**Bio Informatics Computing**

MUKHASIR SHAH SYED

109137268

Task: Using the stimulated reads from selected 250 gene sequences of HG19 chr1.fa which are mapped on the genome using BowTie and BowTie output files are passed to a program(code) which will calculate mappability percentile for each read length of 50, 70 and 100. Using the mappability percentile, we will create graphs for mappability analysis which would be done using R.

Steps of operation:

1. Firstly, we have to prepare simulated read files of read lengths 50, 70 and 100 from chr1-250.fa which contains selected 250 gene sequences. Filenames: reads50.fa, reads70.fa and reads100.fa
2. Secondly, we have to map the simulated reads on the genome using BowTie. Below mentioned are steps are used to generate output (BED format) which Chromosome Name, Map Start index, Map End Index, Tile ID, dummy score and strand value.
3. We will create indexes of Chr1.fa file using below command and place the indexes in the indexes folder.

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ ./bowtie-build chr1.fa CHR1

1. Now map the reads50.fa to chr1.fa using Bowtie

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ ./bowtie -f -m1 -v2 CHR1 reads50.fa > BTout50-v2-m1 # reads processed: 1022516

# reads with at least one reported alignment: 760080 (74.33%)

# reads that failed to align: 0 (0.00%)

# reads with alignments suppressed due to -m: 262436 (25.67%)

Reported 760080 alignments to 1 output stream(s)

1. Now get the required 6 columns and make the bowtie output in BED format.

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ cut -f1-5,7 < BTout50-v2-m1 >BTout-6col-50

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ awk '{print $3 "\t" $4 "\t" $4+length($5) "\t" $1 "\t" $6 "\t" $2;}' BTout-6col-50 > BTout-BED-50

1. Now map the reads70.fa to chr1.fa using Bowtie

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ ./bowtie -f -m1 -v2 CHR1 reads70.fa > BTout70-v2-m1 # reads processed: 1017554

# reads with at least one reported alignment: 795946 (78.22%)

# reads that failed to align: 0 (0.00%)

# reads with alignments suppressed due to -m: 221608 (21.78%)

Reported 795946 alignments to 1 output stream(s)

1. Now get the required 6 columns and make the bowtie output in BED format.

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ cut -f1-5,7 < BTout70-v2-m1 >BTout-6col-70

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ awk '{print $3 "\t" $4 "\t" $4+length($5) "\t" $1 "\t" $6 "\t" $2;}' BTout-6col-70 > BTout-BED-70

1. Now map the reads100.fa to chr1.fa using Bowtie

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ ./bowtie -f -m1 -v2 CHR1 reads100.fa > BTout100-v2-m1 # reads processed: 1010575

# reads with at least one reported alignment: 816456 (80.79%)

# reads that failed to align: 0 (0.00%)

# reads with alignments suppressed due to -m: 194119 (19.21%)

Reported 816456 alignments to 1 output stream(s)

1. Now get the required 6 columns and make the bowtie output in BED format.

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ cut -f1-5,7 < BTout100-v2-m1 >BTout-6col-100

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7/bowtie-0.12.7

$ awk '{print $3 "\t" $4 "\t" $4+length($5) "\t" $1 "\t" $6 "\t" $2;}' BTout-6col-100 > BTout-BED-100

1. Thirdly, as we have got Bowtie output for each read length these files would be passed to the program to calculate the mappability percentile for each read length and place it all values in a file for corresponding.
2. Finally, output file from above step would be used to build histograms and scatter plot for mappability analysis using "R".

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* Date : April 14 2015 Author : Mukhasir Shah Syed

\* Below program is used for checking the mappability of the 250 gene sequences within the scope of HG19 chr1.

\* In this program we used 3 different read lengths, which are 50, 70 and 100. As reads50.fa and reads70.fa are

\* provided , we have prepared reads100.fa by creating program for that.

\* Now we have used Bowtie to get the mappability reads for each gene by using reads50.fa, reads70.fa and reads100.fa files individually.

\* We would create BowTie output (BED format) for each readlength.

\* In this program we will use BowTie output and get the Unique GeneID's and

\* mapped read count and GeneLength to calculate the Mappability Percentile value for each GeneID.

\* Mappability Percentile = ((# of mapped reads)/(GeneLength-readLength+1))\*100 formula used for calculation.

\* Above step would be repeated for Bowtie Output of read length 50,70 and 100.

\* After getting mappability values for all 3 read length 50, 70 and 100, we will combile all values into one file.

\* Final file would contains all values for read length 50, 70 & 100 for respective geneId's

\* Output file format below:

\* geneID readLen50 readLen70 readLen100

NR\_046018 100.0 100.0 100.0

NR\_026818 94.66 99.79 100.0

NM\_001005484 100.0 100.0 100.0

NR\_039983 0.57 1.87 6.46

\* To complie/run the code we have to create a Console Application using Visual Studio 2013 and compile/build this application which will create an executable file then run the

".exe" application to execute the code. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GeneMap\_Bowtie\_R

{

class Program

{

static void Main(string[] args)

{

string Seq = string.Empty;

string ID = string.Empty;

//Initialize KeyValuePair List for storing query sequences.

List<KeyValuePair<string, string>> CHRData = new List<KeyValuePair<string, string>>();

//Loop through chr1-250.fa file to store all query sequences into list.

foreach (string Chr250Data in File.ReadLines(@"E:\Bio Informatics\Assignment 7\chr1-250.fa"))

{

if (Chr250Data.Contains(">"))

{

if (Seq != string.Empty)

{

CHRData.Add(new KeyValuePair<string, string>(ID, Seq));

Seq = string.Empty;

}

ID = Chr250Data;

}

else

{

Seq += Chr250Data;

}

}

CHRData.Add(new KeyValuePair<string, string>(ID, Seq));

//Initialize dictionary to store unique gene ID's from BowTie output (BED format) file.

Dictionary<string, int> GeneData = new Dictionary<string, int>();

//Initialize dictionary to store Gene ID's and its gene length.

Dictionary<string, int> GeneDataLength = new Dictionary<string, int>();

//Variable used to store header of combined file.

string Header = "geneID" + "\t" + "readLen50" + "\t" + "readLen70" + "\t" + "readLen100" + "\n";

string Values\_Check = string.Empty;

string Values50 = string.Empty;

string Values70 = string.Empty;

string Values100 = string.Empty;

//Call MappabilityFunction meenthod to get the mappability percentages for respective GeneID's with read length as '50'

MappabilityFunction(@"E:\Bio Informatics\Assignment 7\BTout-BED-50", GeneData, GeneDataLength, CHRData, ref Values50, 50);

//Call MappabilityFunction meenthod to get the mappability percentages for respective GeneID's with read length as '70'

MappabilityFunction(@"E:\Bio Informatics\Assignment 7\BTout-BED-70", GeneData, GeneDataLength, CHRData, ref Values70, 70);

//Call MappabilityFunction meenthod to get the mappability percentages for respective GeneID's with read length as '100'

MappabilityFunction(@"E:\Bio Informatics\Assignment 7\BTout-BED-100", GeneData, GeneDataLength, CHRData, ref Values100, 100);

//Variable and arrays used to combile data for read lengths '50','70' & '100' and display them in one file.

int i=0;

string[] Arr50\_GeneId = Values50.Split(';');

string[] GeneIDs = new string[Arr50\_GeneId.Length-1];

string[] Arr50 = new string[Arr50\_GeneId.Length - 1];

for(int j=0;j < (Arr50\_GeneId.Length-1);j++)

{

GeneIDs[j] = Arr50\_GeneId[j].Split('$')[0].Split('.')[3];

Arr50[j] = Arr50\_GeneId[j].Split('$')[1];

}

string[] Arr70\_GeneId = Values70.Split(';');

string[] Arr70 = new string[Arr70\_GeneId.Length - 1];//Values70.Split(';');

for (int j = 0; j < (Arr70\_GeneId.Length-1); j++)

{

Arr70[j] = Arr70\_GeneId[j].Split('$')[1];

}

string[] Arr100\_GeneId = Values100.Split(';');

string[] Arr100 = new string[Arr100\_GeneId.Length - 1];

for (int j = 0; j < (Arr100\_GeneId.Length-1); j++)

{

Arr100[j] = Arr100\_GeneId[j].Split('$')[1];

}

//Loop through GeneId's for which mappability value has been calculated.

for(int k=0; k<(GeneIDs.Length);k++)

{

string GeneName\_Value = GeneIDs[k];

if(i<Arr50.Length)

Values\_Check += GeneName\_Value + "\t" + Arr50[i] + "\t" + Arr70[i] + "\t" + Arr100[i] + "\n";

i++;

}

//Write the GeneID and its mappability value for each read length into a file.

File.WriteAllText(@"E:\Bio Informatics\Assignment 7\mappability-output-250", Header + Values\_Check);

}

/// <summary>

/// In this method we will take one dictionary to populate GeneId's and mapped read count and another dictionary

/// to populate GeneID's and their length. Take a list CHRData with data from chr1-250.fa file.

/// Pass read length value and a string variable which will return the GeneID and their corresponding mappability value.

/// </summary>

/// <param name="FilePath"></param>

/// <param name="GeneData"></param>

/// <param name="GeneDataLength"></param>

/// <param name="CHRData"></param>

/// <param name="Values"></param>

/// <param name="readLength"></param>

public static void MappabilityFunction(string FilePath ,Dictionary<string, int> GeneData, Dictionary<string, int> GeneDataLength, List<KeyValuePair<string, string>> CHRData, ref string Values, int readLength)

{

//Clear the list as there might be values for previous read length Bowtie file.

GeneData.Clear();

//Loop through the Bowtie output (BED format) to get the unique GeneId's and there mapped read count

foreach (string GeneID in File.ReadAllLines(FilePath))

{

string[] BToutDetails = GeneID.Split('\t');

string[] GeneNameDetails = BToutDetails[3].Split('.');

//Build the gene name

string GeneName = ">" +GeneNameDetails[0] + "." + GeneNameDetails[1] + "." + GeneNameDetails[2] + "." + GeneNameDetails[3] + "." + GeneNameDetails[4];// GeneNameDetails[3];

//Check if there exists Gene Name in dictionary and add 1 to the value.

if (GeneData.ContainsKey(GeneName))

{

//Add 1 to existing value for particular GeneID. If Map start and end are in range of Gene ID.

if ((Convert.ToInt32(BToutDetails[1]) >= Convert.ToInt32(GeneNameDetails[1])) &&

(Convert.ToInt32(BToutDetails[2]) <= Convert.ToInt32(GeneNameDetails[2])))

{

GeneData[GeneName] = GeneData[GeneName] + 1;

}

}

else

{

//Add GeneID and 1 as initial value. If Map start and end are in range of Gene ID.

if ((Convert.ToInt32(BToutDetails[1]) >= Convert.ToInt32(GeneNameDetails[1])) &&

(Convert.ToInt32(BToutDetails[2]) <= Convert.ToInt32(GeneNameDetails[2])))

{

GeneData.Add(GeneName, 1);

}

}

}

//Clear the dictionary before we put some information into it.

GeneDataLength.Clear();

//Loop through CHRData which contains information of Chr1-250.fa file and populate the unique GeneID's and their length.

foreach (KeyValuePair<string, string> GeneLengthData in CHRData)

{

string[] GeneAnnotData\_Array = GeneLengthData.Key.Split('.');

string GeneName = GeneLengthData.Key;//GeneAnnotData\_Array[3];

int GeneLength = 0;

//If we have duplicate GeneID then we will consider only the last GeneID found and its corresponding length.

if (GeneDataLength.ContainsKey(GeneName))

{

//Replace new length with the existing Value for particular GeneID which is duplicate.

GeneDataLength[GeneName] = Convert.ToInt32(GeneAnnotData\_Array[2]) - Convert.ToInt32(GeneAnnotData\_Array[1]);

}

else

{

//Add GeneID and its length.

GeneLength = Convert.ToInt32(GeneAnnotData\_Array[2]) - Convert.ToInt32(GeneAnnotData\_Array[1]);

GeneDataLength.Add(GeneName, GeneLength);

}

}

//Linq query to get data present in disctionary.

var items = from pair in GeneData

select pair;

//Initialize KeyValuePair List for storing data from GeneData dictionary.

List<KeyValuePair<string, int>> med1 = new List<KeyValuePair<string, int>>(items);

//Linq query to get data present in disctionary.

var items1 = from pair in GeneDataLength

select pair;

//Initialize KeyValuePair List for storing data from GeneDataLength dictionary.

List<KeyValuePair<string, int>> med2 = new List<KeyValuePair<string, int>>(items1);

//Set read\_Length value.

int read\_length = readLength;

double FinalValue = 0.0;

bool present = false;

//Loop through the list which contains GeneID and its length values.

foreach (KeyValuePair<string, int> GeneLengthValue in med2)

{

//Loop through the list which contains GeneID and its mapped reads.

foreach (KeyValuePair<string, int> GeneDataValue in med1)

{

//Check if GeneID exists or not

if (GeneDataValue.Key == GeneLengthValue.Key)//&& GeneLengthValue.Key == "NR\_026818")

{

//Calculate the mappability value in percentile.

double totaltiles = 0.0;

//if gene length less than or equal to current read length then set total tiles as 1 other wise calculate.

if (GeneLengthValue.Value <= read\_length)

{

totaltiles = 1.0;

}

else

{

totaltiles = GeneLengthValue.Value - read\_length + 1;

}

FinalValue = ((Convert.ToDouble(GeneDataValue.Value) / totaltiles)) \* 100;

string vl = Math.Round(FinalValue, 2).ToString();

if (vl.IndexOf('.') != -1)

{

Values += GeneDataValue.Key + "$" + vl + ";";

}

else

{

Values += GeneDataValue.Key + "$" + vl + ".0" + ";";

}

present = true;

}

}

//Based on flag value above, if GeneID doesnt exists then we will set value as '0.0'

if(!present)

{

Values += GeneLengthValue.Key + "$" + "0.0;";

present = false;

}

else

{

present = false;

}

}

}

}

}

***OUTPUT:***

|  |  |  |  |
| --- | --- | --- | --- |
| geneID | readLen50 | readLen70 | readLen100 |
| NR\_046018 | 100.00 | 100.00 | 100.00 |
| NR\_026818 | 94.66 | 99.79 | 100.00 |
| NM\_001005484 | 100.00 | 100.00 | 100.00 |
| NR\_039983 | 0.57 | 1.87 | 6.46 |
| NR\_028327 | 0.00 | 0.54 | 2.35 |
| NM\_001005221 | 0.00 | 0.00 | 0.00 |
| NM\_001005221 | 0.00 | 0.00 | 0.00 |
| NR\_028327 | 0.00 | 0.00 | 0.00 |
| NR\_024321 | 100.00 | 100.00 | 100.00 |
| NR\_027055 | 74.10 | 85.76 | 91.89 |
| NR\_026874 | 100.00 | 100.00 | 100.00 |
| NM\_198317 | 95.93 | 96.31 | 96.88 |
| NM\_001160184 | 96.87 | 98.11 | 98.47 |
| NR\_027693 | 81.67 | 86.93 | 90.36 |
| NM\_021170 | 100.00 | 100.00 | 100.00 |
| NM\_001142467 | 100.00 | 100.00 | 100.00 |
| NM\_005101 | 100.00 | 100.00 | 100.00 |
| NM\_001205252 | 100.00 | 100.00 | 100.00 |
| NR\_038869 | 68.18 | 71.99 | 77.99 |
| NR\_029639 | 100.00 | 100.00 | 100.00 |
| NR\_029834 | 100.00 | 100.00 | 100.00 |
| NR\_029957 | 100.00 | 100.00 | 100.00 |
| NM\_153254 | 95.39 | 95.70 | 96.18 |
| NM\_004195 | 100.00 | 100.00 | 100.00 |
| NM\_003327 | 100.00 | 100.00 | 100.00 |
| NM\_080605 | 100.00 | 100.00 | 100.00 |
| NM\_001014980 | 98.11 | 100.00 | 100.00 |
| NM\_153339 | 97.55 | 99.07 | 100.00 |
| NM\_001029885 | 100.00 | 100.00 | 100.00 |
| NM\_152228 | 100.00 | 100.00 | 100.00 |
| NM\_032348 | 97.45 | 99.22 | 100.00 |
| NM\_001127230 | 100.00 | 100.00 | 100.00 |
| NM\_001127229 | 100.00 | 100.00 | 100.00 |
| NM\_017900 | 100.00 | 100.00 | 100.00 |
| NM\_001039577 | 80.03 | 93.31 | 99.07 |
| NR\_015434 | 100.00 | 100.00 | 100.00 |
| NM\_017971 | 80.29 | 91.91 | 98.93 |
| NM\_001145210 | 91.06 | 96.24 | 100.00 |
| NM\_001146685 | 98.88 | 100.00 | 100.00 |
| NM\_022834 | 100.00 | 100.00 | 100.00 |
| NM\_001114748 | 99.60 | 100.00 | 100.00 |
| NM\_001242659 | 100.00 | 100.00 | 100.00 |
| NR\_002946 | 2.03 | 3.17 | 4.92 |
| NM\_006983 | 12.88 | 13.82 | 15.26 |
| NR\_002946 | 2.09 | 3.22 | 4.97 |
| NM\_001198995 | 84.52 | 86.28 | 88.16 |
| NM\_138705 | 100.00 | 100.00 | 100.00 |
| NM\_178545 | 100.00 | 100.00 | 100.00 |
| NM\_182533 | 95.06 | 98.18 | 99.88 |
| NR\_024489 | 100.00 | 100.00 | 100.00 |
| NM\_153818 | 96.71 | 99.13 | 100.00 |
| NM\_001010926 | 100.00 | 100.00 | 100.00 |
| NR\_026927 | 24.85 | 32.59 | 40.71 |
| NR\_037844 | 100.00 | 100.00 | 100.00 |
| NM\_003820 | 99.54 | 100.00 | 100.00 |
| NR\_036638 | 100.00 | 100.00 | 100.00 |
| NM\_001195738 | 100.00 | 100.00 | 100.00 |
| NM\_080431 | 100.00 | 100.00 | 100.00 |
| NR\_015440 | 100.00 | 100.00 | 100.00 |
| NR\_024371 | 100.00 | 100.00 | 100.00 |
| NR\_036215 | 100.00 | 100.00 | 100.00 |
| NR\_030277 | 100.00 | 100.00 | 100.00 |
| NM\_182752 | 93.73 | 98.09 | 100.00 |
| NM\_001163724 | 100.00 | 100.00 | 100.00 |
| NR\_039612 | 100.00 | 100.00 | 100.00 |
| NR\_039838 | 100.00 | 100.00 | 100.00 |
| NM\_001024598 | 100.00 | 100.00 | 100.00 |
| NM\_019089 | 93.96 | 96.60 | 100.00 |
| NR\_036218 | 0.00 | 0.00 | 0.00 |
| NM\_003790 | 94.13 | 96.76 | 100.00 |
| NM\_001039664 | 100.00 | 100.00 | 100.00 |
| NM\_005341 | 90.35 | 96.99 | 99.96 |
| NM\_001195753 | 84.55 | 96.18 | 99.56 |
| NM\_001195752 | 84.03 | 96.05 | 99.54 |
| NM\_006786 | 87.28 | 96.14 | 99.78 |
| NM\_021995 | 88.25 | 96.44 | 99.79 |
| NR\_038351 | 100.00 | 100.00 | 100.00 |
| NR\_029610 | 100.00 | 100.00 | 100.00 |
| NR\_027045 | 100.00 | 100.00 | 100.00 |
| NM\_001302 | 100.00 | 100.00 | 100.00 |
| NM\_139208 | 94.69 | 99.92 | 100.00 |
| NM\_003132 | 86.24 | 95.68 | 99.12 |
| NM\_021146 | 96.34 | 98.72 | 100.00 |
| NM\_012168 | 98.04 | 99.45 | 100.00 |
| NM\_183412 | 83.52 | 93.37 | 99.03 |
| NM\_033182 | 82.58 | 92.99 | 98.97 |
| NM\_006341 | 85.06 | 93.16 | 99.59 |
| NR\_037806 | 89.06 | 93.80 | 98.19 |
| NM\_006172 | 76.64 | 84.69 | 95.65 |
| NM\_002521 | 100.00 | 100.00 | 100.00 |
| NM\_138346 | 93.09 | 97.11 | 99.75 |
| NR\_039775 | 100.00 | 100.00 | 100.00 |
| NR\_003025 | 100.00 | 100.00 | 100.00 |
| NM\_001080830 | 82.13 | 91.82 | 97.24 |
| NM\_023013 | 15.36 | 24.33 | 37.19 |
| NM\_001146344 | 9.84 | 17.58 | 32.89 |
| NM\_001146181 | 13.01 | 18.65 | 29.01 |
| NM\_001013631 | 17.26 | 27.22 | 39.46 |
| NM\_023014 | 22.74 | 35.04 | 53.38 |
| NM\_001009611 | 4.72 | 8.17 | 15.69 |
| NM\_001039361 | 0.00 | 0.00 | 0.00 |
| NM\_001012277 | 0.00 | 0.00 | 0.00 |
| NM\_001012276 | 0.00 | 0.00 | 0.00 |
| NM\_001010889 | 0.36 | 0.59 | 0.92 |
| NM\_001100631 | 0.00 | 0.00 | 0.00 |
| NM\_001013407 | 0.11 | 0.33 | 0.66 |
| NM\_001136561 | 16.16 | 27.35 | 41.80 |
| NM\_001100631 | 0.00 | 0.00 | 0.00 |
| NM\_001013692 | 0.00 | 0.00 | 0.12 |
| NM\_001010889 | 0.00 | 0.00 | 0.00 |
| NM\_001013407 | 0.00 | 0.00 | 0.00 |
| NM\_001012276 | 0.00 | 0.00 | 0.00 |
| NM\_001098376 | 0.00 | 0.00 | 0.00 |
| NM\_001010890 | 0.00 | 0.00 | 0.00 |
| NM\_001024661 | 0.00 | 0.00 | 0.00 |
| NM\_001099850 | 0.00 | 0.00 | 0.00 |
| NM\_001045480 | 1.05 | 1.06 | 1.07 |
| NM\_001100114 | 0.53 | 1.36 | 2.68 |
| NM\_001012277 | 0.00 | 0.00 | 0.00 |
| NM\_001012276 | 0.00 | 0.00 | 0.00 |
| NM\_001098376 | 0.00 | 0.00 | 0.00 |
| NM\_001010890 | 0.00 | 0.00 | 0.00 |
| NM\_001099854 | 0.00 | 0.00 | 0.00 |
| NM\_001099850 | 0.00 | 0.00 | 0.00 |
| NM\_001099851 | 0.07 | 0.07 | 0.07 |
| NM\_001100114 | 0.52 | 1.18 | 3.08 |
| NM\_007272 | 91.11 | 97.83 | 100.00 |
| NM\_006511 | 100.00 | 100.00 | 100.00 |
| NM\_207348 | 89.16 | 97.60 | 99.70 |
| NM\_001013641 | 85.54 | 96.19 | 100.00 |
| NM\_001089591 | 35.99 | 52.67 | 75.85 |
| NM\_178840 | 87.40 | 95.51 | 99.79 |
| NM\_014424 | 100.00 | 100.00 | 100.00 |
| NM\_001165945 | 66.33 | 74.81 | 82.16 |
| NM\_030907 | 88.97 | 94.62 | 99.42 |
| NR\_037446 | 0.00 | 0.00 | 0.00 |
| NR\_027504 | 8.69 | 15.40 | 23.86 |
| NR\_037446 | 0.00 | 0.00 | 0.00 |
| NR\_002729 | 7.17 | 13.15 | 21.68 |
| NR\_037446 | 50.00 | 100.00 | 100.00 |
| NM\_001135248 | 90.99 | 91.42 | 91.87 |
| NM\_001135247 | 92.15 | 92.53 | 92.93 |
| NM\_152375 | 98.40 | 99.64 | 100.00 |
| NR\_039844 | 100.00 | 100.00 | 100.00 |
| NM\_016183 | 93.25 | 98.69 | 100.00 |
| NR\_040288 | 56.35 | 69.54 | 80.10 |
| NM\_012067 | 50.41 | 64.18 | 75.15 |
| NR\_037847 | 94.51 | 99.05 | 100.00 |
| NM\_003689 | 85.46 | 92.26 | 97.54 |
| NM\_019062 | 100.00 | 100.00 | 100.00 |
| NM\_014589 | 100.00 | 100.00 | 100.00 |
| NM\_001161729 | 100.00 | 100.00 | 100.00 |
| NM\_001161728 | 100.00 | 100.00 | 100.00 |
| NM\_012400 | 95.92 | 99.07 | 100.00 |
| NM\_152376 | 99.64 | 100.00 | 100.00 |
| NM\_018584 | 99.95 | 100.00 | 100.00 |
| NM\_024544 | 91.43 | 97.61 | 100.00 |
| NM\_207334 | 100.00 | 100.00 | 100.00 |
| NM\_005216 | 94.30 | 97.74 | 100.00 |
| NR\_038404 | 96.43 | 98.76 | 100.00 |
| NR\_023918 | 85.12 | 93.96 | 99.76 |
| NR\_023919 | 84.23 | 93.60 | 99.75 |
| NM\_015991 | 100.00 | 100.00 | 100.00 |
| NM\_172369 | 100.00 | 100.00 | 100.00 |
| NM\_000491 | 98.26 | 99.78 | 100.00 |
| NR\_039832 | 100.00 | 100.00 | 100.00 |
| NR\_036214 | 100.00 | 100.00 | 100.00 |
| NM\_001242521 | 80.48 | 93.67 | 99.63 |
| NR\_036057 | 100.00 | 100.00 | 100.00 |
| NM\_000864 | 94.90 | 98.99 | 100.00 |
| NM\_030634 | 90.55 | 96.56 | 99.71 |
| NR\_033690 | 100.00 | 100.00 | 100.00 |
| NR\_033691 | 100.00 | 100.00 | 100.00 |
| NM\_002167 | 100.00 | 100.00 | 100.00 |
| NM\_000975 | 98.89 | 100.00 | 100.00 |
| NM\_020362 | 82.26 | 92.53 | 98.56 |
| NM\_007260 | 100.00 | 100.00 | 100.00 |
| NM\_001127621 | 95.59 | 99.03 | 100.00 |
| NM\_000403 | 96.47 | 99.22 | 100.00 |
| NM\_001008216 | 96.65 | 99.26 | 100.00 |
| NR\_039615 | 100.00 | 100.00 | 100.00 |
| NM\_017761 | 41.75 | 46.12 | 52.25 |
| NR\_045409 | 88.94 | 94.63 | 98.36 |
| NM\_020317 | 96.09 | 99.52 | 100.00 |
| NR\_024498 | 92.54 | 93.78 | 98.03 |
| NM\_178422 | 83.98 | 92.25 | 98.61 |
| NM\_203401 | 97.04 | 99.01 | 99.93 |
| NR\_037481 | 100.00 | 100.00 | 100.00 |
| NM\_005563 | 97.21 | 99.07 | 99.94 |
| NM\_203399 | 97.36 | 99.12 | 99.94 |
| NM\_032513 | 96.17 | 98.37 | 100.00 |
| NM\_024869 | 94.63 | 99.10 | 100.00 |
| NM\_015871 | 100.00 | 100.00 | 100.00 |
| NM\_031286 | 100.00 | 100.00 | 100.00 |
| NM\_001803 | 80.70 | 89.63 | 100.00 |
| NM\_005517 | 82.29 | 86.48 | 91.07 |
| NR\_031740 | 100.00 | 100.00 | 100.00 |
| NM\_006142 | 100.00 | 100.00 | 100.00 |
| NM\_022078 | 84.19 | 93.24 | 99.38 |
| NM\_021969 | 95.28 | 98.53 | 100.00 |
| NM\_001013642 | 94.22 | 98.96 | 99.93 |
| NM\_052943 | 96.18 | 98.67 | 100.00 |
| NR\_033926 | 82.29 | 90.67 | 99.06 |
| NM\_173452 | 83.55 | 92.97 | 99.77 |
| NM\_207397 | 100.00 | 100.00 | 100.00 |
| NM\_005281 | 100.00 | 100.00 | 100.00 |
| NM\_002038 | 89.12 | 95.97 | 99.85 |
| NR\_002997 | 100.00 | 100.00 | 100.00 |
| NM\_018053 | 96.51 | 99.48 | 100.00 |
| NR\_027268 | 100.00 | 100.00 | 100.00 |
| NM\_178190 | 100.00 | 100.00 | 100.00 |
| NM\_017638 | 79.15 | 90.79 | 99.08 |
| NR\_036473 | 90.49 | 98.18 | 100.00 |
| NR\_003077 | 100.00 | 100.00 | 100.00 |
| NR\_002987 | 100.00 | 100.00 | 100.00 |
| NR\_002976 | 93.98 | 100.00 | 100.00 |
| NR\_003035 | 100.00 | 100.00 | 100.00 |
| NR\_024127 | 93.30 | 99.75 | 100.00 |
| NM\_152304 | 83.46 | 90.73 | 97.32 |
| NM\_001193532 | 81.06 | 89.37 | 96.92 |
| NR\_004407 | 100.00 | 100.00 | 100.00 |
| NM\_001171868 | 100.00 | 100.00 | 100.00 |
| NM\_001003682 | 100.00 | 100.00 | 100.00 |
| NR\_034182 | 100.00 | 100.00 | 100.00 |
| NR\_039616 | 100.00 | 100.00 | 100.00 |
| NR\_033295 | 0.00 | 0.00 | 0.00 |
| NR\_004054 | 0.00 | 0.00 | 0.00 |
| NR\_003066 | 100.00 | 100.00 | 100.00 |
| NR\_002762 | 100.00 | 100.00 | 100.00 |
| NM\_004102 | 97.72 | 99.77 | 100.00 |
| NR\_034112 | 100.00 | 100.00 | 100.00 |
| NR\_027085 | 82.00 | 94.95 | 99.81 |
| NM\_001204415 | 89.78 | 93.57 | 96.25 |
| NM\_001525 | 98.01 | 99.16 | 100.00 |
| NR\_036216 | 100.00 | 100.00 | 100.00 |
| NM\_024296 | 94.20 | 99.47 | 100.00 |
| NM\_001160042 | 95.07 | 98.56 | 99.90 |
| NM\_001099434 | 79.15 | 91.71 | 99.10 |
| NM\_019118 | 85.87 | 94.92 | 99.70 |
| NM\_003757 | 75.69 | 90.81 | 99.38 |
| NM\_032648 | 100.00 | 100.00 | 100.00 |
| NM\_023009 | 100.00 | 100.00 | 100.00 |
| NM\_001167676 | 100.00 | 100.00 | 100.00 |
| NM\_052841 | 100.00 | 100.00 | 100.00 |
| NM\_001198973 | 96.52 | 98.38 | 99.57 |
| NM\_001171940 | 93.56 | 96.18 | 99.19 |
| NM\_002143 | 100.00 | 100.00 | 100.00 |
| NM\_033504 | 99.70 | 100.00 | 100.00 |
| NR\_037400 | 100.00 | 100.00 | 100.00 |
| NM\_145205 | 99.41 | 100.00 | 100.00 |
|  |  |  |  |

Steps to create graphs using **R:**

* Type **R**  it will open R console then execute below mentioned commands to create Histogram and Scatter plot graphs for the data that is provided.

mukhasir@Mukhasir /cygdrive/e/Bio Informatics/Assignment 7

$ R

R version 3.1.3 (2015-03-09) -- "Smooth Sidewalk"

Copyright (C) 2015 The R Foundation for Statistical Computing

Platform: x86\_64-unknown-cygwin (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

> x<-read.table("mappability-output-250",header=TRUE,sep="")

**//Histograms**

> pdf("Hist\_50.pdf")

> hist(x[,2],breaks=seq(0,100,1),main="Histogram for read Len 50",xlab="len50")

> dev.off()

null device

1

>

> pdf("Hist\_70.pdf")

> hist(x[,3],breaks=seq(0,100,1),main="Histogram for read Len 70",xlab="len70")

> dev.off()

null device

1

>

> pdf("Hist\_100.pdf")

> hist(x[,4],breaks=seq(0,100,1),main="Histogram for read Len 100",xlab="len100")

> dev.off()

null device

1

**//Scatter plots**

> pdf("ScatterPlot\_50\_70.pdf")

> plot(x[,2],x[,3],main="Scatter plot for read Len 50 vs. read Len 70",xlab="len50",ylab="len70")

> dev.off()

null device

1

>

> pdf("ScatterPlot\_50\_100.pdf")

> plot(x[,2],x[,4],main="Scatter plot for read Len 50 vs. read Len 100",xlab="len50",ylab="len100")

> dev.off()

null device

1

>

> pdf("ScatterPlot\_70\_100.pdf")

> plot(x[,3],x[,4],main="Scatter plot for read Len 70 vs. read Len 100",xlab="len70",ylab="len100")

> dev.off()

null device

1

***Mappability Analysis:***

1. Histogram for Read Len 50: (fine grained)

This graphs denotes relation between Number of records[Frequency] (y-axis) present within certain map percentages (x-axis) for read length "50". From the graph we can say that there are approximately 35 records with percentage 0 till 1 and we have 100 records with 100% mappability.

1. Histogram for Read Len 70: (fine grained)

This graphs denotes relation between Number of records[Frequency] (y-axis) present within certain map percentages (x-axis) for read length "70". From the graph we can say that there are approximately 120+ records with percentage 99 till 100 mappability.

1. Histogram for Read Len 100: (fine grained)

This graphs denotes relation between Number of records[Frequency] (y-axis) present within certain map percentages (x-axis) for read length "100". From the graph we can say that there are approximately 150+ records with percentage 99 till 100 mappability.

1. Scatter plot for Read Len 50 vs. Read Len 70:

This graphs denotes relation of ReadLen 50 vs. ReadLen 70, for a given record the percentile for readlen50 is marked on x-axis and percentile for readlen70 is marked on y-axis. Location where two intersect is plotted on the graph. From this plot we can say that a record has 50% mappability for readlen 50 (x-axis) and on other hand it has 100% mappability for readlen 70 (y-axis). There are more records accumulated from 80 to 100 mappability percent.

1. Scatter plot for Read Len 50 vs. Read Len 100:

This graphs denotes relation of ReadLen 50 vs. ReadLen 100, for a given record the percentile for readlen50 is marked on x-axis and percentile for readlen100 is marked on y-axis. Location where two intersect is plotted on the graph. From this plot we can say that a record has 56% mappability for readlen 50 (x-axis) and on other hand it has 80% mappability for readlen 100 (y-axis). There are records ranging from 80 to 100 for readlen50 which have 100% mappability for readlen100.

1. Scatter plot for Read Len 70 vs. Read Len 100:

This graphs denotes relation of ReadLen 70 vs. ReadLen 100, for a given record the percentile for readlen70 is marked on x-axis and percentile for readlen100 is marked on y-axis. Location where two intersect is plotted on the graph. From this plot we can say that a gene ID has 52%(approx.) mappability for readlen 70 (x-axis) and on other hand it says 75%(approx.) mappability for readlen 100 (y-axis). There are records ranging from 90 to 100 for readlen70 which have 100% mappability for readlen100.