Need and Role of Scala Implementations in Bioinformatics

Abbas Rehman
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Muhammad Atif Sarwar
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Ali Abbas
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Javed Ferzund
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

***Abstract*—Next Generation Sequencing has resulted in the generation of large number of omics data at a faster speed that was not possible before. This data is only useful if it can be stored and analyzed at the same speed. Big Data platforms and tools like Apache Hadoop and Spark has solved this problem. However, most of the algorithms used in bioinformatics for Pairwise alignment, Multiple Alignment and Motif finding are not implemented for Hadoop or Spark. Scala is a powerful language supported by Spark. It provides, constructs like traits, closures, functions, pattern matching and extractors that make it suitable for Bioinformatics applications. This article explores the Bioinformatics areas where Scala can be used efficiently for data analysis. It also highlights the need for Scala implementation of algorithms used in Bioinformatics.**

Keywords—Scala; Big Data; Hadoop; Spark; Next Generation Sequencing; Genomics; RNA; DNA; Bioinformatics

I. Introduction

Today, we are living in the world of Big Data. Huge amount of data is being produced on daily basis. Major sources of data include social media, enterprise systems, sensor based applications, Bioinformatics sequencing machines, smart phones, digital videos or pictures and World Wide Web. Big Data's characteristics are Veracity, Velocity, Variety, Volume and Potential Value (these are known as 5 V's). To make this data useful, it needs to be stored and analyzed with accuracy and speed. Traditional techniques are unable to store and analyze such large amount of data. These techniques are better for a limited amount of data analyses as the cost of analysis increases with increment in data volume.

To deal with this hurdle, Big Data platforms and tools are introduced which can analyze a large amount of data with accuracy, speed and scalability. Using Big Data Platforms like Hadoop, cost of analysis is also reduced as it runs on commodity hardware. Major challenges for Big Data are speed, performance, efficiency, scalability and accuracy. Big Data platforms and tools like Hadoop (distributed management System) and Apache Spark (for big data analysis) address these issues. NGS (Next Generation Sequencing) machines bring an evolutionary change in data generation of different sequences. NGS machines are generating a huge amount of sequence data per day that needs to be stored, analyzed and managed well to seek the maximum advantages from this. Existing bioinformatics techniques, tools or software are not keeping pace with the speed of data generation. Old Bioinformatics tools have very less performance, accuracy and scalability while analyzing large amount of data. When storing, managing and analyzing large amount of data which is being generated now a days, these tools require more time and cost with less accuracy.

Apache Hadoop is best Platform for Big Data processing. Hadoop is open source Java Platform that contains thousands of clusters that is used for parallel processing and execution of Big Data. Its main components are Pig, HBase, Hive, HDFS (Hadoop Distributed File System), MapReduce and Apache Spark Framework. Pig is High level language that is used for scripts. It includes load store operators and provides users the capability of creating own built-in-functions (extensible). HBase is used for automatic sharding and sparse data processing by replacing RDBMS (Relational Database Management System). Hive is not used for real time processing but it is used for large analytics and efficient query processing with the help of meta-store unit. HDFS is file system that is developed for processing and execution of large files in database that is created by Hadoop components. Its two units are data node and name node. MapReduce is designed for parallel execution and processing of large datasets in Hadoop Platform. Apache Spark is framework especially designed for Analytics by using the Languages Java, Python, C and Scala. Its main components are caching, action and transformation.

Many Bioinformatics Algorithms are implemented in Scala language for Apache Spark Framework. Scala is functional, statically typed and object oriented language. It is better for concurrent processing. Its main features are traits, closures and functions that are used for processing of multiple Genome Sequencing Algorithms. Scala mostly works like C++ language.

Scala consists of Arrays, Loops, Strings, Classes, Objects, collections, Pattern Matching and Extractors. All of these structures and statements are used for Bioinformatics Algorithmic comparison by Scala in Spark Framework. Scala also contains many Built-in-Methods, Libraries and Functions that are very useful for designing Bioinformatics Algorithms. Scala language plays an imperative role in Bioinformatics Applications.

Genome Sequencing, Motif Finding, Pairwise Alignment and Multiple Alignment are main features for Bioinformatics. Scala language is very important for these Algorithms. In Genome and Multiple Sequencing, a lot of algorithms are used for handling Biological Sequences. These Algorithms are implemented in Scala language. In Apache Spark, Motif Finding Algorithms are implemented using Scala language. In Pairwise Alignment, Scala language is very significant for pattern Matching.

Spark provides the facility of Scala shell for the implementation of these Bioinformatics Algorithms. Primitive Types and anonymous functions in Scala perform well for managing arrangements of Multiple Sequences. Anonymous functions are used in transformations, actions and loading files for Analytics of Bioinformatics datasets in Apache Spark Framework. Shared variables and key-value pairs are used in Hadoop using Scala language for Bioinformatics Algorithms.

For implementing Bioinformatics Algorithms in Scala language on Hadoop Platform, datasets are stored in specific format. Different storage formats are used for different Algorithms on Hadoop and Spark Platform for example, Fasta, Fastq, CSV, ADAM, BAM (Binary Alignment Map)/ SAM (Sequence Alignment Map) and ADAM.

The objectives of this study are:

- To explore the Supported Languages and Supported Platforms for Genome Sequencing, Motif Finding, Pairwise Alignment and Multiple Alignment Algorithms
- To analyze the need for Scala language for the implementation of Bioinformatics Algorithms on Hadoop Platform
- To explore the Scala Language used in existing Bioinformatics tools

The rest of the paper is organized as follows: Section II explains the related work in this field. Section III describes the tools for Bioinformatics. Section IV represents Role of Scala Implementations in Bioinformatics.

II. Related Work

Ali et al. [1] have explained study in which many Machine Learning classification and clustering Algorithms are implemented in Hadoop MapReduce and Apache Spark using Scala language. They also describe the Performance comparison of different Machine Learning Techniques and Algorithms in the perspective of Hadoop and Spark. It illustrates further research ideas in his paper in which Machine Learning Techniques and Algorithms are implemented in Hadoop and Spark Framework. Sarwar et al. [2] have proposed review study about Bioinformatics Tools. They demonstrate the implementations of Tools for Alignment Viewers, Database Search and Genomic Analysis on Hadoop and Apache Spark Framework using Scala language. It also describes further research domains for the implementation of Bioinformatics Tools on Hadoop and Apache Spark using various languages such as Java, Scala and Python.

SeqPig is a library and tool for Analysis and query sequencing data with scalability [3]. It uses the Hadoop engine, Apache Pig, that automatically parallelizes and distributes tasks that are translated into sequence of MapReduce jobs. It provides extension mechanism for library functions supported by languages (Python, Java and JavaScript) and also provides import and export functions for file format such as Fastq, Qseq, FASTA SAM and BAM. It allows the user to load and export sequencing data. SeqPig provides five read statistics. (a) average base quality read; (b) length of reads; (c) base by position inside the read; (d) GC content of read. Finally combined with single script, it is also used for ad-hoc Analysis but SparkSeq is the best option for ad-hoc analysis.

Wiewiorka et al. [4] have launched bioinformatics tool used to build genome pipeline in Scala and for RNA and DNA sequence analysis. The purpose of this work was to determine scalability and very fast performance by analysis of large datasets such as protein, genome and DNA. A new MapReduce model has been developed for parallel and distributed execution in Spark. Data cannot be stored in HDFS without BAM library (for direct access data and support formats). After data storage in Hadoop, Spark queries applied to sequencing datasets and data is analyzed.

Nordberg et al. [5] proposed the BioPig, used for analysis of large sequencing datasets in the perspective of Scalability (scale with data size), Programmability (reduced development time) and portability (without modification Hadoop). To evaluate these three perspectives, Kmer application was implemented to check its performance and compare with other methods. BioPig uses methods (pigKmer, pigDuster and pigDereplicator). Dataset size for Biopig ranges from 100 MB to 500 GB. Biopig is same as SeqPig in such a way that both use Hadoop and Pig environment and same functions (import and export) and similar run time performance. Only difference is that BioPig includes many Kmer applications and wrapper for BLAST that the SeqPig does not have. The limitation of BioPig is the startup latency of Hadoop. This problem is solved by Spark.

Sun et al. [6] presented the Mapping of long sequence by Bwasw-cloud algorithm with the help of Hadoop MapReduce implementation. Many single processor algorithms like BLAST, SOAP and MAQ are struggling for quick reads. Many multiprocessor algorithms perform much better work like BlastReduce and short reads but some problems occur as its performance and expense for equipment. These problems are decreased by Bwasw-cloud algorithm. This algorithm contains three phases (Map, Shuffle and Reduce) by using seed-and-extend technique and sequence alignment functions are mostly implemented in Map phase. The scaling is measured by length of reads, different mismatches and different number of reference chunks, whereas performance is measured as the speedup over this algorithm.

Taylor et al. [7] focused the next-generation sequencing data and its use in bioinformatics field. Hadoop and MapReduce play an important role in NGS. In this work, he has discussed some terminologies such as Hadoop, MapReduce, HBase, Hive, pig and Mahout then their role in bioinformatics field such as CloudBurst software same as BlastReduce (for NGS short read mapping into reference genome), Bowtie crossbow (for genome re-sequencing analysis), Contrial (for assembly DNA short reads without reference genome), R/Bioconductor (for calculating different gene expression in large RNA-seq dataset). Hadoop and HBase also used for Biodoop tool that consist of three algorithms (BLAST, GSEA and GRAMMAR). Hadoop also used for multiple sequence alignment.

Srinivasa et al. [8] have proposed a technique to classify sequences with the help of Distance matrix formula (m*m) and to understand the relationship among different species during evolution using MapReduce model by dividing the sequences into blocks. Dynamic algorithms Needleman-Wunsch and Smith-waterman are limited to number and size of sequence. So, new MapReduce model developed to reduce these limitations. The input format is FASTA format and output in the custom type. It includes three MapReduce jobs: (a) Data preprocessing (b) Cartesian product (c) Sequence alignment. After these three phases, hierarchical clustering is performed by UPGMA (to produce rooted trees). Due to scalability of Hadoop framework, the proposed method for Phylogenetic is suited for large scale problems.

III. Tools for Bioinformatics

There are several Bioinformatics tools those are used for the analysis of small and large datasets. Every tool performs specific function. Different tools are used for sequence analysis, motif finding, database search and genome analysis. These tools require the data to be stored in a specific format for any kind of analysis. These tools are built using different programming languages. It is important to know the specific language in order to customize the tools. The skills in a programming language are more helpful when extending these tools for Hadoop MapReduce or Apache Spark framework.

A. Motif Finding Tools

Sequence motifs are repeated patterns that are of biological significance. Many tools are available for motif finding in the nucleotide or protein sequence. These tools are also implemented using different programming languages like C, C++, Java, Perl, FORTRAN, Python, and R. a list of the motif finding tools is presented in TABLE I.

Like the alignment viewer and genomics Analysis, the motif finding tools also implemented in Apache spark and Hadoop MapReduce Framework for the experimentation of Big Data analysis. PMS and BLOCKS are implemented in a Hadoop MapReduce Framework for the Big Data analysis.

TABLE I. MOTIF FINDING TOOLS

Name	**Sequence Type**	**Language**	**Data Format**	**MapReduce**	**Spark**
PMS [9]	Protein or nucleotide sequence	*Perl, Python, Java, C++*	Fasta	YES [10]	NO
FMM [11]	Nucleotide sequence	*Python, Java*	Fasta	NO	NO
BLOCKS	Protein or nucleotide sequence	*Perl, Python, Java*	Fasta	YES [12]	NO
eMOTIF	Protein or nucleotide sequence	Java	Fasta	NO	NO
Gibbs motif sampler [13]	Protein or nucleotide sequence	C, C++, Java, and Fortran, Python, R [3]	Fasta	NO	NO
HMMTOP [14]	Protein sequence	Perl, Python, C or Fortran	Fasta	NO	NO
I-sites [15]	Protein sequence	Python, C++	Fasta	NO	NO
JCoils	Protein sequence	C++	Fasta	NO	NO
MEME/MAST [16]	Protein or nucleotide sequence	Ruby, Python	Fasta	NO	NO
CUDA-MEME [17]	Protein or nucleotide sequence	*Python*, *Perl*, Fortran, *Java, Ruby* C, C++	Fasta	NO	NO
MERCI	Protein or nucleotide sequence	C, C++	Fasta	NO	NO

B. Multiple Sequence Alignment

These tools are used for the alignment of more than two nucleotide or protein sequences. These tools are helpful in finding the homology and evolutionary relationships between the studied sequences. A number of multiple sequence alignment tools are developed using Ruby, C, C++ and Python.

ABA, ALE, AMAP, anon, BAli-Phy are implemented in Ruby, C, Python and C++. Multiple sequence alignment tools support different format of data for storage and alignment purpose of protein and nucleotide. ABA, ALE, AMAP, anon, BAli-Phy tools have the different data format like Fasta GenBank, EMBL, GDBM, PHYLIP, MFA. With the growing technologies in Bioinformatics, the tools of Multiple Sequence Alignment are also implemented in Modern technology like Hadoop MapReduce and Apache Spark. MSA, SAGA MSAProbs are tools of Multiple Sequence Alignment category that are implemented in Hadoop MapReduce and Apache Spark.

TABLE II presents the available multiple sequence alignment tools.

TABLE II. MULTIPLE SEQUENCE ALIGNMENT TOOLS

Name	**Sequence Type**	**Language**	**Data Format**	**MapReduce**	**Spark**
ABA [18]	Protein sequence	Ruby	Fasta	NO	NO
ALE	Nucleotides	C Python	GenBank, EMBL Fasta GDBM Phylip,	NO	NO
AMAP	Protein and Nucleotides sequence	Python	MFA Fasta	NO	NO
Anon	Nucleotides	Python	-	NO	NO
BAli-Phy	Protein and Nucleotides sequence	C++	Fasta	NO	NO
Base-By-Base [19]	Protein and Nucleotides sequence	Java	Fasta GenBank	NO	NO
CHAOS/DIALIGN	Protein and Nucleotides sequence	Java	Fasta	NO	NO
ClustalW	Protein and Nucleotides sequence	C++	Fasta	NO	NO
CodonCode Aligner	Nucleotides	C++	Fasta Fastq, Sam, GenBank, or EMBL	NO	NO
Compass [20]	Protein sequence	C, C++, Java Python	Fasta	NO	NO
DECIPHER	Protein and Nucleotides sequence	R	Fasta Fastq , QSEQ, RAW, Miro, and Seq	NO	NO
DIALIGN-TX and DIALIGN-T	Protein and Nucleotides sequence	C	Fasta	NO	NO
DNA Baser Sequence Assembler	Nucleotides	Java	SCF, ABI, Fasta SEQ, TXT, GBK	NO	NO
DNASTAR Lasergene Molecular Biology Suite	Protein and Nucleotides sequence	C, C++ Python	EMBL, GenBank	NO	NO
DNA Alignment	Protein and Nucleotides sequence	Python Perl Javascript	Fasta	NO	NO
EDNA [21]	Nucleotides	Java	GeneMappe	NO	NO
FSA	Protein and Nucleotides sequence	C++	Fasta	NO	NO
Geneious	Protein and Nucleotides sequence	C++	Fasta Genbank	NO	NO
Kalign	Protein and Nucleotides sequence	C	Fasta GCG, EMBL, GenBank, PIR,NBRF, Phylip, Swiss-Prot	NO	NO
MAFFT	Protein and Nucleotides sequence	C	Fasta	NO	NO
MARNA	RNA sequence	C++	Fasta	NO	NO
MAVID [22]	Protein and Nucleotides sequence	C++	Fasta	NO	NO
MSA	Protein and Nucleotides sequence	C	Genepop , Msvar, Structure, Arlequin, Migrate, IM-format	YES [23]	NO
MSAProbs	Protein sequence	C++ bioPerl	Fasta	YES [23]	NO
MULTALIN	Protein and Nucleotides sequence	C	MultAlin, Fasta , GenBank , EMBL, SwissProt	NO	NO
Multi-LAGAN [24]	Protein and Nucleotides sequence	C C++	Fasta	NO	NO
MUSCLE	Protein and Nucleotides sequence	C++	Fasta	NO	NO
Opal	Protein and Nucleotides sequence	Java	data maNOger file (*.odm)	NO	NO
Pecan [25]	DNA sequence	Python	Fasta	NO	NO
Phylo	Nucleotides	R, Javascript	Fasta	NO	NO
PMFastR	RNA sequence	C++	Fasta	NO	NO
Praline	Protein sequence	Ruby Javascript	Fasta or PIR	NO	NO
PicXAA [26]	Protein and Nucleotides sequence	C++	Fasta	NO	NO

POA	Protein sequence	C	Fasta	NO	NO
Probalign	Protein sequence	C++	Fasta	NO	NO
ProbCons	Protein sequence	C++	Fasta	NO	NO
PROMALS3D	Protein sequence	Python	Fasta	NO	NO
PRRN/PRRP [27]	Protein sequence	Ruby	Fasta	NO	NO
PSAlign	Protein and Nucleotides sequence	C	Fasta	NO	NO
RevTrans	DNA or Protein	Python	ASTA, MSF and ALN	NO	NO
SAGA [28]	Protein sequence	C	Fasta	YES [29]	NO
SAM	Protein sequence	Perl, C	Fasta	NO	NO
Se-AL	Protein and Nucleotides sequence	Java	Nexus, Phylip, MEGA, NBRF, Fasta GDE and GDE 97	NO	NO
StatAlign [30]	Protein and Nucleotides sequence	Java	Fasta	NO	NO
Stemloc	RNA sequence Alignment	-	Fasta	NO	NO
UGENE	Protein and Nucleotides sequence	C++, Qt	Fasta, GenBank , EMBL , GFF	NO	NO
VectorFriends	Protein and Nucleotides sequence	Assembly	EMBL, Fasta Nexus	NO	NO
GLProbs [31]	Protein sequence	C++	Fasta	NO	NO
T-Coffee	Protein and Nucleotides sequence	C, biopython C++, Perl and python	Fasta , PIR	NO	NO

C. Pairwise Alignment

These tools are used for the identification of similarity regions between two biological sequences that can indicate functional, structural or evolutionary relationships. Pairwise Alignment tools are also implemented in different programming languages. ACANA is implemented in C++, AlignMe in Python and Perl, Bioconductor in PHP, Perl and Java, BioPerldpAlign in Perl, BLASTZ, LASTZ in C and CUDAlign is implemented in C++. A list of the available pairwise alignment tools is presented in TABLE III.

Pairwise alignment tools also support different data formats for the storage and analysis of biological data. These formats include Fasta Fastq, BAM, gtf, bed, wig, nib, hsx, GenBank, Raw DNA file formats, and Primers (.csv). Some of the tools also support Hadoop MapReduce and Apache Spark. Matcher, JAligner, Genome Compiler, Bioconductor, BioPerldpAlign are tools that are implemented for Big Data Platforms.

TABLE III. PAIRWISE ALIGNMENT TOOLS

Name	Sequence Type	Language	Data Format	MapReduce	Spark
ACANA [32]	Protein or nucleotide sequence	C++	Fasta	NO	NO
AlignMe	Protein sequence	Python,Perl	Fasta	NO	NO
Bioconductor	Protein or nucleotide sequence	PHP, Perl Java	Fasta fastq, BAM, gtf, bed, and wig	YES [78]	NO
BioPerl [33]	Protein or nucleotide sequence	Perl	Fasta	YES [79]	NO
BLASTZ,LASTZ	Nucleotides	C, C++	Fasta fastq, nib, 2bit or hsx	NO	NO
CUDAlign	Nucleotides	C++	Fasta	NO	NO
DNADot	Nucleotides	Java	Fasta	NO	NO
DNASTAR Lasergene Molecular Biology Suite	Protein or nucleotide sequence	Java	GenBank	NO	NO
DOTLET	Protein or nucleotide sequence	Java	Fasta	NO	NO
FEAST [34]	Nucleotides	Java	Genbank	NO	NO
Genome Compiler [35]	Nucleotides	*C, Perl, PHP, Java,* ruby Python, Perl	GenBank, Fasta	YES [80]	YES [81]
G-PAS	Protein or nucleotide sequence	C++	Fasta	NO	NO
GapMis	Protein or nucleotide sequence	C	Fasta	NO	NO
GGSEARCH, GLSEARCH	Protein sequence	C, C++	Fasta	NO	NO
JAligner [36]	Protein or nucleotide sequence	Java	Fasta	YES [82]	NO

K*Sync	Protein sequence	Java	Fasta	NO	NO
LALIGN	Protein or nucleotide sequence	Python	Fasta	NO	NO
NW-align [37]	Protein sequence Alignment	Java	Fasta PDB format	NO	NO
mAlign	Nucleotides	Java, C	Genbank Fasta	NO	NO
Matcher	Protein or nucleotide sequence	C, C++	Fasta msf, trace, srs	YES [83]	YES [84]
MCALIGN2 [38]	DNA sequence Alignment	C++	Fasta	NO	NO
MUMmer	Nucleotides	-	Fasta delta	NO	NO
Needle	Protein or nucleotide sequence	C, C++, Python	Fasta msf, clustal, mega, meganon, nexus,,nexus	NO	NO
Ngila [39]	Protein or nucleotide sequence	C++	Fasta	NO	NO
NW	Protein or nucleotide sequence	C, C++, Python	Fasta	NO	NO
Parasail	Protein or nucleotide sequence	C, C++, Python	Fasta Fastq	NO	NO
Path [40]	Protein sequence Alignment	Java	Fasta	NO	NO
PatternHunter	Nucleotides	Java	Genbank Fasta	NO	NO
ProbA (also propA)	Protein or nucleotide sequence	C	Fasta	NO	NO
PyMOL	Protein sequence Alignment	C, C++	Fasta Genbank	NO	NO
REPuter [41]	Nucleotides	Json web service	Fasta Genbank	NO	NO
SABERTOOTH	Protein sequence Alignment	Java	FastaGenbank, EMBL, SWISSPROT	NO	NO
Satsuma	DNA sequence	C++	Fasta	NO	NO
SEQALN [42]	Protein or nucleotide sequence	-	genbank, newat, Fasta, pir, swissprot	NO	NO
SIM, GAP, NOP, LAP	Protein or nucleotide sequence	C/C++/Python	Fasta	NO	NO
SIM	Protein or nucleotide sequence	C/C++/Python	Fasta	NO	NO
SPA: Super pairwise alignment	Nucleotides	C++	Fasta Genbank	NO	NO
SSEARCH	Protein sequence	C#, Java, Perl	Fasta	NO	NO
Sequences Studio [43]	Generic Sequence	Java	Fasta	NO	NO
SWIFT suit	DNA sequence	Swift	Fasta	NO	NO
Stretcher	Protein or nucleotide sequence	Ruby	Fasta Genbank	NO	NO
ss	Nucleotides	R	Embl, Imgt Refseqn Genbank	NO	NO
UGENE	Protein or nucleotide sequence	C++	FASTA GenBank, EMBL, GFF	NO	NO
Water [44]	Protein or nucleotide sequence	R	Fasta	NO	NO
WordMatch	Protein or nucleotide sequence	R	Fasta msf, clustal, mega, meganonexus	NO	NO
YASS [45]	Nucleotides	C	Fasta Axt	NO	NO

IV. Importance of Scala Implementations

A lot of programming languages are being used for the implementation of Bioinformatics Algorithms on Hadoop Platform and Apache Spark. Most of Bioinformatics tools are implemented using Java, Python, C++, Perl, FORTRAN, R, Ruby, C, Bioperl, Assembly, JavaScript, PHP and Swift languages. Some Algorithms are used in Multiple Sequence Alignment, Pairwise Alignment and Motif Finding. These Algorithms are implemented by using Hadoop and Apache Spark framework. Many languages are used to implement these Bioinformatics Algorithms. Most commonly used languages are Java, Python and Scala.

Our goal is to use best language for the implementation of Bioinformatics Algorithms. Scala language is superlative language in Hadoop Platform and Apache Spark for the implementation of Bioinformatics Algorithms. By using Scala language for Bioinformatics Algorithms, we will achieve better Performance, Scalability and Accuracy. This language plays imperative role in all benchmarks. When we implement Bioinformatics Algorithms in Spark Framework, Scala language give better results. Closure, Traits, Pattern Matching and Functions are main features of Scala language.

Some Motif Finding tools such as PMS, FMM, BLOCS, eMOTIF, Gibbs motif sampler, HMMTOP, JCoils, MEME/MAST, CUDA-MEME and MERCI are available in Bioinformatics. Algorithms in these tools are not implemented in Spark using Scala language. We can use Scala language for the implementation of these Motif Finding Bioinformatics Algorithms to attain better outcomes. Scala is state of the art language that associates Object Oriented and Functional programming concepts.

Most of tools such as ABA, ALE, AMAP, Anon, Bali-Phy, Base-By-Base, CHAOS, ClustalW, CodonCode Aligner, Compass, DECIPHER, DNA Alignment, Geneious, Kalign, EDNA, FSA, MAFFT, MARNA, MAVID, MSA, MUSCLE, Opal, Pecan, Phylo, Praline, POA, PicXAA, ProbCons, PSAlign, SAGA, SAM, Se-AL, StemAlign, UGENE and VectorFriends are available for Multiple Sequence Alignment in Bioinformatics. Some Pairwise Alignment tools such as ACANA, AlignMe, Bioconductor, BioPerl, BLASTZ, CUDAlign, DNADot, DOTLET, FEAST, Genome Compiler, G-PAS, GapMis, JAligner, K*Sync, LALIGN, NW-Align, Matcher, MUMmer, Needle, Ngila, NW, Parasail, Path, ProbA, PyMOL, REPuter, Satsuma, SIM, GAP, NOP, LAP, SIM, SSEARCH, Sequences Studio, SWIFT suit, Stretcher, SPA, ss, UGENE, Water and YASS are available for Bioinformatics. Algorithms in these tools are not implemented in Spark using Scala language. We can use Scala language for the implementation of these Multiple Sequence Alignment and Pairwise Alignment Bioinformatics Algorithms to attain better outcomes.

Many Bioinformatics Algorithms are based on Greedy and Dynamic Programming paradigm. Some Bioinformatic sequences are Map/Align with Local, Global, Multiple and Pairwise method. Nussinov-Algorithm and Viterbi-Algorithm also require Scala language for their implementation. SCABIO is the best framework for Bioinformatics Algorithms in Scala language. It includes many built-in-methods and libraries that are helpful for Scala implementation. It also provides Greedy and Dynamic Programming approach for Bioinformatic sequences. We can use SCABIO for Global, Local, Multiple and Pairwise Alignment. Pattern Matching is best performed with the help of SCABIO because SCABIO includes Scala language implementation concepts.

V. Conclusion

Keeping in view the data analysis demands in Bioinformatics, Big Data Platforms and tools are an obvious choice. Among these platforms, Spark is most efficient platform for rapid analysis of large data sets. Spark itself is implemented in Scala languages and supports programs in Java, Scala and Python. Majority of the tools in bioinformatics are not designed for Big Data Platforms. As discussed in previous sections, most of the Multiple Alignment tools, Pairwise Alignment tools and Motif Finding tools still need to be enhanced for use on Big Data Platforms like Hadoop and Spark. So, there is need of time to implement bioinformatics tools on Big Data Platforms. Several languages are available for implementation of bioinformatics tools like Java, C, Perl, Python and Scala. Among these languages, Scala is a good choice especially for Spark Implementations. It provides structures and constructs that are suitable for Bioinformatics applications. It provides support for dynamics programming and pattern matching. It can provide efficient implementations of machine learning algorithms. We recommend that Scala must be used for future implementations of Bioinformatics tools on Big Data Platforms.

References

[1] M. U. Ali, S. Ahmad and J. Ferzund, "Harnessing the Potential of Machine Learning for Bioinformatics using Big Data Tools," International Journal of Computer Science and Information Security (IJCSIS), vol. 14, no. 10, pp. 668-675, 2016.

[2] M. A. Sarwar, A. Rehman and J. Ferzund, "Database Search, Alignment Viewer and Genomics Analysis Tools: Big Data for Bioinformatics," International Journal of Computer Science and Information Security (IJCSIS), vol. 14, no. 12, 2016.

[3] S. Andre, P. Luca, N. Matti and K. Aleksi, "SeqPig: simple and scalable scripting for large sequencing data sets in Hadoop," Oxford.

[4] S. Oehmen, "ScalaBLAST 2.0: rapid and robust BLAST calculations on multiprocessor systems," Oxford.

[5] N. Henrik, B. Karan, W. Kai and W. Zhong, "BioPig: a Hadoop-based analytic toolkit for large-scale sequence data," oxford, September 10, 2013.

[6] S. Mingming, Z. Xuehai and Y. Feng, "Bwasw-Cloud: Efficient sequence alignment algorithm for two big data with MapReduce," in Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference, 2014.

[7] R. C. Taylor, "An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics," BioMed Central, 2010.

[8] G. Siddesh, K. Srinivasa and M. Ishank, "Phylogenetic Analysis Using MapReduce Programming Model," in Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International, 2015.

[9] "http://motifsearch.com/," [Online]. Available: http://motifsearch.com/.

[10] "A MapReduce-based Algorithm for Motif Search," ResearchGate.

[11] "Learn Motifs from Unaligned Sequences," [Online]. Available: https://genie.weizmann.ac.il/pubs/fmm08/fmm08_learn_unalign.html.

[12] Y. Liu, X. Jiang, H. Chen, J. Ma and X. Zhang, "MapReduce-Based Pattern Finding Algorithm Applied in Motif Detection for Prescription Compatibility Network," Springer Link.

[13] "Using the Gibbs motif sampler to find conserved domains in DNA and protein sequences.," PubMed.

[14] "The HMMTOP server," [Online]. Available: http://www.enzim.hu/hmmtop/html/document.html.

[15] "I-sites Libraries 2008," [Online]. Available: http://www.bioinfo.rpi.edu/bystrc/Isites2/.

[16] [Online]. Available: https://en.wikipedia.org/wiki/Multiple_EM_for_Motif_Elicitation.

[17] "cuda-meme," [Online]. Available: https://sites.google.com/site/yongchaosoftware/Home/cuda-meme.

[18] R. Benjamin, D. Zhi, H. Tang and P. Pevzner, "A novel method for multiple alignment of sequences with repeated and shuffled elements," PubMed Central.

[19] "Virology.ca Tools," [Online]. Available: http://athena.bioc.uvic.ca/virology-ca-tools/base-by-base/.

[20] G. N. Sadreyev R, "COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance.," PubMed .

[21] [Online]. Available: https://sourceforge.net/projects/msa-edna/.

[22] N. B. Pachter and Lior, "MAVID multiple alignment server," Oxford.

[23] Jurate, O. D. Aisling and D. S. Roy, "An Overview of Multiple Sequence Alignments and Cloud," ISRN Biomathematics, 2013.

[24] Brudno M et al. "LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA.," PubMed.

[25] [Online]. Available: https://github.com/benedictpaten/pecan/blob/master/doc/pecan/README_PECAN.txt.

[26] S. Sahraeian and B. Yoon, "PicXAA: a probabilistic scheme for finding the maximum expected accuracy alignment of multiple biological sequences.," PubMed.

[27] "prrn," [Online]. Available: http://www.genome.jp/tools/prrn/prrn_help.html.

[28] "SAGA HOME PAGE," [Online]. Available: http://www.tcoffee.org/Projects/saga/.

[29] M. Chris and M. Michael, "Programming Abstractions for Data Intensive Computing on Clouds and Grids," CCGRID '09 Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid.

[30] "An Extendable Software Package for Joint Bayesian Estimation of Alignments and Evolutionary Trees," [Online]. Available: http://statalign.github.io/.

[31] Y. Yongtao, W.-l. C. David and W. Yadong, "GLProbs: Aligning Multiple Sequences Adaptively," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2014.

[32] W. Huang, D. Umbach and L. Li, "Accurate anchoring alignment of divergent sequences.," PubMed.

[33] "BioperlOverview," [Online]. Available: http://www.ebi.ac.uk/~lehvasla/bioperl/BioperlOverview.html.

[34] [Online]. Available: http://monod.uwaterloo.ca/feast/..

[35] "Genome Compiler," [Online]. Available: http://www.genomecompiler.com/about-genome-compiler/.

[36] [Online]. Available: http://jaligner.sourceforge.net/.

[37] [Online]. Available: http://zhanglab.ccmb.med.umich.edu/NW-align/.

[38] W. Jun, D. K. Peter and J. Toby, "MCALIGN2: Faster, accurate global pairwise alignment of non-coding DNA sequences based on explicit models of indel evolution," BMC Bioinformatics.

[39] [Online]. Available: http://scit.us/projects/ngila/ .

[40] [Online]. Available: http://bioinfo.lifl.fr/path/path.php.

[41] [Online]. Available: http://bibiserv.techfak.uni-bielefeld.de/reputer/.

[42] [Online]. Available: http://thegrantlab.org/bio3d/html/seqaln.html.

[43] [Online]. Available: http://www.bioinformatics.org/sStu/doc/index.html.

[44] [Online]. Available: http://emboss.sourceforge.net/apps/release/6.6/emboss/apps/water.html.

[45] [Online]. Available: https://en.wikipedia.org/wiki/Yass_(software).