**Bio Informatics Computing**

**(CSCI-291T)**

Assignment – 1

Mukhasir Shah Syed

109137268

*Question:*

From HG19 chr1.fa, extract following 5 RefSeq gene sequences and make a fasta file. In each extracted gene sequence, mark exon regions in upper\_case letters and intron regions in lower\_case letters.

*Solution:*

Language used: ***C#***

Files mentioned below were used in the code:

1. *Chr1.fa*

This file has genome data, which contains Chromosome Name and bases.

1. *HG19-refseq-gene-annot*

This file has gene annotation data, which contains Gene IDs, starting base, ending base and gene strand (“+” or “-”).

1. *hg19-refseq-exon-annot-chr1*

This file has exon annotation data, which contains exons starting position, ending position, and Gene ID and strand.

To 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. When Executable file(application) is run it will check App.config file and get details if any then execute the code and place the output in a file with “.fa” extension as mentioned.

Program.cs:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Date: Feb 02, 2015. Author: Mukhasir Shah Syed

\* Following code is written to get the bases of the specific geneID and place that string

into a file with '.fa' extension.

\* Get details of file locations to read and which Gene ID are required from App.config file.

\* Read 'chr1.fa' file to get genome data using File.ReadAllLines() and place bases into string.

\* After that get Gene Details from 'HG19-refseq-gene-annot' file using File.ReadLines().

\* Check if the required Gene Id is present or not. If no then no operation is required.

If yes then proceed further and get details of exons from "hg19-refseq-exon-annot-chr1"

file using File.ReadAllLines().

\* Based on exons start and end position get substring and convert them into UpperCase and

concat that into a string. Similarly to get introns, code will check end position of previous

exon and starting position of next exon and convert then into LowerCase. This process

will be continued until all the exon and introns are concatenated into a string.

\* If gene strand is '+' then concatenate the above string with respective GeneID and strand value.

If gene strand is '-' then Complement and Reverse the above string and then concatenate

above string with respective GeneID and strand value.

\* When all required data is collected then put that into a file with '.fa' extension using

File.WriteAllText() method.

\* To run this code goto bin\Debug folder of this Console Application folder and

run GeneData.exe(application) file.If any changes in locations or Gene IDs then

modify the values in GeneData.exe.config file as code has been written in generic way.

Location: E:\Bio Informatics\Assignment 1\GeneData\GeneData\bin\Debug

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.Configuration;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GeneData

{

/// <summary>

///

/// </summary>

class Program

{

static void Main(string[] args)

{

string label = string.Empty;

string bases = string.Empty;

//Get Details of locations and Gene Ids from App.cofig file

string ChromosomeData = ConfigurationManager.AppSettings["ChromosomeData"];

string GeneData = ConfigurationManager.AppSettings["GeneLocationData"];

string ExonData = ConfigurationManager.AppSettings["ExonLocationData"];

string GeneIds = ConfigurationManager.AppSettings["GeneIDs"];

string FinalLocation = ConfigurationManager.AppSettings["FinalLocation"];

//Get Label and Bases details from Ch1.fa file

foreach (string line in File.ReadLines(@ChromosomeData))

{

if (line.Contains(">"))

{

label = line;

}

break;

}

//Read bases from file and make it as string from array

string[] lines = File.ReadAllLines(@ChromosomeData);

List<string> list = new List<string>(lines);

list.RemoveAt(0);

bases = string.Join("", list.ToArray());

//Variables used to store bases data for exon(s) and intron(s)

string subBases = string.Empty;

string BasesValue = string.Empty;

string BasesLabel = string.Empty;

//Get GeneID's on which operations have to be done.

string[] GeneIDValue = null;

if(GeneIds != null)

{

GeneIDValue = GeneIds.Split(';');

}

//Get Annotaion details - Start , Last , Strand and Gene Name from HG19-refseq-gene-annot

foreach (string GeneAnnotDetails in File.ReadLines(@GeneData))

{

subBases = string.Empty;

string[] geneAnnotDetailValues = GeneAnnotDetails.Split('\t');

//Check if the wanted Gene ID is present or not

bool GenePresent = false;

for (int i = 0; i < GeneIDValue.Length;i++)

{

if(geneAnnotDetailValues[3] == GeneIDValue[i])

{

GenePresent = true;

}

}

//Check condition to get exon details for the mentioned genes

if (GenePresent)

{

int geneStartBase = Convert.ToInt32(geneAnnotDetailValues[1]);

int geneEndBase = Convert.ToInt32(geneAnnotDetailValues[2]);

string geneStrand = geneAnnotDetailValues[5];

BasesLabel = ">" + geneAnnotDetailValues[3];

int baseStart = geneStartBase;

//Get Annotation details for mentioned genes

foreach (string ExonAnnotDetails in File.ReadAllLines(@ExonData))

{

string[] exonAnnotDetailValues = ExonAnnotDetails.Split('\t');

int exonStart = Convert.ToInt32(exonAnnotDetailValues[1]);

int exonEnd = Convert.ToInt32(exonAnnotDetailValues[2]);

string exonStrand = exonAnnotDetailValues[5];

bool exit = false;

//Check if required gene id is present or not and based on start and end positions of exons, code will

//fetch data of exons and introns and place it into a string.

//if the gene id has '-' strand then bases will to be complimented and all gene bases string will be reversed.

if (exonAnnotDetailValues[3].Contains(geneAnnotDetailValues[3]))

{

if (baseStart == exonStart)

{

int length = (Convert.ToInt32(exonEnd)) - (Convert.ToInt32(exonStart));

if (geneStrand == "+")

{

subBases += bases.Substring(Convert.ToInt32(exonStart), length).ToUpper();

}

else if (geneStrand == "-")

{

subBases += ComplementBases(bases.Substring(Convert.ToInt32(exonStart), length).ToUpper());

}

baseStart = Convert.ToInt32(exonEnd);

exit = true;

}

if(!exit)

{

if (baseStart != exonStart)

{

int length = (Convert.ToInt32(exonStart)) - Convert.ToInt32(baseStart);

if (geneStrand == "+")

{

subBases += bases.Substring(Convert.ToInt32(baseStart), length).ToLower();

}

else if (geneStrand == "-")

{

subBases += ComplementBases(bases.Substring(Convert.ToInt32(baseStart), length).ToLower());

}

baseStart = Convert.ToInt32(exonStart);

}

if (baseStart == exonStart)

{

int length = (Convert.ToInt32(exonEnd)) - Convert.ToInt32(exonStart);

if (geneStrand == "+")

{

subBases += bases.Substring(Convert.ToInt32(exonStart), length).ToUpper();

}

else if (geneStrand == "-")

{

subBases += ComplementBases(bases.Substring(Convert.ToInt32(exonStart), length).ToUpper());

}

baseStart = Convert.ToInt32(exonEnd);

}

}

}

}

//Check if gene strand is '-' or not, and Reverse the string with all bases for specified Gene

if (geneStrand == "-")

{

subBases = ReverseBases(subBases);

}

//Concat Gene Label and respective bases for All gene ids mentioned.

BasesValue += BasesLabel + "(" + geneStrand + ")"+ "\n" + subBases + "\n";

}

}

//Write the final value into a File with .fa extension

File.WriteAllText(@FinalLocation, BasesValue);

}

/// <summary>

/// In this method, if the gene strand is '-' then compliment as A=T,C=G,G=C,T=A and a=t,c=g,g=c,t=a

/// and return the complimented string.

/// </summary>

/// <param name="s"></param>

/// <returns></returns>

public static string ComplementBases(string s)

{

char[] arr = s.ToCharArray();

for (int i = 0; i < arr.Length; i++)

{

switch (arr[i])

{

case 'A':

case 'a':

if (arr[i] == 'A')

{

arr[i] = 'T';

}

else if (arr[i] == 'a')

{

arr[i] = 't';

}

break;

case 'C':

case 'c':

if (arr[i] == 'C')

{

arr[i] = 'G';

}

else if (arr[i] == 'c')

{

arr[i] = 'g';

}

break;

case 'G':

case 'g':

if (arr[i] == 'G')

{

arr[i] = 'C';

}

else if (arr[i] == 'g')

{

arr[i] = 'c';

}

break;

case 'T':

case 't':

if (arr[i] == 'T')

{

arr[i] = 'A';

}

else if (arr[i] == 't')

{

arr[i] = 'a';

}

break;

}

}

return new string(arr);

}

/// <summary>

/// In this method, if the gene strand is '-' then after complimenting the string the following code

/// will reverse the whole string and return the reversed string.

/// </summary>

/// <param name="s"></param>

/// <returns></returns>

public static string ReverseBases(string s)

{

char[] arr = s.ToCharArray();

Array.Reverse(arr);

return new string(arr);

}

}

}

App.config:

<!-- This file contains key value pairs of information on locations and GeneIds required for processing

If any change in any of the values, it should be modified here so that it gets reflected in code execution-->

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />

</startup>

<appSettings>

<add key="ChromosomeData" value="d:\chr1.fa"/>

<add key="GeneLocationData" value ="d:\HG19-refseq-gene-annot"/>

<add key="ExonLocationData" value="E:\Bio Informatics\Assignment 1\hg19-refseq-exon-annot-chr1"/>

<add key ="GeneIDs" value="NM\_032291;NM\_024066;NM\_001199739;NM\_003689;NM\_001201547"/>

<add key="FinalLocation" value ="E:\Bio Informatics\Assignment 1\GeneDetailsFile.fa"/>

</appSettings>

</configuration>