**/\*Program description:** RNA-Seq data analysis pipeline, we manipulate process1 and process3, i.e., \*processing without considering junction region mappings. In practice, junction mapping covers about \*6% of total mappings.

**\*Input :** HG19 RefSeq exon annotation, chr1.fa, collapsed\_exon\_annot.fa

**\*Task1: Exon region mask complement**

\*1. Prepare HG19 chr1 exon annotation:

\*2. Make a collapsed exon annotation: From the resulting annotation from Step1, collapse all \*overlapped regions and make a collapsed annotation with each collapsed exon name “x” and strand \*“+”.(program for this.)

\*3 . Mask-complement the genome: Using the collapsed exon annotation (from Step2), mask all non-\*exon regions of HG19 chr1 with ‘N’s.(program for this)

**\*Task 2: read mapping and read count**

\*1. Using Bowtie, map a reads file (fastq format) onto the genome

\*2. Using the original exon annotation file, count mapped reads on each exon.

\*3. Convert the exon-level read counting to gene-level counting

\*/

package assignment7;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Exon\_File\_Create {

public static void main(String args[]){

String fileName1 = "C:/Users/sruthi/Desktop/hg19-refseq-exon-annot-chr1.fa";

String fileName = "C:/Users/sruthi/Desktop/collapsed\_exon\_annot.fa";

try {

// FileReader reads text files in the default encoding.

FileReader fileReader1 = new FileReader(fileName1);

FileWriter fileWriter = new FileWriter(fileName);

// Always wrap FileReader in BufferedReader.

BufferedReader bufferedReader1 = new BufferedReader(fileReader1);

BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

String line;

//reads the content of chr1.fa line by line

while((line = bufferedReader1.readLine()) != null ) {

String[] parts= line.split("\\s");

String[] exon\_Start = new String[1000];

String[] exon\_End= new String[1000];

int i=0;

exon\_Start[i] = parts[1];

int j=0;

exon\_End[j]=parts[2];

System.out.println(exon\_Start[i]+" "+exon\_End[j]);

bufferedReader1.readLine();

String[] parts2 = line.split("\\s");

String[] exon\_Start1 = new String[1000];

String[] exon\_End1 = new String[1000];

int i1=0;

exon\_Start[i1] = parts2[1];

int j1=0;

exon\_End[j1] = parts2[2];

if(Integer.parseInt(exon\_Start1[i1])>Integer.parseInt(exon\_Start[i])) {

exon\_Start[i1]= exon\_Start[i];

}

fileWriter.write(exon\_Start[i]);

fileWriter.write(exon\_End[i]);

fileWriter.write("x");

fileWriter.write("0");

i++;

j++;

i1++;

j1++;

}

// Always close files.

bufferedReader1.close();

} catch (FileNotFoundException ex1) {

System.out.println("Unable to open file '" + fileName1 + "'");

} catch (IOException ex) {

System.out.println("Error reading file '" + fileName1 + "'");

// Or we could just do this:

// ex.printStackTrace();

} }

}

//Pgrm2:

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Mask {

public static void main(String[] args) {

// TODO Auto-generated method stub

String fileName = "C:/Users/sruthi/Desktop/collapsed\_exon\_annot.fa";

String fileName2= "C:/Users/sruthi/Desktop/chr1.fa";

try {

// FileReader reads text files in the default encoding.

FileReader fileReader1 = new FileReader(fileName);

FileReader fileReader2 = new FileReader(fileName2);

// Always wrap FileReader in BufferedReader.

BufferedReader bufferedReader1 = new BufferedReader(fileReader1);

BufferedReader bufferedReader2 = new BufferedReader(fileReader2);

String line;

String line1;

String arr="";

String arr1;

String new\_arr;

while(line = bufferedReader.readLine()) != null){

arr= arr+line;

}

while((line = bufferedReader1.readLine()) != null ) {

line.replace("A","N");

line.replace("C","N");

line.replace("G","N");

line.replace("T","N");

line.replace("a","N");

line.replace("c","N");

line.replace("g","N");

line.replace("t","N");

}

while((line1 = bufferedReader2.readLine()) != null ) {

String[] parts= line.split("\\s");

String[] exon\_Start = new String[1000];

String[] exon\_End= new String[1000];

}

while((line = bufferedReader1.readLine()) != null ){

arr.replaceAll(arr1, new\_arr);

}

// Always close files.

bufferedReader1.close();

} catch (FileNotFoundException ex1) {

System.out.println("Unable to open file '" + fileName + "'");

} catch (IOException ex) {

System.out.println("Error reading file '" + fileName + "'");

// Or we could just do this:

// ex.printStackTrace();

} }

}

291T BIO INFORMATICS COMPUTING

Assignment-7

NAME : SRUTHI YELLAPATI

ID: 109153960