**Bio Informatics Computing**

MUKHASIR SHAH SYED

109137268

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

\* Date: Feb 07, 2015. Author: Mukhasir Shah Syed

\* Using the below code we will implement the Smith-Waterman local alignment with the

linear gap scoring scheme.

\* As dynamic programming would require large table which seems impractical to implement

such a O(nm) alignment.

\* In the code below we implement space optimized version by using two columns.

\* With this space optimized version, we cannot tract the alignment, but can obtain the

optimum score and ending positions of the optimum local alignment.

\* Following code will read database file (chr1.fa) and

query file(Prog2-input-NM\_032291-10exon-seqs.fa) which includes multiple fasta format

sequences. Once reading is done then code calls Smith-Waterman function for

each query sequence.

\* We would consider score as "-1" for each indel (insertion or deletion) and score for

match as "+2" and score for mis-match as "-1".

\* Based on the inputs and above mentioned score we calculate optimal score using

Smith-Waterman algorithm for first 10 exon sequence for NM\_032291.

\* Inputs Provided:

\* Chr1.fa

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

database file for this code.

\* Prog2-input-NM\_032291-10exon-seqs.fa

This is query file which contains gene NM\_032291's first 10 exon sequences.

\* Output contains optimum local alignment score for each exon sequence against database

file "chr1.fa", it will aso show corresponding end positions of the query and

database sequences.

\* In the code we have Main() method and SmithWatermanScore().

\* Main() method is used to read database file and query sequence file. Get exon label

and sequence then call SmithWatermanScore() method to get optimum score, database index

and query sequence index. After getting all values display the output in below

mentioned format.

>chr1.66999824.67000051.NM\_032291\_exon\_0\_0\_chr1\_66999825\_f.+ len=227

ref(1-10) = NNNNNNNNNN ; patern(1-10) = TTTCTCTCAG

--- Optimum Smith-Waterman score = 454 (i=227, j=67000051)

\* SmithWatermanScore(string,string) this takes Reference sequence(database) and

Query sequence strings as input. By using Smith-Waterman algorithm we calculate

optimum score for local alignment.

\* 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 GeneMaxScore

{

class Program

{

/// <summary>

/// This is the method that gets hit first when this application is run.

/// In this method we read data from Chr1.fa for database and from Prog2-input-NM\_032291-10exon-seqs.fa for

/// query sequence for matching. Get all the query sequences into Keyvalue pair list.

/// Loop the list so that optimum score is calucluated for each exon by calling SmithWatermanScore(string,string)

/// </summary>

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

static void Main(string[] args)

{

string bases = string.Empty;

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

string[] lines = File.ReadAllLines(@"D:\BioInfo Activity\chr1.fa");

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

list.RemoveAt(0);

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

string RefSeq = bases;

//Variables used to store query sequence label, bases and final text format to be displayed.

string PatternID = string.Empty;

string Pattern\_Seq = string.Empty;

string FinalText = string.Empty;

//Initialize KeyValuePair List for storing query sequences.

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

//Loop through Prog2-input-NM\_032291-10exon-seqs.fa file to store all query sequences into list.

foreach (string ExonData in File.ReadLines(@"E:\Bio Informatics\Assignment 2\Prog2-input-NM\_032291-10exon-seqs.fa"))

{

if (ExonData.Contains(">"))

{

if (Pattern\_Seq != string.Empty)

{

PatternData.Add(new KeyValuePair<string, string>(PatternID, Pattern\_Seq));

Pattern\_Seq = string.Empty;

}

PatternID = ExonData;

}

else

{

Pattern\_Seq = ExonData;

}

}

PatternData.Add(new KeyValuePair<string, string>(PatternID, Pattern\_Seq));

//Loop through List of query sequences and get Optimum Score and ending positions of optimum local alignment.

//For the string the required format mentioned below.

//>chr1.66999824.67000051.NM\_032291\_exon\_0\_0\_chr1\_66999825\_f.+ len=227

//ref(1-10) = NNNNNNNNNN ; patern(1-10) = TTTCTCTCAG

//--- Optimum Smith-Waterman score = 454 (i=227, j=67000051)

foreach (KeyValuePair<string, string> PatternValue in PatternData)

{

//Forming string to show the output in required format.

string Header = PatternValue.Key + "\t len=" + PatternValue.Value.Length.ToString() + "\n";

string MiddleText = "ref(1-10) = " + RefSeq.Substring(0, 10) + " ; pattern(1-10) = " + PatternValue.Value.Substring(0, 10) + "\n";

string score\_indexText = SmithWatermanScore(RefSeq,PatternValue.Value);//calling SmithWatermanScore method

FinalText += Header+MiddleText+score\_indexText;

}

//Store the output/complete result for all 10 exon sequences into Result.txt file.

File.WriteAllText(@"E:\Bio Informatics\Assignment 2\Result.txt", FinalText);

}

/// <summary>

/// SmithWatermanScore(string,string) method uses Smith-Waterman Algorithm to get Optimum Score.

/// Parameters passes are database string and query sequence string.

/// An 2 dimensional array ScoreIndex is initializeed which has "zero(0)" as default value and

/// which have row length as (query sequence length+1) and column value as 2. As default value

/// is zero we need not have to set value first row as zero in the begining.

/// Code will match each bases of query sequence to each base in database string.

/// Loop through the matrix and by implementing Smith-Waterman alogorithm we get Optimum Score

/// for the sequence.This method returns a string which have Optimum score and ending positions

/// of optimum local alignment which is used in above method (Main()).

/// </summary>

/// <param name="Refseq"></param>

/// <param name="PatternSeq"></param>

/// <returns></returns>

public static string SmithWatermanScore(string Refseq,string PatternSeq)

{

//Variables which wil be used in the method for calculation

int maxValue = 0,optimalScore = 0,PatternIndex = 0, RefSeqIndex=0;

//Array which has row value as query sequence+1 and column size as 2.

//Here default value in rows and columns in ScoreIndex array is “zero(0)”.

int[,] ScoreIndex = new int[PatternSeq.Length + 1, 2];

int i, j;

//Loop through the size of database(Column)

for (j = 1; j < Refseq.Length; j++)

{

//Loop through the size of query sequence.(Row)

for (i = 0; i <= PatternSeq.Length; i++)

{

//If it is first row then value in [0,1] should be set as "0"

//Else check each base of query sequence with each base in database string

//If the bases are matched then score will be 2 or for mismatch as "-1"

if (i == 0)

{

ScoreIndex[i, 1] = 0;

}

else

{

int scoreValue = 0;

//Compare the query sequence base with database sequence and get scoreValue as "2" if match.

//else "-1" for mis-match.

if (PatternSeq[i - 1] == Refseq[j - 1])

{

scoreValue = 2;

}

else

{

scoreValue = -1;

}

//Implementation of Smith-Waterman Alogorithm.

//For insertion we will substract "1" from value in side section.

//For deletion we will substract "1" from value in upper section.

//For match/mis-match we will add scoreValue fetched from above code to value in section diagonally.

int value1 = ScoreIndex[i, 0] - 1;//Insertion

int value2 = ScoreIndex[i - 1, 1] - 1;//Deletion

int value3 = ScoreIndex[i - 1, 0] + scoreValue;//Diagonal

//Code will check the max of value1(insertion), value2(deletion), value3(match/mis-match)

//and "zero(0)" and place it in respective index

maxValue = Math.Max(Math.Max(Math.Max(value1, value2), value3), 0);

ScoreIndex[i, 1] = maxValue;

//Check if the new score is greater than optimum score or not.

//If yes then go inside and assign new optimum score to optimalScore and ending positions to PatternIndex

//and RefSeqIndex respectively from i,j..

if (ScoreIndex[i, 1] >= optimalScore)

{

optimalScore = ScoreIndex[i, 1];

PatternIndex = i;

RefSeqIndex = j;

}

}

}

//Move 2nd column to 1st column, so that query sequence bases are compared with next base in the database string.

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

{

ScoreIndex[k, 0] = ScoreIndex[k, 1];

}

}

//Once all bases in database string are looped and checked then return the optimum score and ending positions.

return "--- Optimum Smith-Waterman score = " + optimalScore.ToString() + " (i=" + PatternIndex.ToString() + ", j=" + RefSeqIndex.ToString() + ")\n";

}

}

}

***Output for given nucleotide data:***

>chr1.66999824.67000051.NM\_032291\_exon\_0\_0\_chr1\_66999825\_f.+ len=227

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = TTTCTCTCAG

--- Optimum Smith-Waterman score = 454 (i=227, j=67000051)

>chr1.67091529.67091593.NM\_032291\_exon\_1\_0\_chr1\_67091530\_f.+ len=64

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = GATTGAAAAA

--- Optimum Smith-Waterman score = 128 (i=64, j=67091593)

>chr1.67098752.67098777.NM\_032291\_exon\_2\_0\_chr1\_67098753\_f.+ len=25

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = AGGTTCACCA

--- Optimum Smith-Waterman score = 50 (i=25, j=67098777)

>chr1.67101626.67101698.NM\_032291\_exon\_3\_0\_chr1\_67101627\_f.+ len=72

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = CAGCCCAGCC

--- Optimum Smith-Waterman score = 144 (i=72, j=67101698)

>chr1.67105459.67105516.NM\_032291\_exon\_4\_0\_chr1\_67105460\_f.+ len=57

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = AAGAAAAGCA

--- Optimum Smith-Waterman score = 114 (i=57, j=67105516)

>chr1.67108492.67108547.NM\_032291\_exon\_5\_0\_chr1\_67108493\_f.+ len=55

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = AACTCACCTG

--- Optimum Smith-Waterman score = 110 (i=55, j=67108547)

>chr1.67109226.67109402.NM\_032291\_exon\_6\_0\_chr1\_67109227\_f.+ len=176

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = CTACCAAAGG

--- Optimum Smith-Waterman score = 352 (i=176, j=67109402)

>chr1.67126195.67126207.NM\_032291\_exon\_7\_0\_chr1\_67126196\_f.+ len=12

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = AGGAAAAGTC

--- Optimum Smith-Waterman score = 24 (i=12, j=171314484)

>chr1.67133212.67133224.NM\_032291\_exon\_8\_0\_chr1\_67133213\_f.+ len=12

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = AGGCGCAGCC

--- Optimum Smith-Waterman score = 24 (i=12, j=214153268)

>chr1.67136677.67136702.NM\_032291\_exon\_9\_0\_chr1\_67136678\_f.+ len=25

ref(1-10) = NNNNNNNNNN ; pattern(1-10) = GGTGCAATT

--- Optimum Smith-Waterman score = 50 (i=25, j=67136702)