

CSC / BIO 310

Bioinformatics

Instructor: Dr. Laurie J. Heyer

Assignment #3

Due Thursday, Jan 31

Instructions: Create a Word document to answer the questions below, and email the Word document to me. You may not consult with anyone outside of your team, other than me. To do this assignment, you will need to modify the files `regex.pl` or `regex_whole_genome.pl` with the appropriate regular expressions described in chapters 4 and 12.

1.

- a. Explain what an EcoRI site is, and how these sites can be used by biologists.

An EcoRI site is a restriction site where a particular enzyme can cut DNA. Biologists can use these sites to “cut and paste” DNA together, and to infer things about the sequence of a molecule by what fragment lengths are produced by cutting the sequence with a particular restriction enzyme (or set of restriction enzymes).

- b. Use a regular expression to find all EcoRI sites in the E. coli genome. Paste the line of your Perl script containing this regex into your Word document.

```
my $regex = 'GAATTC';
```

- c. List the number of EcoRI sites that were found in the genome.

645 matches found. Note that if we wanted to search the other strand for EcoRI sites, we would first find the reverse complement of the given strand (so that it read 5' to 3'). But because the reverse complement of GAATTC is also GAATTC, we **do not** have to perform this search. We would just find the exact same locations that we did in the search of the first strand.

2.

- a. Modify either `regex.pl` or `regex_whole_genome.pl` to read the file `TOM2_GeneList_noblink.txt` rather than the E. coli data.
- b. Write a regular expression that finds lines that start with “blank”. Paste the line of your Perl script containing this regex into your Word document.

```
my $regex = '^blank';
```

- c. List the number of lines in `TOM2_GeneList_noblink.txt` that start with “blank”, and the number of lines that do not start with “blank”.

512 lines start with “blank”, the remaining $12672 - 512 = 12160$ lines do not.

3.

- a. Do a web search to find a disease associated with dinucleotide repeats.

Here are some that I found. There are conflicting reports about some of these, and association does not mean cause. You may have found others.

Mental illness:

<http://content.karger.com/ProdukteDB/produkte.asp?Doi=79971>

Cystic fibrosis:

<http://www.pnas.org/cgi/content/full/101/10/3504>

Diabetic nephropathy:

<http://www.nature.com/ki/journal/v60/n4/abs/4492562a.html>

Breast cancer:

http://www.ncbi.nlm.nih.gov/pubmed/18161656?ordinalpos=2&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum

Colorectal cancer (responsiveness to therapy):

http://www.ncbi.nlm.nih.gov/pubmed/18206383?ordinalpos=1&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum

Endometriosis:

http://www.ncbi.nlm.nih.gov/pubmed/16169423?ordinalpos=1&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum

- b. Use regular expressions to find all occurrences of dinucleotide repeats in *E. coli*. List the number of occurrences for each observed number of repeats.

Search the whole genome to find patterns in the genome. The list of 7-mers does not cover the whole genome (note that none are repeated in the list).

The regex might look like `'(..)\1{n}'` where n is the number of repeats of the starting dinucleotide.

1 – 196632

2 – 12933

3 – 729

4 – 30

5 – 1

4. Use regular expressions to find all occurrences of strings of dinucleotides in which each dinucleotide consists of a repeated base. TTGGTTTT and AATTGGAACC are examples of this motif, of length 6 and 10, respectively. List the number of non-overlapping occurrences of this motif of each length greater than or equal to 8.

The regex I used was

`my $regex = '((.)\2){4}';`

where 4 indicates half the length of the resulting motif. This search was repeated for various lengths.

Number of occurrences of each length:

8 – 13259

10 – 3203

12 – 730

14 – 173

16 – 41

18 – 8

20 – 1

>20 – 0

5. Explain the similarities and differences in each of the following regular expressions:

a. `((.)\2){4}`

This matches a 6-base mirror repeat, e.g. ACTTCA. If two such motifs are adjacent to each other, they would both be counted separately.

b. `((.)\2){4}`

This matches motifs that repeat the *same* 6-base mirror repeat exactly once, e.g.

ACTTCAACTTCA

This regex returns only patterns of length 12.

c. `((.)\2){4}`

This matches a repeat (`\1`) of the 6-base mirror repeat, 0 or more times (`*`). The `*` always modifies the pattern just before it, in this case, the `\1`. This regex returns patterns of this type that have length 6, 12, 18, etc. The number found in this case is less than in (a) because if there are 12 or 18 matching, the greediness counts only 1 occurrence rather than 2 or 3 occurrences.

d. `((.)\2)+`

This matches 1 or more repeats of a 6-base mirror repeat, but note that because there is no `\1` in the regex, the 6 base motif may be different in each repeat of the “pattern”, e.g.

ACTTCAGTCCTGAATTAA