Amy Rose Lazarte
Bio 131
Anna Ritz
May 10, 2015

Comparing Blast Alignments of Mouse Genes and Human Chromosomes

Originally, the goal of this project was to create synteny blocks that would compare rearrangements in human and mouse chromosomes. However, that ended up being a little too ambitious, and I was forced to focus in on a smaller set of data. I ended up creating dot plots that compared different Blast alignments of the mouse Ubn1 gene and human chromosome 16. While all of the alignments I found for Ubn1 had above 80% identity, I was able to find alignments of the mouse gene gt(ROSA)26 that had a smaller %identity. The relatively bad gt(ROSA)26 alignments provided a baseline to compare my Ubn1 alignments to, which confirmed that the Ubn1 alignments were good. Ultimately, my program can take any table of blast results and produce dot plots that visualizes the quality of the alignments.

To get my data, I downloaded the mouse genes I was interested in from NCBI. I originally had some issues running my program because I accidentally downloaded the wrong gene file. For future reference, the best way to download a gene sequence is to select “Go to nucleotide: FASTA” under Genomic Regions, Transcripts, and Products on the full report of the gene. Afterwards, click “Send to” and you can download the sequence as a fasta file. Once I had my gene sequence, I blasted it against the human genome. I downloaded the alignment results as table(csv) files, which opened in excel. The table in excel didn’t have any titles, so after some searching I found the title for each column and added them to a row in excel. I converted that file into a .txt file, which I was able to use in my program. Lastly, I downloaded the sequences for the human chromosomes that aligned with my mouse gene.

The majority of my programming was focused on pulling data from the alignment chart so that it could be used to compare sequences. After writing a function that read the sequence files and converted them into strings, I wrote functions that converted the alignment text file into a list. I separated this list by rows so that all of the data for each alignment was grouped together. I wrote functions that stored all of the query start, query end, search start, and search end coordinates into their own lists. These coordinates were then used to find the corresponding segments of either the mouse gene or human chromosome sequence. I modified a function that I had created for an earlier homework assignment that created a matrix comparing two strings and their reverse complements. Two segments for each alignment, the matrix, and a k-value were inputted into the dot plot function Anna made, although a few minor adjustments were made to allow for using two different strings.

After creating dot plots for different alignments using different values of k, I was ultimately able to find a good baseline for determining the quality of an alignment. While typically a k-value of 4 tended to give off too much noise to be able to understand the data, with smaller alignment lengths this was not the case. And similarly, while a k-value of 10 gave the most focused view, on smaller alignment lengths it was entirely too sparse. This shows that though making a dot plot of an alignment will ultimately show you the quality of the alignment, it may take two or three different plots to get an accurate idea of the alignment. The best alignments have a strong diagonal line running from the origin to the top right corner, whereas less accurate alignments will either have a broken up line or no pattern at all.