Needles in a Haystack: Finding structural transcription factor-binding sites in Enteropathogenic *E. coli*

Lily Ben-Avi

Enteropathogenic *Escherichia coli* (EPEC) is a gram-positive, rod-shaped bacterium that is a major cause of diarrheal disease in the developing world[i] (Ochoa paper). A major research target is a 35kb mobile genetic element known as the locus of enterocyte effacement (LEE), without which the bacteria are significantly less effective pathogens. This locus contains all the genes necessary for a molecular syringe known as a type-III secretion system (T3SS), which translocates virulence factors into the host epithelium to cause disease[ii]. The first gene in this element, known as LEE1, encodes the regulatory protein Ler, which regulates not only the rest of the genes in the LEE, but also a number of other genes that determine the pathogenicity and/or survival of the bacterium. There is a high level of interest in identifying which genes are regulated by Ler, as well as the nature of that regulation. However, this is somewhat challenging because Ler does not bind to any conserved regulatory motif. Instead, it binds to AT-rich sequences in the promoter regions of the genes that it directly regulates[iii]. While this is suspected to be a fairly common binding pattern for transcription factors involved in the regulation of horizontally transferred genetic elements[iv] , the programs that exist to look for binding sites require a conserved sequence as input. Thus, the aim of this code is to identify AT-rich regions in promoter regions of genes shown experimentally via microarray analysis to be either directly or indirectly regulated by Ler.

Sequence data for this project was obtained from the EPEC E2348/69 full-genome nucleotide sequence on NCBI[v]. Using the results of a microarray analysis of the Ler regulon[vi], a list of genes that were strongly upregulated by Ler were identified, with 'strongly' referring to a fold-change greater than or equal to 20. The pre-promoter regions, defined as the 330 nucleotides directly preceding the promoter, were copied into 'EPECgenes.txt'. The odd-numbered lines are the names of the genes, and the even numbered lines are the sequences, thus allowing the code to pair the names and sequences. This data collection strategy did take a fair amount of time on my part because I had to locate the genes and manually identify where the pre-promoter region was. Then I would copy the FASTA sequence into a text document and make sure it was all on one line, with the preceding line being the name of the gene. However, because the UCSC genome browser didn't have EPEC, this was the fastest and most efficient way to obtain the sequences. I also was working with pretty small strings, so it wasn't so hard to do.

The intent of this code is to calculate the AT-content of a given section of the promoter sequence by taking a sliding window along the code. It will then graphically display the AT-content of each kmer agains the position in the sequence at which that kmer begins. The code's only input—besides the file containing the sequence data, which is actually imbedded in the function—is a k value. This corresponds to the length

of the k-mers that the code will use, and the larger this value is, the less noisy the graph will be.

So the code begins by reading the file 'EPECgenes.txt' and storing the names of the genes in one list called 'listofnames' and the sequences of the promoter regions in another list, 'listofstrings'. Then, two functions will be called, 'makeATD()' and 'makeindexD'. The first of these utilizes another function, 'calcAT()', to get the AT-content for each successive kmer in a string and store them in a sequential list. This list, called the 'ATlist', is used to make the dictionary 'ATD', in which the key is each gene name in 'listofnames', and the value for each genes is the 'ATlist'. The makeindexD() function also makes a dictionary 'indexD' in which the key is the name of each gene, but instead of lists of AT-content as values, they are instead lists called the 'iList' in which the position of the start of each corresponding kmer are stored. Thus, using one key, a final graphing function can be used to plot the AT-content by position for a given gene. The code will simply loop through for each name (listofnames[i]), and each gene will have its own graph produced. The graphing code specifically is adapted from HW4 from class. A slight alteration I also made to my code is the option of using the 'easyAT()' function. If used, insead of getting a list of %AT values called 'ATlist', there will be a binary list in which there is either a 0.1, if the AT content of a given kmer is below the threshold value of 0.7, or a 0.8 if it is above. This was done in order to make it easier to discriminate between peaks and noise in the graphs.

For a first attempt, my code had fairly positive results. When tested on toy sequences called meow, woof, and quack, the graphs came out well using the original 'calcAT()' function (Fig 1).

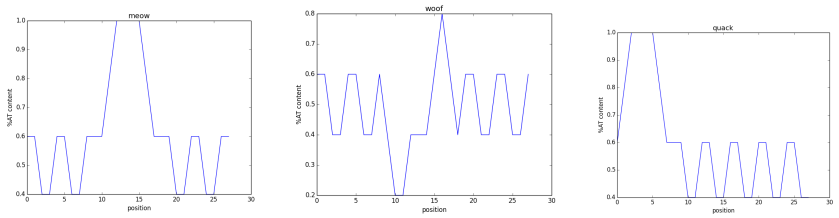

Figure 1. Graphs showing %AT-content vs. position in sequence for three toy sequences. Both 'meow' and 'quack' have AT-rich regions, which show up as peaks in the graphs. Woof was supposed to be random and thus have no peaks, but it appears that there are slight negative and positive peaks. Regardless though, it is clearly less definitive than the other two

Once I got to the real data, things got a little messier, but still seemed to work well. A number of the genes I tested had AT-rich regions, and the easyAT function made these easier to spot (Fig 2).

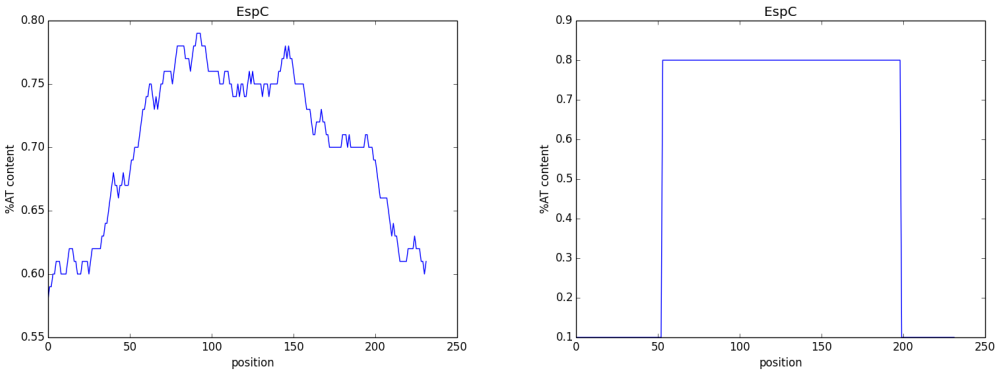

Figure 2. Graphs showing (left) the %AT-content by position in the promoter region of EspC, and (right) the same data, but with a peak only where the %AT exceeds a threshold value of 0.7. The one on the right is certainly more convincing, although it wasn't necessary in order to interpret the first graph. The presence of a peak indicates an AT-rich region of the sequence, suggesting direct regulation of this gene by Ler.

The 'easyAT()' function also made it easier to pick out the genes that do not have a significant AT-rich region, such as in eaeA (Fig 3).

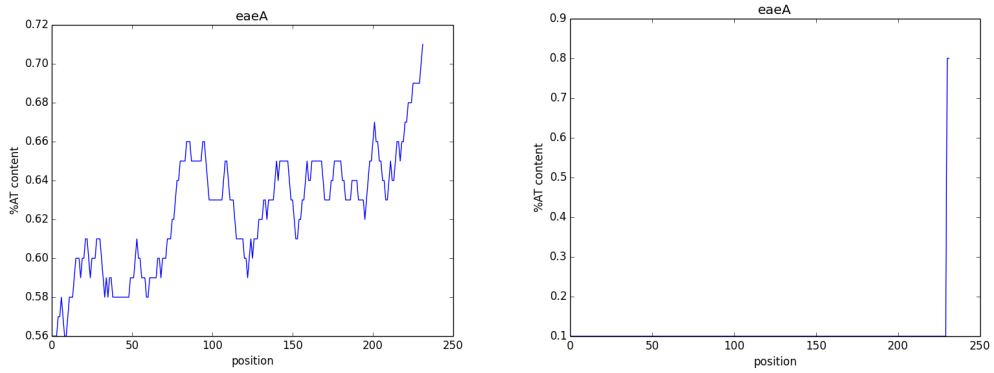

Figure 3. . Graphs showing (left) the %AT-content by position in the promoter region of EspC, and (right) the same data, but with a peak only where the %AT exceeds a threshold value of 0.7. . The left one is not easy to interpret, and it is not clear until one sees the right graph that there is no appreciable AT-rich region.

The main assumption that I made that may prove to be problematic is that the Ler binding site will be within the 300nt of the promoter. I based this off of the paper from which I got the original idea, however, there is no reason that it could not be farther. I also assumed that 0.7 was a reasonable threshold for AT-richness, but it is also possible that Ler may be able to bind with less than that. Finally, I assumed that the genes that showed the highest fold change in the microarray with the Ler-deletion strain would be those most likely to have Ler bind. However, the regulon of Ler is quite complex, and it is entirely possible that there are some that it binds to that simply have multiple regulatory inputs and are thus not showing a huge net change.

The approach I used could certainly be improved upon so that some kind of peak picking program can determine the statistical significance of a given peak. Were this done, it could be applied to a lot of regulatory factors in prokaryotes. Many bacteria, like EPEC, store their virulence factors in mobile genetic elements. Because there is evidence that horizontally transferred genes might more commonly have this kind of binding site for their transcription factors, there is a huge need for programs that can find a similar system in other bugs, perhaps even with less input requirements. Theoretically, the overall concepts could eventually be used to make a code that simply searched for AT-rich regions in the genome, and used other criteria to help identify genes that might be regulated by Ler besides those we already suspect.

REFERENCES:

[i] Ochoa TJ, Contreras CA. "Enteropathogenic *E. coli* (EPEC) infection in children". *Curr Opin Infect Dis.* 2011 October. 24(5): 478–483. doi:10.1097/QCO.0b013e32834a8b8b

[ii] Franzin F.M., Sircili M.P. "Locus of Enterocyte Effacement: A Pathogenicity Island Involved in the Virulence of Enteropathogenic and Enterohemorragic *Escherichia coli* Subjected to a Complex Network of Gene Regulation". BioMed Research International, vol. 2015, Article ID 534738, 10 pages, 2015.

[iii] Sperandio V, Mellies JL, Delahay RM, Frankel G, Crawford JA, Nguyen W, and Kaper JB. "Activation of enteropathogenic *Escherichia coli* (EPEC) LEE2 and LEE3 operons by Ler". *Molecular Microbiology.* (2000), 38(4), 781-793.

[iv] Will RW, Bale DH, Reid PJ, Libby SJ, and Fang FC. "Evolutionary expansion of a regulatory network by counter-silencing". *Nature Communications.* (2014). 5:5270 doi: 10.1038/ncomms6270.

[v] Reference: NC_011601.1

[vi] Bingle LH et al. "Microarray Analysis of the Ler Regulon in Enteropathogenic and Enterohemorrhagic Escherichia coli Strains". *Plos One.* 2014. 9(1) e80160.