**Bio Informatics Computing**

MUKHASIR SHAH SYED

109137268

Initially we have to setup BLAST+ for using makeblastdb and blastn for formatting database and checking homology respectively.

We use following threshold values and criteria for checking homology:

1. BLASTn Evalue = 10^-5 [This would be checked using BLASTn]
2. Alignment Identity >= 95% [This would be checked using BLASTn]
3. 90%<=alignment\_length/query\_length<=110% [This would be checked using code]

Step 1:

* Checking homology of MetaHIT-20000.fa against HMP-2000.fa using NCBI BLASTn gives MetaHIT-nr-HMP.fa file.
* Code reads output file and removes matching sequences in MetaHIT-20000.fa and stored as “MetaHIT-nr-HMP\_95.fa”

Step 2:

* We used modified MetaHIT data for checking homology of HMP-2000.fa against MetaHIT-nr-HMP\_95.fa using NCBI BLASTn gives HMP-nr-(MetaHIT-nr-HMP).fa
* Code reads output file and removes matching sequences in HMP-2000.fa and stored as “HMP-nr-(MetaHIT-nr-HMP)\_95.fa”

Finally store the merged data into “Union\_HMP\_MetaHIT.fa” file which contains query label and respective sequence length.

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

\* Date: 03-March-2015. Author: Mukhasir Shah Syed

\* Using the below code we will build a non-redundant microbiome database from HMP (Human Microbiome

Project) and MetaHIT databases, i.e., union of the two databases.

\* HMP and MetaHIT databases contain prokaryotic genomic contigs (sequence) and there exists

redundant ones. Our objective is to produce union of HMP and MetaHIT databases by eliminating

redundant genomic contigs (sequences).

\* We would use BLASTn to determine the homology of sequences and below code will access the output

file fetch after BLAST operation.

\* Step 1:

\* We will check homology of "MetaHIT-20000.fa" against "HMP-2000.fa" using NCBI BLASTn.[Cygwin Used]

a. format database (reference database) fasta file:

$ ./makeblastdb -in HMP-2000.fa -dbtype nucl -out HMP2000

b. run BLASTn using the homology checking criteria (Identity:95% or more, Evalue threshold:10^-5)

$ ./blastn -evalue 1e-5 -perc\_identity 95 -db HMP2000 -query MetaHIT-20000.fa > MetaHIT-nr-HMP.fa

\* Now using code below we would traverse through MetaHIT-nr-HMP.fa file to get the query value

which have "alignment\_length/query\_length" value between 90 and 110[including both] and

\* remove the query sequences from "MetaHIT-20000.fa" file.

\* RemoveMatchings() method which takes List which contains MetaHIT-20000.fa file data and the

\* homology file "MetaHIT-nr-HMP.fa", is used to remove the matching sequences from MetaHIT-20000.fa file.

\* Now we will use modified MetaHIT-20000.fa file ("MetaHIT-nr-HMP\_95.fa") which doesnot contain

matching sequences as Database for checking homology.

\* Step 2:

\* We will now check homology of "HMP-2000.fa" against "MetaHIT-nr-HMP\_95.fa" using NCBI BLASTn.

a. format database (reference database) fasta file:

$ ./makeblastdb -in MetaHIT-nr-HMP\_95.fa -dbtype nucl -out MetaHITnrHMP20000

b. run BLASTn using the homology checking criteria (Identity:95% or more, Evalue threshold:10^-5)

$ ./blastn -evalue 1e-5 -perc\_identity 95 -db MetaHITnrHMP20000 -query HMP-2000.fa > HMP-nr-(MetaHIT-nr-HMP).fa

\* Now we will place "HMP-nr-(MetaHIT-nr-HMP).fa" file in working folder[E:\Bio Informatics\Assignment 4]

and Type "Y" and if file exists then we proceed with operation on HMP-2000.fa data, else exit.

\* Now using code below we would traverse through HMP-nr-(MetaHIT-nr-HMP).fa file to get the query value

which have "alignment\_length/query\_length" value between 90 and 110[including both] and

\* remove the query sequences from "HMP-2000.fa" file.

\* RemoveMatchings() method which takes List which contains HMP-2000.fa file data and the

homology file "HMP-nr-(MetaHIT-nr-HMP).fa", is used to remove the matching sequences from HMP-2000.fa file.

\* Now we will place modified HMP-2000.fa file into "HMP-nr-(MetaHIT-nr-HMP)\_95.fa" which doesnot contain

matching sequences.

\* Now we merge two lists into one list which contain modified HMP data and MetaHIT data.

\* Sort data present in List based on query value.

\* Place the data from merged list into "Union\_HMP\_MetaHIT.fa" file in the format mentioned below:

QueryLabel SequenceLength

>C3636389\_1\_MH0002 500

>C3636421\_1\_MH0002 500

>C3636483\_1\_MH0002 500

\* 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.IO.MemoryMappedFiles;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace NonRedundantDB

{

class Program

{

/// <summary>

/// In this main() method we would read MetaHIT-20000.fa file and BLASTn output file from homology of

/// MetaHIT-20000.fa against HMP-2000.fa and remove matching sequences from MetaHIT-20000.fa based on the criteria.

/// and store in MetaHIT-nr-HMP\_95.fa file which would be used to get homology of HMP-2000.fa against this file.

/// Remove the matching sequence from HMP-2000.fa file and store in HMP-nr-(MetaHIT-nr-HMP).fa file.

/// Merge both modified MetaHIT and HMP data then sort baesd on querylabel value.

/// Finally put querylabel and corresponding sequence length in "Union\_HMP\_MetaHIT.fa" file.

/// </summary>

/// <param name="args"></param>

static void Main(string[] args)

{

//Variables used in the program

string HMPLabel = string.Empty;

string HMPSeq = string.Empty;

string MetaHitLabel = string.Empty;

string MetaHitSeq = string.Empty;

string HMPnrData = string.Empty;

string MetaHITnrData = string.Empty;

bool FilePresent = false;

//STEP : 1

//Initialize KeyValuePair List for storing MetaHIT-20000 sequences.

List<KeyValuePair<string, string>> MetaHit20000\_Data = new List<KeyValuePair<string, string>>();

//Loop through file to store the label and sequences respective in key-value pair format.

foreach (string MetaHitLine in File.ReadLines(@"E:\Bio Informatics\Assignment 4\MetaHIT-20000.fa"))

{

if (MetaHitLine.Contains(">"))

{

if (MetaHitSeq != string.Empty)

{

MetaHit20000\_Data.Add(new KeyValuePair<string, string>(MetaHitLabel, MetaHitSeq));

MetaHitSeq = string.Empty;

}

MetaHitLabel = MetaHitLine;

}

else

{

MetaHitSeq += MetaHitLine;

}

}

MetaHit20000\_Data.Add(new KeyValuePair<string, string>(MetaHitLabel, MetaHitSeq));

//Pass List with MetaHIT data and BLASTn output file of MetaHIT and HMP data.

//This method will traverse through the MetaHit-nr-HMP.fa file and get the query value and its length

//and get identity length value of each matching sequence. Remove the matching sequence which have

//align\_length/query\_length value between 90 and 110 [both included] from the List.

RemoveMatchings(ref MetaHit20000\_Data, @"E:\Bio Informatics\Assignment 4\MetaHit-nr-HMP.fa");

//Convert the complete list into string and place it in a file which will be used for checking homology in STEP-2.

var result = String.Join("\n", MetaHit20000\_Data.ToList().ToArray());

string nrData = result.Replace("[", "").Replace("]", "").Replace(",", "\n");

File.WriteAllText(@"E:\Bio Informatics\Assignment 4\MetaHIT-nr-HMP\_95.fa", nrData);

//Message is displayed and asked for input, so that user can use MetaHIT-nr-HMP\_95.fa and check homology of HMP-2000.fa file data.

//Once we get output from BLASTn, we would place HMP-nr-(MetaHIT-nr-HMP).fa into working folder and press "Y".

Console.WriteLine("Press 'Y' after Homology of HMP2000.fa against MetaHIT-nr-HMP\_95.fa is checked using BLASTn and placing 'HMP-nr-(MetaHIT-nr-HMP).fa' file in corresponding folder.");

ConsoleKeyInfo FileKeyValue = Console.ReadKey();

//We will check if file is present or not. If present then we will set flag as true, else display message.

if(File.Exists(@"E:\Bio Informatics\Assignment 4\HMP-nr-(MetaHIT-nr-HMP).fa"))

{

FilePresent = true;

}

else

{

Console.WriteLine("\n File Not Present, please verify.");

}

//We will check if the file is present and user has press "Y" then proceed for removing matching sequence from HMP data.

if (FileKeyValue.Key.ToString().ToLower() == "y" && FilePresent == true)

{

Console.WriteLine("File found and processing started");

//STEP : 2

//Initialize KeyValuePair List for storing HMP-2000 sequences.

List<KeyValuePair<string, string>> HMP2000\_Data = new List<KeyValuePair<string, string>>();

//Open HMP-2000.fa file and read line by line data and place data into list as key value pair.

using (FileStream fs = File.Open(@"E:\Bio Informatics\Assignment 4\HMP-2000.fa", FileMode.Open, FileAccess.Read, FileShare.ReadWrite))

using (BufferedStream bs = new BufferedStream(fs))

using (StreamReader sr = new StreamReader(bs))

{

string line;

while ((line = sr.ReadLine()) != null)

{

if (line.Contains(">"))

{

if (HMPSeq != string.Empty)

{

HMP2000\_Data.Add(new KeyValuePair<string, string>(HMPLabel, HMPSeq));

HMPSeq = string.Empty;

}

HMPLabel = line;

}

else

{

HMPSeq += line.Trim();

}

}

}

HMP2000\_Data.Add(new KeyValuePair<string, string>(HMPLabel, HMPSeq));

//Pass List with HMP data and BLASTn output file of HMP and MetaHIT-nr-HMP\_95.fa data ["HMP-nr-(MetaHIT-nr-HMP).fa"].

//This method will traverse through the HMP-nr-(MetaHIT-nr-HMP).fa file and get the query value and its length

//and get identity length value of each matching sequence. Remove the matching sequence which have

//align\_length/query\_length value between 90 and 110 [both included] from HMP List.

RemoveMatchings(ref HMP2000\_Data, @"E:\Bio Informatics\Assignment 4\HMP-nr-(MetaHIT-nr-HMP).fa");

//Traverse through modified HMP2000\_Data list and place query label and sequence length in "HMP-nr-(MetaHIT-nr-HMP)\_95.fa" file.

foreach (KeyValuePair<string, string> V in HMP2000\_Data)

{

//Append data to text already present in the file.

using (StreamWriter sw = File.AppendText(@"E:\Bio Informatics\Assignment 4\HMP-nr-(MetaHIT-nr-HMP)\_95.fa"))

{

//Write/Append Query Label

sw.WriteLine(V.Key);

//Write/Append Sequence

sw.WriteLine(V.Value);

}

}

//Merge HMP List into MetaHIT list.

MetaHit20000\_Data.AddRange(HMP2000\_Data);

//Sort Merged list MetaHit20000\_Data based on key value(query label)

MetaHit20000\_Data.Sort(Compare);

//Initialize a generic list of type string.

List<string> FinalData = new List<string>();

//Traverse through merged list and place query label and sequence length into List

foreach (KeyValuePair<string, string> PatternValue in MetaHit20000\_Data)

{

FinalData.Add(PatternValue.Key + "\t" + PatternValue.Value.Length);

}

//Convert list into single string

var Finalresult = String.Join("\n", FinalData.ToList().ToArray());

//Place string into Union\_HMP\_MetaHIT.fa file.

File.WriteAllText(@"E:\Bio Informatics\Assignment 4\Union\_HMP\_MetaHIT.fa", Finalresult);

Console.WriteLine("End");

}

}

/// <summary>

/// This method is used a delegate funtion to sort the list based on the key.

/// </summary>

/// <param name="a"></param>

/// <param name="b"></param>

/// <returns></returns>

static int Compare(KeyValuePair<string, string> a, KeyValuePair<string, string> b)

{

return a.Key.CompareTo(b.Key);

}

/// <summary>

/// This method will traverse through the BLASTn output file and get the query value and its length

/// and get identity length value of each matching sequence. Remove the matching sequence which have

/// align\_length/query\_length value between 90 and 110 [both included] from query sequence file.

/// </summary>

/// <param name="Data"></param>

/// <param name="Path"></param>

public static void RemoveMatchings(ref List<KeyValuePair<string,string>> Data, string Path)

{

//Variables used in this method.

string query = string.Empty;

double length = 0.0;

double align = 0.0;

bool QueryLength = false;

bool MatchDelete = false;

string Seq = string.Empty;

string Label = string.Empty;

//Traverse through BLASTn output file line by line.

foreach (string lines in File.ReadAllLines(Path))

{

//Check if the line contains "Query="

if (lines.Contains("Query="))

{

string[] QueryValue = lines.Split('=');

query = QueryValue[1];//stores label name

query = ">" + query.Trim();

//Flags used to check whether to traverse for further matchings or not.

QueryLength = true;

MatchDelete = true;

}

else if (lines.Contains("Length=") || lines.Contains("Identities = "))//Check if line contains Length= and Identities=

{

if (lines.Contains("Length="))//If line contains Length= then go inside and based on QueryLength flag store length.

{

//if querylength flag is true then set the length.

if (QueryLength)

{

string[] LengthValue = lines.Split('=');//stores data/bases of protein.

length = Convert.ToInt32(LengthValue[1]);

QueryLength = false;

}

}

else if (lines.Contains("Identities = "))//if line contains "Identities = " then go inside statement and based on MatchDelete flag perform some operations.

{

//If matchdelete is true then get the align length and calculate the identity percentage.

if (MatchDelete)

{

double Value = 0.0;

//In the line which contains Identities we get substring of the number by taking starting point as

//index of '/' + 1 and length as [index of '(' - 1] substract [index of '/' - 1] and trim if any spaces.

string value = lines.Substring(lines.IndexOf('/') + 1, ((lines.IndexOf('(') - 1) - (lines.IndexOf('/') + 1))).Trim();

//Convert string value into integer value.

align = Convert.ToInt32(value);

int i = 0;

//Traverse through the list of key value pair and calculate align/length value and remove sequences baeed on value.

foreach (KeyValuePair<string, string> DataValue in Data.ToList())

{

if (query == DataValue.Key)

{

//calculate align/length value

Value = (align / length) \* 100;

//Check if value in between 90 and 110 or not.

//If yes then remove the query label and sequence from the respective file/list.

if (Value >= 90.0 && Value <= 110.0)

{

Data.RemoveAt(i);

MatchDelete = false;

break;

}

}

i++;

}

}

}

}

}

}

}

}