**291T BIOINFORMATICS COMPUTING**

(**Assignment 1**)

**NAME:YELLAPATI SRUTHI**

**ID:109153960**

**Input**:hg19\_chr1.fa and hg19\_chr1\_refseq\_exon\_annotation

**Output**: a fasta file having 5 extracted gene sequences in which intron regions are masked with N's.

**Procedure**:

1. First extract the given gene IDs from hg19 file, and store them into different file. –Using

**awk** command and file name – gene\_exon.

2. By using C program, take chr1.fa file, read whole the file, then read the data into string array.

By considering gene id, store its start point, end point, and strand sign into separate arrays.

**cut** command is used.

4.Now look for the exon positions ,let the exons remains the same and regions other than exon mask them as "**N**"

5. write C program for reverse complement for the gene sequence having negative strand.

For -ve strand:

do (A or a ->T;

T or t------->A;

G or g----->C;

C or c----->G)

6.Read the resultant output to another result.fa file.

**Source Code:**

**awk commands used:**

At command prompt performed a function called awk to extract the given gene\_ids :

**$ awk ‘{if($4 == "NM\_032291") print $0 ;} hg19 > x1**

{Function – to extract the given id –NM\_032291 from the file using fourth column and store the

output into x1 file.}

**$ awk ‘{if($4 == "NM\_024066") print $0 ;}' hg19 > x2**

**$ awk ‘{if($4 == "NM\_001199739") print $0;}' hg19 > x3**

**$ awk ‘{if($4 == "NM\_003689”) print $0;}' hg19 > x4**

**$ awk ‘{if($4=="NM\_001201547”)print $0;}' hg19 > x5**

**$ cat x1 x2 x3 x4 x5 > 5gene\_exon**

script to extract only the starting number from gene\_exon and store in other file - start.:

**cut -f $2 gene\_exon > start**

script to extract only the ending number from gene\_exon and store in other file - end:

**cut -f $4 gene\_exon > end**

script to extract only the ending number from gene\_exon and store in other file - strand:

**cut -f $6 gene\_exon > strand**

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#include <string.h>

#define MAX 50/\* for defining and limiting the parameters of gene sequence \*/

#define SIZE 30 /\*for defining and limiting the parameters of gene sequence\*/

int main()

{

int n,i,x,y,k,a;

char start[MAX][SIZE],end[MAX][SIZE],strand[MAX][SIZE];

char \*text,\*st;

FILE \*f\_chr1 ;

FILE \*f\_start,\*f\_end,\*f\_strand;// for storing the start, end and strand file.

f\_chr1 = fopen( "chr.fa" , "r+");

fseek(f\_chr1, 0, SEEK\_END);

int chr1\_file\_size = ftell(f\_chr1);

text=(char\*)malloc(chr1\_file\_size+1);

st=(char\*)malloc(chr1\_file\_size+1);

text[chr1\_file\_size]='\0';

f(f\_chr1 != NULL)

{

fread( text, sizeof(char), chr1\_file\_size,f\_chr1);/\* reading the entire data into array \*/

}

fclose(f\_chr1);

f\_strt= fopen("start", "r");

for(n=0;(n<MAX)&& fgets(start,30, f\_strt);n++); /\* reading the starting numbers into the array\*/

f\_end= fopen("end", "r");

for(n=0;(n<MAX)&& fgets(start,30, f\_end);n++);/\* reading the ending numbers into the array \*/

f\_strand= fopen("strand", "r");

for(n=0;(n<MAX)&& fgets(start,30, f\_strand);n++);/\* reading the strands into the array\*/

while(n!=MAX)

{

int k=0;

x=atoi(\*(f\_start+a)); // considering starting number

y=atoi(\*(f\_end+a)); // considering ending number

a++;

for(k=0;k<chr1\_file\_size;k++)

{

if((k>x && k<y) ||( k==x ||k==y)) // exon region

{

(\*(text+k));

}

else{

(\*(text+k))= "N"; // mask intron region as "N"

}

}

n++;

}

do

{

int k=0;

x=atoi(\*(f\_start+a));// starting number

y=atoi(\*(f\_end+a)); //ending number

printf(">%d.%d.%d.%c",chr1,x,y,\*( f\_strand));

if(f\_strand[a] == “-“) /\*looking for the strand if it is + or - , if negative do reverse complement\*/

{

for(k=0;k<chr1\_file\_size;k++)

{

if((k>x && k<y) ||( k==x ||k==y))

{

(\*(st+(i++)))=(\*(text+k));

}

}

int length = 0;

while(\*st)

{

length++;

\* st++;

}

i=0;

while (i < length)

{

temp = \*(st+i);

\*(st+i) = \*(st+j);

\*(st+j) = temp;

i++;

j--;

}

for(i=0;i<length;i++) // reverse complement logic (A or a ->T; T or t--->A; G or g----->C; C or c----->G

{

switch (\*(st+i))

{

case 'A':

c = 'T';

break;

case 'T':

c = 'A';

break;

case 'G':

c = 'C';

break;

case 'C':

c = 'G';

break;

case 'a':

c = 't';

break;

case 't':

c = 'a';

break;

case 'g':

c = 'c';

break;

case 'c':

c = 'g';

break;

}

}

i=0;

for(k=0;k<chr1\_file\_size;k++)

{

if((k>x && k<y) ||( k==x ||k==y))

{

(\*(text+k))=(\*(st+(i++)));

}

}

}

a++;

n++;

}while(n!=MAX);

FILE \*res = fopen("result.fa", "w+"); /\*storing the resultant file into the other file result.fa\*/

For(i= atoi(\*(f\_start+0)); i<=atoi(\*(f\_end+MAX));i++)

{

fputc(\*(text+i), res);

fclose(res);

}

return 0;

}