all the local nodes where the input splits are stored. The output of Map Task gets distributed over the nodes, where the Reducer Task is performed.

$< k_{1P}, v_{1P} >$, which will act as input to this procedure. Each $< k_{1P}, v_{1P} >$ pair is assigned to a reducer node depending upon k_{1P}.base-position value. If this k_{1P}.base-position value lies in $[p, q]$, $< k_{1P}, v_{1P} >$ pair will be assigned to the corresponding reducer node. In this way the reduce task will get distributed over the commodity hardware. The value pairs $< K_{1P}, V_{1P} >$ form output of partitioner procedure. Here K_{1P} value is the same as that of k_{1P}, while $V_{1P} = v_{1P}$ corresponding to a reducer node.

2.1.2 Reduce task

The reduce task is divided into following 2 subtasks: Grouping and Sorting (Grouper procedure), and Reducing (Reducer procedure). The outputs of partitioner procedures executing in all the slave nodes, in the form of $< K_{1P}, V_{1P} >$ pairs, are emitted in parallel by several instances of the map task through the respective partitioner, will be accumulated in the reducer nodes, on the basis of k_{1P}.baseposition values. Hence, $< K_{1P}, V_{1P} >$ represent all those pairs accumulated in a reducer node corresponding to an interval $[p, q]$.

Grouper procedure

In each reducer node, the collection of different data accumulated from the different partitioner procedures is grouped by grouper procedure. In this procedure let us represent $< K_{1P}, V_{1P} >$ in the form of $< k_{1G}, v_{1G} >$, which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Algorithm 1 Mapper procedure: Mapping and isolation of base level information.

1: Set i=1.
2: **for** each record r in a split, where r represents the alignment information of the read corresponding to the record **do**
3: Input $< k_{1M}, v_{1M} >$ for r
4: **for** each base-position $b = r$.start-coordinate to r.end-coordinate of alignment **do**
5: Extract nucleotide from sequence of record r, which has been aligned at the base position.
6: Extract base-quality of the nucleotide of record r, which has been aligned at the base position.
7: **if** base-quality lies in $[0, 10]$ (in Phred Scale) **then**
8: set base-quality $= 0$.
9: **end if**
10: **if** base-quality lies in $[11, 20]$ (in Phred Scale) **then**
11: set base-quality $= 1$.
12: **end if**
13: **if** base-quality lies in $[21, 30]$ (in Phred Scale) **then**
14: set base-quality $= 2$.
15: **end if**
16: **if** base-quality lies in $[31, 40]$ (in Phred Scale) **then**
17: set base-quality $= 3$.
18: **end if**
19: Extract cigar-information from record r, which corresponds to the base position b.
20: Emit a key value pair $< K_{1M}, V_{1M} >$, where key K_{1M} comprises $<$ reference-id, base-position b, nucleotide, cigar-information of b, sample-id, integrated base-quality $>$, and $V_{1M} = 1$ (initial count of occurrence of the key K_{1M}).
21: Increase i by 1.
22: **end for**
23: **end for**

act as input to the grouper procedure. Each group G_b will contain several $< k_{1G}, v_{1G} >$ pairs with identical k_{1G}.base-position values b. Thus, several groups will be formed for different b values, which lie in $[p, q]$ corresponding to the reducer node. The output of grouper procedure is of the form $< K_{1G}, V_{1G} >$, where $K_{1G} = k_{1G}$ and $V_{1G} = v_{1G}$.

Reducer procedure

The grouper procedure forms groups of $< K_{1G}, V_{1G} >$ pairs with identical K_{1G}.base-position values, which become input $< k_{1R}, v_{1R} >$ to the reducer procedure, where $k_{1R} = K_{1G}$ and $v_{1R} = V_{1G}$. The reducer procedure (3) will process each group G_b. It will calculate the statistics involving base coverage and nucleotide coverage of the corresponding base position, on the basis of the $< k_{1R}, v_{1R} >$ values of each group. In addition, a summarized statistics of cigar and discretized base quality information, associated with a base position, will also be evaluated for each group representing a base position. Finally, a reducer node will produce a $< key, value >$ pair corresponding to each group G_b, of the form $< K_{1R}, V_{1R} >$ as output, where the key K_{1R} is the combination of reference-id, base-position b and sample-id. The value of V_{1R} corresponding to group G_b comprises the values of all the parameters computed in lines 4-16 of Algorithm 3. Thus the key K_{1R} of the output $< key, value >$ pair represents a particular base position of an individual sample, whose alignment information with respect to the reference sample has been summarised in the associated value V_{1R}. The summarised values in V_{1R} comprises the frequency count or occurrence of each possible nucleotide at the base position for the sample, as obtained from alignment information of all the reads. In addition, the summarisation of integrated base quality and cigar will also be evaluated for each of their possible category, as components of V_{1R}.

Algorithm 2 Partitioner procedure: Partitioning each $< k_{1P}, v_{1P} >$ pair depending upon its k_{1P}.base-position value.

1: **for** each $< k_{1P}, v_{1P} >$, generated in combiner procedure, **do**
2: **if** k_{1P}.base-position lies in $[p, q]$ **then**
3: Assign/Emit $< K_{1P}, V_{1P} >$ pair to $node_j^{pq}$ (j^{th} reducer node representing genomic interval $[p, q]$), $1 \leq j \leq n$), where $K_{1P} = k_{1P}$ and $V_{1P} = v_{1P}$.
4: **end if**
5: **end for**

2.2 Phase SeqClust: Clustering samples having the same nucleotide sequence/patterns in various genomic segments

Most of the alignment techniques consider a traditional reference sequence without considering the diversity of genomes, and therefore, alignment results incur error [26]. Thus developing a multiple genome based reference sequence on the basis of similar class is essential to get better performance of any NGS based analytical process which depends on the result of short read alignment. Phase SeqClust will devise another reduce operation, where the genomic profiles of several individuals accumulated as output of phase BaseProfiler, will be integrated to generate a set of genomic contigs or patterns that are commonly found in several sets of individuals. A set is formed on the basis of some common features, such as ethnicity, presence of similar diseases, or presence of some similar phenotypes. The output of this phase, in the form of contigs containing integrated DNA profile across multiple samples, would act as a rich knowledge base to different alignment tools, most of which suffer from biases induced from usage of a traditional reference sequence.

All the base pair level information of each individual obtained from phase BaseProfiler becomes the input to this phase. In other words, the DNA profiles of all the samples collectively constitute the input to the phase. The primary objective of this phase is to generate a repository comprising genomic segments of varying size, where a segment (with a particular nucleotide pattern) is present in a set of samples. It may happen that a subsegment of the segment is also present in a set of some other samples. On the other hand, each individual will also be associated with many segments, establishing a many-to-many association between segments and samples (Fig. 3).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

Algorithm 3 Reducer procedure: Reducing each group G_b.

1: **for** each group G_b, corresponding to each base position b lying in the genomic interval $[p, q]$ of a reducer node **do**
2: Input $< k_{1R}, v_{1R} >$
3: **for** all $< k_{1R}, v_{1R} >$ in G_b **do**
4: Generate base-cov = $\sum v_1 R$.
5: Generate base-nucleotideA-cov= $\sum v_{1R}$, such that k_{1R}.nucleotide='A'
6: Generate base-nucleotideT-cov= $\sum v_{1R}$, such that k_{1R}.nucleotide='T'
7: Generate base-nucleotideC-cov= $\sum v_{1R}$, such that k_{1R}.nucleotide='C'
8: Generate base-nucleotideG-cov= $\sum v_{1R}$, such that k_{1R}.nucleotide='G'
9: Generate base-cigar-match= $\sum v_{1R}$, such that k_{1R}.cigar='match'
10: Generate base-cigar-mismatch= $\sum v_{1R}$, such that k_{1R}.cigar='mismatch'
11: Generate base-cigar-deletion= $\sum v_{1R}$, such that k_{1R}.cigar='deletion'
12: Generate base-cigar-insertion= $\sum v_{1R}$, such that k_{1R}.cigar='insertion'
13: Generate base-quality-0= $\sum v_{1R}$, such that k_{1R}.base-quality$= 0$
14: Generate base-quality-1= $\sum v_{1R}$, k_{1R}.base-quality$= 1$
15: Generate base-quality-2= $\sum v_{1R}$, k_{1R}.base-quality$= 2$
16: Generate base-quality-3= $\sum v_{1R}$, such that k_{1R}.base-quality$= 3$
17: **end for**
18: Emit $< K_{1R}, V_{1R} >$ pair as output
19: **end for**

Thus it requires a nested data structure to store information on samples associated with the segment as well as that on additional samples, if any, being associated with a subsegment in a non-redundant manner. The novel data structure, introduced here for this purpose, will serve diverse information requirement, involving single or group of individuals. On one hand, the storage structure will provide the whole genome based information of any individual in the form of collection or set of genomic segments or contigs. On the other hand, the same data structure will also provide information regarding several clusters of individuals and their inter relationship in the context of sharing similar genomic segments or contigs, and vice versa.

Algorithm 4 will determine the association between genomic segments and the individuals sharing common pattern in those segments. The method will be based on a bottom up approach of forming larger genomic segments from smaller ones, where each genomic segment will be associated with an interval of genomic coordinate and all the nucleotide sequences (patterns) associated in that interval. The method will treat each genomic segments as a group and all possible nucleotide sequences corresponding to the group or segment as subgroups, where each subgroup is shared by a set of individuals. In other words, multiple individuals having the same nucleotide pattern will be the

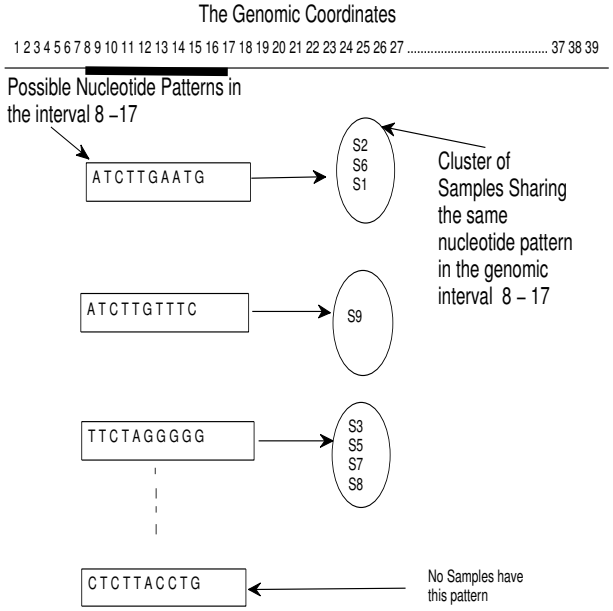

Fig. 3. Clustering of samples based on the nucleotide patterns being shared in a genomic segment. Here the segment considered has genomic coordinates in the interval[8, 17]. The first pattern 'A T C T T G A A T G' is shared by samples S2, S6 and S1, i.e., all these samples have the said pattern in the interval [8, 17]. Thus, a cluster of samples will be associated each pattern. In this way a repository is generated with all possible segments across the genome, along with all possible nucleotide patterns, where each is shared by a set of samples.

members of the subgroup, under the group associated with a particular genomic interval. For instance, $G^m_{x,y}$ denotes an mth subgroup, associated with a unique nucleotide pattern or subsequence that is present in genomic interval $[x, y]$ in some individuals.

The method will start initially with subgroups associated with genomic intervals or segments of size 1 as shown in Fig. 4.

Eventually, each subgroup, associated with a nucleotide subsequence of particular size will be paired with another, if the genomic intervals associated with these subgroups of the pair are adjacent to each other with no-overlap (Fig. 5).

Set intersection operation to find out the common individual present in both the pairing subgroups will be done to obtain a new subgroup that will represent a genomic interval or segment of double size (even length, i.e., $2 \times size$) with respect to that of a subsequence pattern, associated with a parent subgroup. This new subgroup, associated with a larger genomic segment, will be created, provided that the number of individuals or members of this subgroup is more than one. All the individuals or samples of the newly created subgroups will be removed from their parent subgroup, as these individuals will become the member of a larger genomic segment, which contains both the subsequence pattern associated with the paired parent subgroups. A parent subgroup will be removed, in case the subgroup (set) becomes empty, i.e., no individual is a member of the subgroup any more, as each of them is now member of a larger (even sized) genomic segment. If there exists only one member in the parent subgroup, the corresponding subsequence pattern will be embedded into newly created larger genomic segment, using a storage structure as illustrated in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

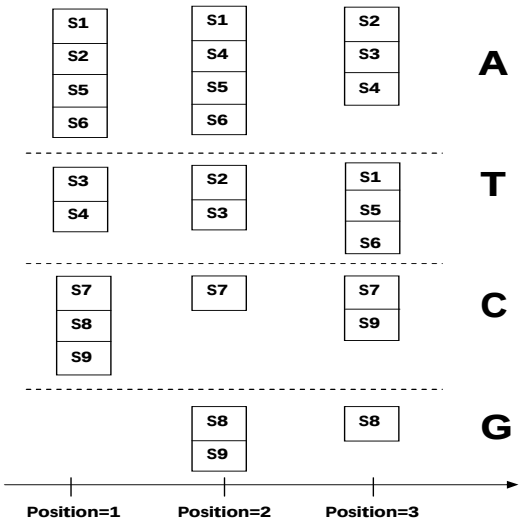

Fig. 4. Initial subgroups associated with genomic intervals or segments of size 1. Here S1-S9 represent sample id's. 'Pos' represents the position or genomic coordinate. The first column represents that at first base position, the samples having nucleotide A are S1, S2, S5 and S6. The samples having nucleotide T at that position are S3 and S4, whereas S7, S8 and S9 have nucleotide C at position 1. It is observed that there are no samples that contain nucleotide G in base coordinate 1. Similarly the second and third column represents the clusters of samples containing a particular nucleotide at base position 2 and 3 respectively.

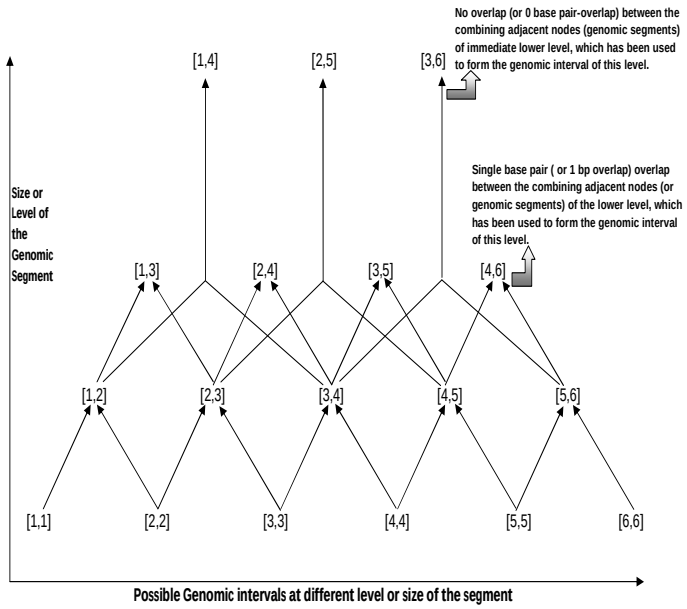

Fig. 5. Bottom up approach of forming larger genomic segments from smaller ones, where each genomic segment will be represented by an interval of genomic co-ordinates. In the first level or bottom most level each segment size is 1. Each subgroup, associated with a nucleotide subsequence of particular size, will be paired with another one, if the genomic intervals associated with these subgroups of the pair are adjacent to each other with no-overlap or having overlap of 1 base pair. In the next level, a new segment of size 2 is formed. For instance, in this level a pairing between the patterns in segments $[2, 3]$ and $[3, 4]$ results in a new pattern in the segment [2, 4]. Each internal node $[a, b]$ in this tree like structure, represents a set of all possible genomic patterns occurring between the genomic positions a and b. Each such pattern in the interval $[a, b]$ is exhibited in some sets of genomes, which are members of the pattern occurring at the interval. As indicated in the figure, each such interval or internal node in the tree structure is a combination of two adjacent nodes in the immediate lower level. These two lower level adjacent nodes either have 1 base pair ovelap or no overlap (0 overlap) in terms of genomic position, as shown in the label of the figure. Precisely, members of a pattern of an interval $[a, b]$ are common members of patterns represented by adjacent pair of interval $[a, floor((a + b)/2)]$ and $[(ceiling((a + b)/2), b)]$ in the lower level.

the Data Structure section described below.

Similarly, for remaining individuals (samples) of these subgroups associated with genomic segments of a particular size, pairing will be done once again, on the basis of adjacency of their corresponding genomic intervals, but this time, with an overlap of 1 base-pair. Set intersection operation on these pairing subgroups will now produce subgroups, representing an odd sized segment, whose length (i.e, $2 \times size - 1$) is smaller than the previously formed even sized segment by one base-pair. As shown in Fig. 5, in a bottom up manner, this process will continue with subgroups of each subsequent higher level, where the level indicates the specific size of the genomic intervals or segments associated with the pairing subgroups of the level. This process will continue till a level (or size) k, for which there will be no genomic pattern (i.e, associated subgroup) that appears in more than one individual (i.e., membership count of subgroup less than 2).

Data Structure: The notion of subgroup, used in Algorithm 4, represents the association between a unique subsequence pattern corresponding to a particular genomic segment, and the corresponding set of individual members exhibiting the whole or part of the sequence pattern at the genomic segment. A novel technique based on data structures, like interval tree and bitmap index structure, will be devised to implement the notion of group or subgroup. Each subgroup will be a $< key, value >$ combination, where key constitutes genomic$-$interval and nucleotide$-$sequence, and value will be a compressed bitmap index of the individual samples. Each bit of the bitmap will represent a sample, and the corresponding value of 0 or 1 in the bit will indicate the membership or non-membership of the individual sample in the cluster corresponding to the key. In other words, bits of bitmap will contain 1 for those individuals, who will be sharing a nucleotide pattern at a genomic segment. The bitmap, being some permutation of 0's and 1's, will be kept in compressed form. It will be more likely exhibiting some sparseness in one of these two states. Thus in most of the cases, the bitmap will be compressed on the basis of sparseness in 0 or 1, and depending upon which, a tag (0-based or 1 based compression) will be associated with the compressed bitmap. The advantage of incorporating a bitmap to represent each cluster of individuals, sharing a nucleotide sequence pattern in a particular genomic segment, will be realised during phase ContigClust, where analysis based on inter-relationship between various clusters will be obtained. The use of bitwise operators among the bitmaps corresponding to various clusters, will allow one to compute the inter relationship between these clusters efficiently.

The $< key, value >$ pair will have an additional feature or capability of embedding multiple sub-genomic intervals into it, where each of these sub-genomic intervals will be completely contained in the key, associated with the genomic interval of the subgroup. In other words, each of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

these $< key, value >$ pairs will act as a nested structure that will contain other $< key, value >$ pairs of other subgroups. The use of interval tree data structure will allow to store all the sub genomic intervals with respect to the key.genomic-interval, in the value part of the $< key, value >$ pair. An interval tree is a height balanced Red-Black Tree, where each node represents an interval, and therefore, an array representation of this tree can be realised efficiently. In addition to all the information maintained in each node of a standard interval tree, a compressed bitmap will also be maintained to track the samples associated with the interval. Thus the value part of $< key, value >$ pair will comprise a compressed bitmap index $bitmap_{interval}$ as discussed above, and an interval tree in form of a list or array of nodes, where each node represents a sub genomic interval and corresponding bitmap. This new form of $< key, value >$ pair has been illustrated in Supplementary Figure S2.

2.3 Phase ContigClust: Generating contigs with similar features in an individual

Phase ContigClust will process the output of phase BaseProfiler to generate a repository of contigs or genomic segments. Phase BaseProfiler provides per base detailed information of each sample, based on the alignment information with respect to a reference sequence. In phase ContigClust, clustering will be performed to generate genomic segments or contigs with similar features, where a feature may denote similarity in the form of structural variations, any biological phenotypes like diseases, read mappability bias, GC-content profile or combination of any of these features, among others. In addition, any customized Map-Reduce job can use the base pair statistics, evaluated in phase BaseProfiler, to generate contigs as per requirement. The output of phase ContigClust will provide contigs or segments of genome exhibiting similarity on the basis of various combinations of base level statistics. These segments or contigs will capture important biological features of different variety, and provide the user a rich set of information for further analyses. This information can be a genomic segment having presence of any type of structural variations, like CNV [27] or SNP [28]. It may reveal a segment with similar cigar pattern. Moreover, the qualitative information of each such segment, in terms of phred scale quality of sequencing, can also be revealed. The output of the phase, i.e., genomic segments or contigs will be stored and indexed, using the notion of multilevel disk block indexing, to provide faster and efficient access to these data. Other automated tools will easily and efficiently access these segments and their associated information, for further analysis through this index. Moreover, the storage mechanism of these segments will provide better interactivity in terms of query evaluation.

2.3.1 Map task

The map task in phase ContigClust will be performed in parallel, on the output of all the reducer nodes of phase BaseProfiler. The output of the reducer procedure of phase BaseProfiler, in the form of $< key, value >$ pairs, will be the input to the mapper procedure (corresponding to map task) of phase ContigClust. In other words, all the base level information become the input. Let us represent the input

Algorithm 4 SeqClust: Combination Procedure (Clustering of samples having similar genomic patterns)

for $size$ = 1 to k **do** [k is the maximum possible size of a segment that is shared by a set of samples (individuals). n is the genome length. $size$ denotes the length of the genomic segment that is used for creating larger segments of length $2 \times size$ or $2 \times size$ -1]
  for overlap = 0 to 1 **do** [Controlling the overlapping (1 base-pair) or non overlapping of the pairing segments while formation of larger segment]
    for position pos=1 to $n - size + 1$ increment by 1 at each step **do**
      l_1=pos, [here l_1 represents the start coordinate of the left segment involved in the pairing]
      r_1=pos+$size$-1, [here r_1 represents the end coordinate of the left segment involved in the pairing]
      l_2=pos+$size$-$overlap$, [here l_2 represents the start coordinate of the right segment involved in the pairing]
      r_2=pos+$2 \times size$-1-overlap, [here r_2 represents the end coordinate of the right segment involved in the pairing]
      Let N_1 and N_2 be the number of possible combination of sequences, i.e, all possible nucleotide patterns in the genomic interval [l_1, r_1] and [l_2, r_2] respectively.
      for i=1 to N_1 **do**
        for j=1 to N_2 **do**
          if $|G^i_{l_1,r_1}| \geq 2$ and $|G^j_{l_2,r_2}| \geq 2$, [$G^i_{l_1,r_1}$ and $G^j_{l_2,r_2}$ represent ith and jth subsequence at genomic intervals [l_1,r_1] and [l_2,r_2] respectively] **then**
            if $|G^i_{l_1,r_1} \bigcap G^j_{l_2,r_2}| \geq 2$ [which implies count of common individuals between the pair.] **then**
              Create $G^t_{l_1,r_2} = G^i_{l_1,r_1} \bigcap G^j_{l_2,r_2}$. [Here t is a counter, tracking the number of possible combination of sequences, i.e, all possible nucleotide patterns in the genomic interval [l_1, r_2], of length ($2 \times$ $size$-$overlap$), where the newly created subgroup have more than one individuals as members]
              $G^i_{l_1,r_1} = G^i_{l_1,r_1}$ - $G^t_{l_1,r_2}$ and $G^j_{l_2,r_2} = G^j_{l_2,r_2}$ - $G^t_{l_1,r_2}$. [After creating the group remove the common members from both of the parent groups]
              if $|G^i_{l_1,r_1}| = 0$ or $|G^j_{l_2,r_2}| = 0$ or both [i.e., $G^i_{l_1,r_1}$ or $G^j_{l_2,r_2}$ is empty] **then**
                Remove $G^i_{l_1,r_1}$ or $G^j_{l_2,r_2}$ or both, accordingly.
              end if
              if $|G^i_{l_1,r_1}| = 1$ or $|G^j_{l_2,r_2}| = 1$ or both [i.e., only one individual in $G^i_{l_1,r_1}$ or $G^j_{l_2,r_2}$, or both] **then**
                Embed group $G^i_{l_1,r_1}$ or $G^j_{l_2,r_2}$, or both into $G^t_{l_1,r_2}$, using the data structure.
              end if
            end if
          end if
        end for
      end for
    end for
  end for
end for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

of map procedure of phase ContigClust as $< k_{3M}, v_{3M} >$ which is the same as $< K_{1R}, V_{1R} >$. The mapper procedure of this phase is depicted in Algorithm 5.

Algorithm 5 Phase ContigClust: Mapper procedure

1: **for** all k_{3M}.base-position (b) **do**
2: Calculate percentage of insertion in the position b, using the data v_{3M}.base-coverage and v_{3M}.base-cigar-insertion.
3: Calculate percentage of deletion in the position b, using the data v_{3M}.base-coverage and v_{3M}.base-cigar-deletion.
4: Calculate percentage of mismatch in the position b, using the data v_{3M}.base-coverage and v_{3M}.base-cigar-mismatch.
5: Calculate percentage of match in the position b, using the data v_{3M}.base-coverage and v_{3M}.base-cigar-match.
6: **Assign X= Class Label 'L', if any, determined by a trained Classifier using base coverage data of position b and percentage data of a particular cigar information 'L'('insertion', 'deletion', 'mismatch' or 'match') covering position b.**
7: Emit $< key, value >$ in the form of $< K_{3M}, V_{3M} >$, where K_{3M} comprises (sample-id, reference-id, base-position), and V_{3M} comprises (base-coverage, Label (X), base-quality).
8: **end for**

The map task will generate $< key, value >$ pairs in the form of $< K_{3M}, V_{3M} >$, where K_{3M} comprises (sample-id, reference-id, base-position), and V_{3M} comprises base-coverage, Label (X), base-quality. For example, if a contig corresponding to K_{3M} includes an insertion event, Category/Label will be appropriately depicted by V_{3M}. This value will be determined on the basis of predefined range of percentage values of the above parameters. Thus the map task will provide the base level statistics of an individual.

2.3.2 Reduce task

The reduce task of phase ContigClust will generate contigs or genomic segments, using the output $< K_{3M}, V_{3M} >$ of corresponding map task as its input, which have been represented as $< k_{3R}, v_{3R} >$. The segmentation will be done on these $< key, value >$ pairs on the basis of similar percentage count of cigar information or individual nucleotide coverage. An instance of reduce task generating contig or genomic segment using CIGAR information of alignment have been explained in Reducer Algorithm 6.

In addition, the quality information will also be processed for each $< key, value >$ pair or base position to associate a probability of correctness of the information associated with that base position. The reducer will generate output in form of $< K_{3R}, V_{3R} >$ pair, after considering similarity using all these information. Here, K_{3R} comprises (reference-id, sample-id, start-coordinate, end-coordinate) of each segment, and V_{3R} represents segment having similar summarized feature. In Algorithm 6 the reducer outputs (contig-nucleotide-sequence,Label) as value that represents a contig with similar CIGAR data exhibited by the sample. Moreover, these genomic segments or contigs will be stored and organized using a multilevel indexing structure. Fig. 6 represents the detailed workflow of phase ContigClust. These genomic segments or contigs will be stored and organized using a multilevel indexing structure.

Algorithm 6 Phase ContigClust: Reducer procedure

1: Perform grouping (g) of all $< k_{3R}, v_{3R} >$ on basis of k_{3R}.Label
2: **for** each sub-group g **do**
3: Sort $< k_{3R}, v_{3R} >$ on basis of k_{3R}.base-position (b)
4: Generate a contig or genomic segment from group g by combining series of $< k_{3R}, v_{3R} >$ using any single pass segmentation algorithm [29],on basis of v_{3R} values of consecutive k_{3R}.base-positions.
5: **for** each of these generated contigs **do**
6: Emit $< K_{3R}, V_{3R} >$, where K_{3R} comprises (start-coordinate, end-coordinate), and V_{3R} comprises (sample-id, reference-id, contig-nucleotide-sequence, feature-annotations).
7: **end for**
8: **end for**

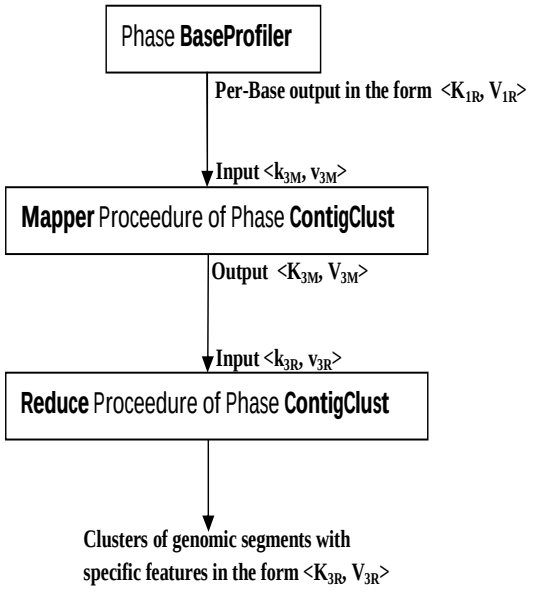

Fig. 6. The execution workflow of Map-Reduce phase ContigClust.

Storage and indexing of genomic segments or contigs: A multilevel index structure similar to B+ tree has been maintained on $< K_{3R}, V_{3R} >$ pairs. These pairs represent several contigs from different individuals. This index structure will be maintained in the local disk of the data node, where all $< K_{3R}, V_{3R} >$ pairs will occupy several local disk blocks in sorted order of their K_{3R}.start-coordinate. These disk blocks of the index structure is categorized into two types, viz., a) Leaf index block b) Internal index block. Each index entry in the leaf index block would contain following three information: *minimum start-coordinate* of the key component among all $< K_{3R}, V_{3R} >$ pairs, stored in the referred data block of the index entry; *maximum end-coordinate* of the key component among all $< K_{3R}, V_{3R} >$ pairs, stored in the referred data block of the index entry; *local disk data*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

block pointer **containing disk address of the referred data block. Similarly, the parent node of several leaf index nodes of the tree will have similar index entries, except, the pointer will refer to disk address of leaf index blocks. The other internal nodes of various level of the tree will also contain similar entries of their immediate lower level index blocks. Since, entries of each leaf index block points to several data blocks of $< K_{3R}, V_{3R} >$ pairs, thus the total number of entries across all index blocks are same as the total number of local disk data blocks. Unlike B+Tree index, the information of each data block will be organized as an interval tree. Since in a data block, $< K_{3R}, V_{3R} >$ pairs belongs to different individuals or samples, hence there is high probability of overlap among the corresponding genomic segments or contigs. Thus an interval tree similar to the data structure used in the phase SeqClust has been used to organize the $< K_{3R}, V_{3R} >$ pairs in a local data block. This novel way of organizing the data block improves data retrieval process**

3 Results and Comparative Performance

The effectiveness of the model has been demonstrated on a prototype using a cluster of commodity computers (PCs). The HDFS File System of Hadoop has been used to reposit data, as described in the model. Thus, Hadoop 2.51 Eco-System has been installed on top of Ubuntu 14.04 LTS operating System in each of these computers. One of these computers has been configured as a master node and the remaining machines as slave nodes. Each machine, either a master or slave node, has a 4 core commodity processor with a clock speed of 3.11 GHz. The master node has a main memory of 16GB, while the slave nodes have 4GB each. The configuration of all the nodes in the cluster are taken to be inexpensive and simple, considering the economic constraints in implementing the model.

3.1 Experimental Data Set

The data considered here comprise a combination of real data set and a simulated data set. Read alignment information of chromosome 1 and 20, obtained in the form of BAM file of sixteen human individuals, has been used as real data set in this experiment. Simulated data set have also been considered for experimentation with this model. Reads of 184 individuals have been simulated by manipulating the reads obtained in the real data set. A read of a simulated individual has been generated from the reads of two real individuals, using the following strategy. To generate a simulated read in the region starting from location x bp to location $(x + y)$ bp of any chromosome (1 or 20), we have combined a substring of a read covering the region x to $(x + \lfloor y/2 \rfloor)$ bp, with the substring of a read covering the region $(x + \lfloor y/2 \rfloor + 1)$ bp to $(x + y)$bp of the same chromosome. The substrings used to form the simulated read has been taken from 2 different individuals. Thus, all possible $m \times n$ combinations or crossover of substrings have been obtained, where m and n are the number of reads covering the region x to $(x + \lfloor y/2 \rfloor)$ bp and the region $(x + \lfloor y/2 \rfloor + 1)$ bp to $(x + y)$ bp of a particular chromosome of two real individuals respectively. After this, 30 reads have been chosen randomly from this set of $m \times n$ combined strings, considering a $30\times$ read coverage of the simulated individual. In addition, every base-pair and mapping information of these chosen reads have been accumulated from corresponding alignment data of the two real samples. Using the above procedure, we have simulated read information of chromosome 1 and 20 of 184 individuals and created BAM file for each of them.

3.2 Results

The prototype of the model provides fast projection of summarized data of any queried genomic region. As an instance, Fig. 7 presents a few summarized information of a particular queried region starting from 2.1 Mb to 3.1 Mb of chromosome 1 of a sample from the real dataset. One of the genomic information called mappability measure [30] or score of each base location is an important parameter, having significant effect on the depth of coverage information of each base or base coverage data. This mappabilty score lies between 0 to 1 , where a higher value indicates a good chance of a unique mapping or alignment of reads to the base. A summarization of mappability score information of the query region from 2.1 Mb to 3.1 Mb of chromosome 1 has been depicted in Fig. 7a. Fig. 7b depicts the base coverage, i.e., number reads mapped or aligned to a particular base position, lying in the query region starting from location 2.1 Mb to 3.1 Mb, where the sample has a high base coverage at a region starting from 2.143mb to 2.246mb due to presence of the CNV. The correlation between mappability score and the z-score of read depth, for each base position of the queried region, across all the samples in the repository has been depicted in Fig. 8. This information can be effectively used by an alignment algorithm to improve its mapping quality, as mappability bias provides an insight on the level of uniqueness of the pattern in the region.

The prototype of the model also projects the summarized data on GC content of reads, aligned to a certain region of the genome. Fig. 7c depicts the average GC content of reads covering each base position of a queried region from location 2.1 Mb to 3.1 Mb of chromosome 1, for all the samples. The GC content level incorporates a bias in the process of NGS read generation. More precisely, it affects the PCR amplification process. This bias during DNA read generation, has significant effect in read depth data of a genome, especially affecting the process of structural variant analysis.The relationship between GC content data and the base coverage data provides a further insight in probing structural variations like the CNV, novel insertion, deletion, etc.

3.3 Performance Comparison

The performance of the model prototype has been evaluated against popular NGS read summarization tools, like GATK and HADOOP-BAM, as these two tools have also addressed the big data aspect of NGS data, and like our model, they are also designed for a generalized purpose to provide information to other NGS based analytical tools for various downstream analyses. Since, both GATK and HADOOP-BAM provide base wise summarized information across the genome, viz., genome traversal (in GATK) or on the fly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

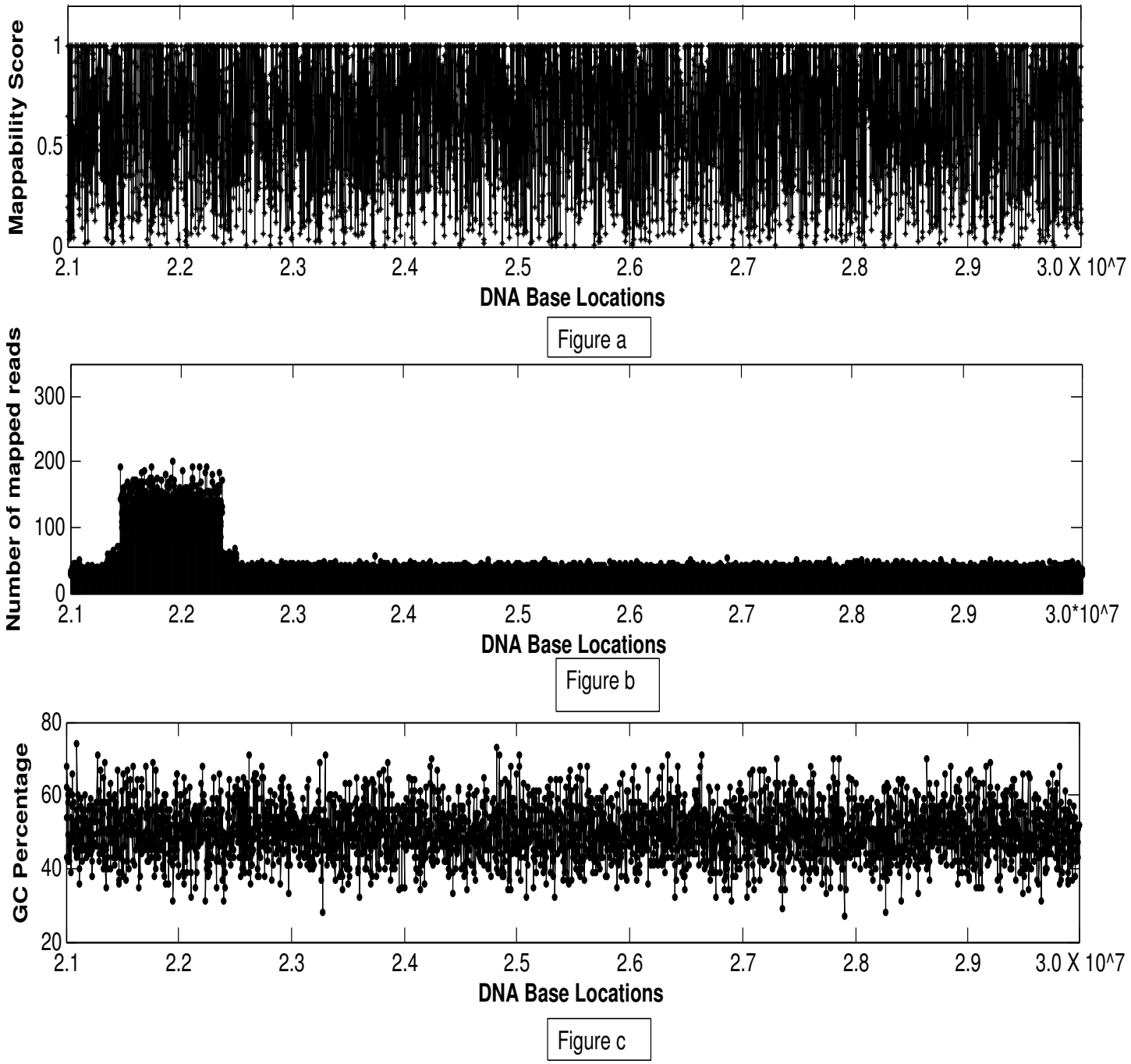

Fig. 7. (a)The figure depicts the average Mappability Score of a query region starting from 2.1 Mb to 3.1 Mb of chromosome 1. (b) The figure represents the base coverage data of each base in the same genomic region. (c) The figure depicts GC Content of reads in percentage, covering each base position of a particular region of chromosome 1, for all the samples.

base coverage calculation (in HADOOP-BAM), this base-wise information retrieval is chosen as a common task, to be used for performance evaluation of these three competing tools, including the present one. This base wise information retrieval or genome walk will provide summarized data, such as depth of coverage, mappability bias, GC-content Profile, and RPKM [31] (Reads per kilo base per million mapped reads) of each base or region, falling under the walk. Here the performance issue is dictated by the time required to fetch or generate information of a base or region, falling under the genome walk, so that the downstream analytical tools can consume this data quickly for further analysis.

We have used execution time speed up as the performance parameter for all these competing tools. Here, execution time speedup is defined as $SpeedUp = (ET(n,1))/(ET(n,r))$, where $ET(n,1)$ and $ET(n,r)$ are the average execution time required for retrieving information of n base pairs by any one of these competing tool in a uni-processor system and r-processor Hadoop cluster respectively.

In Fig. 9, this speed up represented in the vertical axis

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

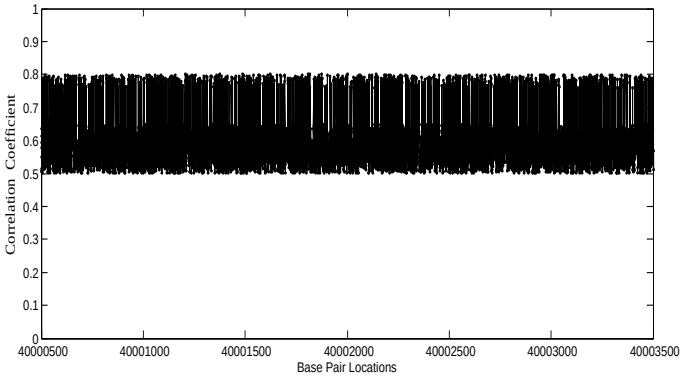

Fig. 8. The Figure represents the correlation of mappability bias with the z-score of read depth, for each base of the region, across all the samples in the repository.

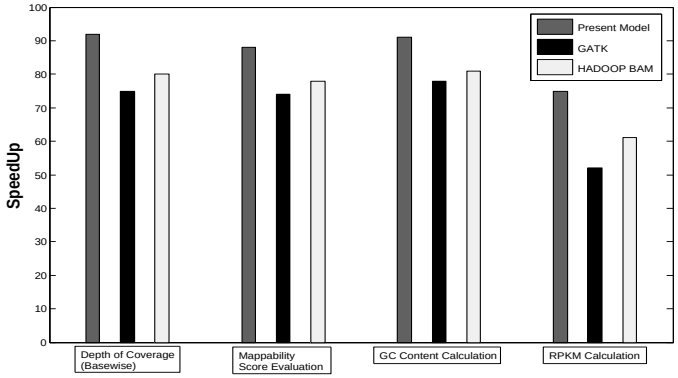

Fig. 9. The performance comparison of the present model with HADOOP-BAM and GATK.

depicts the performance of each of these competing models in retrieving or generating some important genomic information, like depth of coverage, mappability bias and RPKM. The performance of the present model has been found to be superior, in terms of speedup, over both HADOOP-BAM and GATK tools. In comparison to GATK or HADOOP-BAM, the present model will eventually produce more accurate summarized information, with continuous evolution of genomic repository. **Unlike GATK and HADOOP-BAM, integrated information, obtained by the phases SeqClust and ContigClust, from multiple genomes with various ethnic origins will enable other analytical tools to use the data more accurately due to their superior overview of the data. The evolving data repository will facilitate the work-flow modules of the present model to enhance its performance with respect to the competing tools, due to eventual growth of knowledge base. In addition, the use of specialized data structure based on interval tree in the model will enhance the storage and retrieval mechanism in comparison to the above tools. Moreover, deployment of index-based retrieval mechanism in the present model, in collaboration with HDFS file system has attributed to its superiority in terms of speedup. Thus, the dynamic knowledge base design, along with a supporting fast index based access at local disk, provides an upper edge to the present model against the competing ones.** In addition, the Hadoop setup used in this experimentation is quite inexpensive and commodity type. However, the present model would also be capable of outperforming the competing tools in high end cluster setup.

4 Applications

The present theoretical model can be used as a huge repository of alignment information in two different forms generated by: i) phase SeqClust and ii) phase ContigClust. They will provide information for several types of analytical processes based on the Map-Reduce paradigm to do the following types of analyses.

Analyzing association between base coverage and distribution of nucleotides at each base position: Mapper procedure of phase ContigClust evaluates the percentage of different nucleotides covering each base position as depicted in Algorithm 5. This information can be used to perform analyses, like determining the distribution of G,C or A,T nucleotides, and its possible relationship with the corresponding base coverage information, at each base position. Thus this information will help all NGS based analyses that require consideration of mappability bias information, caused by abundance or deficiency of A, T or G, C combinations in a genomic segment. In addition, the clusters created in phase SeqClust, on the basis of similar nucleotide sequence of a genomic segment, can be combined with these feature based analytical information of phase ContigClust, to obtain new association or relationship between nucleotide pattern and mappability bias among multiple individuals.

Determining presence of SNPs: The output of phase SeqClust combining with that of phase ContigClust, which contains the base wise statistics related to cigar information of different genomic segments, can provide a better insight into the patterns of structural variations and its association with other clusters of individuals. Information containing the relative presence of each cigar information (such as insertion, deletion, mismatch or match), in terms of percentage, associated with a base position as obtained from phase ContigClust, can be used to find similar structural variations in other individuals, using the clusters obtained from phase SeqClust. Analyzing this cigar information along with the nucleotide coverage information, associated with a base of different samples of the cluster, will help in determining presence of SNPs. In addition, the analysis of the associated base quality information will provide a better insight regarding the true positive presence of an SNP. For instance, if the percentage count of base-cigar-mismatch at base-position b is greater than a certain threshold value and associated percentage count of base-nucleotideX-cov is greater than a certain threshold value with higher base quality, then the base may be categorized or labelled as SNP. The cause-effect relationship between SNP and disease can also be extracted by analysing the clusters depicting the structural variations in the form of SNP.

Qualitative analysis of the sequencing process: Qualitative analysis of the sequencing process can be performed, using the base-quality information, to explore any pattern of relationship between DNA sequence and the quality of data. Moreover, analysis of quality information associated with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2816022, IEEE/ACM Transactions on Computational Biology and Bioinformatics

that base will provide support to other type of analytical results obtained for that base position.

Analysing mappability bias: Mappability bias is another interesting feature that can be analyzed using the base level information. Presence of G or C nucleotide in a genomic region may influence the read generation from that region, which gets reflected in the coverage information, derived from read alignment. Analyzing the relationship between coverage information of a base position with the percentage count of G or C nucleotide in the reads, aligned to that position, will provide an insight into this aspect. This mappability bias is an important parameter used by analytical tools for finding large structural variations, including copy number variation, insertion or deletion of sequence. This information can be associated with cluster of the samples depicting similar genomic pattern, as obtained from phase SeqClust and the corresponding coverage of these samples, the map task processing of phase ContigClust evaluates the percentage of different nucleotides covering each base position as depicted in Algorithm 5. This information can be used to perform analyses, like determining the distribution of G,C or A,T nucleotides, and its possible relationship with the corresponding base coverage information, at each base position. Now, the reducer procedure (of phase ContigClust) generates repository of contigs or genomic segments using $< key, value >$ pairs of the above map task. The repository provides genomic segments with similar read coverage and their association with GC content profile. This information will provide an insight into the genomic regions suffering from mappability bias [32]. Hence these regions can be taken care off and further be normalized.

Determining the presence of Structural Variations - Copy number variations: Copy number variation (CNV) is a type of genomic structural variation, which is caused by either duplication or deletion of a large DNA segment. The presence of CNVs affects the human health, and is associated with some neurological diseases as well as cancer. Through the repository of genomic segments, generated in reducer process of phase ContigClust, the presence of copy number variation, insertion of novel sequence or deletion, can be identified. The regions with similar cigar-info (insertion or deletion), and base-coverage with high deviation will support the presence of such structural variations. For instance, if there exists a large segment with the percentage count of base-cigar-insertion greater than a certain threshold value for each base-position of that segment, and associated percentage count of base-coverage of each base of that segment is greater than a certain threshold value, then the segment may be categorized or labelled as CNV. Presence of common CNVs or rare CNVs can be determined by combining the information associated with cluster of individuals, as obtained from phase SeqClust. In addition, presence of a particular CNV in an ethnic race and effect of evolution can also be studied by analysing the clusters of the phase, and their association with the segments of phase ContigClust.

Association between disease and genomic information: Recent study shows that genomic structural variation is one of the major cause of many diseases. Analysis of genomic information as available in the repository, will help in exploring genomic features commonly associated with disease-affected individuals. The extraction of disease associated genomic features, such as any structural variation, presence or absence of some nucleotide sequence pattern, will provide biologists and pharmacists a better insight regarding the study of a disease, or related drugs and medicines.

Predictive analysis regarding several unknown features of an individual's genome: The model will also allow to predict unknown features of an individual, through predictive analysis of information available in the repository. As an instance, an individual with unknown ethnic origin, can be predicted by the model, by investigating presence of some genomic features, commonly associated with individuals of different ethnic group. On the other hand, this analysis can also be used to predict or diagnose the occurrence of some diseases in an individual.

Prediction of Ethnicity of an individual of unknown origin: The model along with the support of the repository will facilitate the use of machine learning tools to predict the ethnic-similarity of an individual of unknown origin, through analysing of the known features of a sample. It will be done by developing a decision support system that can evaluate the similarity in genomic features exhibited by a particular ethnic group with that of a sample (individual) of unknown ethnic origin. Based on such analysis, the ethnic similarity of the sample (unknown origin) can be determined and further prediction regarding unknown features of the sample can be done, which were exhibited by the similar ethnic groups.

Integrating External databases:

(a) Integration of external databases, where genomic coordinates of the segments with structural variations are reported, will further enrich the DNA feature analyses of these segments overlapping with those available in these databases. New features or patterns will be explored through this integration.

(b) The gene expression data, associated with different genomic segments in the respective databases, can also be integrated with the phase SeqClust and phase ContigClust of the model. This will help in exploring interesting relationships between variations in the segment and gene expression level.

(c) This integration can facilitate predictive analysis regarding unknown features of samples, by using machine learning based techniques on the output of phase SeqClust and phase ContigClust.

5 Conclusion and Discussion

The primary objective of this work is to develop a model that will provide big data solutions to huge volume of NGS alignment data and create an appropriate repository, in order to facilitate analyses on these data in a big data platform. The framework is modularised into three phases based on Map-Reduce paradigm. Each of these phases has pre-defined objectives, and provides output which can be used by subsequent phases to perform further analyses. Moreover, in this work, different scope of analyses that can be performed in this framework, have also been discussed in the analysis part.

A comparative study of the model has been done by evaluating its performance against two competing tools,

viz., HADOOP-BAM and GATK. The experimental data used for performance comparison, was a mixture of both simulated data and publicly available real data. Base wise summarized information has been retrieved for each of these competing models. The execution time speed up with respect to a single processor, clearly reveals the better performance of the model in comparison to other two competing tools, however, the clusters used for this purpose was formed with commodity machines(PCs). Thus there remains a scope of improving the model further in a larger scale using high-end machines.

REFERENCES

[1] G. J. Hua and C. L. Hung, *Local alignment tool based on Hadoop framework and GPU architecture*,BioMed Res. Int., vol 7, 2014.

[2] M. L. Metzker,*Sequencing technologies - the next generation*, *JNat. Rev. Genet.*, vol 11, pp. 31-46, 2010.

[3] A. O'Driscoll, J. Daugelaite and R. D. Sleator, *'Big data', Hadoop and cloud computing in genomics*, *J. Biomed. Inform.*, vol 46, pp. 774-781, 2013.

[4] Illumina, *An introduction to next generation sequencing technology*,Available: www.illumina.com/NGS

[5] R. Tripathi, P. Sharma, P. Chakraborty, P. K. Varadwaj *Next-generation sequencing revolution through big data analytics*, *Frontiers in Life Science.*, (Online) Journal homepage: http://www.tandfonline.com/loi/tfls20, 2016.

[6] S. Ghemawat, H. Gobioff, and S. T. Leung *GFS: A Fault-Tolerant File System for Large Distributed Applications 19th ACM Symposium on Operating Systems Principles, Lake George, NY.*, October, 2003.

[7] W. Ding and C. Lin, *Attribute Equilibrium Dominance Reduction Accelerator (DCCAEDR) Based on Distributed Coevolutionary Cloud and Its Application in Medical Records*, *IEEE Trans. Syst., Man, and Cybern: Syst.*, vol 46, 2016.

[8] The apache software foundation, *Hadoop*, Available: https://hadoop.apache.org.

[9] D. Miner and A. Shook, *'Map-Reduce Design Patterns*, Oreilly, 2012.

[10] T. Wang, W. Zhang, C. Ye, J. Wei, H. Zhong and T. Huang, *FD4C: Automatic Fault Diagnosis Framework for Web Applications in Cloud Computing*, *IEEE Trans. Syst., Man, and Cybern: Syst.*, vol 46, 2016.

[11] P. Hu and W. Dai, *Enhancing fault tolerance based on hadoop cluster*, *IJDTA*, vol 7, pp. 37-48, 2014.

[12] Y. Xun, J. Zhang and X. Qin, *FiDoop: Parallel Mining of Frequent Itemsets Using Map-Reduce*, *IEEE Trans. Syst., Man, and Cybern: Syst.*, vol 46, 2016.

[13] M. Meyerson, S. Gabriel and G. Getz, *Advances in understanding cancer genomes through second-generation sequencing*, *Nat. Rev. Genet.*, vol 10, 2010.

[14] M. C. Schatz, *CloudBurst: highly sensitive read mapping with MapReduce*, *BIOINFORMATICS.*, vol 25, 2009.

[15] L. Pireddu, S. Leo and G. Zanetti, *SEAL: a distributed short read mapping and duplicate removal tool*, *BIOINFORMATICS.*, vol 27, 2011.

[16] R. V. Pandey and C. Schltterer, *DistMap: A Toolkit for Distributed Short Read Mapping on a Hadoop Cluster*, *PLOS One.*, vol 8, 2013.

[17] B. Langmead, M. C. Schatz, J. Lin, M. Pop and S. L. Salzberg, *Searching for SNPs with cloud computing*, *Genome Biology.*, vol 10, 2009.

[18] *Crossbow: Whole Genome Resequencing using Cloud Computing*.[http://bowtie-bio.sourceforge.net/crossbow/].

[19] R. Li, Y. Li, X. Fang, H. Yang, J. Wang and K. Kristiansen, *SNP detection for massively parallel whole-genome resequencing.*, *Genome Res.*, vol 19, 2009.

[20] L. Wang, Y. Si, L. K. Dedow, Y. Shao, P. Liu and T. P. Brutnell, *A Low-Cost Library Construction Protocol and Data Analysis Pipeline for Illumina-Based Strand-Specific Multiplex RNA-Seq*, *PLOS One.*, vol 6, 2011.

[21] B. D. O'Connor, B. Merriman and S. F. Nelson, *SeqWare Query Engine: storing and searching sequence data in the cloud*, *BMC Bioinformatics*, vol 11, pp. 1471-2105, 2010.

[22] B. Langmead, K. D. Hansen and J. T. Leek, *Cloud-scale RNA-sequencing differential expression analysis with Myrna*, *PLOS One.*, vol 8, 2013.

[23] A. McKenna, *et al.*, *The genome analysis toolkit: A mapReduce framework for analyzing next-generation DNA sequencing data*, *BMC Bioinformatics*, vol 20, pp. 1165-1173, 2010.

[24] M. Niemenmaa, *et al.*,*Hadoop-BAM: directly manipulating next generation sequencing data in the cloud*, *Bioinformatics*, vol 28, pp. 876-877, 2012.

[25] The SAM/BAM format specification working group. *Sequence alignment/map format specification*, Available: https://samtools.github.io/hts-specs/SAMv1.pdf.

[26] R. Li, Y. Li, X. Fang, H. Yang, J. Wang and K. Kristiansen, *Simultaneous alignment of short reads against multiple genomes.*, *Genome Biology.*, vol 10, 2009.

[27] P. Stankiewicz and J. R. Lupski, *Structural variation in the human genome and its role in disease*, *Annu Rev Med.*, vol 61, pp. 437-455, 2010.

[28] N. J. Schork, D. Fallin and J. S. Lanchbury, *Single nucleotide polymorphisms and the future of genetic epidemiology*, *Clin Genet.*, vol 58, pp. 250-264, 2000.

[29] R. Sinha, S. Samaddar and R. K. De, *CNV-CH: A Convex Hull Based Segmentation Approach to Detect Copy Number Variations (CNV) Using Next-Generation Sequencing Data*, *PLoS ONE*, 10(8): e0135895. doi:10.1371/journal.pone.0135895, 2015.

[30] Y. Benjamin, T. P. Speed *Summarizing and correcting the GC content bias in high-throughput sequencing*, *Nucleic Acids Research*, 2012, Vol. 40, No. 10.

[31] D. Ramskld, E. T. Wang, C. B. Burge and R. Sandberg, *An Abundance of Ubiquitously Expressed Genes Revealed by Tissue Transcriptome Sequence Data*, *PLOS Computational Biology*, vol 5, 2009.

[32] J. P. Szatkiewicz, W. Wang, P. F. Sullivan and W. Sun, *Improving detection of copy-number variation by simultaneous bias correction and read-depth segmentation*, *Nucleic Acids Res.*, vol 41, pp. 1519-32, 2013.

Mr. Sandip Samaddar is currently working as an Assistant Professor in the Department of Computer Science and Engineering, at Heritage Institute of Technology, Kolkata. He obtained his M.Tech. degree from National Institute of Technical Teachers Training and Research, Kolkata, in the year 2008. He has 10 years of teaching experience. He has published 2 International journals. His research interest includes Computational Biology, Pattern recognition, Data mining and Algorithms in Big data paradigms.

Ms. Rituparna Sinha is currently working as an Assistant Professor in the Department of Information Technology, at Heritage Institute of Technology, Kolkata. She obtained her M.Tech. degree from National Institute of Technical Teachers Training and Research, Kolkata, in the year 2008. She has 10 years of teaching experience. She has published 4 International Journals. Her research interest includes Computational biology, Data mining, Distributed Databases and Algorithms in Big data paradigms.

Dr. Rajat K De is a Professor of Indian Statistical Institute, Kolkata, India. He obtained his Ph.D. degree from the same Institute in 2000. He was a Distinguished Postdoctoral Fellow at the Whitaker Biomedical Engineering Institute, Johns Hopkins University, USA, during 2002-2003. During the last 10-12 years, Professor De has been working in the area of bioinformatics and in silico systems biology. Recently, he has started working on Big Data Analytics in the domain of bioinformatics and systems biology. He has published about 90 research papers in international journals, conference proceedings and in edited books, and co-edited three books.