

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **ABSTRACT**

### **EXACT GENOME ALIGNMENT**

**by**  
**Nandini Ghosh**

The increase in the volume of genomic data due to the decrease in the cost of whole genome sequencing techniques has opened up new avenues of research in the field of Bioinformatics, like comparative genomics and evolutionary dynamics. The fundamental task in these studies is to align the genome sequences accurately. Sequence alignment helps to identify regions of similarity between the sequences to establish their functional, evolutionary and structural relationship. The thesis investigates the performance of two sequence alignment programs LASTZ, a hash table based faster method and SSEARCH, a slower but more rigorous Smith-Waterman based approach, on whole genome sequences from primates and mammals. An exact genome alignment technique is used by breaking the entire genome into fragments and aligning these fragments with the reference genome using the Smith-Waterman based method. A comparison of the two methods reveals that the second approach performs better for genomes from closely related species.

# **EXACT GENOME ALIGNMENT**

**by**

**Nandini Ghosh**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Bioinformatics**

**Department of Computer Science**

**May 2015**

Blank Page

**APPROVAL PAGE**

**EXACT GENOME ALIGNMENT**

**Nandini Ghosh**

---

Dr. Usman Roshan, Thesis Advisor Date  
Associate Professor of Bioinformatics and Computer Science, NJIT

---

Dr. Jason T. Wang, Committee Member Date  
Professor of Bioinformatics and Computer Science, NJIT

---

Dr. Zhi Wei, Committee Member Date  
Associate Professor of Bioinformatics and Computer Science, NJIT

## **BIOGRAPHICAL SKETCH**

**Author:** Nandini Ghosh  
**Degree:** Master of Science  
**Date:** May 2015

### **Undergraduate and Graduate Education:**

- Master of Science in Bioinformatics,  
New Jersey Institute of Technology, Newark, NJ, 2015
- Master of Science in Physics,  
Central Michigan University, Mt. Pleasant, MI, 2001
- Bachelor of Science in Physics,  
Banaras Hindu University, Varanasi, India, 1996

**Major:** Bioinformatics

*In loving memory of my father, Nihar Nath Ghosh, who always motivated me to enlighten myself with the warmth of knowledge.*



## ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to my advisor, Dr. Usman Roshan, for his guidance and support for this thesis work. His encouraging suggestions and valuable discussions have been indispensable in the completion of this work.

I wish to thank the members of my thesis committee, Dr. Jason Wang and Dr. Zhi Wei for their support and feedback during the course of this work. I gratefully acknowledge the help and cooperation extended by Ling Zhong, a doctoral student in the department.

This research would not have been possible without the High Performance Computing resources at NJIT. I would like to thank Gedaliah Wolosh, Manager of ARCS Computing Resources for his valuable suggestions that helped running the jobs on Kong machines faster and easier.

This thesis was made possible because of data from the Alignathon project and a variety of software from the UCSC Genome Browser website and Miller Lab, Penn State University Center for Comparative Genomics and Bioinformatics website. I would like to express my appreciation for making these resources freely available.

Finally, I thank my family for their constant support and encouragement without which none of this would have been possible.

## TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Background.....	2
1.3 Overview.....	3
2 METHODS.....	5
2.1 Datasets.....	5
2.2 Pairwise Alignment with LASTZ.....	7
2.3 Pairwise Alignment with SSEARCH.....	12
3 CONCLUSION.....	18
3.1 Results.....	18
3.2 Discussion.....	26
APPENDIX A SUMMARY OF SSEARCH RESULTS FOR MAMMALS DATASET.....	28
APPENDIX B PYTHON SCRIPTS FOR PREPROCESSING FILES.....	40
B.1 Script for Dividing the Query Sequence into Same Size Fragments.....	40
B.2 Script to Find the Alignments for a Given Score Ratio.....	41
B.3 Script to Convert SSEARCH Output to AXT Format.....	42
REFERENCES.....	45

## LIST OF TABLES

<b>Table</b>		<b>Page</b>
2.1	Primates Data Summary.....	6
2.2	Mammals Data Summary.....	7
2.3	Details of Fields in the AXT file.....	8
2.4	Attributes of the Chain Format File.....	10
2.5	Attributes of the MAF File.....	12
2.6	Scoring Scheme for SSEARCH36.....	13
3.1	LASTZ Comparison Summary of the Primate Dataset.....	20
3.2	SSEARCH Comparison Summary of the Primate Dataset.....	21
3.3	LASTZ Comparison Summary of the Mammal Dataset.....	22
3.4	Fragment and Sliding Window Sizes for SSEARCH Mammal Data.....	23
3.5	SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.95).....	23
3.6	SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.9).....	24
3.7	SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.85).....	24
3.8	SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.8).....	25
3.9	SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.75)....	25
A.1	SSEARCH Mammals Summary (Size = 250, Width = 125, Ratio < 0.95).....	28
A.2	SSEARCH Mammals Summary (Size = 250, Width = 125, Ratio < 0.9).....	29
A.3	SSEARCH Mammals Summary (Size = 500, Width = 250, Ratio < 0.95).....	29
A.4	SSEARCH Mammals Summary (Size = 500, Width = 250, Ratio < 0.9).....	30
A.5	SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.95).....	30

**LIST OF TABLES**  
(Continued)

<b>Table</b>	<b>Page</b>
A.6 SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.9).....	31
A.7 SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.85).....	31
A.8 SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.8).....	32
A.9 SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.95).....	32
A.10 SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.9).....	33
A.11 SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.85).....	33
A.12 SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.8).....	34
A.13 SSEARCH Mammals Summary (Size = 1000, Width = 500, Ratio < 0.95)....	34
A.14 SSEARCH Mammals Summary (Size = 1000, Width = 500, Ratio < 0.9).....	35
A.15 SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.95)....	35
A.16 SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.9).....	36
A.17 SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.85)....	36
A.18 SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.8).....	37
A.19 SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.75)....	37
A.20 SSEARCH Mammals Summary (Size = 3000, Width = 1500, Ratio < 0.95)...	38
A.21 SSEARCH Mammals Summary (Size = 3000, Width = 1500, Ratio < 0.9)....	38
A.22 SSEARCH Mammals Summary (Size = 5000, Width = 2500, Ratio < 0.95)...	39
A.23 SSEARCH Mammals Summary (Size = 5000, Width = 2500, Ratio < 0.9)....	39

## LIST OF FIGURES

<b>Figure</b>	<b>Page</b>
2.1 Phylogenetic tree for the simulated primates dataset.....	5
2.2 Phylogenetic tree for the simulated mammals dataset.....	6
2.3 Two alignment blocks from an AXT format file.....	8
2.4 Part of a chain file format.....	9
2.5 Example of a MAF format file.....	11
2.6 Example of SSEARCH36 output.....	14
2.7 Example of the SGE array job script.....	17

## LIST OF ABBREVIATIONS

DNA	Deoxyribonucleic Acid
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
HPC	High Performance Computing
LASTZ	Large Scale Genome Alignment Tool
MAF	Multiple Alignment Format
MSA	Multiple Sequence Alignment
PPV	Positive Predictive Value
SGE	Sun Grid Engine
TP	True Positive
WGA	Whole Genome Alignment

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation

Whole Genome Alignment (WGA) is a much more challenging problem than protein alignment due to the size and complexity of whole genomes. Apart from the mutational events of nucleotide substitution, insertion and deletion, whole genomes also undergo subsequence deletion, subsequence insertion and genome rearrangements like inversion, translocation, chromosome fusion and chromosome fission during the process of evolution. All these factors have to be addressed when performing whole genome sequence alignments.

The motivation for this work came from the ‘Alignathon’ project, a collaborative project to assess the state of the art tools in Whole Genome Sequence Alignment. Multiple Sequence Alignments (MSA) of whole genomes were performed for three datasets by various teams using different alignment pipelines. In all, three datasets were used, two simulated mammalian and primate phylogenies and a real fly dataset. A competitive evaluation was carried out once all the submissions were received to determine the best alignment tools for Whole Genome Alignment.

For closely related species, the tools performed well, but the performance was not competitive enough for divergent genomes. The alignment method used by most of these pipelines was a hash table based fast local alignment program, LASTZ (Large-Scale Genome Alignment tool). The thesis investigates a more exact approach to the fundamental problem of whole genome alignment by using a slow but rigorous Smith

Waterman based method on the simulated primates and mammalian dataset from the Alignathon project. This study focuses on only pairwise alignment of DNA sequences, which are made up of only four alphabets A, C, G and T.

## **1.2 Background**

The increased availability of whole genome sequences has opened up new opportunities for the phylogenetic and evolutionary analyses of species. To understand the evolutionary homology of species, Multiple Sequence Alignments (MSA) are performed assuming that all aligned sequences have diverged from a common ancestor. The evolutionary distance increases over time due to a series of mutational processes like substitution, insertion or deletion of nucleotides in the sequence. It is difficult to assess the quality of WGA methods due to a dearth of standard reference alignments. The Alignathon project was a step towards addressing this issue.

Some of the existing tools for whole genome alignment are AutoMz (Miller et al., 2007), Cactus (Paten et al., 2011), EPO (Paten et al., 2008), Pecan (Paten & Birney, 2009), GenomeMatch, Mugsy (Angiuoli and Salzberg, 2011), Multiz (Miller et al. 2007), PSAR-Align (Kim & Ma, 2013), progressiveMauve (Darling et al., 2010), Robusta (Notredame, 2012), TBA (Blanchette et al., 2004), and Vista-Lagan (Brudno et al., 2003).

There are four different ways of doing multiple sequence alignment. These are using simulation, expert information, statistical assessments and downstream analysis. The thesis focuses on the simulation based method, where a set of sequences and alignments are generated using an evolution model. The simulated sequences are aligned



using different tools and the predicted alignments are compared to the true simulated alignment.

### **1.3 Overview**

All aspects of genome evolution have to be considered when simulating genomes for whole genome alignment. This includes sequence evolution and genome rearrangements. For the Alignathon study, the EVOLVER software (Edgar et al., 2009), a whole genome sequence evolution simulator was used to create the datasets. This software simulates full sized, multi-chromosome genome evolution in forward time. Evolver simulates the long term effects of mutation and selection over an entire species and generates a representative genome of a species.

Most of the tools that were evaluated used LASTZ, a hash table based local alignment program to perform the pairwise alignment of the genomes. In this method the target sequence is read into memory to build a seed word position table so that it can be mapped to all the positions it appears in the query. The query is read as a word and the position table is used to find matches in the target. These matches are then extended to longer matches using a seed and extend technique without allowing gaps. Finally, each gap-free match that exceeds a certain threshold is extended by a dynamic programming algorithm that allows gaps. Hash table based methods are computationally very fast but their accuracy diminishes when there are a lot of mismatches and gaps in the alignment. Therefore, for the whole genome alignment of distantly related species such methods might have low accuracy. In the thesis, an alternate method has been developed using the Smith-Waterman algorithm which performs an exact genome alignment and can take

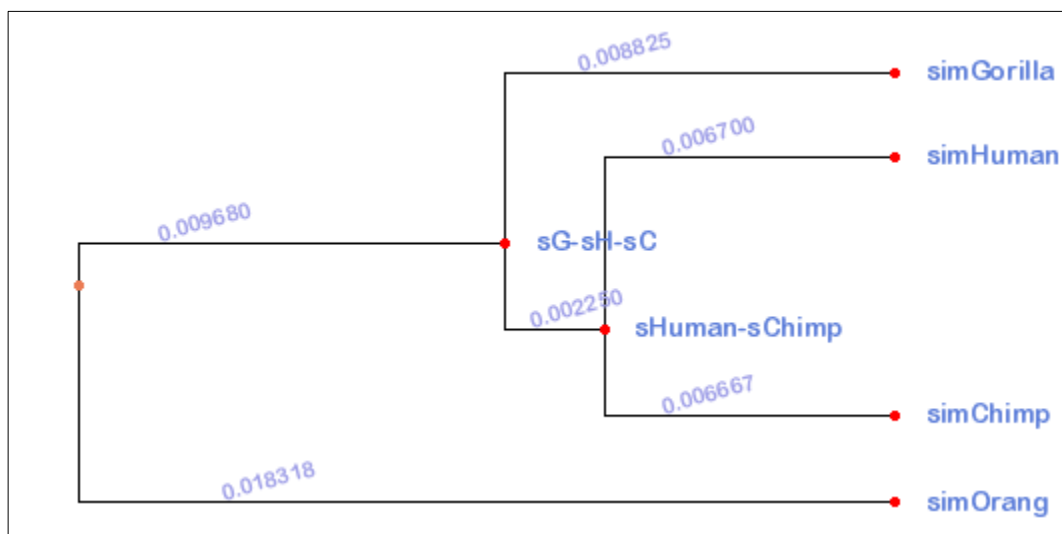
days or even months to align two whole genomes. The Smith-Waterman algorithm performs local sequence alignments for determining regions of maximum similarity between two strings, protein or nucleotide sequences. Unlike global alignments that consider the entire sequence, this algorithm compares subsequences of all possible lengths and finds the optimal local alignment of the two sequences. The reduction in runtime was achieved by dividing the query genome into fragments and then aligning each fragment to the target genome using a parallel approach.

## CHAPTER 2

### METHODS

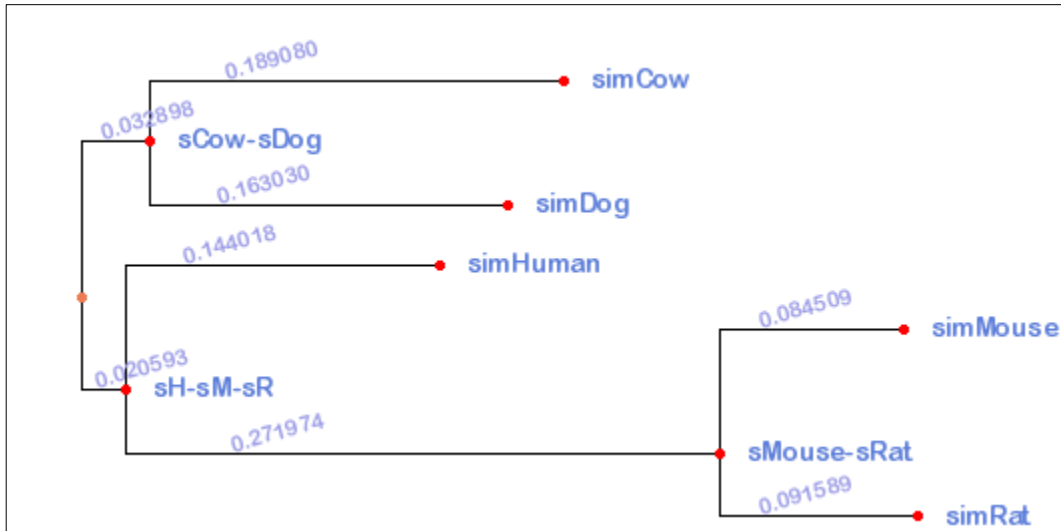
#### 2.1 Datasets

All the datasets for the analysis was obtained from the Alignathon project website (<http://compbio.soe.ucsc.edu/alignathon/>). The datasets were simulated using the Evolver tool. The simulated genomes were created from a 1/20th scale mammalian genome of 120 megabases (Mb) based upon a subset of the human genome hg19/GRC37. The entire chromosome sequences for the chromosomes 20, 21 and 22 were used for simulation. The primate dataset consists of a great ape phylogeny in which the genomes share the same evolutionary relationships as humans, chimpanzees, gorillas and orangutans. The simulated genomes in the mammalian phylogeny have the same evolutionary relationship as humans, cows, dogs, mice and rats.



**Figure 2.1** Phylogenetic tree for the simulated primates dataset.

Source: [http://compbio.soe.ucsc.edu/alignathon/set\\_primate.html](http://compbio.soe.ucsc.edu/alignathon/set_primate.html)



**Figure 2.2** Phylogenetic tree for the simulated mammals dataset.

Source: [http://compbio.soe.ucsc.edu/alignathon/set\\_mammal.html](http://compbio.soe.ucsc.edu/alignathon/set_mammal.html)

The phylogenetic trees of the simulated primate and mammal datasets are shown in Figures 2.1 and 2.2. Since WGA is computationally very expensive, the size of the genomes was important for the analysis. Four datasets were chosen from the simulated data for this study. Two sets each were chosen from the primate and mammalian phylogenies and their details are listed in Tables 2.1 and 2.2.

**Table 2.1** Primates Data Summary

Genome	Chromosome	Size (base pairs)
Human	Chr D	10,572,275
Gorilla	Chr D	10,570,608

Source: [http://compbio.soe.ucsc.edu/alignathon/set\\_primate.html](http://compbio.soe.ucsc.edu/alignathon/set_primate.html)

**Table 2.2** Mammals Data Summary

<b>Genome</b>	<b>Chromosome</b>	<b>Size (base pairs)</b>
Cow	Chr C	33,408,597
Mouse	Chr O	3,949,899

Source: [http://compbio.soe.ucsc.edu/alignathon/set\\_mammal.html](http://compbio.soe.ucsc.edu/alignathon/set_mammal.html)

A pairwise genome alignment was computed for the chosen datasets using LASTZ, which is an established program for performing sequence alignment of DNA sequences. The genomes were also aligned using the method developed in this thesis, which is an exact genome alignment strategy using the SSEARCH36 package from the FASTA suite of programs developed by Pearson and Lipman (W. R. Pearson and D. J. Lipman (1988), "Improved Tools for Biological Sequence Analysis", PNAS 85:2444-2448). The program uses a rigorous Smith-Waterman algorithm to find the best scoring local alignment of the two sequences.

## **2.2 Pairwise Alignment with LASTZ**

After obtaining the datasets from the Alignathon project, the pairwise alignment of the primate and mammalian data was done using the LASTZ program (version 1.02.00) with all default parameters. For the primate dataset, the Human chromosome D was chosen as the reference sequence and the Gorilla chromosome D was chosen as the query sequence for computing the alignment. For the mammalian dataset, Cow chromosome C was chosen as the reference and Mouse chromosome O was chosen as the query sequence. The LASTZ output files were in AXT format where each alignment block comprises of three lines. The first line is a summary line and the next two lines are the sequence lines.

```

17 simHuman.chrD 8507246 8507305 simGorilla.chrD 7593 7652 + 3231
CTTGGATACGTCGTGTTGCTTCTGCACTTTCAAGTAAATGTGCCAGTTGTCACACCACTT
CTTAGTCTCATCATGTTGCTTCTGCACTATCAAGTTATTGTGCCAGTTGTCATGCGACTT

18 simHuman.chrD 6528902 6528946 simGorilla.chrD 7606 7650 + 3038
TGTTGCTTCTGCATTTTCAAATTATTGTGCCAGTTGTCACACCAC
TGTTGCTTCTGCACTATCAAGTTATTGTGCCAGTTGTCATGCGAC

```

**Figure 2.3** Two alignment blocks from an AXT format file. Each block consists of a summary line followed by two sequence lines.

The summary lines consist of nine required fields that contain chromosomal and position information about the alignment. The details of the fields are listed in Table 2.3.

**Table 2.3** Details of Fields in the AXT File

Field	Explanation
Alignment Number	Numbering starts from 0 and increments by 1
Chromosome	Name of the reference genome
Alignment start	Position of the reference genome where the alignment starts
Alignment end	Position of the reference genome where the alignment ends
Chromosome	Name of the query genome
Alignment start	Position of the query genome where the alignment starts
Alignment end	Position of the query genome where the alignment ends
Strand	Can be '+' or '-'. If the value is '-', the query genomes start and end positions are relative to the reverse-complemented coordinates of its chromosome
LASTZ score	Score of the alignment

Source: <http://genome.ucsc.edu/goldenPath/help/axt.html>

After obtaining the pairwise alignments in the AXT format, the publicly available `axtChain` program from the UCSC Genome Bioinformatics Site (<https://genome.ucsc.edu/>) was used to chain together the alignments. The program joins two matching alignments next to each other into one fragment if they are close enough. The output is in a chain format which describes a pairwise alignment that allows gaps in both the sequences simultaneously. Every set in the chain alignments starts with a header, one or more alignment data lines and ends with a blank line. An example of the chain file is shown in Figure 2.4.

```

chain 16834 Human 10572275 + 10114562 10118991 Gorilla 10570608 + 7705900 7710524 4
258 0 2
13 0 8
2219 297 300
554 0 180
315 0 1
692 0 1
81

chain 9464 Human 10572275 + 7636105 7638153 Gorilla 10570608 + 7689915 7691970 5
1079 0 7
969

```

**Figure 2.4** Part of a chain file format.

The header line starts with the keyword `chain` followed by 12 attribute values and ends with a blank line. The attribute values are described in Table 2.4.

**Table 2.4** Attributes of the Chain Format File

<b>Attribute</b>	<b>Description</b>
Score	Chain score
tName	Name of the reference sequence chromosome
tSize	Size of the reference sequence chromosome
tStrand	Reference sequence strand value ('+' or '-')
tStart	Alignment start position in the reference sequence
tEnd	Alignment end position in the reference sequence
qName	Name of the query sequence chromosome
qSize	Size of the query sequence chromosome
qStrand	Query sequence strand value ('+' or '-')
qStart	Alignment start position in the query sequence
qEnd	Alignment end position in the query sequence
Id	Identification number of the chain

Source: <https://genome.ucsc.edu/goldenPath/help/chain.html>

If the strand value is '-', position coordinates are listed in terms of the reverse complemented sequence. The alignment data lines contain three required attribute values, the size of the ungapped alignment, the difference between the end of one block and the beginning of the next block in the reference sequence genome, the difference between the end of one block and the beginning of the next block in the query sequence. The last line of the alignment section contains the ungapped alignment size of the last block. The default minimum chain score was 1000 and the default linearGap was 'loose' for the primates and 'medium' for the mammal dataset.



After the alignments were chained, they were converted to AXT format and then to a MAF format file for comparison with the true alignment. These conversions were done using the freely available chainToAxt and axtToMaf programs from the UCSC Genome Bioinformatics Site (<https://genome.ucsc.edu/>). MAF stands for Multiple Alignment format and stores a series of multiple alignments in a format that is easy to parse and read. The file has a line oriented format in which each multiple alignment is a separate paragraph that begins with an "a" line and consists of lines that begin with an "s" for each sequence in the multiple alignment. An example of the MAF file format is shown in Figure 2.5.

```
##maf version=1 scoring=blastz

a score=176721.000000
s simCow.chrC 30152664 2236 + 33408597
AATGCTTGTGAATTAAGCCACTCCTTGCCCAGGGCATTATTAGAGATAGAGTCTCTCTGTGCATCCAGGCTGCAGTG
s simMouse.chrO 3116534 2238 + 3949899
AGTACTTTGAAATCATAGCCACCCCTTGCCCAATGCATTGCCGGAGATAGAGTCGCACTCTGTGCATCCAGGCTGCAATG
a score=76513.000000
s simCow.chrC 30153174 952 + 33408597
TGTGGATGCCTTTGGGGATAGAAATAACCAGACCACTGCACAAAGACAAGTTAGCGGGGACGCCTGGGACTAATACCC
s simMouse.chrO 3117046 952 + 3949899
TGGAGATGCCTTTGGGGATAGAAATAGCCAGACCACTGCACAGAGAAAAGTTAGCAGAGATGCCTGGGACTACTACCCA
```

**Figure 2.5** An example of a MAF format file.

The first line of the MAF file version number and the scoring scheme used for the alignment. Each alignment block begins with 'a' followed by the score of the pairwise alignment. The next two lines beginning with 's' have six fields which are explained in Table 2.5.

**Table 2.5** Attributes of the MAF File

<b>Attribute Name</b>	<b>Description</b>
src	Name of the genomes being aligned
start	The start position for the aligning region
size	The size of the aligning region in the sequence
strand	'+' or '-'. If '-' then the alignment is reverse complemented
srcsize	The size of the genome
text	The aligned nucleotides

Source: <https://genome.ucsc.edu/FAQ/FAQformat.html#format5>

The MAF files were compared with the true alignment for the predicted versus true comparison and the accuracy of the alignments were determined.

### **2.3 Pairwise Alignment with SSEARCH**

The primate and mammalian phylogeny data described above in the 'Datasets' section was used to generate pairwise sequence alignment using the SSEARCH36 program (version 36.3.6f). The query genome sequence was divided into same size fragments using a sliding window approach. A fixed sliding window size was chosen and when dividing the genome into fragments, the sliding window was used to move backward by the size of the window and then the genome was divided into a fragment of the same size. This was done to obtain overlap of nucleotides in the different fragments so as to achieve maximum genome coverage for the pairwise alignment. For example, if the fragment size is 500 and the sliding window has a size of 250, then the first fragment will consist of nucleotides from position 1 to 500. For constructing the second fragment, the sliding

window of size 250 will be used to go back 250 nucleotides from position 500 and then move forward by the fragment size of 500 to generate the second fragment. Hence the second fragment will have nucleotides from position 251 to position 750 of the genome. All the remaining fragments were constructed similarly. The number of fragments obtained from the entire genome is given by,

$$n = \frac{x}{(y - z)} \quad (2.1)$$

where  $n$  = number of fragments,  $x$  = genome size,  $y$  = fragment size,  $z$  = sliding window.

For the purpose of this study, Human chromosome D was chosen as the reference genome and Gorilla chromosome D was chosen as the query genome from the primates dataset. The mammalian analysis was done with Cow chromosome C as the reference genome sequence and Mouse chromosome O as the query genome sequence. An exact pairwise whole genome alignment was performed by aligning each individual fragment to the reference genome with the SSEARCH36 program. The scoring scheme used in this study is listed in Table 2.6.

**Table 2.6** Scoring Scheme for SSEARCH36

<b>Parameter</b>	<b>Value</b>
Match	+5
Mismatch	-4
Gap open	-26
Gap extend	-1

```

# ../fasta-36.3.6f/bin/ssearch36 -s ../score -f -26 -g -1 -n -b 1 -m 3 -W 0 window1.fa ../Human.ChrD.fa
SSEARCH performs a Smith-Waterman search
version 36.3.6 Aug., 2014(preload9)
Please cite:
T. F. Smith and M. S. Waterman, (1981) J. Mol. Biol. 147:195-197;
W.R. Pearson (1991) Genomics 11:635-650
Query: window1.fa
1>>>window1 - 500 nt
Library: ../Human.ChrD.fa
10572275 residues in 1 sequences
Statistics: (shuffled [500]) MLE statistics: Lambda= 0.1613; K=0.09174
statistics sampled from 71 (71) to 500 sequences
Algorithm: Smith-Waterman (SSE2, Michael Farrar 2006) (7.2 Nov 2010)
Parameters: ../score matrix (5:-4), open/ext: -26/-1
Scan time: 13.020
The best scores are:
simHuman.chrD (10572275) [f] 2431 569.3 2.3e-162
Smith-Waterman score: 2431; 98.8% identity (98.8% similar) in 497 nt overlap (4-500:9-505)
>window ..
CTTTAGATCTTGATAATGCTAATATGGCAGATTGCATTACTAGATACTAG
AGAGGTGAGCCAGGTTGATAGCTGCCACTCCATTATGCTGAAGATTCTCT
CTCTGGATTTTGCCACATGGCTTTTGCACCTAAGCTTCTAAGGGGTGGGT
TAAAAATGTGCTAATACTTAGAGGGGTAAAAAGGGGTCAATCAAGTTTAGG
CCCTTCTCCTCTTCTATAAAAAATTATAAGAATATTTTAAATTGATCAC
TGTGAGCCCAAAGACACAAGTGGAGTCAACACTTTCCAATAGGTTAGAAG
GCAATTTGAGATTTGTGTGGATCTCACCTCTCAGCTAGGGTCATGCTGAT
AGGGTGTCTGGATTTACAGAACACTATCATAATTGATTGTTGCAGGAA
TGTAGGACACTTGCAATTTATTGGATTTATCTGCTGGCTTGATCCAGAGTA
ATTCATTTGTACTTTTATTCATCAGGTGATGGGTTAATAATAGACTA
>simHum ..
CTTTAGATCTTGATAATGCTAATATGGCAGAGTGCATTACTAGATACTAG
AGAGGTGAGCCATGTTGATAGCTGCCACTCCATTATGCTGAAGATTCTCT
CTCTGGATTTTGCCACATGGCTTTTGCACCTAAGCTTCTAAGGGGTGGGT
TAAAAATGTGCTAATACTTAGAGGGGTAAAAAGGGGTCAATCAAGTTTATG
CCCTTCTCCTCTTCTATAAAAAATTATAAGAATACTTTTAAATTGATCAC
CGTGAGCCCAAAGACACAAGTGGAGTCAACACTTTCCAATAGGTTAGAAG
GCAATTTGAGATTTGTGTGGATCTCACCTCTCAGCTAGGGTCATGCTGAT
GGGGTGTCTGGATTTACAGAACACTATCATAATTGATTGTTGCAGGAA
TGTAGGACACTTGCAATTTATTGGATTTATCTGCTGGCTTGATCCAGAGTA
ATTCATTTGTACTTTTATTCATCAGGTGATGGGTTAATAATAGACTA
500 residues in 1 query sequences
10572275 residues in 1 library sequences
Tcomplib [36.3.6 Aug., 2014(preload9)] (8 proc in memory [0G])
start: Thu Feb 12 23:28:40 2015 done: Thu Feb 12 23:28:43 2015
Total Scan time: 13.020 Total Display time: 0.960
Function used was SSEARCH [36.3.6 Aug., 2014(preload9)]

```

**Figure 2.6** Example of SSEARCH36 output. The pairwise alignment of a fragment of the query genome to the reference genome is shown here.

For the downstream analysis, the SSEARCH36 output had to be converted to the AXT format which has been described in the previous section. Figure 2.6 shows the result of a SSEARCH36 alignment. The highest scoring alignment was chosen from each fragment-reference genome pairwise alignment as these were the best optimal local alignment found by the program.

The best alignment from each fragment was then written to the AXT file. Different combinations of fragment and sliding window sizes were used to find their optimal size that gave the best pairwise alignment. The `axtChain` program was used to chain the AXT output alignment blocks and `chainToAxt` and `axtToMaf` programs were used for the file format conversions for further analysis. The default parameters used for chaining were 1000 for the minimum chain score and `linearGap` was chosen as 'loose' for the primates. For the mammals data, the `linearGap` was 'medium', but, the minimum chain score had to be chosen to determine chains with the best accuracy. Since the mammals had divergent genomes, finding the optimal alignment was a challenging task. The pairwise genome alignment predicted by this method was then compared to the true alignments to ascertain their accuracy.

SSEARCH36 was set up to output only the two highest scoring alignments for each fragment-reference genome pair. For the primate analysis, only the highest scoring alignment was considered while for the mammalian dataset as the genomes were divergent a different strategy was employed. The highest scoring alignment was taken into account in this case only if the ratio of the second highest score to the highest score was less than a chosen threshold otherwise it was discarded. Various thresholds ranging from 0.7 to 0.95 were chosen to find the threshold that yielded the best accuracy. This

was done to eliminate duplicate alignments. This was followed by the chaining process where the default parameters were set for the primate data but for the mammals dataset, the minimum score for chaining had to be varied from 0 to 30 to find the score that provided the best accuracy. Thus, for the mammalian phylogeny comparisons were made with different thresholds for the ratio and then for each ratio, the alignments were chained by changing the minimum chain score. After the necessary file format conversions, the pairwise genome alignments predicted by this method were compared to the true alignments to ascertain their accuracy.

Since SSEARCH36 uses the Smith-Waterman algorithm to perform the exact local alignment, it can take days or even months to align two genomes and the process is computationally very expensive. All SSEARCH36 computations were done on the High Performance Computing (HPC) Kong cluster. The runtime was reduced considerably, by running them as a Sun Grid Engine (SGE) array job on the Kong machines. This was possible because the query genome was divided into fragments and SSEARCH36 would perform the fragment - reference alignment for each of these fragments. Each fragment-reference alignment was one task in the array. Only one shell script was written to submit the array job. If we had 10,000 fragments, then submitting an array job to do 10,000 computations is equivalent to submitting 10,000 separate scripts. Figure 2.7 shows an example script of an array job. The maximum tasks that the array could handle at a time were 60,000. Running the job as an array reduced the runtime from a few days to a few hours.

```
#!/bin/sh
# Name of job
#$ -N SSearch
# Tell the SGE that this is an array job, with "tasks" to be numbered 1 to
39498
#$ -t 1-39498
# When a single command in the array job is sent to a compute node,
# its task number is stored in the variable SGE_TASK_ID,
# so we can use the value of that variable to get the results we want:
# Make sure that the .e and .o file arrive in the working directory
#$ -cwd
# Send mail to these users
#$ -M ng245@njit.edu
##$ -m beas

../ssearch36 -r +5/-4 -f -26 -g -1 -n -b 2 -m 3 -W 0 window${SGE_TASK_ID}.fa
../../Cow.ChrC.fa > output/win${SGE_TASK_ID}.out
```

**Figure 2.7** Example of the SGE array job script. This was used to align the Cow and Mouse genomes using SSEARCH36 exact alignment program.

All other computations including the LASTZ genome alignment and the comparison of the true and predicted alignments were done on Open Source Lab (OSL) machines. The input files containing the genome sequences were in FASTA format. The scripts used for preprocessing the files for the downstream analysis were written in Python.

## CHAPTER 3

### CONCLUSION

#### 3.1 Results

The comparison of the predicted alignment to the true simulated alignment was done using a comparison tool called mafComparator from the suite of Multiple Alignment Format (MAF) tools that were developed for the Alignathon project (<https://github.com/dentearl/mafTools/tree/master/mafComparator>). This program takes two MAF files as input and compares the set of aligned pairs of nucleotides to one another. For each ordered pair of sequences in the first MAF file, mafComparator samples a set of homology tests. If we have two sets of pairwise alignments A and B, and we pick a pair of aligned positions in A called a homology pair, the AB homology test returns true if the homology pair is present in B otherwise it returns false. The set of possible homology tests for the ordered pair (A, B) may not be equivalent to the set of possible homology tests in the ordered pair (B, A). After sampling the homology pairs from the first MAF file, the program then reads the second MAF file to check if any of the sampled pairs from the first MAF is present in the second file. This comparison gives results for the (A, B) homology test. For the (B, A) homology test, the homology pairs were sampled from File B and checked to see if the same pairs were present in File A. The statistics were reported in an XML formatted file after the completion of the comparison of the true and predicted alignments.

If we consider two MAF files A and B, where A is the simulated true alignment and B is the predicted alignment created by a WGA program then the ratio of the number



of pairs in the intersection of A and B to the number of pairs in A is the recall or sensitivity of the prediction. The ratio of the number of pairs in the intersection of B and A to the number of pairs in B is known as the precision or positive predictive value of the prediction. The accuracy of the alignments was judged by calculating the F-Score, given in Equation 3.1, which is the harmonic mean of the precision and recall. The higher the F-Score the better is the accuracy of the pairwise genome alignment.

$$\text{F Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (3.1)$$

The XML output obtained from the mafComparator program was summarized using the comparatorSummarizer utility that was used in the Alignathon project for the analysis of the results submitted by the various teams. (<https://github.com/dentearl/mwgAlignAnalysis/blob/master/evaluations/src/comparatorWrapper/comparatorSummarizer.py>). The comparatorSummarizer utility calculates the true positives (TP), false negatives (FN), and false positives (FP) from the simulated truth and the predicted MAF files A and B, respectively. After which the program computes the precision, recall and F-Score of the true and predicted alignments. Equations 3.2 and 3.3 show the formula used for calculating the Precision and Recall by the comparatorSummarizer program.

$$\text{Precision} = \frac{\text{TP(B)}}{\text{TP(B)} + \text{FP(B)}} \quad (3.2)$$

$$\text{Recall} = \frac{\text{TP(A)}}{\text{TP(A)} + \text{FN(A)}} \quad (3.3)$$

The values of TP(A) and FN(A) are evaluated from the (A,B) homology test and the values of TP(B) and FP(B) are calculated from the (B,A) homology test.

The results of all the comparisons are listed in the Tables below. Table 3.1 summarizes the results for the Human chromosome D and Gorilla chromosome D using the LASTZ program. The alignments were done with gaps and without gaps and the accuracy was tested for both types of alignments with and without the chaining process. The F-Score is a little better for the alignments after chaining and the ungapped alignment has a better accuracy over the gapped one.

**Table 3.1** LASTZ Comparison Summary of the Primate Dataset

Method	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
LASTZ (with gaps, no chaining)	976177	274834	724482	9862	0.27502	0.99	0.430459
LASTZ (with gaps, after chaining)	976066	284820	714905	9822	0.2849	0.99004	0.442472
LASTZ (no gaps, no chaining)	889105	378937	621276	97086	0.37886	0.90155	0.533519
LASTZ (no gaps, after chaining)	888167	379274	620404	97762	0.3794	0.90084	0.533929

**Table 3.2** SSEARCH Comparison Summary of the Primate Dataset

Method	Fragment Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
SSEARCH36 (no chaining)	50	20	922142	910426	68152	64010	0.93036	0.93509	0.932719
SSEARCH36 (after chaining)	50	20	913137	997958	1310	74379	0.99869	0.92468	0.960261
SSEARCH36 (no chaining)	100	50	923245	918619	60617	60972	0.9381	0.93805	0.938075
SSEARCH36 (after chaining)	100	50	920348	999084	1925	65010	0.99808	0.93402	0.964988
SSEARCH36 (no chaining)	500	100	923386	930974	54467	62954	0.94473	0.93617	0.940431
SSEARCH36 (after chaining)	500	100	921226	996586	2639	63806	0.99736	0.93522	0.965291
SSEARCH36 (no chaining)	500	250	923655	922490	54734	61428	0.94399	0.93764	0.940804
SSEARCH36 (after chaining)	500	250	924703	979877	2846	62637	0.9971	0.93656	0.965882
SSEARCH36 (no chaining)	500	400	925946	911038	54490	59563	0.94356	0.93956	0.941556
SSEARCH36 (after chaining)	500	400	924878	955561	3337	60702	0.99652	0.93841	0.966592
SSEARCH36 (no chaining)	1000	500	925397	925436	52667	61696	0.94615	0.9375	0.941805
SSEARCH36 (after chaining)	1000	500	922873	975182	2400	62909	0.99754	0.93618	0.965886
SSEARCH36 (no chaining)	1000	900	926788	907632	52647	59357	0.94518	0.93981	0.942487
SSEARCH36 (after chaining)	1000	900	923752	939534	2288	60820	0.99757	0.93823	0.96699
SSEARCH36 (no chaining)	3000	2900	927826	907384	52097	60008	0.9457	0.93925	0.942464
SSEARCH36 (after chaining)	3000	2900	924059	926300	2249	61646	0.99758	0.93746	0.966586

Table 3.2 compares the results of the SSEARCH36 alignments of primates for various fragment and sliding window widths. There is an improvement in the F-Scores by a factor of approximately 0.02 after the alignments are chained because there is an increase in the number of true positives and a decrease in the number of false positives in the (B,A) homology test. In this case, a fragment size of 1000 nucleotides with a sliding window width of 900 nucleotides has the best accuracy for the pairwise whole genome alignment. For the primate dataset, the Smith-Waterman based SSEARCH36 program performs better than the hash table based LASTZ method.

**Table 3.3** LASTZ Comparison Summary of the Mammal Dataset

Method	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
LASTZ (no chaining)	135649	571253	428739	305034	0.57126	0.30782	0.400067
LASTZ (after chaining)	135466	593491	393325	305390	0.60142	0.30728	0.406744

Table 3.3 shows the results of the pairwise alignment using the LASTZ program for the mammal dataset. In this method, the accuracy is slightly better after chaining. The pairwise alignment of the mammals data with SSEARCH36 was performed for different sets of fragment and sliding window sizes. The fragment size and the corresponding sliding window size that were used are listed in Table 3.4. A summary of the output results for the fragment size and window size combinations for a minimum chaining score of 0 to 30 can be found in APPENDIX A. In this chapter, Tables 3.5 to 3.9 summarizes the results of the SSEARCH36 output for different chaining scores for a fragment size of 1000 nucleotides and a sliding window size of 900 nucleotides for a given ratio of the second best alignment score to the best alignment score.

**Table 3.4** Fragment and Sliding Window Sizes for SSEARCH Mammal Data

Fragment Size	Sliding window width
250	125
500	250
500	300
500	400
1000	500
1000	800
1000	900
3000	1500
5000	2500

**Table 3.5** SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	900	112678	439800	336534	327982	0.56651	0.2557	0.352359
25	1000	900	116035	436679	351293	324520	0.55418	0.26338	0.357062
20	1000	900	118450	432368	366045	323571	0.54153	0.26797	0.358527
15	1000	900	119151	426672	378450	320936	0.52995	0.27074	0.358388
10	1000	900	119870	421082	391504	320577	0.5182	0.27216	0.356884
5	1000	900	120562	413311	407215	320132	0.50371	0.27357	0.354570
0	1000	900	120862	405179	422882	319578	0.48931	0.27441	0.351625

From Table 3.5, it can be seen that the best accuracy for second best score to best score ratio 0.95 is for a minimum chaining score of 20.

**Table 3.6** SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	900	111368	467578	281149	328999	0.6245	0.2529	0.360009
25	1000	900	114615	464273	298011	325744	0.60906	0.26028	0.364705
20	1000	900	117775	459575	314144	322889	0.59398	0.26727	0.368657
15	1000	900	118237	453701	328694	322249	0.57989	0.26842	0.366975
10	1000	900	119371	447014	343446	321967	0.56551	0.27048	0.365935
5	1000	900	119310	438997	361294	320678	0.54855	0.27117	0.362930
0	1000	900	118916	430576	377970	320717	0.53253	0.27049	0.358756

**Table 3.7** SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.85)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	900	110437	480523	253430	329980	0.65471	0.25076	0.362630
25	1000	900	114184	477196	271445	326622	0.63742	0.25903	0.368366
20	1000	900	116368	472011	289009	324208	0.62023	0.26413	0.370486
15	1000	900	116942	466240	304232	322804	0.60514	0.26593	0.369488
10	1000	900	118019	458928	319892	321969	0.58926	0.26823	0.368651
5	1000	900	118199	450136	338641	321262	0.57068	0.26896	0.365609
0	1000	900	118232	441894	356029	321712	0.55381	0.26874	0.361877

Table 3.6 and Table 3.7 show that the highest F-Score is for a minimum chaining score of 20 for score ratios 0.9 and 0.85.

**Table 3.8** SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.8)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	900	109914	478899	230346	332108	0.67522	0.24866	0.363468
25	1000	900	112612	484812	254820	327285	0.65548	0.256	0.368199
20	1000	900	114690	479914	272496	325357	0.63784	0.26063	0.370052
15	1000	900	116972	473830	288810	324844	0.6213	0.26475	0.371286
10	1000	900	116700	466569	304622	323921	0.605	0.26485	0.368418
5	1000	900	117849	457661	324416	323429	0.58519	0.26706	0.366749
0	1000	900	117489	449144	342648	323806	0.56725	0.26624	0.362391

**Table 3.9** SSEARCH Mammals Summary (Size = 1000, Width = 900, Ratio < 0.75)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	900	107949	471854	215577	332844	0.6864	0.2449	0.360999
25	1000	900	110387	484866	243926	329902	0.6653	0.25071	0.364182
20	1000	900	113257	483159	263900	328065	0.64675	0.25663	0.367454
15	1000	900	114357	476914	280496	325910	0.62966	0.25974	0.367771
10	1000	900	114977	469797	296566	325807	0.61302	0.26085	0.365973
5	1000	900	116045	460968	315968	325482	0.59332	0.26283	0.364287
0	1000	900	115884	451599	334990	324827	0.57412	0.26295	0.360698

In Tables 3.8 and 3.9, a minimum chaining score of 15 provides the best accuracy for score ratios 0.8 and 0.75, respectively. The F-Score increases as the score ratio decreases from 0.95 to 0.8, but for a ratio of 0.75 the F-Score values start decreasing. With a fragment size of 1000 and window size of 900 the highest F-Score value was observed for a score ratio of 0.8.

It was observed that the F-Score increased as the fragment sizes increased, with 1000 and 900 being the optimal fragment and sliding window sizes for this dataset also. Increasing the fragment size to 3000 and 5000 or decreasing it to 250 or 500 reduced the F-Score considerably. A fragment size of 1000 nucleotides with an overlap of 900 nucleotides, a minimum chain score of 15 and a ratio of the second best to best score of 0.8 yielded the highest F-Score and hence the best accuracy of the pairwise alignment of the genomes. The exact alignment method scored less than LASTZ by a factor of 0.03 for the mammalian phylogeny.

### **3.2 Discussion**

An exact alignment approach was employed to align two whole genomes from the primate and mammalian phylogenies and the result compared to the output of an established hash table based alignment procedure. The exact method, using a rigorous Smith-Waterman local alignment algorithm provided much better accuracies than the hash table based LASTZ program for the pairwise alignment of the primate genomes. However, for the mammalian genomes, the exact alignment program SSEARCH yields accuracies that are lower than those of LASTZ.

While the exact alignment method improves the accuracy for the primates, improvement in the F-Score and in turn the accuracy is not observed for the mammalian dataset. The genomes from the primate dataset are from closely related species and so it is much easier to align two similar genome sequences. The genomes from the mammal dataset are much more divergent and hence difficult to align. This could be one of the reasons for not obtaining a higher accuracy when aligning the Cow and Mouse genomes



from the mammal dataset. Another possibility could be the chaining of the alignments after the local pairwise alignment with SSEARCH36. The chaining process produces a sequence of gapless blocks with no overlapping of blocks in the target or query positions in the chain. Since the species are divergent, the initial alignment can have a lot of gaps and this might result in abnormalities during the chaining step which reduces the number of true positives and as a result the F-Score values are also low. However, the Smith-Waterman algorithm based programs, have performed better than hash table based methods for some other Bioinformatics problems like mapping divergent short reads to a genome (Turki, T. and Roshan, U. (2014) MaxSSmap: a GPU program for mapping divergent short reads to genomes with the maximum scoring subsequence, *BMC genomics*, **15**, 969)

Further investigation is needed to determine the reasons for not obtaining a better accuracy of alignment with the proposed method. The chaining process needs to be evaluated and necessary corrections will have to be implemented to improve the performance of the proposed method for divergent genomic data. This study has formulated an exact sequence alignment method for whole genomes and laid the foundation for further research in this rapidly growing area of whole genome analysis to determine the evolutionary relationship at the nucleotide level between two or more genomes.

## APPENDIX A

### SUMMARY OF SSEARCH RESULTS FOR MAMMALS DATASET

Tables A.1 to A.23 compares the results of the pairwise alignment of the Cow Chr C and Mouse Chr O. Each table compares the accuracy for different chaining scores for a given fragment size, sliding window width and ratio of the second best alignment score to the best alignment score. In Tables A.1 to A.4, NC stands for No Chaining of the alignments.

**Table A.1** SSEARCH Mammals Summary (Size = 250, Width = 125, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
NC	250	125	103118	227497	723913	336406	0.23912	0.23461	0.236844
30	250	125	98591	416481	491083	341945	0.4589	0.2238	0.30087
25	250	125	100525	416210	494363	341557	0.45709	0.22739	0.303698
20	250	125	101653	415120	498090	339132	0.45457	0.23062	0.305997
15	250	125	102433	413870	501227	337493	0.45227	0.23284	0.307415
10	250	125	103665	412654	504232	337646	0.45006	0.2349	0.308687
5	250	125	103227	410326	507839	337969	0.4469	0.23397	0.30714
0	250	125	103151	407726	511646	337640	0.44348	0.23401	0.306362

**Table A.2** SSEARCH Mammals Summary (Size = 250, Width = 125, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
NC	250	125	98586	263659	670861	342198	0.28213	0.22366	0.249515
30	250	125	94662	413424	301108	345590	0.57859	0.21502	0.313525
25	250	125	96139	421523	313047	345778	0.57384	0.21755	0.315493
20	250	125	97965	426964	323891	343397	0.56864	0.22196	0.31929
15	250	125	98322	430250	332278	342007	0.56424	0.22329	0.31996
10	250	125	98765	431936	338748	342935	0.56046	0.2236	0.319666
5	250	125	99084	432480	345285	341512	0.55605	0.22489	0.320255
0	250	125	99322	432754	352534	340852	0.55108	0.22564	0.320182

**Table A.3** SSEARCH Mammals Summary (Size = 500, Width = 250, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
NC	500	250	113681	258645	678623	327232	0.27596	0.25783	0.266587
30	500	250	106982	467135	343449	334030	0.57629	0.24258	0.341437
25	500	250	109557	479235	364551	330976	0.56796	0.24869	0.345916
20	500	250	111932	488506	384314	329144	0.55969	0.25377	0.349206
15	500	250	112610	493536	400593	327980	0.55197	0.25559	0.349393
10	500	250	112999	495827	413296	327316	0.54539	0.25663	0.349027
5	500	250	113394	492854	420079	327916	0.53986	0.25695	0.348181
0	500	250	113152	488251	426766	326780	0.5336	0.2572	0.347096

**Table A.4** SSEARCH Mammals Summary (Size = 500, Width = 250, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
NC	500	250	112022	281131	646431	328393	0.30309	0.25436	0.276595
30	500	250	104709	459827	239374	335867	0.65765	0.23766	0.349146
25	500	250	107574	471661	258364	332800	0.64609	0.24428	0.35452
20	500	250	110021	480773	276280	329981	0.63506	0.25005	0.358818
15	500	250	110899	485663	291225	328942	0.62514	0.25213	0.359334
10	500	250	111796	487909	302763	329277	0.61708	0.25346	0.359329
5	500	250	112188	488998	312472	328037	0.61013	0.25484	0.359517
0	500	250	111869	489345	323316	328953	0.60215	0.25377	0.35706

**Table A.5** SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	300	110004	481949	380763	330834	0.55864	0.24953	0.344971
25	500	300	113164	485822	396461	328489	0.55064	0.25623	0.349723
20	500	300	115115	482126	404907	325406	0.54353	0.26132	0.352948
15	500	300	115895	478569	413244	324609	0.53662	0.2631	0.353085
10	500	300	117076	474606	420026	323257	0.5305	0.26588	0.354226
5	500	300	117520	470735	426297	323093	0.52477	0.26672	0.353679
0	500	300	117140	465585	433737	323087	0.51771	0.26609	0.351512

**Table A.6** SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	300	108126	473311	264058	331846	0.64189	0.24576	0.355435
25	500	300	110960	485419	284935	329736	0.63012	0.25178	0.359795
20	500	300	113417	494559	304231	327984	0.61914	0.25695	0.363177
15	500	300	114305	499949	320495	326755	0.60936	0.25916	0.363657
10	500	300	114814	502397	333051	325630	0.60135	0.26068	0.363699
5	500	300	115004	503597	344420	325583	0.59385	0.26102	0.362644
0	500	300	115371	503906	357665	325696	0.58487	0.26157	0.361477

**Table A.7** SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.85)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	300	105799	463294	214510	334668	0.68352	0.2402	0.355479
25	500	300	109169	475078	233793	331763	0.67019	0.24759	0.361595
20	500	300	111134	484063	251940	329983	0.65769	0.25194	0.36432
15	500	300	112208	489357	267085	327792	0.64692	0.25502	0.365828
10	500	300	112595	491745	278949	328550	0.63805	0.25523	0.36461
5	500	300	112782	492908	289611	328424	0.6299	0.25562	0.363662
0	500	300	112672	493217	302217	327706	0.62006	0.25585	0.362234

**Table A.8** SSEARCH Mammals Summary (Size = 500, Width = 300, Ratio < 0.8)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	300	103299	452069	188961	336470	0.70522	0.23489	0.352404
25	500	300	106503	463646	207164	334320	0.69117	0.2416	0.358045
20	500	300	107745	472336	224287	332874	0.67804	0.24453	0.359433
15	500	300	109018	477368	238617	331436	0.66673	0.24751	0.361004
10	500	300	109206	479727	249888	330760	0.65751	0.24821	0.360378
5	500	300	110505	480876	260118	331369	0.64896	0.25008	0.361034
0	500	300	109715	481169	272126	330738	0.63875	0.2491	0.358422

**Table A.9** SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	400	113903	433130	395416	326967	0.52276	0.25836	0.345812
25	500	400	117161	431550	405156	324470	0.51577	0.26529	0.350366
20	500	400	119866	429093	412840	321151	0.50965	0.27179	0.354519
15	500	400	121077	425905	421500	319212	0.5026	0.27499	0.355483
10	500	400	121921	422349	428266	319320	0.49652	0.27631	0.355042
5	500	400	121643	418154	437677	318246	0.48859	0.27653	0.353173
0	500	400	121863	414175	445059	319510	0.48203	0.2761	0.351097

**Table A.10** SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	400	113221	475621	312327	327967	0.60362	0.25663	0.360144
25	500	400	115346	474178	324261	325379	0.59388	0.26172	0.363325
20	500	400	118261	471741	335395	322407	0.58446	0.26837	0.367838
15	500	400	118614	467495	345707	321990	0.57488	0.26921	0.366699
10	500	400	119492	464371	354435	320893	0.56713	0.27134	0.367062
5	500	400	119791	459099	364527	321374	0.55741	0.27153	0.365174
0	500	400	120180	454015	374183	321479	0.5482	0.27211	0.363693

**Table A.11** SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.85)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	400	110940	485208	263363	329799	0.64818	0.25171	0.362607
25	500	400	113354	494597	284149	326153	0.63512	0.25791	0.366849
20	500	400	115324	491981	296507	324439	0.62395	0.26224	0.369277
15	500	400	116953	488474	307463	322903	0.61371	0.26589	0.371031
10	500	400	117481	484064	316748	322616	0.60447	0.26694	0.370336
5	500	400	118308	478856	327867	321841	0.59358	0.26879	0.370023
0	500	400	118165	473311	339046	322272	0.58264	0.26829	0.367402

**Table A.12** SSEARCH Mammals Summary (Size = 500, Width = 400, Ratio < 0.8)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	500	400	108236	473176	236670	332203	0.66659	0.24575	0.359108
25	500	400	110924	485034	258215	330514	0.65259	0.25128	0.362846
20	500	400	113194	494650	278755	327524	0.63957	0.25684	0.3665
15	500	400	114597	494205	292158	326041	0.62847	0.26007	0.367898
10	500	400	114970	490249	301890	324759	0.61889	0.26146	0.367615
5	500	400	115339	484623	313388	325722	0.60729	0.2615	0.36558
0	500	400	114782	479547	324359	324671	0.59652	0.26119	0.36331

**Table A.13** SSEARCH Mammals Summary (Size = 1000, Width = 500, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	500	103329	452387	262397	337551	0.6329	0.23437	0.342068
25	1000	500	106096	465002	287595	334369	0.61786	0.24087	0.346614
20	1000	500	108412	474785	310981	331514	0.60423	0.24643	0.350082
15	1000	500	109751	480161	329975	330604	0.59269	0.24923	0.350903
10	1000	500	111241	482855	345781	329935	0.58271	0.25215	0.351988
5	1000	500	110456	484053	358746	330112	0.57434	0.25071	0.349052
0	1000	500	110594	484456	372015	330263	0.56564	0.25086	0.347572



**Table A.14** SSEARCH Mammals Summary (Size = 1000, Width = 500, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	500	105648	462137	234912	334202	0.66299	0.24019	0.352629
25	1000	500	105582	462137	234912	335120	0.66299	0.23958	0.351971
20	1000	500	107582	471808	257007	332067	0.64736	0.2447	0.355153
15	1000	500	109296	477138	275154	331649	0.63425	0.24787	0.356440
10	1000	500	110071	479825	290283	331524	0.62306	0.24926	0.356071
5	1000	500	110336	481009	302794	330604	0.61369	0.25023	0.355504
0	1000	500	109816	481412	315465	329731	0.60412	0.24984	0.353490

**Table A.15** SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	800	110215	482084	336223	330876	0.58912	0.24987	0.350906
25	1000	800	113000	477974	352203	328060	0.57575	0.2562	0.354606
20	1000	800	115659	472225	366127	325263	0.56328	0.26231	0.357935
15	1000	800	117844	466127	378664	323937	0.55177	0.26675	0.359636
10	1000	800	117756	459648	389586	323079	0.54125	0.26712	0.357704
5	1000	800	117977	452948	401196	323426	0.53029	0.26728	0.355419
0	1000	800	118263	446462	412683	322414	0.51966	0.26837	0.353949

**Table A.16** SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	800	108630	477879	263227	331542	0.64482	0.24679	0.356961
25	1000	800	113084	491000	291951	328212	0.62711	0.25625	0.363831
20	1000	800	115615	500583	318723	326264	0.61098	0.26164	0.366384
15	1000	800	116421	493961	332320	324506	0.59781	0.26404	0.366295
10	1000	800	116714	487191	344725	324622	0.58563	0.26446	0.364375
5	1000	800	116970	479700	357953	323611	0.57267	0.26549	0.362790
0	1000	800	117166	472705	370027	324512	0.56092	0.26527	0.360196

**Table A.17** SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.85)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	800	108215	472820	228694	331900	0.674	0.24588	0.360315
25	1000	800	111097	485798	256361	329619	0.65457	0.25208	0.363986
20	1000	800	113520	496379	283069	326133	0.63683	0.2582	0.367428
15	1000	800	114294	501853	305400	325966	0.62168	0.25961	0.366268
10	1000	800	115403	500134	322237	324521	0.60816	0.26232	0.366539
5	1000	800	115808	492605	336142	324252	0.5944	0.26316	0.364808
0	1000	800	115428	485539	348334	325689	0.58227	0.26167	0.361074

**Table A.18** SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.8)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	800	106772	468164	206999	333664	0.69341	0.24242	0.359246
25	1000	800	110117	481050	233713	330950	0.67302	0.24966	0.364213
20	1000	800	112698	491554	259705	328029	0.65431	0.25571	0.367714
15	1000	800	113598	497009	281463	326550	0.63844	0.25809	0.367584
10	1000	800	114211	499871	300994	326234	0.62416	0.25931	0.366398
5	1000	800	114733	501190	321449	326065	0.60925	0.26028	0.364739
0	1000	800	115118	494409	334586	326024	0.5964	0.26095	0.363050

**Table A.19** SSEARCH Mammals Summary (Size = 1000, Width = 800, Ratio < 0.75)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	1000	800	105831	462666	190477	334990	0.70837	0.24008	0.358618
25	1000	800	108336	475401	216473	332770	0.68712	0.2456	0.361859
20	1000	800	111897	485809	241899	329786	0.66759	0.25334	0.367297
15	1000	800	112214	491240	263254	327667	0.65109	0.2551	0.366574
10	1000	800	112598	494018	282348	327698	0.63632	0.25573	0.364836
5	1000	800	112900	495313	302164	327154	0.6211	0.25656	0.363123
0	1000	800	113652	495641	320488	327796	0.60731	0.25745	0.361608

**Table A.20** SSEARCH Mammals Summary (Size = 3000, Width = 1500, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	3000	1500	90004	393719	205535	350388	0.65702	0.20437	0.311764
25	3000	1500	92803	405297	233537	347282	0.63443	0.21088	0.316543
20	3000	1500	94967	414254	261328	345110	0.61318	0.2158	0.319246
15	3000	1500	95856	419431	282623	344900	0.59743	0.21748	0.318880
10	3000	1500	96805	421813	299433	344217	0.58484	0.2195	0.319199
5	3000	1500	96334	422882	312981	344206	0.57467	0.21867	0.316795
0	3000	1500	96639	423341	327358	344053	0.56393	0.21929	0.315784

**Table A.21** SSEARCH Mammals Summary (Size = 3000, Width = 1500, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	3000	1500	89131	388820	174957	351703	0.68967	0.20219	0.312705
25	3000	1500	90868	400277	201792	350269	0.66484	0.20599	0.314528
20	3000	1500	93199	409115	228432	346698	0.6417	0.21187	0.318561
15	3000	1500	94737	414268	248828	346496	0.62475	0.21471	0.319587
10	3000	1500	95445	416628	265052	345788	0.61118	0.21631	0.319531
5	3000	1500	95303	417643	278159	345542	0.60023	0.21618	0.317874
0	3000	1500	95631	418102	292096	344306	0.58871	0.21737	0.317507

**Table A.22** SSEARCH Mammals Summary (Size = 5000, Width = 2500, Ratio < 0.95)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	5000	2500	82706	362320	181386	358368	0.66639	0.18751	0.292668
25	5000	2500	86002	372755	209047	355597	0.64069	0.19475	0.298703
20	5000	2500	87455	381187	236888	353241	0.61673	0.19845	0.300277
15	5000	2500	95856	419431	282623	344900	0.59743	0.21748	0.318880
10	5000	2500	88744	388209	274569	352682	0.58573	0.20104	0.299338
5	5000	2500	88618	389196	287463	350847	0.57517	0.20165	0.298610
0	5000	2500	88616	389608	300935	351775	0.56421	0.20122	0.296645

**Table A.23** SSEARCH Mammals Summary (Size = 5000, Width = 2500, Ratio < 0.9)

Minimum Chaining Score	Size	Width	TP (A)	TP (B)	FP (B)	FN (A)	Precision (PPV)	Recall (Sensitivity)	F-Score
30	5000	2500	81711	356206	155986	359703	0.69545	0.18511	0.292393
25	5000	2500	83982	366406	182432	356668	0.6676	0.19059	0.296526
20	5000	2500	85904	374690	209162	355092	0.64176	0.1948	0.298878
15	5000	2500	87032	379477	229793	354664	0.62284	0.19704	0.299372
10	5000	2500	87654	381634	245696	353084	0.60835	0.19888	0.299763
5	5000	2500	87608	382567	258175	352916	0.59707	0.19887	0.298362
0	5000	2500	87711	382979	271448	353966	0.58521	0.19859	0.296547

## APPENDIX B

### PYTHON SCRIPTS FOR PREPROCESSING FILES

#### B.1 Script for Dividing the Query Sequence into Same Size Fragments

```
# Makewindows.py
# Chop the sequence into fragments of size x with a sliding window of size y

import os
import math
fileo = open('../Mouse.ChrO.fa', 'r')
line1 = fileo.readline().strip()
line2 = fileo.readline().strip()
seq1_length = len(line2)
width = 1000
overlap = 900
num_windows = math.floor(seq1_length/(width-overlap))
print (seq1_length)
print (num_windows)
fileo.close()
fileo = open('../Mouse.ChrO.fa', 'rb')
line1 = fileo.readline()
position = fileo.seek(0,1)
strand = fileo.read(width)
foname = "window1.fa"
fo = open(foname, 'wb')
fo.write(b'>window1\n')
fo.write(strand);
fo.close()
count = 2
while (count <= num_windows+1):
    position = fileo.seek(-overlap,1)
    strand = fileo.read(width)
    foname = "window"+str(count)+".fa"
    fo = open(foname, "wb")
    outtitle = ">window"+ str(count)+"\n"
    fo.write(outtitle.encode('utf-8'));
    fo.write(strand);
    fo.close()
    count = count + 1
fileo.close()
print("Done")
```

## B.2 Script to Find the Alignments for a Given Score Ratio

# Find windows that have the best score. The ratio of the second best score to the best #score should not be greater than a given ratio

```
from __future__ import division
count = 1
scoreratio = 0.95
#scoreratio = 0.9
#scoreratio = 0.85
#scoreratio = 0.8
#scoreratio = 0.75
#scoreratio = 0.7

numwindows = 39498
str1 = "simCow.chrC"
str2 = "!! No sequences with E() < 2"
fo = open("Bestscorewindows.out", 'w')
while (count <= numwindows):
    bestscore1=[]
    bestscore2=[]
    scout = 0
    lineno = 0
    windownum = 0
    ratio = 0
    fname = "output/" + "win"+str(count)+".out"
    f = open(fname, 'r')
    for line in f:
        lineno = lineno + 1
        if (line[0:11].strip() == str1 and scout == 0):
            bestscore1 = line.split()
            found = lineno
            scout = 1
            windownum = count
        if (line[0:11].strip() == str1 and scout == 1 and lineno > found):
            bestscore2 = line.split()
            break
        if ( line.strip() == str2):
            break
    f.close()
    if (len(bestscore1) != 0 and len(bestscore2) != 0):
        a = int(bestscore1[3])
        b = int(bestscore2[3])
        ratio = b/a
        if (ratio < scoreratio):
            fo.write(str(windownum) + "\n")
```

```

    if (len(bestscore1) != 0 and len(bestscore2) == 0):
        fo.write(str(windownum) + "\n")
        count = count + 1
fo.close()
print ("Done")

```

### B.3 Script to Convert SSEARCH Output to AXT Format

```

# Convert ssearch output to AXT format. Only those windows having the best scores
# (ratio of second best score to best score is less than a given ratio) are considered in the
# AXT file

```

```

import os
import math

```

```

fileo = open('Bestscorewindows.out', 'r')
width = 1000
overlap = 900

```

```

genome1 = "simCow.chrC" # Target
genome2 = "simMouse.chrO" # Query
checkstr1 = "Smith-Waterman score"
checkstr2 = ">window" # Query
checkstr3 = ">simCow" # Target
flag = 0
fo = open('CO_SSsw1000ov900br8.axt', 'w')

```

```

count = 1

```

```

for line in fileo:

```

```

    windownum = line.strip()
    lineno = 0
    qfound = 0
    tfound = 0
    query = "" # Mouse
    target = "" # Cow
    querystartpos = 0
    queryendpos = 0
    targetstartpos = 0
    targetendpos = 0
    writestr = ""
    fname = "output/" + "win"+str(windownum)+".out"
    f = open(fname, 'r')
    for line in f:

```



```

    lineno = lineno + 1
    if line[0:20] == checkstr1:
        index0 = line.index(";")
        score = line[22:index0]
        index1 = line.index("in")
        index2 = line.index("nt overlap (")
        length = len(line[index2+12:])
        str1 = line[index2+12:index2+12+length-2]
# Extract position of Query
        sub1 = str1.find('-')
        sub2 = str1[0:sub1]
        sub3 = str1.find('.')
        sub4 = str1[sub1+1:sub3]
# Extract positions of Target
        sub5 = str1.rfind('-')
        sub6 = str1[sub3+1:sub5]
        sub7 = str1[sub5+1:]
# Query start position
        querystartpos = ((int(windownum)-1)*(width-overlap))+int(sub2)
# Query end position
        queryendpos = ((int(windownum)-1)*(width-overlap))+int(sub4)
# Target start position
        targetstartpos = int(sub6)
# Target end position
        targetendpos = int(sub7)
        if (queryendpos < querystartpos):
            strandvalue = "-"
            writestr = str(count-1) + " " + genome1 + " " + str(targetstartpos)+ " " +
str(targetendpos) + " " + genome2 + " " + str(queryendpos) + " " + str(querystartpos) + " "
+ str(strandvalue) + " " + score
        else:
            strandvalue = "+"
            writestr = str(count-1) + " " + genome1 + " " + str(targetstartpos)+ " " +
str(targetendpos) + " " + genome2 + " " + str(querystartpos) + " " + str(queryendpos) + " "
+ str(strandvalue) + " " + score

    if (line[0:7] == checkstr2):
        qfound = lineno
        flag = 1
    elif (line[0:7] == checkstr3):
        flag = 2
        tfound = lineno
    if (flag == 1 and lineno > qfound):
        query = query + line.rstrip('\n')
    if (flag == 2 and lineno > tfound):
        target = target + line.rstrip('\n')

```

```
    if (line == '\n' and qfound != 0):
        fo.write(writestr + '\n')
        fo.write(target + '\n')
        fo.write(query + '\n')
        fo.write('\n')
        f.close()
        flag = 0
        break
    count = count + 1
fo.close()
fileo.close()
print("Done")
```

## REFERENCES

- Angiuoli, S.V. and Salzberg, S.L. (2011) Mugsy: fast multiple alignment of closely related whole genomes, *Bioinformatics*, **27**, 334-342.
- Blanchette, M., *et al.* (2004) Aligning multiple genomic sequences with the threaded blockset aligner, *Genome research*, **14**, 708-715.
- Brudno, M., *et al.* (2003) LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA, *Genome research*, **13**, 721-731.
- Darling, A.E., Mau, B. and Perna, N.T. (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement, *PLoS one*, **5**, e11147.
- Earl, D., *et al.* (2014) Alignathon: a competitive assessment of whole-genome alignment methods, *Genome research*, **24**, 2077-2089.
- Edgar, Robert, *et al.* Evolver: a whole-genome sequence evolution simulator website. <http://www.drive5.com/evolver>
- Harris, R.S. (2007) Improved pairwise alignment of genomic DNA. Ph.D. Thesis, The Pennsylvania State University.
- Kim, J. and Ma, J. (2014) PSAR-align: improving multiple sequence alignment using probabilistic sampling, *Bioinformatics*, **30**, 1010-1012.
- Paten, B., *et al.* (2008) Enredo and Pecan: genome-wide mammalian consistency-based multiple alignment with paralogs, *Genome research*, **18**, 1814-1828.
- Paten, B., *et al.* (2009) Sequence progressive alignment, a framework for practical large-scale probabilistic consistency alignment, *Bioinformatics*, **25**, 295-301.
- Paten, B., *et al.* (2011) Cactus: Algorithms for genome multiple sequence alignment, *Genome research*, **21**, 1512-1528.
- Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison, *Proceedings of the National Academy of Sciences of the United States of America*, **85**, 2444-2448.
- Schwartz, S., *et al.* (2003) Human-mouse alignments with BLASTZ, *Genome research*, **13**, 103-107.
- Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences, *Journal of molecular biology*, **147**, 195-197.
- Turki, T. and Roshan, U. (2014) MaxSSmap: a GPU program for mapping divergent short reads to genomes with the maximum scoring subsequence, *BMC genomics*, **15**, 969.