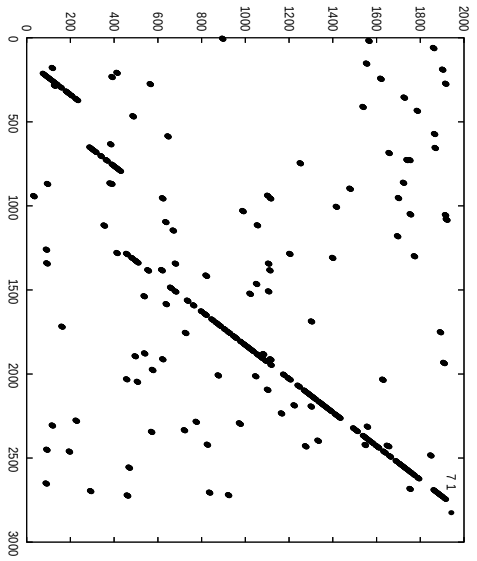
CSci 3003 – Lab 10
Building a Dot Matrix Analysis Tool

1 Introduction

In this lab (and next), we are going to implement a visualization tool that displays sequence alignments on the screen. We are going to use BioPerl and a graphic plotting utility called gnuplot for this project. This fun project calls for efficient algorithm design, good programming techniques, and everything we have learned so far in Perl and maybe more. We hope this project will help and guide you through in solving biology related problems from problem formulation, implementation of the solution to the problem, and testing the final piece of software that achieve our goal, which is to perform sequence analysis through visualization.

1.1 What is Dot Matrix Analysis?

A dot matrix analysis is a method for comparing two sequences to look for possible alignment of characters between two sequences. It displays possible sequence alignments as diagonals on the matrix.



Sample dot plot comparing two sequences XM_536967.2 XM_594524.2.

1.2 Dot Matrix method

In dot matrix method of sequence comparison, one sequence A is listed across the top of the matrix and the other sequence B is listed down the left side. Starting with the first character in B, one moves across the matrix keeping in the first row and placing a dot in any column where the character A is the same. The second character in B is then compared to the entire A sequence, and a dot is placed in row 2 wherever a match occurs. This process is continued until the matrix

is filled with dots representing all the possible matches of A characters with B characters. Any region of similar sequence is revealed by a diagonal row of dots.

2 Building a Dot Matrix analysis tool

2.1 Step 1: Retrieving sequences using BioPerl

You should first write your program to handle short test sequences read in directly from local files. After you have it working, you can then complete this step 1 to retrieve real sequences directly from national databases by name. Once your program is working properly, you will have the chance to showcase it by using it on real sequences. We will use BioPerl to do the fetching of sequences from GenBank. As a test case, retrieve the seguences `'RPBP22 '` and `'RPBPL '` from the `'genbank'` database. The code to retrieve these is exemplified by the following:

```
use Bio::Perl;
$sequence_name_as_a_string = 'RPBP22 '; ## (this is just an example)
$sequence_object = get_sequence('genbank',$sequence_name_as_a_string);
$sequence = $sequence_object->seq();
```

2.2 Step 2: Building a two dimensional matrix

Sequence A \ Sequence B	A	G	C	T	G	T	A	G
G		4			1			1
C			4					
T				4		1		
G		2			4			1
C			2					
A	1						1	

Figure 1: A 6 by 8 dotplot. The numbers are the lengths of diagonal chains.

This is the first step that you should construct, but will become the second step of the final program. The first step is to construct a two dimensional matrix where one can fill in the dots for matched characters. Given 2 sequences A and B, the length of A corresponds to the number of columns of the 2-D matrix and the length of B is the number of row of the 2-D matrix. For example, if A has 12 elements and B has 10 elements, then the 2-D matrix is a 12 rows by 10 columns 2-D matrix
So, you need to:

1. Find the length of each sequence
2. Scan the two strings, filling in the matrix with 1's and 0's, where

```
$matrix[$i][$j] = 1 if (substr($seqA,$i,1) eq substr($seqB,$j,1));
```

2.3 Step 3: Graphic display

Generate the coordinates of the chain of dots that represent alignment of A and B. You will need to write the coordinates of dots to a file so that gnuplot can then generate a "postscrpt" file for display. Once the postscript file is in place, you can then use "gv" utility to view it. From the *shell* command line, type: `gv plot.ps` where `plot.ps` is the postscript file.
The format of the coordinate file is illustrated by the following file, used to plot the example in Fig. 1. The `pointsize` can be adjusted for large plots.

```
set terminal postscript
set pointsize 1.0
plot '-' title "0 1.0" with points pt 7
1 2
1 5
1 8
2 3
3 4
3 6
4 2
4 5
4 8
5 3
6 1
6 7
e
```

To generate the `plot.ps` file, at the command line, type the *shell* command

```
gnuplot plot > plot.ps
```

where "`plot`" is your coordinate file. This will generate the required postscript file for the "gv" utility.

3 Next Week: Find the alignments

Do *NOT* do this step during the first lab. Once you complete the rest of this assignment, you can then consider this step, and think about how you will design the code before you attempt to write anything. This is because this part can be expensive and slow if done in a naive way.

In this step, you will need to locate the chains of dots that represent the alignments of sequence A and B. This code will prune the matrix so that only those dots corresponding to long chains will be plotted. You will need to come up with a scheme that scans through the sequences to find the chains of dots. You will have to design a scheme that does not scan the entire matrix more than necessary, because this will be expensive for long strings. Therefore you should think a little how you will design this scan before you come into the lab.

One suggested way is to scan the entire matrix, and every time you reach a 1, you scan down the diagonal to find how long the chain is, and then replace all the 1's in that chain with the length of that chain. One must also build up a list of all the i,j-positions within a chain above a given threshold. This can be done in the same scan or in a subsequence scan of the matrix. The coordinates can be recorded by writing them out in a file as shown in the section on graphical displays. One question is how to design the code so that one can decide to adjust the threshold or the pointsize in the plots while keeping to a minimum the amount of computation that must be repeated. So this simple mechanism just described may not be sufficient.