**Program input**: chr1.fa and a query file containing gene NM\_032291’s first 10 exon sequences.

**Program output**: The optimum local alignment score for each exon sequence against chr1.fa, together with the corresponding ending positions of the query and database sequences.

**Description:** This program is to implement the space optimized version of Smith-Waterman local alignment algorithm with the linear gap scoring scheme, which is using only two columns for the score table. This program reads a database file(chr1.fa) and a query file, which includes gene NM\_032291's first 10 exon sequences, and then call Smith-Waterman algorithm to output the optimum score.

**SOURCE CODE:**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <string>

#include <cmath>

#include <sys/time.h>

#include <vector>

//using namespace std;

typedef std::vector<int> intVec;

void smithWaterman(std::string &ref\_str, std::string &pattern) /\*Space optimized Smith-Waterman local alignment algorithm with the linear gap scoring scheme\*/

int main(int argc, const char \* argv[])

{

std::string str;

std::string str1="";

std::string str2;

std::ifstream ref;

//Open reference file

fileOpener(ref, "chr1.fa");

std::getline(ref, str); //Get rid of the first line which is not part of the sequence

while (!ref.eof()) //get whole reference sequence

{

std::getline(ref, str);

str1.append(str);

}

ref.close();

std::ifstream in; // get the query file

fileOpener(in, "input.fa");

while (!in.eof())

{

std::getline(in, str); //get exon id

cout << str;

std::getline(in, str2); //get exon sequence from exon file

std::cout << "\t len = " << str2.length() <<std:: endl;

std::cout << "ref(1-10) = " << str1.substr(0,10) << "; " << "pattern(1-10) = " << str2.substr(0,10) << std::endl;

smithWaterman(str1, str2); // call to the function

}

in.close();

return 0;

}

void smithWaterman(std::string &ref\_str, std::string &pattern)

{

double len\_ref\_str = ref\_str.length();

double len\_pattern = pattern.length();

intVec Vtable[len\_pattern+1][2];

int ins, del, diag,i,j,k;

/\*

ins: current score with insertion

del: current score with deletion

diag: current score with matching or mismatching

\*/

int optimumScore = 0;

double optimumIndex\_i = 0, optimumIndex\_j = 0;

for(i=0;i<=len\_pattern;i++ ) //Initialize the two column to all 0

{

**Vtable[i][0]=0;**

}

**for(j=0;j<=len\_ref\_str;j++)**

{

for**(i=0;i<=len\_pattern;i++)**

{ //Calculate insertion, deletion and diagonal value

ins = VTable[0].at(i) - 1;

del = VTable[1].at(i-1) - 1;

diag = ref\_str.at(j) == pattern.at(j-1) ? VTable[0].at(i-1) + 2 : VTable[0].at(i-1) - 1;

if(i==0)

{

Vtable[i][1]=0;

}

else

{ /\*find the max value from 4 choices(insertion, deletion, diagonal and 0) , Save it to the second column \*/

**VTable[1].at(i) = max( max(ins, del), max(diag, 0));**

}

if (VTable[1].at(j) > optimumScore)

{

optimumScore = VTable[1].at(i);

optimumIndex\_j = j+1;

optimumIndex\_i = i;

}

}

for(k=0;k<=len\_pattern;k++) //Copy the second column to the first column

{

**Vtable[k][0] = Vtable[k][1];**

}

}

std::cout << "--- Optimum Smith-Waterman score = " << optimumScore << " ( i=" << optimumIndex\_i << ", j = " << optimumIndex\_j << " )" << std::endl;

}

**OUTPUT:**

**Compiling the program in cygwin:**

**$g++ filename.cpp -o exection filename**

**$ g++ pgrm2.cpp -o bio2**

**Running the exective file:**

**$./bio2**

>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=67126207)

>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=67133224)

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

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

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

291T(BIOINFORMATICS COMPUTING)

(Assignment 2)

NAME : YELLAPATI , SRUTHI

ID: 109153960